

UNIVERSITY OF NEW ENGLAND

DOCTORAL THESIS

Computer Vision Approaches for Weed Detection: Using Traditional and Deep Learning Methods in Southeast Australian Pastures

Author:

Jonathan FORD BCompSc
(Hons 1) (UNE)

Supervisors:

Dr. Edmund SADGROVE
Assoc. Prof. David PAUL



*A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy*

in the

Computational Science
School of Science & Technology

*This research has been conducted with the support of the Australian Government
Research Training Program Scholarship*

16th October 2025

Declaration of Authorship

I, Jonathan FORD, declare that this thesis titled, “Computer Vision Approaches for Weed Detection: Using Traditional and Deep Learning Methods in Southeast Australian Pastures” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 16th October 2025

"A weed is but an unloved flower!"

Ella Wheeler Wilcox

UNIVERSITY OF NEW ENGLAND

Abstract

Computational Science

School of Science & Technology

Doctor of Philosophy

**Computer Vision Approaches for Weed Detection: Using Traditional and Deep
Learning Methods in Southeast Australian Pastures**

by Jonathan FORD

Weeds pose a significant challenge in Australia, costing the livestock industry an estimated AUD 2,409 million annually due to control measures and lost production. The application of computer vision for weed detection in agriculture is an expanding area of research, offering the potential to develop autonomous, site-specific weed management (SSWM) solutions that minimize herbicide use and reduce the labour needed for weed control. Using images collected by autonomous ground vehicles in Southeast Australian pastures infested with a range of weeds, several computer vision techniques were considered.

In one study, a pipeline was developed using local binary patterns, histograms of oriented gradients, and colour features, and an extreme learning machine, to produce a fine, patch-based segmentation of images containing serrated tussock, thistle species, white horehound, and Bathurst burr, yielding a mean Intersection over Union (mIOU) of 87.6, 81.6, 79.5 and 87.1 respectively. The fine-grained segmentation produced by this model enabled more accurate localization of individual weed plants, which should enable a greater precision in herbicide application.

In another study, a dual-purpose network for plant segmentation and spray point detection was developed using a novel convolutional neural network (CNN) architecture incorporating ConvNeXt modules. This network was tested on a dataset of serrated tussock and achieved a mean Intersection over Union (mIOU) of 80.7 for segmentation and an F1-score of 79.6 for spray point detection, even under field conditions not encountered during training. These results outperformed state-of-the-art dual-purpose networks. Furthermore, when a novel normalization technique, termed HistMatch, was applied prior to input, the network's performance improved to an mIOU of 85.4 and an F1-score of 80.6, respectively.

The dual-purpose network was then tested on a multispecies dataset of serrated tussock, white horehound, variegated thistle and stinging nettle, yielding mIOU values of 88.8 and 82.6 for the plant and spraypoint tasks respectively. This demonstrated that the network was able to generalize well to the situation in which multiple species of weed are present in a field.

The studies presented in this thesis have advanced the understanding of computer vision techniques for weed localization in pastures in several ways. Firstly, the ELM-based segmentation approach demonstrated greater precision in weed localization compared to previous methods that relied on classification or bounding box techniques. Additionally, the development of a dual-purpose network for plant segmentation and spray point localization in pastures represents the first of its kind, enabling more flexible treatment options depending on whether herbicide is applied to the entire plant or just its center. HistMatch normalization proved to be a superior technique compared to other normalization methods, particularly in scenarios where illumination and growing conditions in the test set differ from those encountered during training. Further testing of this dual-purpose network showed that it remains robust even in complex scenarios where multiple weed species are present in the field.

Acknowledgements

I would like to begin by expressing my deepest gratitude to my parents, who first encouraged me to learn and who fostered in me a lasting curiosity about the world. Their steady support and belief in the value of knowledge have been a constant source of inspiration throughout my life.

I am sincerely grateful to my supervisors, whose expert advice in the field of computer vision and agricultural applications has been invaluable. Their guidance through the many stages and challenges of this PhD—from shaping research questions to interpreting results—has not only strengthened this work but also deepened my understanding of what it means to conduct meaningful research.

I also wish to acknowledge the many teachers I have had over the years, at both school and university, who generously shared their knowledge and helped to shape my academic journey.

To all of those who have guided, encouraged, and supported me along the way, I extend my heartfelt thanks.

Contents

Declaration of Authorship	iii
Abstract	viii
Acknowledgements	ix
1 Introduction	1
1.1 Motivation	1
1.2 Background	4
1.3 Southeast Australian Pastures	6
1.4 Research Questions	7
1.5 Outline of Thesis	8
2 Literature Review	11
2.1 Introduction	11
2.2 Computer Vision	12
2.2.1 Introduction to Computer Vision	12
2.2.2 Colour Spaces and Hyperspectral Imagery	13
2.2.3 Feature Extraction and Classification	13
Segmentation through Thresholding	14
Feature Extraction	14
Classification	16
2.2.4 Conventional Classifiers	16
Artificial Neural Networks	16
Support Vector Machines	19
Random Forests	20
Extreme Learning Machines	20
Variations of Extreme Learning Machines	21
2.2.5 Deep Learning - Combining Feature Extraction and Classifica- tion	23
2.2.6 Attention Mechanisms	28
2.2.7 From Classification to Segmentation	30
The Continuum of Localization	30
Converting a Classifier into a Segmenter	31
Models Used in Segmentation	33
Improving Spatial Context and Receptive Field Size	35
Detection Models	36
2.2.8 Summary	37
2.3 Weed Localization in Farmland.	39
2.3.1 Introduction	39

2.3.2	Image Capture	39
2.3.3	Weed Datasets used in the Literature	42
2.3.4	Common Metrics	43
2.3.5	Methods Specific to Crop Fields	48
2.3.6	Methods used in Pastures and Other Complex Environments	50
2.3.7	Conventional Machine Learning Methods	53
2.3.8	Deep Learning Methods	56
2.3.9	Hybrid Methods	64
2.3.10	Transfer Learning	65
2.3.11	Data Augmentation Techniques	66
2.3.12	Generalization of Models	67
2.3.13	Models with Multiple Outputs	68
2.3.14	Comparison Studies of Conventional and Deep Methods	70
2.3.15	Summary	71
2.4	Conclusions	72
3	Developing an Extreme Learning Machine Based Approach to Weed Segmentation in Pastures	77
3.1	Introduction	77
3.2	Published Paper	78
3.3	Supplementary Material	91
4	Joint Plant-Spraypoint Detector with ConvNeXt Modules and HistMatch Normalization	101
4.1	Introduction	101
4.2	Published Paper	102
4.3	Supplementary Material	125
5	Dual-Task Multi-Species Network Design and Training Using Advanced Augmentation Techniques	137
5.1	Introduction	137
5.2	Published Paper	139
6	Discussion and Conclusion	151
6.1	Introduction	151
6.2	Research Summary	151
6.3	Can a shallow network be used to segment weeds in pastures in SE Australia?	156
6.4	Can CNNs be used to segment weeds in pastures in SE Australia – focusing on modern architectures?	157
6.5	Can segmentation be done in real time?	157
6.6	Can we produce a combined plant/spraypoint detector?	158
6.7	Can we produce a multi-species detector?	159
6.8	How well do segmentation models generalize to fields or conditions that they have not been trained on?	159
6.9	Can advanced data augmentation techniques improve results?	160
6.10	Broader Implications	160

6.11 Conclusion	161
A Exploring Context and Attention in CNN Model Design	163
A.1 Introduction	163
A.2 Methodology	163
A.3 Results	164
A.4 Discussion	166
Bibliography	167

List of Figures

1.1	(a) An image from the CWFID dataset [85], with small carrot plants and weeds against a background of soil. (b) An image taken for the CRC showing variegated thistle and nettle growing in pastoral land.	3
1.2	The two vehicles used for image capture in this study. On the left is the Polaris Ranger EV, showing the data collection rig. On the right is the Kubota RTV 900XT.	4
1.3	Research progression roadmap showing the key stages of this thesis, including baseline approaches, joint plant and spraypoint detection, multispecies extension, training optimisation, and condition-invariance evaluation.	9
2.1	An example of overfitting, where a simple linear model (solid line) models the relationship between inputs and targets better than a more complex model (dashed curve). The complex model performs poorly when given new data (hollow points), even though it fits the training samples (solid points) perfectly.	19
2.2	An example CNN which takes 48 x 48 pixel RGB images as input, and outputs a vector of 6 values, with each value corresponding to a different class. The input is green, convolutional layers are red, pooling layers are blue, flattening layers are grey, and fully connected layers are yellow. The flattening layer flattens the 3-dimensional feature map (or tensor) into a 1-dimensional vector. The dimensions of the outputs of each layer are written inside the layer.	24
2.3	The convolutional operator for a single filter. The input is often padded with zeros to preserve the spatial dimensions of the output. A single output value (shown in red), is calculated by element-wise multiplication of the filter with a section of the input (red). The elements of the resultant matrix are added together, along with a bias term, producing the final result.	25
2.4	Different modules used in CNNs: (a) an inception module [236], (b) a dense module [98], (c) a residual module [86], (d) a MobileNetV2 modules [213], (e) a ConvNeXt module [142] Kernel sizes are shown for conv, dconv (depthwise convolution) and pool layers. Filter dimension of output is shown in brackets where required to demonstrate bottleneck and inverted bottleneck designs.	27

2.5	A classification model, such as VGG-16 (a) [222], can be converted into an FCN-32s segmentation model (b) [143], by removing the flattening layer, replacing the fully connected layers with convolutional layers, and adding an upsampling layer. Layers are coloured green (input), red (convolutional), blue (downsampling), gray (flattening), yellow (dense), and purple (upsampling). For convolutional layers, the filter size is shown above.	32
2.6	Some common CNNs used in segmentation problems share a similar macro structure, consisting of an encoder arm, a decoder arm, and skip connections that link the two.	33
2.7	Examples of skip connections (a-c) and upsampling operations (c-e) used in fully convolutional networks. (a) FCN-16s/FCN-8s [143] skip, (b) U-Net [200] skip, (c) SegNet [9] skip, (d) Transposed convolution [268], (e) upscaling. For (c), the skip connections and upsampling are combined into a single unpooling operation. For (e), nearest interpolation is shown for simplicity. However, bilinear interpolation is more commonly found in the literature.	34
2.8	The unpooling operation of SegNet [9]. The max-pooling indices from the encoder arm are passed to the decoder arm and used during up-sampling.	34
2.9	Architecture of PSPNet (from [274])	35
2.10	Dilated convolutional kernels are produced by inserting gaps between the elements of the filter. (a) When the rate is set to 1, a dilated convolution is equivalent to a regular convolution. (b) When the rate is set to 2, the convolution is equivalent to a larger regular convolution, with the gaps filled in with zeros.	36
2.11	Example confusion matrices for (a) binary, and (b) multi-class problems. For the binary task, TP (TN) = true positive (negative), and FP (FN) = false positive (negative).	45
2.12	Example PR (a) and ROC (b) curves. For a PR curve, a baseline model which performed no better than random would achieve a precision equal to the proportion of samples in the positive class (p_+), and a recall equal to the choice of threshold. For the ROC curve, the baseline model would produce a TPR equal to the FPR, and both would be equal to the choice of threshold.	46
A.1	The location within the truncated ConvNeXt (tCN) model at which different modules were inserted. Section (a) shows the insertion of a CBAM module within a residual block, while section (b) shows the insertion of the PPM, ASPP or CBAM modules within the macrostructure of the network. Although all modules are shown in the one diagram, only a single module was included in each of the model variants.	165

List of Abbreviations

AE	Autoencoder
AGV	Autonomous Ground Vehicle
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
AP	Average Precision
ASPP	Atrous Spatial Pyramid Pooling
AUC	Area Under Curve
BCE	Binary Crossentropy
BERT	Bidirectional Encoder Representations from Transformers
BRISK	Binary Robust Invariant Scalable Keypoints
CBAM	Convolutional Block Attention Module
CC	Connected Component
CCE	Categorical Crossentropy
CDELM	Constrained Difference Extreme Learning Machine
CED-Net	Cascading Encoder-Decoder Network
C-ELM	Constrained weights Extreme Learning Machine
CF-ELM	Colour Feature Extreme Learning Machine
CIW	Computed Input Weights
CIW-ELM	Computed Input Weights Extreme Learning Machine
CLCM	Colour Level Co-occurrence Matrix
CMELM	Constrained Mixed Extreme Learning Machine
CNN	Convolutional Neural Network
CRC	Co-operative Research Centre
CRF	Conditional Random Field
CSELM	Constrained Sum Extreme Learning Machine
CV	Computer Vision
DLA	Deep Layer Aggregation
DL	Deep Learning
DPI	Department of Primary Industries
DW-ELM	Deep Wide Extreme Learning Machine
ED	Euclidean Distance
ELM-AE	Extreme Learning Machine Autoencoder
ELM	Extreme Learning Machine
EN-ELM	Ensemble Extreme Learning Machine

ESNet	Extended Skip Network
ExG	Excess Green
ExR	Excess Red
FCN	Fully Convolutional Network
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm
FLOP	Floating Point Operation
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GA	Genetic Algorithm
GE	Generalization (accuracy)
GELU	Gaussian Error Linear Unit
GLCM	Grey Level Co-occurrence Matrix
GPT	Generative Pre-trained Transformer
GSD	Ground Sampling Distance
HOG	Histogram of Oriented Gradients
HSI	Hyperspectral Imagery
HSL	Hue Saturation Lightness colour space
HSV	Hue Saturation Value colour space
ICICM	Integrated Colour and Intensity Co-occurrence Matrix
IOU	Intersection Over Union
IR	Infrared
IRM	Inversted Residual Module
kNN	K-Nearest neighbour
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LIDAR	Light Detection and Ranging
LR	Linear Regression
mAP	mean Average Precision
MEC-ELM	Multiple Expert Colour feature Extreme Learning Machine
mIOU	mean Intersection Over Union
ML	Machine Learning
MLP	Multilayer Perceptron
MP	Megapixels
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
OBIA	Object-Based Image Analysis
OOB	Out Of Bag
PCA	Principal Component Analysis
PLS-DA	Partial Least Squares Discriminant Analysis
PPM	Pyramid Pooling Module
PR	Precision-Recall

RBM	Restricted Boltzmann Machine
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RF-ELM	Receptive Field Extreme Learning Machine
RF	Random Forest
RGB	Red Green Blue colour space
ROC	Receiver Operating Characteristic
RoI	Region of Interest
RSELM	Random Sum Extreme Learning Machine
SAT	Summed Area Table
SCF-ELM	Segmented Colour Feature Extreme Learning Machine
SD	Standard Deviation
SDR	Successful Detection Rate
SELM	Sample Extreme Learning Machine
SLFN	Single hidden Layer Feed-forward Network
SLIC	Simple Linear Iterative Clustering
SMO	Sequential Minimal Optimization
SPM	Spatial Pyramid Matching
SPP	Spatial Pyramid Pooling
SSD	Single Shot MultiBox Detector
SSWM	Site Specific Weed Management
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UAV	Unmanned Aerial Vehicle
UV	Ultraviolet
VGG	Visual Geometry Group
WRN	Wide Residual Network
YOLO	You Only Look Once
ZCA	Zero-phase Component Analysis

Chapter 1

Introduction

1.1 Motivation

The focus of this research is the application of computer vision for the automatic detection of weeds in Southeast Australian pastures. Weeds represent a significant challenge in pastures, as they offer minimal to no nutritional value to livestock and, in some cases, can be harmful [74, 230]. Additionally, weeds compete with beneficial pasture species, leading to decreased productivity. The cost of weeds in Australia, including direct control costs as well as the cost of lost production was estimated to be \$AUD3,927 million per year in 2018, with livestock industries alone accounting for a majority of \$AUD2,409 million [160].

Agricultural land can be broken down into several different categories. Crop fields consist of crop plants, typically planted in rows, which are cultivated for human or livestock consumption, and include such things as grains, vegetables and fruit. Crop fields are generally only planted with a single species of plant at any one time, and the space between the crop plants is often bare soil or stubble, which may or may not contain weeds [96, 139]. Pastures, on the other hand, consist of multiple different plant species growing in a random arrangement, and may be largely comprised of grasses, as well as broadleaf species, such as clover. Pastures do not directly produce human food, but are instead used for grazing animals, such as sheep and cattle. Pastures are sometimes referred to as rangelands, although there is no clear distinction between the two terms. The NSW DPI¹ reserves the term “rangeland” for regions of NSW which are arid or semi-arid, and receive less than 500 mm of annual rainfall on average. In this thesis, the term “pasture” will be used for all grazing land, except where referencing studies which themselves use the term “rangeland”.

One potential method for detecting and controlling weeds in farmland involves the use of an Autonomous Ground Vehicle (AGV). Such a vehicle requires a system for navigating around the field to ensure adequate coverage, a system for avoiding obstacles such as large rocks or trees, a system for detecting weeds, and a system for treating weeds, which may involve the application of herbicide via spray nozzles. These systems need to be interconnected. For example, the system for detecting

¹<https://www.dpi.nsw.gov.au/agriculture/pastures-and-rangelands/rangelands>

weeds will need to communicate with the system for treating weeds, so that an appropriate treatment is applied only when weeds are found. This literature review will focus solely on the systems use for detecting weeds, as this is the subject of the thesis.

The first step in detecting weeds is capturing images of the farmland, which may be carried out by downward facing cameras, although cameras inclined at an angle to the perpendicular are also possible [17]. There are several choices of vehicle to use when capturing images of farmland, which fall into three main categories, being ground vehicle (either manned or unmanned), unmanned aerial vehicle (UAV), and satellite. UAVs can operate at varying heights, ranging from as little as 2 metres [271] to over 100 metres [264] above ground. The choice of vehicle, and hence the height above ground at which the images are captured affects the minimum size of features that can be distinguished in the images. This is typically quantified using the idea of a ground sampling distance (GSD), which is the distance, at ground, between adjacent pixels in the image. Unsurprisingly GSDs vary widely, with one satellite-based study reporting a value of 0.3 m [220], in comparison to one dataset, captured at ground level, reporting a value of 0.11 mm [85].

Another consideration is the kind of camera to use. While one option is a simple colour camera, capable of capturing RGB² images, it is also possible to use infra-red cameras, multispectral cameras capable of capturing multiple bands in the electromagnetic spectrum, as well as sensors which measure depth information. The use of multispectral cameras is referred to as hyperspectral imagery (HSI). One example study utilised HSI to produce images containing 61 bands, ranging from 400 nm - 1000 nm in wavelength [54]. Another example study presented a publicly available dataset, that was captured using multiple sensors, which combine RGB, near infrared (NIR), and depth information to produce a point cloud of the field [4]. NIR images are particularly effective at separating plants from soil due to the high reflectivity of chlorophyll at this wavelength [4, 201].

One of the challenges with weed localization is that weed and non-weed species may have similar colours, morphology or texture [82]. In addition, there may be considerable overlap between different plant components, making it difficult to distinguish where one plant ends and another starts [139].

Traditional methods of weed control involve substantial human effort and extensive use of herbicides, which can adversely affect environmental health. An automatic solution for weed management in pastures presents numerous benefits. Primarily, it would reduce the need for human labour and minimize herbicide usage by employing site-specific weed management (SSWM). SSWM involves spraying herbicides only on the locations where weeds are detected, thus conserving chemicals and reducing environmental impact [155]. One of the key obstacles to the development of an automated SSWM solution is the need for weed identification algorithms that can maintain a high accuracy across the variable conditions found in

²RGB images contain three colour channels, being red, green and blue.

arable land [96]. Approaches to the automation of weed localization include conventional machine learning algorithms, in which hand-crafted features are extracted from the pixel data, and passed to a conventional classifier, such as a Support Vector Machine (SVM) [17, 46, 272] or random forest [5, 112, 147]. More recently, deep learning methods utilizing multi-layered Convolutional Neural Networks (CNNs), which can learn useful representations of data during training, has become more prominent in the applications of computer vision in agriculture [256].

Weed localization in pastures presents significant challenges that are not encountered or are less severe in crop fields. These challenges include the absence of structured plant placement in rows (as seen with crops) and the considerable overlap between different pasture elements, which are typically very similar in colour, making it difficult to distinguish individual plants. These issues can be seen in Figure 1.1. The left image shows an early-stage carrot crop field where the plants are easily distinguishable from the soil due to stark color differences. Additionally, the individual plants have minimal overlap, which helps in clearly identifying the boundaries between crops and weeds. In contrast, the right image depicts a pasture with numerous overlapping plants, including both beneficial species and weeds such as variegated thistle and nettle.

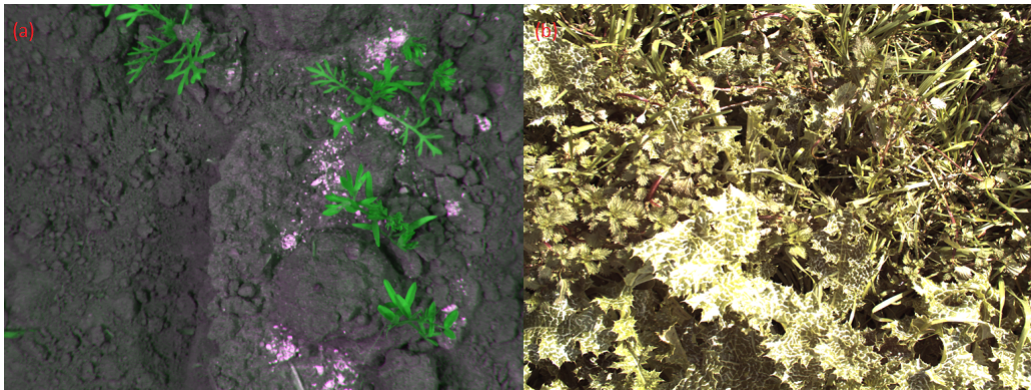


FIGURE 1.1: (a) An image from the CWFID dataset [85], with small carrot plants and weeds against a background of soil. (b) An image taken for the CRC showing variegated thistle and nettle growing in pastoral land.

The proposed solution in this thesis involves the real-time segmentation³ of images captured by an Automated Ground Vehicle (AGV). This real-time image processing enables the AGV's spray nozzles to be activated only in the presence of target weeds, thereby offering a targeted application of herbicides, reducing cost and damage to the environment. The scope of this project is confined to the development and implementation of the image segmentation component for weed detection and does not encompass other aspects of AGV control, such as navigation and obstacle avoidance, or the development of robotic hardware solutions.

³A precise definition of what is meant by real-time will be given later in this chapter.

1.2 Background

The research presented in this thesis was conducted as part of a Cooperative Research Centres Project (CRC-P)⁴, titled “Novel Autonomous Robotic Weed Control to Maximise Agricultural Productivity”. This project (hereafter referred to simply as the CRC), examined real-time applications of computer vision in agriculture, and focused on detection of weeds from images captured by downward facing cameras at ground level. This technology could be used as part of an autonomous weeding platform, and the target application was for site-specific management of weeds of interest in Southeast Australian pastures. A range of different weed species were considered as important, including serrated tussock (*Nassella trichotoma*), African lovegrass (*Eragrostis curvula*), horehound (*Marrubium vulgare*) and thistle species of the family *Asteraceae* such as spear thistle (*Cirsium vulgare*), Scotch thistle (*Onopordum acanthium*) and variegated thistle (*Silybum marianum*).

Datasets used in this thesis were captured by the NSW Department of Primary Industries (DPI). Images were acquired using two off-road ground vehicles, the Polaris Ranger EV and the Kubota RTV 900XT, shown in Figure 1.2. These vehicles are designed for off road use, including sloping and uneven ground, and can handle low obstacles, such as clumps of tussock, and small rocks. The same camera system configuration was used for both vehicles, and is presented in Table 1.1.



FIGURE 1.2: The two vehicles used for image capture in this study. On the left is the Polaris Ranger EV, showing the data collection rig. On the right is the Kubota RTV 900XT.

The complex field conditions, including vehicle movement, plants of varying heights, and varying levels of illumination caused by weather phenomena and vehicle shadow led to some challenges with image generation. With the configuration used, plants were in focus up to 400 mm above ground level, but some higher plants were out of focus. The images were captured with an aspect ratio of 4:3, and a resolution of 2460 by 2052 pixels, which corresponded to a ground patch size of 0.87 by 0.64 m.

⁴Commonwealth Grant CRCPIX000099. Further information about the CRC-P Program can be found at <https://www.cooperativeresearch.org.au/resources/industry-research-grants/crc-p-grants/>

The frame rate used in data capture was 3 FPS. However, in order to maximize the chance of detecting weeds during deployment, this rate could be increased, for example, to ensure that every point on the ground was represented in a minimum of three images. This would give the detection network three chances to detect each weed, reducing the number of weeds missed. In order to prevent a backlog of images that require processing, the frame rate of the detection model must be no less than the frame rate used in data capture. Given an image field of view of 64 cm along the shorter axis, we require three frames per 0.64 m of travel. The AGV travels at a maximum of 1.3 m/s, and using the formula $d = vt$, where d , v and t correspond respectively to distance, velocity and time, we have $0.64 = 1.3t$, or $t = 0.492$ seconds, which is the time taken for a point on the ground to pass entirely through the camera's FOV. In order to take three images in this time, we require an image every $0.492 \div 3 = 0.164$ seconds, which corresponds to a rate of 6.1 FPS. This provides us with an upper bound on the time taken to process a single image, in order to prevent a backlog. For this reason, and for the purpose of this research, real-time image processing is deemed to be any process which can be completed, start to finish, in 164 ms or less.

TABLE 1.1: Camera system configuration settings used in this study.

Specification	Value
Camera	Mako G-507 camera, 5.1 MP 2/3" CMOS Sony IMX264 sensor global shutter
Colour Space	RGB
Camera Height	1200 mm
Camera Angle	0° (downward facing)
Vehicle Speed	1 ± 0.3 m/s
Artificial Lighting	Nil
Frame Rate	3fps
Exposure Time	16 - 1399 μ s
Gain	0 - 30 db
Gamma	0.7
Lens	Goyo Optical GM6HR31214MCN 12mm
Aperture	f5.6
Focal length	1000 mm (0 - 400 mm above ground)
Light Sensor	Yocto-Light-V3 with BH1751FVI Sensor
Measuring range	0 - 100,000 lux
Sensitivity	0.25 lux

1.3 Southeast Australian Pastures

Southeast Australian pastures are shaped by a combination of unique environmental conditions, ancient soils, and grazing management practices. These factors influence pasture species composition, productivity, and resilience, while also affecting the prevalence and management of weeds. Understanding the characteristics of these pasture systems is important for maintaining productive and sustainable grazing enterprises.

Australian soils are among the oldest in the world due to the absence of major glacial disturbance during the last glaciation approximately 21,000 years ago. As a result, many soils have undergone extensive weathering, leading to increased acidity and reduced concentrations of key nutrients such as phosphorus and carbon [49]. In addition, the relatively recent European settlement of Australia, combined with the absence of intensive cultivation practices by Indigenous Australians, has resulted in comparatively few anthropogenic soils - soils that have been substantially altered through direct human activity [49].

The species composition of Southeast Australian pastures has changed considerably over time, with a shift from native warm-season grasses to introduced species that remain green throughout much of the year. However, pasture composition varies according to factors such as the original vegetation, climate, fertiliser inputs, and grazing pressure [68]. The most common improved pasture systems consist of perennial ryegrass (*Lolium perenne*) and white clover (*Trifolium repens*) [181]. Greater species diversity can enhance pasture resilience, with additional species such as lucerne (*Medicago sativa*) and red clover (*Trifolium pratense*) improving drought and heat tolerance while increasing dry matter production during summer and autumn, when ryegrass growth typically declines [181].

Pasture productivity in Southeast Australia is frequently limited by climatic and soil-related issues. During summer, many regions experience periods of high temperatures and limited water availability, which can reduce the growth, persistence, and nutritional value of key perennial grasses such as ryegrass [199]. In areas receiving between 300 and 550 mm of annual rainfall, irrigation is often required to maintain pasture productivity during periods when evaporation exceeds rainfall [199].

Soil acidity is another widespread issue. While lime application can improve soil pH, its use is not always economically viable due to limited returns on investment. Consequently, acid-tolerant legumes such as serradella (*Ornithopus* spp.) and subterranean clover (*Trifolium subterraneum*) are commonly used in these environments [75]. These legumes also play a critical role in pasture systems by contributing biologically fixed nitrogen, which may represent the primary source of nitrogen input in many grazing systems [75].

Weeds are a major challenge in Southeast Australian pastures, with key species including fireweed (*Senecio madagascariensis*), serrated tussock (*Nassella trichotoma*),

Chilean needlegrass (*Nassella neesiana*), and several thistle species [223]. Weed invasion is often associated with declining pasture condition and can be promoted by factors such as overgrazing, drought, reductions in perennial grass cover, poor soil fertility, nutrient imbalances, and pest or disease pressure [223].

Although herbicides remain an important weed control tool, reliance on repeated chemical applications alone can further weaken pastures by reducing desirable species and creating gaps in ground cover that favour weed establishment. Effective weed control therefore requires an integrated weed management approach. Such strategies include maintaining healthy perennial grass and legume populations, reseeding degraded paddocks, preventing overgrazing, and implementing good farm hygiene practices to minimise the spread of weed seeds between locations [223].

Southeast Australian pasture systems operate within a challenging environment characterised by ancient, nutrient-poor soils, periodic drought, and significant weed pressure. Productive and resilient pastures depend on the selection of suitable pasture species, management of soil limitations, and the implementation of integrated weed management practices. Together, these approaches help maintain pasture productivity, improve ecosystem resilience, and support the long-term sustainability of grazing enterprises.

1.4 Research Questions

The research undertaken in this PhD project is centered on developing an advanced computer vision system for the automatic detection of weeds in Southeast Australian pastures. This endeavor requires addressing several critical research questions, each targeting a specific aspect of the system's development and evaluation. These questions aim to explore the feasibility, efficiency, and generalizability of various approaches and techniques within the realm of computer vision and machine learning, particularly focusing on weed detection and management. The following research questions guide this comprehensive investigation:

1. **Can a shallow network be used to segment weeds in pastures in SE Australia?** This question explores the potential of using less complex, shallow neural networks for the task of weed segmentation, aiming to balance accuracy with computational efficiency.
2. **Can CNNs be used to segment weeds in pastures in SE Australia – focusing on modern architectures (e.g., NASNet/ConvNeXt)?** Here, the research investigates the application of state-of-the-art convolutional neural network architectures to enhance weed segmentation accuracy and performance.
3. **Can segmentation be done in real time?** This question addresses the practical aspect of deploying the segmentation model in a real-time setting, as defined above, based on hardware specifications. This is crucial for the effective

functioning of the Automated Ground Vehicle (AGV) in dynamic agricultural environments.

4. **Can we produce a combined plant/spraypoint detector?** This involves developing a system that not only detects weeds but also identifies the precise spray points for targeted herbicide application, optimizing weed management.
5. **Can we produce a multi-species detector?** This question expands the scope to include the detection of multiple weed species, enhancing the versatility and applicability of the system in diverse pasture conditions.
6. **How well do segmentation models generalize to fields/conditions that they have not been trained on?** The generalizability of the developed models is critical for their practical application, and this question assesses their performance in untrained conditions and fields.
7. **Can advanced data augmentation techniques improve results?** This question explores the use of data augmentation techniques that, to date, have not been applied to weed localization studies.

By addressing these questions, this research aims to advance the field of agricultural technology, providing innovative solutions for efficient and sustainable weed management. Table 1.2 provides a summary of which research questions are answered in which chapters.

TABLE 1.2: Research questions and the corresponding thesis chapters.

Research Question	Description	Chapters Addressed
1	Shallow networks	3
2	CNNs	4, 5
3	Real time	3, 4, 5
4	Joint task	4, 5
5	Multispecies	5
6	Generalization	4, 5
7	Augmentation	5

1.5 Outline of Thesis

This thesis is presented as a collection of publications, each addressing specific research questions and contributing to the overall development and evaluation of the

computer vision system for weed detection. Each publication details the methodology, experiments, and findings related to a particular aspect or aspects of the research, ensuring a comprehensive and rigorous exploration of the topic. A technology roadmap outlining the problem, the progression of the research towards a solution, and the outcomes of this research is presented in Figure 1.3. The remaining chapters are structured as follows.

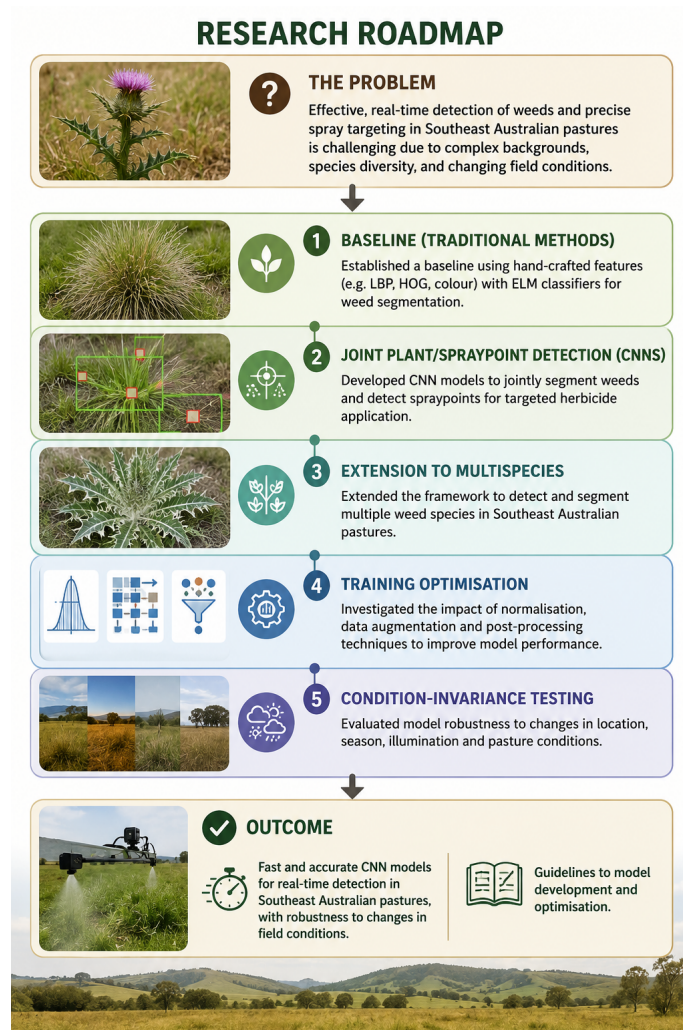


FIGURE 1.3: Research progression roadmap showing the key stages of this thesis, including baseline approaches, joint plant and spray-point detection, multispecies extension, training optimisation, and condition-invariance evaluation.

In Chapter 2, a review of the literature with respect to computer vision applications in weed detection is presented, with a particular focus on those systems designed for use in pastures and/or in image segmentation problems. Chapter 3 presents the first paper, which relates to the use of shallow networks in the segmentation of serrated tussock and other weed species from background pasture. Chapter 4 presents the second paper, which focuses on serrated tussock, and explores the use of convolutional neural networks (CNNs) for the joint tasks of plant segmentation and spraypoint detection. Chapter 5 presents the third paper, in which

the work done in the preceding chapter is extended to the task of multi-species weed detection. This chapter also explores the use of advanced sample mixing data augmentation techniques in agricultural applications. Finally, Chapter 6 presents a comprehensive discussion evaluating whether the research conducted in this thesis successfully addressed the posed research questions. It also identifies any remaining unanswered questions and proposes several potential future research directions, building on the themes explored in this study.

Chapter 2

Literature Review

2.1 Introduction

Weeds in agricultural land are unwanted plants that compete with crops and beneficial pasture elements for water, nutrients and sunlight. Some weeds found in pastures may be eaten by livestock, even though they provide little or no nutrients, or directly harm the animal [74, 230]. Weeds can result in significant loss of production, and their control remains one of the most important activities for farmers. It is estimated that the annual cost of weeds on the livestock industry in Australia, in terms of direct costs involved with control, and lost production is \$2.4 billion [160].

While blanket application of herbicide to an entire field is one method of removing weeds, this is an economically costly solution, as it requires the use of large amounts of herbicide, which additionally may cause damage to beneficial plant species, and affect human health, and the broader environment. Site specific weed management (SSWM) refers to the process of first locating the weeds in the field, and then applying an appropriate treatment solution only where required. The application of herbicide is a typical treatment solution, but other solutions which do not rely on herbicide use are possible. These include physical destruction of the weed, either through mechanical action, or the application of a jet of hot water or laser [39, 57]. SSWM offers a reduction in the use of herbicide, which both reduces the economic cost of treatment, as well as the potential harmful impact of herbicide use on people and the environment [65].

There are multiple studies available in the literature on weed localization in agricultural land. Some examples include the work of Lottes et al. [146, 147, 149–151] on the localization of weeds in sugar beet fields, using images captured at ground level; the work of dos Santos Ferreira et al. [46] on the localization of grass and broadleaf weeds in soybean fields, using images captured by UAV; and the work of Olsen et al. [174] on the localization of multiple weed species in rangeland environments of north-east Australia. This review will present studies of relevance to the thesis, with a particular emphasis on studies that are carried out in complex environments such as pastures, and/or studies that use images captured at or near ground level. Emphasis will also be given to studies on dense (or per pixel) segmentation, in which a particular class (e.g. weed/crop/soil) is assigned to each pixel in the image.

In spite of the abundance of studies on weed localization in agricultural land, there remains much work to be done. For example, there are limited studies on the localization of weeds of interest in Southeast Australian pastures, such as thistle sp., horehound and serrated tussock. Those studies that do exist (e.g. [54, 104, 208–211]) do not consider the task of dense segmentation, as defined in the preceding paragraph, and focus on conventional machine learning techniques. In addition to this, there are new techniques available (for example NASNet [280] and ConvNeXt [142], which will be discussed in Section 2.2.5) whose use in agricultural imagery remain largely unexplored. The high cost of weeds on the livestock industry in Australia, and the limited research carried out in Southeast Australian pastures means that there is very good reason to devote more effort on studying this problem [160].

The remainder of this literature review is structured as follows. Section 2.2 discusses computer vision in general, introducing some of the techniques that have been applied to a wide range of problems. Section 2.3 discusses how computer vision has been used with respect to weed localization in agricultural land. Finally Section 2.4 discusses some of the conclusions that can be drawn from this review of the literature.

2.2 Computer Vision

2.2.1 Introduction to Computer Vision

Machine learning (ML) is a branch of artificial intelligence that focuses on the development of algorithms and models that allow for intelligent behaviour, without the need for explicit programming. Models are trained on example data, and automatically learn useful relationships and patterns, enabling them to make predictions and decision given new, unseen data [106]. Computer Vision (CV) is a subbranch of ML, in which a computer learns to extract relevant information from images. A typical tasks in computer vision is classification, in which a model learns to recognise whether certain objects are present in an image.

ML models can be further broken down into conventional ML, and deep learning. Both of these are used in CV, including weed localization problems. Examples of conventional ML models include support vector machines (SVMs) [34], random forests (RFs) [19] and shallow artificial neural networks (ANNs) [204]. Deep learning models are constructed using ANNs with multiple hidden layers, and include such things as convolutional neural networks (CNNs) [131] and restricted Boltzmann machines (RBM) [90].

Images are very high-dimensional data, with a typical image size used in the literature of $224 \times 224 \times 3$ ¹, yielding a 150528-dimensional space. This would be difficult to work with if raw pixel values were used as input to the CV model. Furthermore, an image which is flipped, rotated, or translated by a small distance will

¹The 3 refers to the number of colour channels

likely have a very similar semantic meaning, even though the raw pixel values are completely different. For this reason, a lot of research into computer vision has gone into extracting relevant features from the image, as a method of reducing dimensionality, while maintaining sufficient discriminative information to be useful for the downstream task. Some features display useful properties such as rotational or translational invariance. Such features may be “hand crafted”, or may be automatically learned during the training process [204].

2.2.2 Colour Spaces and Hyperspectral Imagery

A colour space is composed of one or more intensity channels, and is a way in which different colours are represented in a digital image. The RGB colour space has one channel for each of the three primary colours, red, green and blue, and is likely the most common choice for computer vision studies in agriculture [69, 118, 216, 231]. HSV (Hue, Saturation and Value) is another colour space, that has been used in weed localization studies [79, 209]. The Hue determines the colour (for example, yellow or green), while the saturation determines how much the pure the colour is blended with white light. In comparison, value determines the intensity of the light (i.e. how dark or bright the pixel is) [79]. Y'CbCr has also been used in agricultural applications [110, 207, 209]. Y' is the luma of the colour, which corresponds to a weighted average of the intensities of the red, green and blue components. The Blue-difference Chroma (Cb) and Red-difference Chroma (Cr) values determine how blue, or red the colour is, respectively [209].

In addition to visible light, some specialized cameras can capture electromagnetic radiation in the infrared (IR) spectrum. IR radiation with a wavelength that is close to visible red light is often referred to as near infrared (NIR). Although some studies [85, 146] employ just a single channel of light coming from the IR or NIR spectrum, others have multiple bands of light, covering both visible and IR wavelengths [36, 53, 67]. The latter is referred to as hyperspectral imagery (HSI).

2.2.3 Feature Extraction and Classification

CV can be thought of as a two-step process. In the first step, features are extracted from raw pixel value. A feature can be thought of as any numerical value which has been calculated from the image, and which may or may not be useful for the task at hand. In the second step, the features are passed to a classifier for classification [131]. In rare cases, the first step is skipped, and the raw pixel values become the inputs for the classifier (see for example [159, 209]). This section will discuss some of the “hand crafted” features and conventional classifiers that are encountered in the literature, with a focus on techniques that are used for weed localization. The concept of image segmentation, where regions of interest are extracted from an image, will also be discussed. Deep learning models will not be discussed here, because they do not

use hand crafted features, but instead learn their own representations during the training process. Instead they will be discussed in Section 2.2.5.

Segmentation through Thresholding

Image segmentation is the process by which an image is broken down into multiple segments, where each segment represents either an object of interest, or part of the background. There are multiple methods by which an image can be segmented, including thresholding, clustering, edge detection, Gaussian mixture models, super-pixels, and even methods involving Genetic Algorithms [170]. Thresholding methods vary depending on the choice of colour space. In the field of agriculture, a common method of segmenting vegetation from background soil and crop residue is the use of a vegetative index, which may operate on the RGB colour space, such as excess-green (ExG, equation 2.1) [37], or on red (R) and near-infrared (NIR) colour channels, such as the normalized difference vegetation index (NDVI, equation 2.2) [221]:

$$ExG = 2G - R - B \quad (2.1)$$

$$NDVI = \frac{NIR - R}{NIR + R} \quad (2.2)$$

To this end, a wide range of vegetative indices have been developed, which use visible or infra-red spectral information to distinguish plants from background [257]. A simple thresholding approach, such as Otsu [178], can then be used to complete the segmentation. These methods segment an image based on gross differences between the green colour of vegetation, and the brown or grey colour of the background soil or crop residue, but is not useful in pastures where multiple overlapping plants have similar colours, and so cannot be separated by this means alone.

Feature Extraction

Features used in agricultural imagery fall into the four broad categories of morphological, spectral, textural and spatial [251]. Morphological features, such as area, perimeter, and major/minor axis can be used in cases where leaves can be cleanly segmented from the background [12]. Spectral features, such as vegetative indices can be used in segmentation, but can also be used as a feature during classification [209]. Textural features look at the pattern of light and dark on relatively small scale patches of image, and include Haralick features [81] and Local Binary Patterns (LBP) [173]. An example of a spacial features used in agriculture is the detection of crop rows in aerial imagery [10]. Other features used in image analysis are Histograms of Oriented Gradients (HOG) [38] and K-means features [43].

A Grey Level Co-occurrence Matrix (GLCM) contains information about the comparison of pixel intensities in two (or more) pixels that are separated by a fixed distance and direction. Typically the grey values of pixels are placed into a small number of bins, in order to avoid a situation where the GLCM is too large and sparse. Haralick [81] proposed 28 different features that can be extracted from the GLCM, including contrast, energy, entropy and GLCM mean. The concept of a GLCM can be extended to a colour image, such as in the Integrated Colour and Intensity Co-occurrence Matrix (ICICM) presented by Vadivel et al. [248]. The ICICM is a combination of four sub-matrices, in which the colour or intensity of a pixel is compared with the colour or intensity of its neighbour. For example, the $ICICM_{CI}$ submatrix compares the colour of the pixel with the intensity of its neighbour. As such, the ICICM lends itself to colour spaces with a dedicated intensity channel, such as HSV.

LBP features [173] are another method where the intensity of a pixel is compared with its neighbours. Unlike the GLCM, the pixel is compared with multiple neighbours, and only the relative intensity is considered. As such, the algorithm generates one bit of information for each neighbour as show in equations 2.3 and 2.4.

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.3)$$

$$LBP(g^*) = \sum_{p=0}^{P-1} s(g_p - g^*) 2^p \quad (2.4)$$

Here, $s(x)$ is a sign function, $LBP(g^*)$ is the LBP descriptor for pixel g^* , P is the number of neighbours, and g is the grey level of the pixel. The image can then be broken down into rectangular blocks, and an LBP histogram (LBPH) calculated for each block. If P is set to 8, then the LBPH will have 256 bins, one for each of the possible descriptor values. However, in practice it is typical to pool some of the LBP values into the same bin, in order to achieve a feature vector that is invariant to rotation, and/or combine values which have a low probability of occurring in practical images.

Histograms of Oriented Gradient (HOG) descriptors were originally designed as a solution to the problem of detecting pedestrians in images [38]. In HOG, the image is divided into overlapping blocks of cells, and the occurrence of gradient (or edge) orientation is calculated. The orientation is classified into equal bins from 0° to 180° for unsigned, and 0° to 360° for signed orientation. For the detection of pedestrians, a small block size of 2×2 cells, with each cell being 8×8 pixels, and a stride of 8 was used. Block normalisation was used to reduce the effect of varying levels of illumination, and a total of 9 orientation bins, in the range 0° to 180° were used. It was found that Gaussian blurring reduced accuracy, suggesting that sharp edges at the finest level were important for detecting shape.

An alternative method for feature extraction uses the k-means algorithm to cluster small normalized patches of the image (e.g., 8×8 or 16×16 pixel squares) [33]. Each square can then be mapped to the nearest cluster, or clusters, producing a feature vector to be passed forward to the classifier. Coates et al. [32] demonstrated that the images need to be whitened² prior to clustering in order to produce good results. In fact, their results on the CIFAR-10 [120] and NORB [130] datasets were at the time state of the art. K-means feature learning in agricultural image classification is rare, but has been used to classify soybean from weeds on images collected in China [241, 242]

Classification

Classification is the process whereby images are separated into categories or classes, based on an interesting feature, such as the type of object that is depicted in the image. In agriculture, the classes typically refer to the species of plants that are present in the images, including crop plants and weeds. An image of the background soil without plant components may also constitute an additional class. This section will introduce both conventional and deep classification models, with an emphasis on those which are commonly use in agricultural computer vision applications.

2.2.4 Conventional Classifiers

In the conventional machine learning approach to image classification, features are extracted from the image and passed to a conventional classifier, such as Artificial Neural Networks (ANNs) [204], SVMs [34], or RFs [19]. Each of these classifiers will be discussed in this section, and the ELM algorithm [76] will also be introduced. In Section 2.2.4, variations to the ELM algorithm and network structures will be discussed. In the author's experience, the most common shallow classifiers in agricultural imagery are SVMs and RFs, although ANNs either trained via back propagation or variations of the ELM algorithm have also been used.

Artificial Neural Networks

Artificial neural networks (ANNs) were inspired by the connection of neurons in the human brain. In the brain, a neuron receives inputs from multiple other neurons via its dendritic branches, and converts them into a signal which is passed down its axon to other neurons. Similarly, an ANN is composed of multiple neurons, or nodes, each of which are connected to various input nodes, and output nodes [204]. Nodes are commonly organized into layers, and it is typical for each node to receive inputs from all nodes in the previous layer, and propagate its signal to all nodes in the following layer. This gives us a fully connected layer, as opposed to a partially connected layer, in which only some of the nodes in the previous and/or following

²Whitening refers to a number of methods, such as PCA-whitening and ZCA-whitening, which reduce or eliminate inter-variable correlation.

layer are connected to the neuron. The layers are made up of an input layer, followed by one or more hidden layers, and finally an output layer.

The input layer passes the input data on to the next layer. For example, if our network is designed to analyse 32×32 pixel RGB images, then the input layer will be composed of $32 \times 32 \times 3 = 3072$ nodes, each of which corresponds to a particular pixel and colour channel in the image. A hidden layer combines the information it receives from the previous layer to form a new representation of the data. It does this by generating a linear combination of its inputs, and passing this through a non-linearity, commonly called an activation function. The activation function increases the expressiveness of the model, by allowing it to capture non-linear patterns and relationships in the data. This is demonstrated in equation 2.5, which shows the output y_j of the j th node in a hidden layer. Here x_i is the output of the i th node of the previous layer, b_j is a bias term for the j th node, and g is the activation function.

$$y_j = g\left(\sum_i w_{ij}x_i + b_j\right) \quad (2.5)$$

The output layer is responsible for generating the final output of the network. Its inner workings are the same as a hidden layer, in that each node generates an activated linear combination of the outputs of the previous layer. However, rather than being passed to another layer, its output becomes the output of the entire network. Some example activation functions that have been used in ANNs are provided in Table 2.1. Although sigmoidal functions were used in earlier models, later models, including later CNNs have tended to favour the ReLU activation function and derivatives thereof.

Name	Formula
Sigmoid	$S(x) = \frac{1}{1+e^{-x}}$
Hyperbolic Tangent	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2S(2x) - 1$
Softmax	$\sigma(\vec{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$
Rectified Linear Unit [168]	$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$
Leaky ReLU [154] (for $0 < a < 1$)	$\text{LReLU}(x) = \begin{cases} x & x \geq 0 \\ ax & x < 0 \end{cases}$
Gaussian Error Linear Unit [89]	$\text{GELU}(x) = x\Phi(x) \approx xS(1.702x)$

TABLE 2.1: Some example activation functions used in neural networks. For GELU, Φ is the cumulative probability distribution for a normal Gaussian.

The ANNs discussed so far are called feed forward networks, because their nodes form a directed acyclic graph. There are, however, examples of cyclic neural networks, such as recurrent neural networks [136], which are used to process sequential data. These kinds of networks are seldom used in the literature for problems

involving weed localization, and so they will not be discussed in this review (see however [149], for an example of the processing of sequential image data in weed localization).

ANNs are commonly trained in a fully supervised manner, meaning that we present the network with some labelled training examples, which are input/target pairs, and adjust its parameters so that the model becomes better at outputting a value which is close to the training sample's target, given the training sample's input. Targets are typically encoded as a one-hot vector (meaning that the vector contains a 1 at position n , corresponding to class n , and 0s elsewhere) for multiclass classification, and a scalar, where 0 corresponds to the negative class and 1 corresponds to the positive class, for binary classification. The output layer typically uses a sigmoid activation for binary classification, resulting in a number between 0 and 1, which represents the probability that the input belongs to the positive class. For multiclass classification, a softmax activation is commonly used, because it produces a probability distribution over the predicted classes. A threshold, or *argmax* (i.e. selecting the class with the highest output) for multiclass classification, can be used if an exact prediction is required.

In order to train the ANN, we need a loss function, which is a measure of how far the network's output is from the desired output, and is also known as an error function. A common choice of loss for binary tasks is the binary cross entropy loss, which is given in equation 2.6. Here Y is the real target, $\hat{Y}$ is the predicted target, and N is the number of samples in the training set, or batch.

$$L_{BCE} = -\frac{1}{N} \sum_{i=1}^N (Y_i \log \hat{Y}_i + (1 - Y_i) \log (1 - \hat{Y}_i)) \quad (2.6)$$

For multiclass tasks, the categorical cross entropy loss, given in equation 2.7, can be used, where M is the number of classes.

$$L_{CCE} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M Y_{ij} \log \hat{Y}_{ij} \quad (2.7)$$

The parameters are often initialized randomly, and optimized in an iterative way, by calculating the partial derivative of the loss function, with respect to each of the parameters, and adjusting them as per Formula 2.8, where θ are the parameters of the network, L is the loss function, θ^* are the new parameters of the network after one step of learning, and α is a small value, called the learning rate, which controls how quickly the parameters are adjusted during training.

$$\theta^* = \theta - \alpha \frac{\partial L}{\partial \theta} \quad (2.8)$$

One potential issue with training an ANN is overfitting, in which the network learns to predict the training samples very well, but performs poorly when presented with new, unseen data. A simple example of overfitting is given in Figure

2.1. Overfitting becomes more likely when we are training complex models that contain a large number of parameters. Conversely, having a training dataset which is too small may also contribute to overfitting. There are many techniques which are used to combat overfitting, including early stopping, in which training is halted when the loss on a hold-out validation set reaches a minimum. By comparison, the training data may be augmented in some way, such as through the use of random noise, horizontal and vertical flips, rotations, skews and other image transformations, which have the effect of artificially inflating the number of samples available for training. Finally regularization techniques, such as weight decay, and dropout can also be utilized.

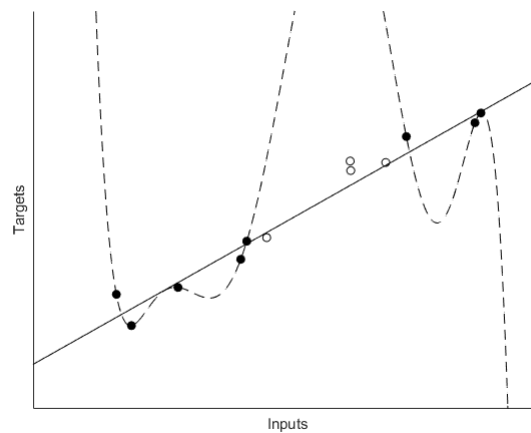


FIGURE 2.1: An example of overfitting, where a simple linear model (solid line) models the relationship between inputs and targets better than a more complex model (dashed curve). The complex model performs poorly when given new data (hollow points), even though it fits the training samples (solid points) perfectly.

Support Vector Machines

SVMs were introduced by Cortes and Vapnik in 1995 [35] as a new learning algorithm for a binary classification problem. They work by computing an optimal hyperplane to separate one class from the other. In practice, the data will most likely not be linearly separable, so SVMs use a nonlinear kernel function, which maps the data into a higher-dimensional space where it can be separated linearly. In the soft-margin approach, datapoints are allowed to fall on the wrong side of the hyperplane, but a penalty is applied for each such error. The optimisation problem is convex, meaning that a global solution can be found. A popular algorithm for training SVMs is Sequential Minimal Optimization (SMO), which can be up to a thousand times faster than other algorithms [183]. SVMs have been a popular choice in machine learning since their inception, and in recent years they have been a very popular choice in agricultural image classification [10, 12, 17, 46, 272].

Random Forests

Random forests (RFs) are ensemble classifier composed of tree-structured units, in which each classifier casts a single vote towards the final classification [19]. Each tree is produced based on a random sampling of the training set. A random selection of input features can also be considered at each node, in order to determine the split. This results in a greater variety of tree structures compared to the case where all features are considered at every node. The addition of more trees to RFs does not result in overfitting [19], but instead, the results converge to a limiting error value. RFs have been used in a number of studies in weed detection in a range of different crop fields [5, 112, 147].

Extreme Learning Machines

The Extreme Learning Machine (ELM) is a learning algorithm for ANNs proposed by Huang et al. [76, 99] as an alternative to the slow process of gradient descent such as the back propagation algorithm [204] typically used by feedforward networks during training. The ELM works on a single hidden layer feedforward network (SLFN) with a fully connected hidden layer, whose weights are randomly assigned and fixed (i.e., do not change during training). The weights of the output layer are analytically determined during training, using the Moore-Penrose pseudoinverse, which generalizes the concept of an inverse to include matrices which are non-square or singular. In this thesis, the term ELM will be used to refer to the algorithm itself, as well as to the networks trained using this algorithm.

We start with a hidden layer with randomly assigned weights $\mathbf{W}$ and biases b , and activation function g . Give a set of training data $\mathbf{X}$ with target output $\mathbf{T}$, we compute the outputs of the hidden layer $\mathbf{H}$ where $\mathbf{H}_{ij} = g(\mathbf{W}_j \cdot \mathbf{X}_i + b_j)$, and then formulate the problem of finding the weights of the output layer, β as:

$$\mathbf{H}\beta = \mathbf{T}$$

We can use the Moore-Penrose pseudoinverse, here denoted by $\mathbf{H}^\dagger$ to calculate the least-squares solution to the problem.

$$\hat{\beta} = \mathbf{H}^\dagger \mathbf{T}$$

Assuming that $\mathbf{H}$ has full rank, the Moore-Penrose pseudoinverse can be calculated as³:

$$\mathbf{H}^\dagger = \mathbf{H}^\top (\mathbf{H}\mathbf{H}^\top)^{-1}$$

In practice, ridge regression [156] is commonly used to avoid a singular or ill-conditioned matrix, which may otherwise lead to instabilities in the solution. In this case, $\hat{\beta}$ for

³We assume here that $\mathbf{H}$ has more columns than rows (or is square). If this is not the case, then we can use the alternative formula $\mathbf{H}^\dagger = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top$

$c > 0$ is calculated as:

$$\hat{\beta} = \mathbf{H}^\top \left(\frac{\mathbf{I}}{c} + \mathbf{H}\mathbf{H}^\top \right)^{-1} \mathbf{T}$$

The ELM was compared with a gradient descent back propagation algorithm, as well as an SVM, and found to have a significantly faster training phase (at times 1000 to 2000 times faster), while maintaining comparable accuracy when tested on a range of datasets [76].

Variations of Extreme Learning Machines

Modifications to ELMs typically involve either an optimization of the hidden layer neurons, or a modification of network structure, while preserving the ELM algorithm for calculating final layer weights. In the first case, the hidden layer weights can be tuned to the training data as in [243] and [278]. Alternatively the ELM algorithm can be combined with back propagation [260], or a process can be used to select better than average hidden layer nodes for the network. The latter can be achieved using either pruning, where nodes that are not contributing much to the accuracy of the network are removed [161], or using incremental addition of nodes, where each node is selected from a number of randomly generated candidates [97]. For network structure, some studies [100, 159, 209] have looked at networks where the hidden layer is not fully connected to the input data. Others have considered deep networks, containing either ELM-based Auto-Encoders [111, 240, 246]⁴, or convolutional layers [158, 185]. Still more studies have considered ensembles of ELMs [169, 210, 211], involving multiple ELMs, and a form of voting system to collate results.

Several algorithms have been designed to select suitable hidden layer weights for ELMs, using the training data at hand. The purpose of these algorithms is to improve performance, such as classification accuracy, or alternatively to get comparable results with a hidden layer that is a fraction of the size required using an ELM with randomly generated weights. For example, Tapson et al. [243] designed the computed input weights (CIW) algorithm, where the weights of a neuron are initialized to a normalized linear combination of the training samples for a particular class, with each class contributing a comparable number of neurons to the overall model. Testing of the CIW-ELM on the MNIST dataset [132] demonstrated a significant improvement over standard ELMs, with an accuracy in excess of 98% achieved with a hidden layer size of 8000. Zhu et al. [278] designed a suite of initialization algorithms, summarized in Table 2.2, based on a combination of random inter- or intra-class samples. As with CIW-ELM, the constrained ELM weights initialization shows a marked improvement in performance compared to standard ELMs, with accuracies of up to 97.6% with a network containing 2000 hidden nodes on the MNIST dataset.

⁴An autoencoder is a network which attempts to map the input to itself, and typically involves passing the data through a lower-dimensional coding layer.

Name	Description	Weight Formula	Bias Formula
Sample ELM (SELM)	Draw one sample randomly from dataset	$\mathbf{w} = \frac{\mathbf{x}}{\ \mathbf{x}\ _{L_2}^2}$	$b \in [0, 1]$
Constrained Difference (CDELM)	Draw two samples from different classes. maps $\mathbf{x}$ to 1 and $\mathbf{y}$ to -1	$\mathbf{w} = \frac{2(\mathbf{y}-\mathbf{x})}{\ \mathbf{y}-\mathbf{x}\ _{L_2}^2}$	$b = \frac{(\mathbf{y}+\mathbf{x})^T(\mathbf{y}-\mathbf{x})}{\ \mathbf{y}-\mathbf{x}\ _{L_2}^2}$
Constrained Sum (CSELM)	Draw two samples from the same class	$\mathbf{w} = \frac{\mathbf{x}_1+\mathbf{x}_2}{\ \mathbf{x}_1+\mathbf{x}_2\ _{L_2}^2}$	$b \in [0, 1]$
Random Sum (RSELM)	Draw two samples randomly from dataset	$\mathbf{w} = \frac{\mathbf{x}_1+\mathbf{x}_2}{\ \mathbf{x}_1+\mathbf{x}_2\ _{L_2}^2}$	$b \in [0, 1]$
Constrained Mixed (CMELM)	Use a mixture of CDELM and CSELM weights		

TABLE 2.2: Constrained weights ELM (C-ELM) [278] used to initialize the hidden layer neurons in a SLFN.

Another method for improving the performance of ELMs is to incorporate back propagation into the training algorithm, as described by Yu and Deng [260]. The simplest method proposed involves repeated application of the ELM algorithm to calculate the final layer weights and back propagation to adjust the hidden layer weights. This, however, does not take into account the dependency of the final layer weights on the values of the hidden layer weights. Therefore, they develop some further algorithms, that take this dependency into account, and use the FISTA algorithm [14] to accelerate convergence. This approach allowed for a best classification accuracy of 98.55% on the MNIST dataset for a network with 2048 nodes.

There have been several ELM variants constructed, in which the hidden layer is not fully connected to the input layer. For example, McDonnell et al. [159] constructed a network, called the receptive field ELM (RF-ELM), in which each hidden unit only received input from a randomly sized and located rectangle of the input image. The RF-ELM, when combined with a clever weight initialization method as discussed above, achieved a best result of >99% on the MNIST dataset with a network containing 15,000 hidden layer units. On the other hand, Sadgrove et. al. [209] constructed a Colour Feature ELM (CF-ELM), in which the hidden layer was partitioned into three sections, each of which received feature information extracted from only one of the colour channels. This approach proved to be beneficial in an agricultural setting, improving the accuracy of classification of thistle, quadbike and cattle on a dataset collected in Southeast Australia.

Several studies have utilised an ensemble of ELMs (EN-ELM) in order to improve classification accuracy [63, 169, 210, 211, 266]. Each ELM is initialized and trained separately, either on a subset of the training images [169], or potentially a

combination of training images and feature sets [210]. One study utilised a novel segmentation approach in which an image is divided into small patches, which are mapped one-to-one to a set of classifiers via an equal k-means algorithm [211]. Another option is to use a boosting algorithm such as AdaBoost [62] to improve ensemble results, such as in [63].

With CNNs dominating image classification and segmentation competitions such as ILSVRC [205], it is not surprising that previous work has been taken to incorporate convolutional layers into ELMs, either with [185] or without [158] back propagation. In the case of [185], a “wide” Residual Network (WRN) was trained via back propagation, and then the output layer replaced and weights determined via the ELM algorithm, producing a deep wide ELM (DW-ELM). It was found that the application of the ELM algorithm to the final layer weights after training with back propagation resulted in a modest but significant improvement in accuracy, but only when a sufficient widening factor was employed. For example the DW-ELM achieved an accuracy of 95.18% and 78.72% respectively, on the CIFAR-10 and CIFAR-100 datasets [120], compared to 94.60% and 76.59% for the WRN. Here width refers to the number of filters employed in a convolutional layer.

The standard ELM algorithm cannot be directly applied to MLPs, because at the time of training, the desired output of all but the final layer is undefined. One method of generalizing the ELM algorithm to deep neural networks relies on the use of an ELM autoencoder (ELM-AE). An auto encoder is a network that attempts to map the input to itself, and an ELM-AE is a SLFN auto encoder that is trained using the ELM algorithm [111]. These ELM-AE units can be stacked sequentially to form a MLP, with the final unit mapping the input to the target output of the training samples, and trained using the ELM algorithm. ELM-AEs have been tested on a number of benchmark datasets, attaining an accuracy of 99% [111, 240] on the MNIST dataset [132], and 91.3% [240] on the NORB dataset [130].

2.2.5 Deep Learning - Combining Feature Extraction and Classification

Neural networks trained with back propagation have the ability to learn their own discriminative features [204], which allows the process of machine learning to develop beyond the use of “hand-crafted” features [131]. As stated in Section 2.2.3, in conventional machine learning, we first need to extract relevant features from the image, before passing these features to a classifier. By contrast, in deep learning, the model is an ANN composed of multiple hidden layers, and each hidden layer “learns” its own useful representation, which becomes increasingly more abstract in later layers, before reaching the final layer, which acts as the classifier [133]. In this respect, deep learning automates the process of feature extraction, and the way in which it achieves this is through the use of back propagation.

Convolutional Neural Networks (CNNs) are a popular choice of deep learning

model in the field of CV. The basic building blocks of a simple CNN are convolutional layers, pooling layers and fully connected layers. An example of a simple CNN is provided in Figure 2.2. A convolutional layer is a partially connected layer, in which each neuron only receives input from a small number of neurons of the previous layer, corresponding to a small, normally square patch of the image. Convolutional layers also employ parameter sharing, meaning that a neuron which corresponds to a certain patch of the image will share weights with neurons that correspond to other patches of the image. This greatly reduces the number of free parameters required by the model. The way in which a convolutional layer works is visualized in Figure 2.3. A bank of filters (or kernels) are passed across the image to build up a feature map from the previous layer, which preserves the spatial information. Pooling layers reduce the dimensionality of the internal representation by combining the outputs of neurons that are spatially adjacent, either by selecting the maximum activation from the adjacent neurons, or taking the average. The fully connected layers effectively destroy spatial information, and are therefore located at the end of the network, in order to further abstract the internal representation. The final fully connected layer produces the final classification, and typically has as many neurons as there are classes to distinguish.

Although CNNs were demonstrated to be effective at classifying handwritten digits in the 1990s [131], there were several factors which made it difficult to train deeper CNNs, being composed of many hidden layers, which were required for more complicated CV problems. These factors included the question of how the layer weights should be initialized [73], which activation functions allowed for efficient training [168], the development of GPUs with sufficient processing power to train more complex networks, and techniques to ameliorate the issue of overfitting.

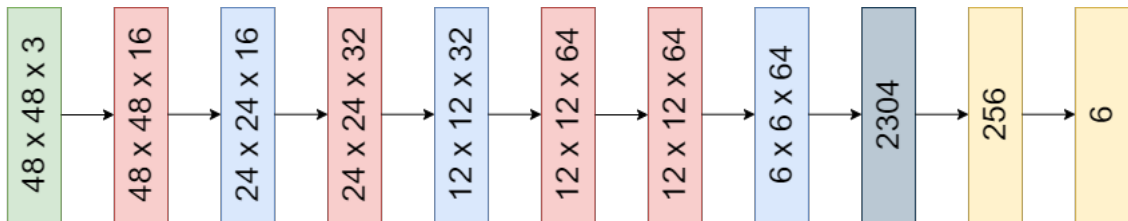


FIGURE 2.2: An example CNN which takes 48×48 pixel RGB images as input, and outputs a vector of 6 values, with each value corresponding to a different class. The input is green, convolutional layers are red, pooling layers are blue, flattening layers are grey, and fully connected layers are yellow. The flattening layer flattens the 3-dimensional feature map (or tensor) into a 1-dimensional vector. The dimensions of the outputs of each layer are written inside the layer.

CNNs did not come to the forefront of more complex CV problems until AlexNet [122] won the ImageNet [205] competition in 2012, achieving a top-5 error rate of 15.3% which was 10.8% lower than the second-placed model. Part of the reason why AlexNet was so successful was because it combined multiple techniques to

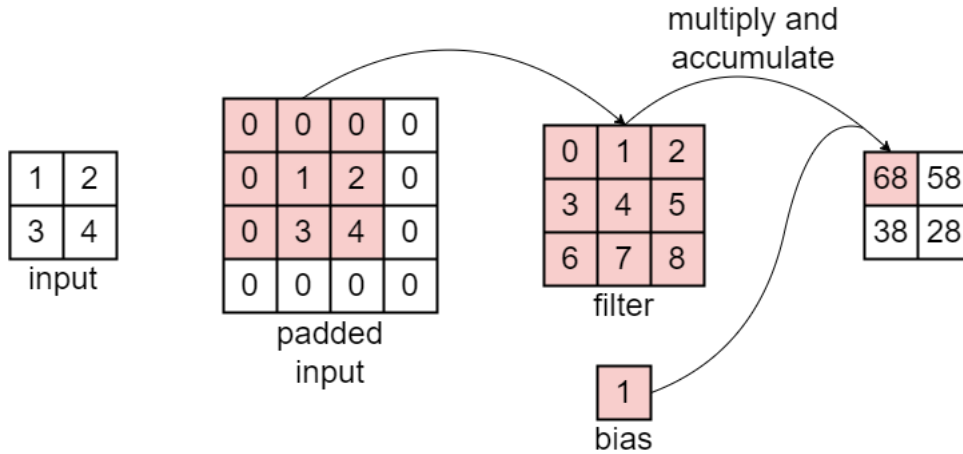


FIGURE 2.3: The convolutional operator for a single filter. The input is often padded with zeros to preserve the spatial dimensions of the output. A single output value (shown in red), is calculated by element-wise multiplication of the filter with a section of the input (red). The elements of the resultant matrix are added together, along with a bias term, producing the final result.

combat issues like overfitting and slow learning, as well as being trained by dividing the model between two GPUs, allowing for a larger model with a greater number of parameters. Although earlier CNNs like LeNet [131] used saturating activation functions, in which the value of the function tends towards a limit for sufficiently large or small input values, such as the hyperbolic tangent, these were known to be linked to slow training, due to the gradient tending towards zero as the activation function approached saturation [131]. For this reason, AlexNet used the ReLU activation function [168], which allowed the network to be trained several times faster than a comparable model with hyperbolic tangent activation [122].

AlexNet contained 5 convolutional and 3 fully connected layers, as well as roughly 60 million parameters. This made it prone to overfitting, even on a large dataset like ImageNet, which contains over a million training samples divided into 1000 classes. The strategies used to prevent overfitting were data augmentation, consisting of random translations, horizontal flips, and crops, as well as dropout. Dropout [228] is a technique where neuron outputs are randomly changed to zero during training with a fixed probability. With dropout, each neuron needs to learn how to function independently of its neighbours, leading to a model that generalizes better over unseen data.

The results of AlexNet were improved upon through the introduction of deeper models, including VGGNets [222], containing between 8 and 16 convolutional layers, in addition to 3 fully connected layers. In contrast to AlexNet, VGGNets use smaller filters of size 3×3 , which are convolved at every pixel (stride of 1). A stack of two (three) such layers has an effective receptive field of 5×5 (7×7), but has fewer parameters, and more non-linearities than a single layer with a larger filter size.

Although the naïve approach to improving model performance is to increase the width and depth of the network, this leads to a number of issues, including an explosion in the number of parameters, resulting in a greater risk of overfitting, as well as an increase in the computational complexity [236]. Furthermore, it was observed that simply adding more layers to a network often resulted in worse (not better) performance, a phenomenon known as the degradation problem [86]. While sparse matrices, in which most entries are set to zero, could in theory be used to reduce the computational burden and the risk of overfitting, in practice it is unclear whether this would result in greater efficiency in GPUs, which are optimized to perform dense calculations quickly [236].

The Inception module [236] was developed as a method to approximate a sparse structure using dense convolutional layers. In this module, as shown in Figure 2.4(a), convolutions with several filter sizes are applied in parallel, in order to capture information at multiple scales, and 1×1 convolutions (also called a pointwise convolution) are used to reduce both the computational complexity of the model as well as the number of parameters. GoogLeNet [236] is an example of an inception network containing 22 layers with parameters, and was found to be significantly more accurate than AlexNet when tested on ImageNet, in spite of the fact that it contained roughly 12 times fewer parameters. The inception architecture has gone through several refinements [237, 238], in which the basic inception module structure has been tweaked to further reduce the size and computational complexity of the model.

As mentioned above, the degradation problem was observed whereby the addition of more layers to a CNN often resulted in a reduction in performance. This is somewhat counter-intuitive, as the additional layers could in principle learn the identity mapping, resulting in a model that performed as well as the original [86]. The residual connection was therefore proposed by He et al. [86, 87] as a method of counteracting this problem, allowing for the training of deeper networks. A residual connection is a shortcut path in neural networks that allows information to bypass one or more layers and be added to the output of those layers. In this way, the skipped layers only need to learn the residual (or difference) between inputs and outputs. This facilitates the learning of a mapping that is close to the identity mapping, thereby overcoming the degradation problem [86]. ResNets utilised a bottleneck design, in which the dimensionality of the feature map is reduced by a pointwise convolution prior to the 3×3 convolution, as can be seen in Figure 2.4(c). This design allows for a reduction in the number of parameters and computational cost while maintaining the depth and modeling capacity of the network.

Inception-ResNet models combine the multi-scale feature extraction capabilities of Inception modules with the improved gradient flow of residual connections, resulting in faster training convergence than pure inception networks [237]. Inception modules have also further been modified into Xception modules (extreme inception), which use depthwise separable convolutions to further reduce computational

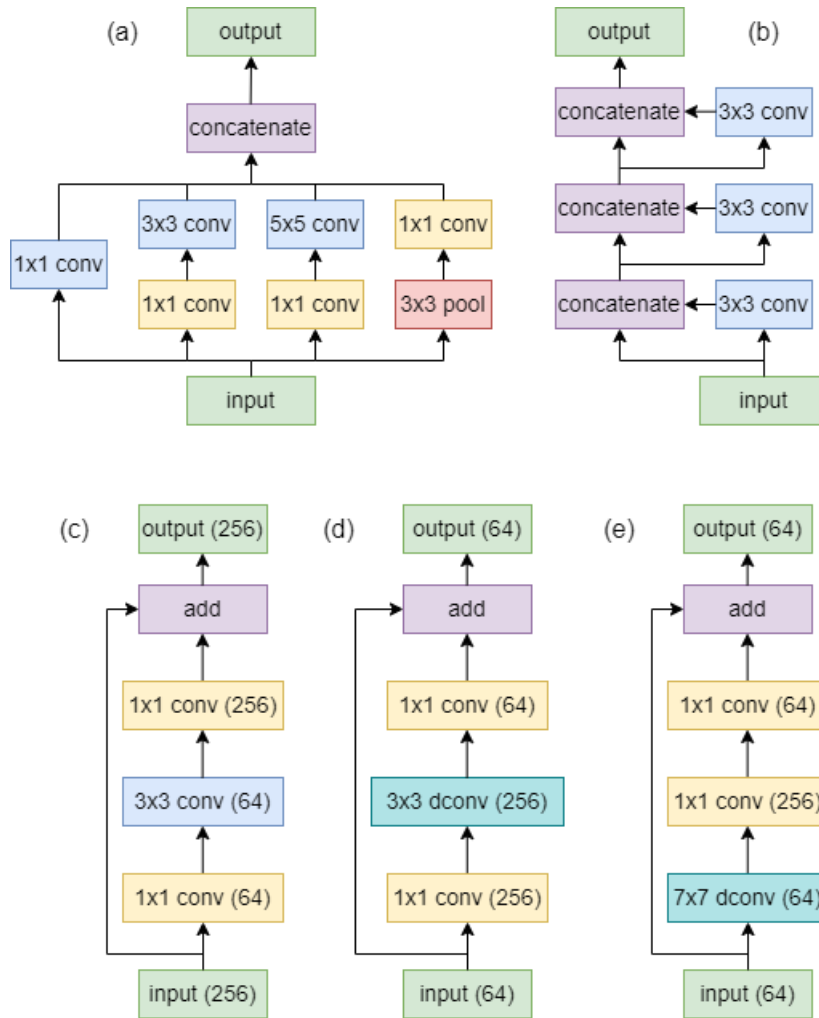


FIGURE 2.4: Different modules used in CNNs: (a) an inception module [236], (b) a dense module [98], (c) a residual module [86], (d) a MobileNetV2 modules [213], (e) a ConvNeXt module [142] Kernel sizes are shown for conv, dconv (depthwise convolution) and pool layers. Filter dimension of output is shown in brackets where required to demonstrate bottleneck and inverted bottleneck designs.

burden [31]. In a depthwise separable convolution, each filter is first convolved with its own convolutional kernel, and then a pointwise convolution is used to project the output onto a new channel space. Xception models perform better when residual connections are also incorporated into the network design [31].

Depthwise separable convolutions are also used in MobileNet [93, 94, 213] blocks. MobileNets are a family of lightweight CNNs, which have been designed for use on mobile devices with limited computational power. MobileNetV2 [213] introduced an inverted bottleneck design for its residual blocks, in which the dimensionality of the feature space is increased prior to the 3×3 convolution, as shown in Figure 2.4(d).

Dense modules as shown in Figure 2.4(b) were introduced as another method for improving the training of deep networks [98]. These modules, as used in DenseNet architectures, connect each layer to every other layer in a feed-forward manner, ensuring maximum information flow between layers. This promotes feature reuse and improves gradient flow during training, and allows for greater performance on complex classification tasks, while simultaneously requiring fewer computational resources than other network designs [98].

One of the difficulties in developing novel network designs is the amount of human effort required to generate and test new structures and designs. Neural Architecture Search Networks (NASNets) were developed in order to reduce the requirement for human intervention during architectural design [280]. Instead, the process is streamlined using reinforcement learning, acting on a search space containing a variety of common network components, in order to determine the best network “unit” for both convolutional and downsampling operations. These basic units are then stacked to form the final network. NASNets can often outperform manually designed networks, even though less human effort has gone into their design and optimization [280].

ConvNeXt is a modern CNN architecture which borrows several transformer-based concepts, such as larger convolutional kernel size, and reduced normalization and activation steps, in order to bridge the gap in performance between pure CNN and transformer networks [142]. Similar to MobileNetV2 [213], ConvNeXt blocks employ an inverted bottleneck design, using a depthwise convolution with a large kernel size, as shown in Figure 2.4(e). This clever design, and modernization of network architecture allows ConvNeXts to outperform Transformer-based networks on complex detection and segmentation tasks, while preserving the simplicity of a standard CNN.

2.2.6 Attention Mechanisms

Attention in deep learning is inspired by human attention and can be understood as the tendency to focus selectively on the most relevant parts of information, rather than processing all input, whether visual, auditory, or otherwise, in full detail [171].

Attention rose to prominence in deep learning with the introduction of the transformer architecture by Vaswani et al. [250], which had the ability to model long-range dependencies more efficiently than previous recurrent structures. Transformers are trained using back propagation, and rely on learning rate optimizers, such as Adam [115] and AdamW [145]

This innovation led to breakthroughs in natural language processing, enabling both representational models like BERT [41] and generative models like GPT [186] to achieve state-of-the-art performance. Since then, attention has been adapted beyond text, leading to advances in computer vision (e.g., Vision Transformers [47]), speech processing [45], and reinforcement learning [179]. The ability of attention to focus on the most relevant parts of input data has made it a core concept across many modern deep learning architectures.

It is often useful to consider attention mechanisms from a data retrieval perspective. Specifically, the concepts of query, key, and value provide a flexible way to compute the relevance of different elements in a set. Each query represents a request for information, while each key describes a piece of available information, and each value contains the actual content to be retrieved. The attention mechanism works by comparing the query to all keys, typically using a similarity measure such as a dot product, to determine how relevant each key is to the query [250]. These relevance scores are then normalized and used to compute a weighted sum of the corresponding values, producing a context-aware output that emphasizes the most relevant information for a given query, while suppressing irrelevant information. This structure allows the model to adjust the focus on different parts of the input, based on the task at hand and broader context [152].

Attention can be split broadly into self-attention and non-self attention. In self-attention, the queries, keys, and values are all derived from the same input, allowing each element to attend to other elements within the same sequence or spatial representation. This is particularly useful for capturing internal relationships, such as dependencies between words in a sentence or regions in an image [250]. In contrast, non-self-attention involves queries coming from one input and keys and values coming from another. A common example is in sequence-to-sequence models, where the decoder generates queries to attend to the encoder's output. This mechanism allows the decoder to selectively and dynamically attend to the most relevant words in the input sequence when generating each word in the output sequence. [11, 152]. Both forms rely on the same underlying mechanism but serve different structural roles in deep learning architectures.

In contrast to Transformer-based architectures, some simpler attention mechanisms are generally not formulated using the query/key/value (QKV) paradigm. The Squeeze-and-Excitation module [95] is a lightweight form of channel-wise attention that uses global average pooling followed by a small multilayer perceptron

(MLP) to assign weights to each feature channel. This allows the network to emphasize channels containing task-relevant information while suppressing less informative ones. The concept was further extended in the Convolutional Block Attention Module (CBAM) [255], which incorporates both channel and spatial attention. CBAM applies these two attention modules sequentially, enabling the network to learn what and where to focus, thereby capturing both feature importance and spatial relevance.

2.2.7 From Classification to Segmentation

Although we are interested to know what is in the image, the question of where in the image the object of interest is located may or may not be of importance. If we do want to know where in the image the object can be found, then we may just want a rough idea (e.g., bottom-left quarter), or we may want a pixel-perfect answer. These different levels of localization form a continuum from classification, in which we simply say whether or not an object of interest exists somewhere in the image, to dense segmentation, in which every pixel in the image is assigned its own class. How much localization we use in our model will depend on how precisely we need to know where the object of interest is.

The Continuum of Localization

As mentioned, for the task of classification, the entire image receives one (or more) classes, depending on which object is (or objects are) present in the image. This method of localization is used in the Deep Weed dataset [174], in which every image is assigned one of nine classes, corresponding to one of the eight weeds of interest used in the study, or the negative class (i.e., no weeds of interest present in the image). Although the authors of this study use images which contain at most one weed species, they design their model so that it is able to detect multiple weeds of interest in an image, should they be present (for example, when testing the model in the field).

Another option for localization is the sliding window approach, where the image is divided into multiple (possibly overlapping) rectangular subimages, and each subimage is passed to the model for classification. This method is used by Zhang et al [272], in which an image of 2048×2048 pixels is divided into six subimages, arranged in two columns and three rows, and each of the subimages classified separately. This choice of subimage size is related to the size of the spraying area of each of two nozzles that are present in their weed control system. Each nozzle covers one of the columns of subimages, and therefore there is no need for a finer localization accuracy. In another study, a sliding window of 130×130 or 250×250 was applied to images captured by UAV in their study on dockleaf growing in a nature reserve in Germany [123].

Moving along the localization scale further, we come to the super-pixel approach, in which pixels are grouped together based on similarity in location and colour. A common technique for the generation of super-pixels is the Simple Linear Iterative Clustering (SLIC) algorithm [2]. Super-pixels differ from the sliding-window approach, because they are typically smaller, may be of variable size, and are not rectangular in shape. This approach is used by dos Santos Ferreira et al. [46] to segment their images captured by a low-flying UAV in a soybean field into small regions, each of which is classified as soybean, grass, broad-leaf weed or soil.

Another approach, which we call patch-based segmentation, seeks to produce a classification for each $N \times N$ pixel patch of the image. This differs from the above approaches in that the patches are small squares, and an area of the image larger than the patch itself is used as input to the network when assigning the patch class. This approach is used by Lottes et al. [146], in which every tenth plant pixel in the horizontal and vertical direction (called a keypoint in their paper) is classified as plant or weed based on an 80×80 pixel patch centred on the pixel. Patch-based segmentation can also be thought of as a downsampled dense segmentation map.

At the bottom of the localization scale is dense segmentation, in which each pixel is assigned its own class. In general, it would be too costly to classify each pixel separately. For this reason, a range of Fully Convolutional Networks, as discussed in Section 2.2.7, are typically used to generate the entire segmentation for the image in a single sweep through the network. Examples of dense segmentation problems for weed localization include the CWFID [85] and GrassClover [224] datasets.

Converting a Classifier into a Segmenter

The CNNs discussed so far have all been classifiers, which assign a single class to an image. These networks are composed of convolutional layers, downsampling layers, and fully connected layers (e.g., Figure 2.2). To convert a CNN from a classifier to a segmenter, we can replace the fully connected layers with convolutional layers, and use upsampling to restore the spatial resolution that was lost in the downsampling layers [143].

Figure 2.5 shows how to convert the classification model VGG-16 into a dense segmentation model, following the procedure of Long et al. [143] in their FCN-32s model. The flattening layer, which converts the $7 \times 7 \times 512$ feature map which is the output of the final downsampling layer, into a 25,088-dimensional vector is removed, and the dense layers are converted into convolutional layers. Importantly, we can take a CNN which has been trained for a classification task, and convert it into a segmentation model, while preserving the pretrained weights. The FCN-32s model, with appropriate padding of the input, can accept arbitrarily-sized images, and output a classification score for every 32×32 pixel square, which can further be upsampled, for example using bilinear interpolation, giving us our dense segmentation map.

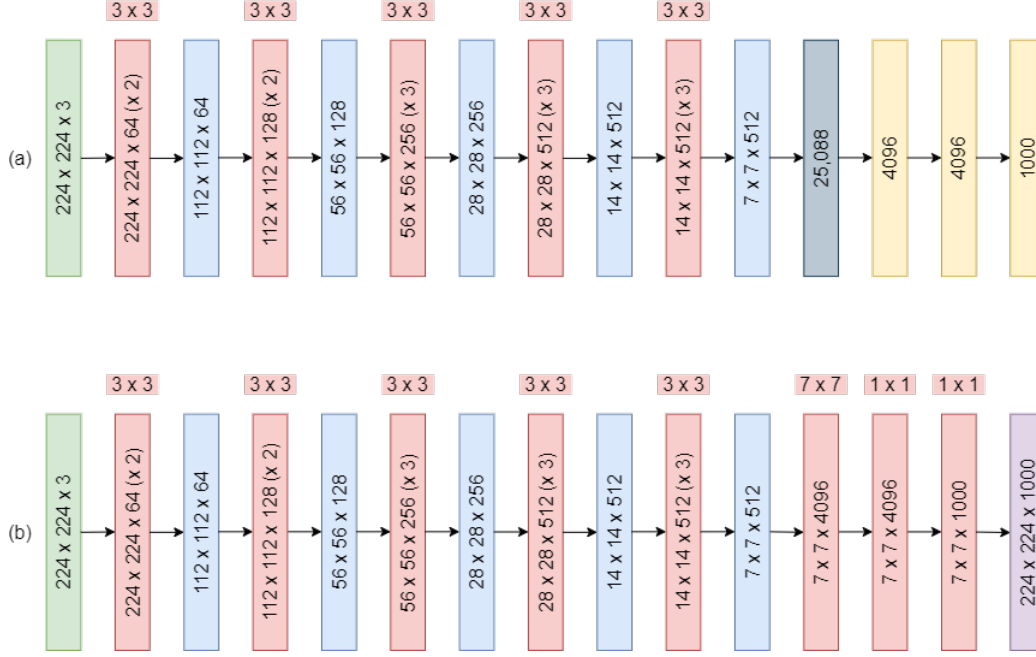


FIGURE 2.5: A classification model, such as VGG-16 (a) [222], can be converted into an FCN-32s segmentation model (b) [143], by removing the flattening layer, replacing the fully connected layers with convolutional layers, and adding an upsampling layer. Layers are coloured green (input), red (convolutional), blue (downsampling), gray (flattening), yellow (dense), and purple (upsampling). For convolutional layers, the filter size is shown above.

The FCN-32s segmentation model discussed above produces a segmentation map that is rather coarse, due to the $32\times$ upsampling required to restore the original resolution of the image. In order to refine the map, most models utilize some kind of skip connection, which passes fine-grained, low level information from lower in the network to later layers. Figure 2.6 provides an overview of the network structure that is common to a lot of fully convolutional networks, including the FCN models of Long et al. [143], U-Nets [200] and SegNets [9]. The network is composed of an encoder arm, composed of multiple convolutional and downsampling stages, which generates an abstract, low resolution representation of the image, and a decoder arm, composed of multiple convolutional and upsampling stages, which is responsible for recapturing fine spacial detail in order to produce a more accurate segmentation result. The way in which spacial awareness is injected back into the decoder arm is through the use of skip connections, which pass forward information from multiple stages of the encoder through to the decoder. Some models, such as nested U-Net [277] and deep layer aggregation (DLA) models [262] have a more complex arrangement of skip connections, in which information is passed up and down the resolution scale multiple times, and across multiple stages.

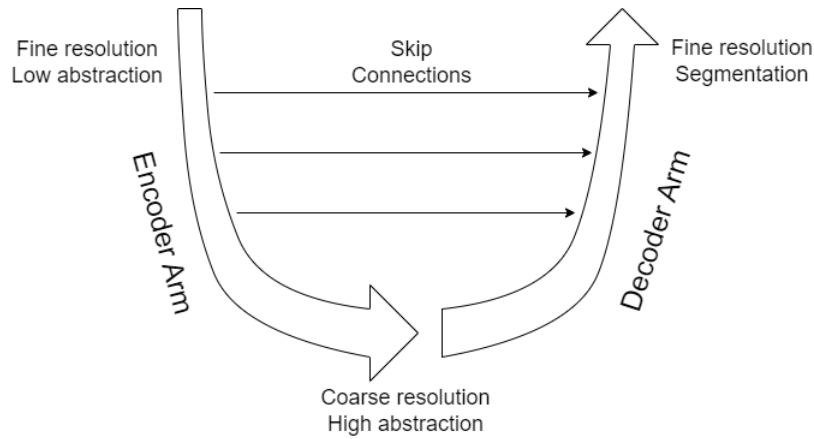


FIGURE 2.6: Some common CNNs used in segmentation problems share a similar macro structure, consisting of an encoder arm, a decoder arm, and skip connections that link the two.

Models Used in Segmentation

As discussed, skip connections can be used to improve segmentation accuracy, by allowing for low level, high resolution features to be considered during the decoding stage. An overview of the kinds of skip connections that are used in common CNN segmenters is given in Figure 2.7. For the FCN-16s and FCN-8s models [143], skips pass high resolution feature maps to the decoder arm for classification at multiple resolutions. These segmentation maps are then upsampled and added together to produce the final segmentation result. For U-Nets [200], the decoder arm is a mirror of the encoder arm, with multiple convolutional layers at each stage. Skip connections concatenate features from the decoder arm with features from the encoder arm along the channels axis, and classification is only carried out in the final stage, when most or all of the input resolution has been restored.

SegNet [9] uses a different approach to skip connections, in which the indices which gave rise to maximum values during the max-pooling operations of the encoder arm are passed to the decoder arm, and use to “unpool” [267] the feature maps during upsampling. This process, shown in Figure 2.8, reduces the number of parameters required in the model, and can improve boundary detection between objects.

The unpooling operation of SegNet handles both the upsampling of information in the decoder arm, as well as the transfer of information from the encoder arm. For other models, upsampling is generally handled either using bilinear interpolation, or transposed convolution [268]. As shown in Figure 2.7, a transposed convolution can be thought of as a convolution with a fractional stride, and can be emulated with a regular convolution by flipping the filter horizontally and vertically, and adding zeros between the rows and columns of the input matrix [268].

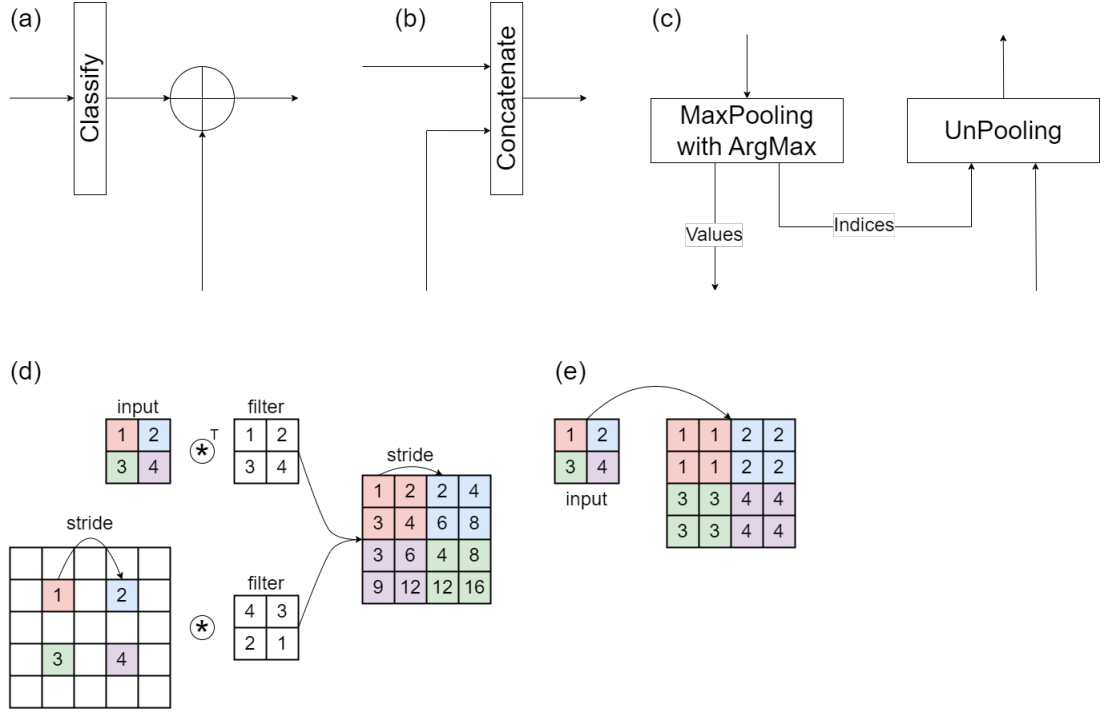


FIGURE 2.7: Examples of skip connections (a-c) and upsampling operations (c-e) used in fully convolutional networks. (a) FCN-16s/FCN-8s [143] skip, (b) U-Net [200] skip, (c) SegNet [9] skip, (d) Transposed convolution [268], (e) upsampling. For (c), the skip connections and upsampling are combined into a single unpooling operation. For (e), nearest interpolation is shown for simplicity. However, bilinear interpolation is more commonly found in the literature.

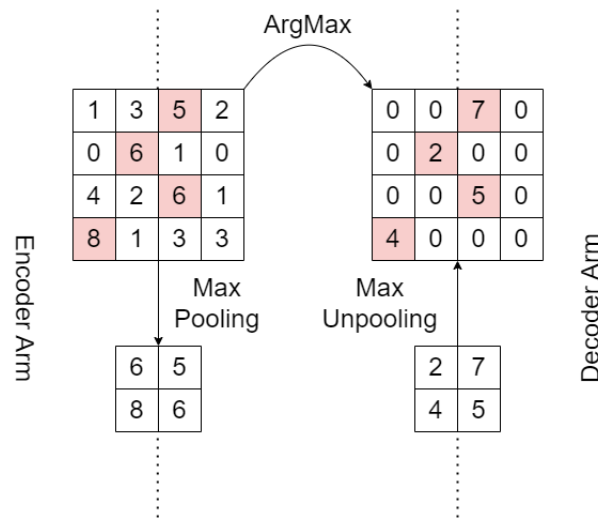


FIGURE 2.8: The unpooling operation of SegNet [9]. The max-pooling indices from the encoder arm are passed to the decoder arm and used during upsampling.

Improving Spatial Context and Receptive Field Size

The Bag of Features approach to image processing [172] ignores the spatial positioning of features during aggregation, allowing for computationally efficient processing at the expense of the loss of spatial awareness. Spatial Pyramid Matching (SPM) [125] was proposed as a method of retaining spatial awareness during aggregation, by considering increasingly finer sections of the image. Rather than considering the aggregate of features across the entire image, a pyramid is constructed showing pooled features across different resolutions. For example a pyramid with three levels would aggregate features over the entire image, over each quadrant (2×2 resolution), and over 16 sub-regions (4×4 resolution).

Zhao et al. [274] proposed a CNN segmentation model called Pyramid Scene Parsing Network (PSPNet) that uses SPM with an average pooling operation to retain contextual awareness during scene parsing. This allows for the separation of similar object categories based on where they are found. For example, a plane is likely to be in the air or on a runway, whereas a car is likely to be on the road or in a garage. It also assists with categories with similar appearances, such as building and skyscraper⁵. In this case, the entire object should be classified as either building or skyscraper. Finally it assists with finding minority objects, such as locating a pillow at the head of a bed, even if the pillow and quilt have a very similar appearance. The architecture of PSPNet is shown in Figure 2.9. As can be seen, PSPNet uses a backbone CNN followed by the injection of spatial context into the feature map through the Pyramid Pooling Module (PPM).

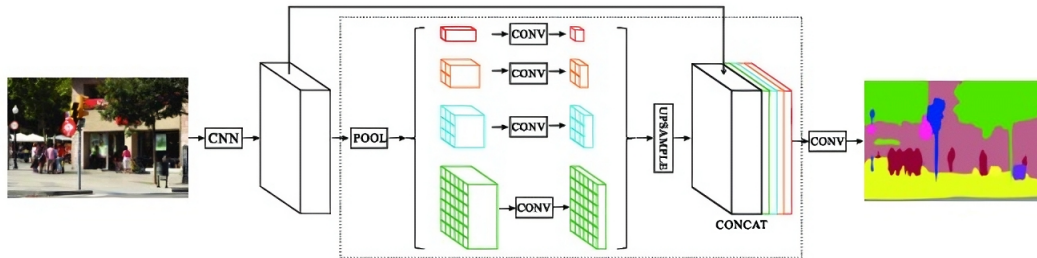


FIGURE 2.9: Architecture of PSPNet (from [274])

Another technique used to improve spatial awareness is the dilated convolution [261] (sometimes called atrous convolution). This is constructed by inserting “holes” or gaps in the convolutional filter, as shown in Figure 2.10, which results in an increased receptive field size. An important consideration for dilated convolutions is the “rate” which is equal to the size of the gaps that are inserted into the convolutional filter. A larger rate will result in a larger receptive field. This technique is used in the DeepLab [27] family of segmentation models, in order to maintain receptive field, while reducing the number of downsampling steps of the backbone model. This assists with dense segmentation, as the output map of the backbone is

⁵For the datasets used in the study by Zhao et al., Skyscraper and Building are distinct classes.

only downsampled by a factor of $8\times$ as compared to $32\times$ for FCN, meaning that there is no requirement for the addition of skip connections. DeepLabs also use a fully connected Conditional Random Field (CRF) as a post-processing step, to refine segmentation results by accounting for long and short range pixel class dependencies. This allows for a more accurate detection of boundaries between objects at the expense of inference time ($4\times$ processing time as reported in the original paper [27]).

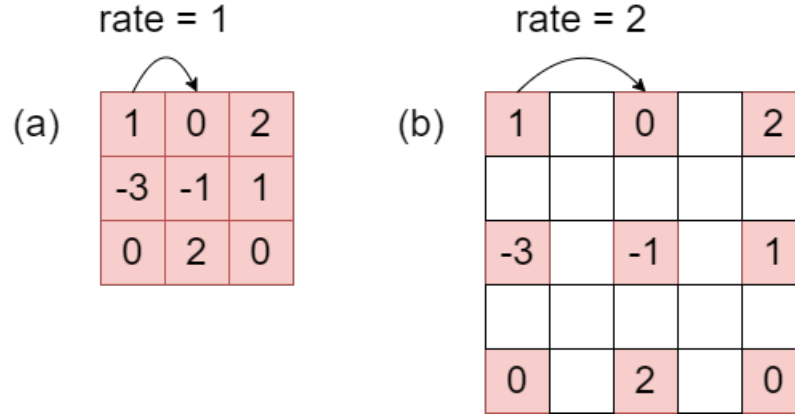


FIGURE 2.10: Dilated convolutional kernels are produced by inserting gaps between the elements of the filter. (a) When the rate is set to 1, a dilated convolution is equivalent to a regular convolution. (b) When the rate is set to 2, the convolution is equivalent to a larger regular convolution, with the gaps filled in with zeros.

DeepLabv2 [26] introduced the atrous spatial pyramid pooling (ASPP) module, in which multiple rates of dilated convolutions are utilized in parallel, in order to capture features from multiple spatial resolutions simultaneously. In DeepLabv3 [29], ASPP modules were further augmented with global context, and different combinations of dilation rates were used to allow for improved performance without the need for CRF post-processing. Finally DeepLabv3+ [28] incorporated elements of the Xception architecture [31], such as depthwise separable convolutions, which reduces computational complexity while improving the network performance.

Detection Models

The models discussed in the preceding sections are designed to generate a dense segmentation map, in which each pixel is assigned a class label. One task that they cannot perform is to distinguish between multiple overlapping objects of the same class. For example, if we consider multiple people standing in a crowd, a segmentation model will produce a single “blob” of class person, without determining where one person ends and the next person begins.

Detection Models, by contrast, are designed to detect the individual objects in an image, and would return one detection, generally defined with bounding box coordinates, for each of the people in the crowd. Detection models include You

Only Look Once (YOLO) [196], Region-based CNN (R-CNN, not to be confused with recurrent CNNs) [71], Faster R-CNN [197], Mask R-CNN [88], and DetectNet ⁶.

R-CNN is a multistage model, consisting of three main modules, being the Region Proposal Module, the CNN, and the classifier. The Region Proposal Module is responsible for producing bounding box “proposals” for regions of the image which may or may not correspond to objects of interest. These regions are then fed through a CNN in order to extract relevant features, which are then passed to the classifier, in order to assign a class to the region (or background). The classifier itself could be implemented as a final layer of the CNN. However, in the original paper, an SVM was used for this purpose [71].

R-CNN is slow, because each region proposal (of which there were 2000 generated per image in the original paper) needs to be passed through the entire CNN. Fast R-CNN [70] solved this issue by the introduction of a Region of Interest (RoI) pooling layer, which sits between the convolutional layers and the fully connected layers of the CNN. The RoI pooling layer generates pooled features of equal size for each region proposal, which are then passed forward through the fully connected layers and the classifier. This means that the convolutional layers need only be used once per image, as opposed to once per region proposal.

Faster R-CNN [197] further improved on Fast R-CNN, by embedding a region proposal network directly into the CNN, allowing the entire process to be completed in a single pass. By comparison, Mask R-CNN [88] builds further on Faster R-CNN by incorporating an additional branch into the network, which simultaneously generates a dense segmentation mask for each object in addition to the bounding box.

The YOLO family of models [195, 196] also consist of detection networks in which the region and classification tasks are performed in a single pass. By treating object detection as a regression problem, YOLO simplifies the task into predicting numerical values for bounding box parameters and class probabilities directly from image features. In comparison to R-CNN, YOLO partitions the image into a grid of regions, and makes predictions for each grid cell. YOLO is well known for its ability to combine speed with accuracy, making it an excellent choice for real-time applications [245].

RetinaNet is a detection model which utilizes a focal loss function, that reduces the weight of well classified examples in order to address the class imbalance between positive and negative detections [137].

2.2.8 Summary

Computer vision refers to the use of artificial intelligence to extract meaningful information from digital images. This is performed in conventional machine learning through the two-stage process of first extracting useful features, such as local binary

⁶<https://developer.nvidia.com/blog/detectnet-deep-neural-network-object-detection-digits/>

patterns (LBPs) or histograms of oriented gradients (HOGs), and then using a classification model, such as SVMs, RFs or ELMs to place the image into one of several classes, which may refer to plant species found in the image.

ELMs are SLFNs, which avoid the time consuming process of back propagation by setting the hidden layer weights randomly and calculating the optimal weights of the final layer in a single step, using the Moore-Penrose pseudoinverse. There are a range of modifications to standard ELMs, including the CF-ELM, in which each of the three colour channels in an image are processed separately in the hidden layer. By comparison C-ELMs and CIW-ELMs compute hidden layer weights using random samples drawn from the training set, and can produce comparable results to standard ELMs with fewer neurons, and hence less computational resources.

Determining which features to use for a particular application can be a time consuming process. Deep learning allows us to skip this step, by combining feature extraction and classification into a single process, which is end-to-end trainable via back propagation. The most common deep learning models used in weed localization are Convolutional Neural Networks (CNNs). There have been a range of CNNs that have been proposed in the past 15 years, including AlexNet and VGG-Nets, which have a simple structure, in which each layer propagates its result to the next layer in the network.

By comparison, models such as Inception and Exception use more complicated structure containing parallel layers, which allow the network to learn more descriptive features, while reducing the number of parameters and Floating Point Operations (FLOPs) of the network. ResNets were developed in order to fix the degradation problem which occurred when training very deep networks. They are designed with identity connections which facilitate the propagation of gradient to earlier layers in the network, and have been used to effectively train very deep CNNs.

Localization tasks vary based on the precision required, and range from classification, in which a single class is assigned to the entire image, and dense segmentation, in which each pixel within the image is assigned its own class. An intermediate task is bounding-box detection, in which the model learns to draw boxes around objects of interest contained within the image.

Classification models can be converted to segmentation models by converting the fully connected layers to convolutional layers, as is the case in FCN-32s [143]. However, during the downsampling stages of classification models, some fine-grained spatial awareness is lost. For this reason, many segmentation models use an encoder-decoder structure, and skip connections, which link earlier stages in the encoder to later stages in the decoder, allowing for the transfer of spatial information prior to segmentation. Examples of this include UNet, nested UNet and SegNet.

Bounding-box detection networks must first generate proposal boxes, where objects of interest may or may not be found, and then determine which class of object, if any, is contained within each of the proposals. Examples of bounding box detection networks include YOLO, R-CNN, and the derivatives thereof.

2.3 Weed Localization in Farmland.

2.3.1 Introduction

There is a considerable body of research literature available on the use of Computer Vision to monitor, locate, and treat weed infestations in agricultural lands, as well as other settings, such as national parks. This section will provide an overview of what has been done to date. Consideration will be given to the purpose of each study, the machine learning techniques that were used, and the conclusions that can be drawn from the study. There are now also a growing number of publicly available datasets, which can be used to train and compare custom models. A selection of datasets, including those that have been particularly influential in the field of weed localization are shown in Table 2.3.

Although the majority of research has been carried out in crop fields, there have also been studies on the localization of weeds in more complex environments such as pastures, and national parks. One example of this is the use of various conventional and deep learning models to detect dockleaf growing in pastures in Europe [17, 39, 119, 226, 272]. Another example is the work of Olsen et al. [174] on the development of a comprehensive dataset of eight weeds of national significance in Australia, as well as background images captured in rangelands of North-East Australia. These studies and others will be discussed in the remainder of this section.

2.3.2 Image Capture

In order to train CV models for weed localization, we first need images to use during training. An important consideration is how the images are to be captured. On the one hand, they could be captured by ground vehicle, unmanned aerial vehicle (UAV) or even satellite. On the other hand, the camera may be a simple RGB camera, or a hyperspectral camera capable of capturing images in multiple bands of the visible and infra-red (IR) spectrum. Another consideration is the resolution of the images, both in terms of the size (in pixels) of the image, as well as the size (in cm or m) of the area of land covered by the image. A key metric is the ground sampling distance (GSD), which is the distance between adjacent pixels, which varies greatly between datasets. For example the CWFID dataset containing images of carrot and weed seedlings captured at ground level has a GSD of 0.1 mm/pixel [85], whereas a study on gamba grass (*Andropogon gayanus*) growing in North Australia uses hyperspectral images (HSI) that were captured by satellite with a GSD ranging from 0.3m to 7.5m depending on the wavelength band [220]. It should also be noted that in cases where real-time processing of images is required, the images will likely be resized down prior to being input into the classifier.

UAV image capture can be performed at a range of altitudes, from as little as 2 metres for one study of weeds growing in wheat fields [271] to 114 metres for another study conducted in pastures [264], which will have a significant effect on

Name and Reference	Environment	Number of Images	Task
CWFID [85]	Carrot (crop)	60	Dense segmentation
Soybean [46]	Soybean (crop)	15,336	Superpixel classification
Plant Seedling [72]	14 species of crop and weed	407	Per-image classification
Chebrolu [4]	Sugar beet (crop)	300 annotated	Dense segmentation
Carrot-Weed [124]	Carrot (crop)	39	Dense segmentation
Sa [206]	Sugar beet	1,026	Dense segmentation
DeepWeeds [174]	8 species of weed (rangeland)	17,509	Per-image classification
GrassClover [224]	Pasture	8,000 synthetic 15 annotated	Dense segmentation Biomass
Oil Radish Growth Dataset	Oil Radish	129	Dense segmentation Biomass + C/N content
OPPD [134]	47 species of weed	7,590	Bounding box
Bender [16]	Brassica (crop)	1,248 annotated	Bounding box and dense segmentation
CropAndWeed [231]	74 species of crop and weed	8,034	Bounding box and dense segmentation
MFWD [69]	2 crop and 28 weed species	94,321	Bounding box, segmentation and instance segmentation
Rai [190]	5 weed species	3975	Bounding box
Turfgrass [229]	Single weeds class	4,203	Bounding box

TABLE 2.3: Example publicly available datasets for weed localization in agricultural land

GSD and field of view. The altitude of the UAV will also depend on the purpose of data collection. If the images are captured simply for surveillance and for the localization of areas with high weed pressure, then a high-flying vehicle which can cover the entire site quickly is preferable. On the other hand, if the UAV is the vehicle responsible for crop or pasture treatment, then it will need to fly at low altitude, so that the treatment is applied precisely, and only where it is needed. For high-flying UAV imagery, the images are often captured with significant overlap so that mosaicking can be performed, in which the images are aligned with one another, and combined into a single overall image of survey site [104, 144].

Another consideration is the angle of the camera. In most, but not all cases, the camera will face directly downwards. However, in one study carried out on dockleaf growing in pastures [17], the camera was pointing forward to an area of pasture directly in front of the vehicle to which it was attached, and the images transformed in order to remove the effect of perspective. The authors note that this approach is only feasible if pasture elements are roughly equal in height, and provided that the camera is not looking too far in front of the vehicle.

Perhaps the most common choice of colour space is RGB, which is used in DeepWeeds [174], GrassClover [224], Soybean [46] and other datasets [72, 166, 244]. By comparison, the CWFID dataset [85] uses images consisting of only two bands, one red and one IR. The reason for this choice is because the plant pixels can easily be separated from background soil pixels through the use of vegetation indices such as Normalized Differential Vegetation Index (NDVI, shown in equation 2.2), due to the fact that the majority of the red light is absorbed by the chlorophyll in the plant, whereas most of the infra-red light is reflected [215]. Most studies using a combination of visible and IR spectra, however, contain far more bands. For example, one study on images of serrated tussock and Chilean needlegrass growing in Southeast Australia contained 11 bands in the near infra-red spectrum, and 16 bands in the visible spectrum [54]. On the other hand, another study on weed species growing in New Zealand pastures used HSI containing a total of 235 separate wavelengths [135].

A further consideration when capturing images is illumination, with the two basic options being natural and artificial. The issue with natural lighting (i.e., sunlight) is that the intensity can vary greatly, depending on weather conditions, and in the case of images captured at ground level, the vehicle's shadow may be present in parts of the image, leading to loss of image quality due to underexposure. For this reason, some experimental frameworks such as the BoniRob robot [85, 203] are fitted with a curtain to block out sunlight, and artificial lighting to provide a consistent illumination for the camera. The disadvantage of this approach is that the curtain may get caught on obstacles such as taller plants, limiting its use to fields without such obstacles (e.g., crop fields at seedling stage). There are also a small number of studies where images have been captured in conditions with limited illumination, such as at night [119, 235].

2.3.3 Weed Datasets used in the Literature

This section will provide an overview of some of the publicly available datasets for weed localization in agricultural land, which are presented in Table 2.3. These datasets vary in such factors as where the images were taken, why the images were taken, how the images were taken, how the images were annotated, the number of weed or crop species represented, and the resolution and number of images.

In the majority of cases, the primary purpose of the datasets were for training models in weed localization. In some cases, however, there were alternative tasks associated with the dataset. For example, the GrassClover dataset [224] primarily focused on the localization of beneficial species of grass and clover in images, including the estimation of species biomass, in order to determine the correct dosage of nitrogen fertilizer to apply. Due to the nitrogen fixing properties of clover, fertilizer dosage is dependent on the quantity of clover present. The Oil Radish Growth Dataset [166] is another example of a dataset with alternative tasks of biomass as well as Carbon and Nitrogen content estimation.

Some of the datasets were collected in a greenhouse setting, including the Plant Seedling Dataset [72], and the Open Plant Phenotyping Dataset [134]. The advantages of this setting is that the researcher can choose the species of interest to include in the dataset, as well as control the number of instances of each class present, to provide a balanced dataset for training. A disadvantage is that the controlled greenhouse environment may not completely reflect conditions that are likely to be found in the field. Although the majority of datasets contained images from glasshouse or crop field settings, there were some that were captured in pastoral or rangeland settings, including the GrassClover dataset [224], and the DeepWeeds dataset [174].

Datasets captured under field conditions may use cameras attached to ground vehicles, or the UAVs, which can be set to fly at a range of different altitudes. In many cases, images are taken in RGB format. However, in some cases, such as for the Chebrolu [4] and CWFID [85] datasets, infrared images are also captured. Chebrolu's dataset also includes LIDAR (light detection and ranging) and laser sensors in order to generate a 3D view of the surroundings for the purpose of collision avoidance.

The datasets discussed in this section use different methods of annotation which have already been covered in Section 2.2.7. These include per image annotation [174, 188], superpixel annotation [46, 176], bounding box annotation [134, 233, 244, 253] and per pixel annotation [4, 72, 85, 124, 153, 166, 224]. Some datasets included multiple types of annotation, such as bounding box and per pixel segmentation [16, 231].

A potential issue with any agricultural dataset is the time taken for a human to produce accurate, high quality annotations. This time is impacted by the kind of annotation used, with dense segmentation maps, in which each pixel is assigned a class, being particularly time-consuming. The complexity of the environment in

which the images are taken can also affect the difficulty of annotating them. In cases where images are of seedlings against a background of soil, annotation may be easier. For example, in the case of the CWFID dataset [85], a plant/soil mask was first generated, and then the plant sections annotated by a human drawing a polygon around regions of vegetation, and labelling as crop or weed, with “hard” sections, containing overlapping crop and weed plants being skipped.

For environments with a more complex arrangements of plants, another alternative is the use of synthetic training images, which increases the number of images available for training, while still maintaining a low cost of human effort, during annotation. This technique was used in the GrassClover dataset [224]. Here individual instances of plants belonging to different species were segmented from images, forming a library of plant components, which were then added to images of soil background with random rotation and scaling adjustments. This process was repeated until the desired level of vegetation cover was achieved. A similar synthetic image generation workflow was adopted in other studies, in which weeds of interest were segmented, and pasted on a background of plain soil, or weed-free crop field images [66, 78, 182, 244].

Species composition is the last thing that will be considered in this section. Some datasets, such as the CWFID [85], Lameski [124] and Chebrolu [4] datasets are concerned with a single species of crop, whereas others contain multiple crop and weed species. For example, the CropAndWeed dataset [231] contains a total of 74 crop and weed species, although due to the under-representation of some classes, a core set of 8 crop and 16 weed species is recommended. The DeepWeeds [174] dataset, by comparison, consists of 17,509 labeled images representing eight significant weed species found in Australian rangelands, as well as many background images of native grasses and broadleaf plants. This dataset was created to support the development of deep learning models for weed classification in real-world conditions, with images coming from a range of different locations with varying environmental conditions.

2.3.4 Common Metrics

A metric is a mathematical formula which determines how well a model achieved the required task, and allows two or more models to be compared with each other. In general, a higher metric score means that the model performed better. There are a range of common metrics used in the CV community, and the choice often depends on the task at hand.

In order to examine the various metrics used in the literature, we first define the confusion matrix, which is a square matrix (or table), which gives the number (or proportion) of samples broken down by their true class and predicted class. The i -th row refers to samples whose true (or ground truth) label is the i -th class, and

the j -th column refers to samples whose predicted label is the j -th class. Therefore, those values which are found along the major diagonal of the matrix are those which have been predicted correctly, and those values which are not have been predicted incorrectly. Figure 2.11 provides example confusion matrices for a binary and a multi-class problem.

For the binary classification problem the term true negative (TN) refers to the number of samples which are negative, and which were classified as negative, the term false negative (FN) refers to the number of samples which are positive, and which were classified as negative. The terms true positive (TP) and false positive (FP) can be defined analogously. For multiclass problems, we will refer to the value in the i -th row and j -th column as M_{ij} , the number of classes as N , the set of classes as $[N] = \{1, \dots, N\}$, and the set of classes excluding i as $[N] \setminus \{i\} = \{1, \dots, i-1, i+1, \dots, N\}$. In this case, the terms TP, FP, FN and TN can be defined in a class-wise fashion as:

$$\begin{aligned}
 TP_i &= M_{ii} \\
 FP_i &= \sum_{j \in [N] \setminus \{i\}} M_{ji} \\
 FN_i &= \sum_{j \in [N] \setminus \{i\}} M_{ij} \\
 TN_i &= \sum_{j, k \in [N] \setminus \{i\}} M_{jk}
 \end{aligned} \tag{2.9}$$

The confusion matrix can be normalized, by dividing each row by the sum of its elements. In this case, M_{ij} refers to the proportion of samples whose true label is i that have been classified as j by the model. For the remainder of this section, we will discuss the multiclass problem, except where otherwise specified. The binary class, however, is analagous, with $N = 2$, and the negative and positive classes corresponding to classes 1 and 2 respectively, although the metrics would commonly only be evaluated for the positive class.

Perhaps the simplest metric is accuracy (A), given in equation 2.10, which refers to the proportion of samples that were classified correctly. This metric is used in the DeepWeeds paper [174]. Accuracy is an acceptable metric to use if the classes are reasonably well balanced, but is not suitable for very unbalanced classes. To see why, consider a binary classification problem, in which the positive class corresponds to only 1% of samples. In this case, a naïve classifier which predicts all samples to be negative would achieve an accuracy of 99%, in spite of the fact that it would be

completely useless for any practical purposes.

$$A = \frac{\sum_{i \in \mathcal{N}} M_{ii}}{\sum_{i,j \in \mathcal{N}} M_{ij}} \quad (2.10)$$

		Predicted	
		Neg	Pos
Actual	Neg	12 (TN)	3 (FP)
	Pos	4 (FN)	8 (TP)

		Predicted			
		1	2	3	4
Actual	1	10	1	0	0
	2	0	8	0	1
	3	2	3	12	0
	4	1	0	2	9

FIGURE 2.11: Example confusion matrices for (a) binary, and (b) multi-class problems. For the binary task, TP (TN) = true positive (negative), and FP (FN) = false positive (negative).

For this reason, studies that utilize unbalanced datasets often use precision (P, also called sensitivity, true positive rate or TPR), recall (R), false positive rate (FPR, also called specificity), intersection over union (IOU), or F_1 score (also called the Dice Coefficient) as a metric. The formulae for these for a given class, i , are given below:

$$\begin{aligned}
 P_i &= \frac{TP_i}{TP_i + FP_i} \\
 R_i &= \frac{TP_i}{TP_i + FN_i} \\
 FPR_i &= \frac{FP_i}{FP_i + TN_i} \\
 IOU_i &= \frac{TP_i}{TP_i + FP_i + FN_i} \\
 F_{1,i} &= \frac{TP_i}{TP_i + \frac{1}{2}(FP_i + FN_i)} \\
 &= 2 \cdot \frac{P_i \cdot R_i}{P_i + R_i}
 \end{aligned} \quad (2.11)$$

Mean and frequency weighted mean versions can also be calculated for any metric f , as below, where p_i is the proportion of images whose ground truth label is class i :

$$\begin{aligned}
 mean(f) &= \frac{1}{N} \sum_{i=1}^N f_i \\
 mean_{fw}(f) &= \sum_{i=1}^N p_i f_i
 \end{aligned} \quad (2.12)$$

Another way of presenting the results is by plotting one metric against another, with two common examples being precision recall (PR) and receiver operating characteristic (ROC) curves. A ROC curve is a plot of the TPR against the FPR. To understand how these work, we consider the binary classification problem. The classification layer of the network will produce a final classification score, which will be between 0 and 1 if the sigmoid activation function is used (which it commonly is for binary problems [22, 39, 114, 273]). Although a threshold value of 0.5 might be the default for determining whether a sample is positive, this threshold can be varied anywhere between 0 and 1.

A low threshold will result in most samples being classified as positive, which will give us a high recall and TPR, whereas a high threshold will result in most samples being classified as negative, giving us a low recall and TPR. On the other hand, a low threshold will result in many negative samples being classified as positive, giving us a low precision, and a high FPR, whereas a high threshold will result in few negative samples being classified as positive, giving us a high precision, and a low FPR. Figure 2.12 gives example PR and ROC curves. Given a PR or ROC curve, we can calculate the area under the curve (AUC), which is also sometimes used as a metric.

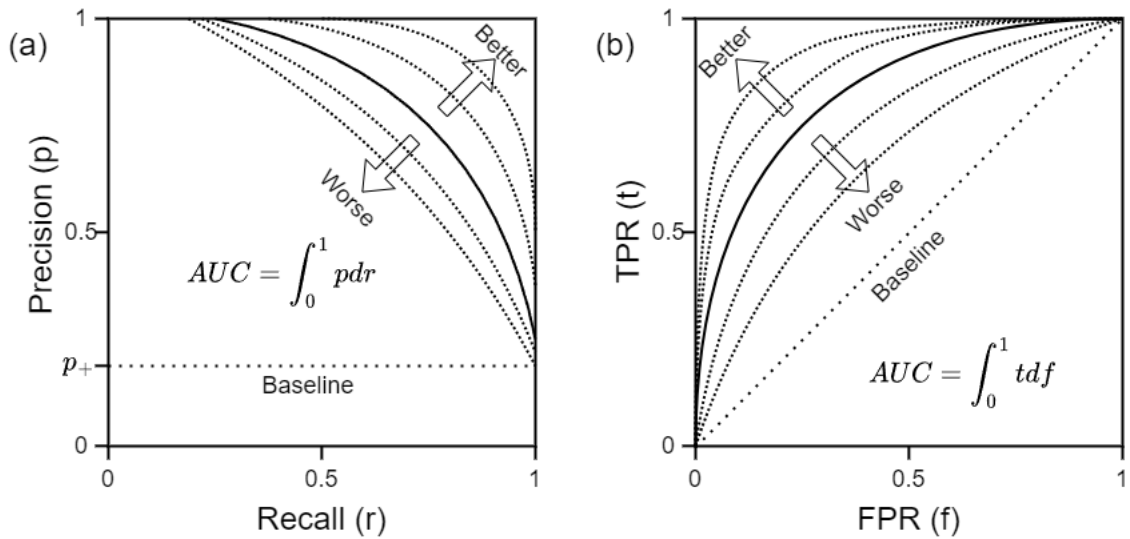


FIGURE 2.12: Example PR (a) and ROC (b) curves. For a PR curve, a baseline model which performed no better than random would achieve a precision equal to the proportion of samples in the positive class (p_+), and a recall equal to the choice of threshold. For the ROC curve, the baseline model would produce a TPR equal to the FPR, and both would be equal to the choice of threshold.

Another metric that is closely related to the AUC of a PR curve is average precision (AP), which is similar to a discrete approximation of the AUC. If we define the $p_{interp}(r)$ function, as the maximum precision for any $\tilde{r} \geq r$, then we can define the AP as the mean value of p_{interp} , calculated at n equally spaced values for recall,

between 0 and 1 inclusive [52]. This is shown below, for $n = 11$:

$$AP = \frac{1}{11} \sum_{r \in [0, 0.1, \dots, 1]} p_{interp}(r) \quad (2.13)$$

$$p_{interp}(r) = \max_{\tilde{r} \in [r, 1]} p(\tilde{r})$$

We can similarly defined the mean AP (mAP) as the mean across classes of the AP metric.

The final consideration when calculating any of the above metrics is what constitutes a sample. This is very much dependent on the kind of localization problem being carried out. For a classification task, a sample will be an entire image. By contrast, for a segmentation problem, a sample may be a single pixel in the case of dense segmentation, or square patch in the case of patch-based segmentation. Alternatively, we might use a Connected Component (CC) based approach, and say that every CC (e.g., individual plant) is a sample. This will produce a similar problem to the bounding-box case, except that the CCs of course do not need to be rectangular.

In the CCs or bounding-box cases, an additional consideration is needed. Clearly it would be too strict for us to expect a model's predicted CC, or bounding box co-ordinated to exactly match the ground truth label, and so, a measure of "closeness" is required when determining whether a detection is a true positive. The measure that is typically used is the IOU, which is defined similarly to the cases above, but for a single instance. Given a ground truth label L and a prediction P , both of which are defined as a set of pixels, we can define the IOU as follows, where *card* is the cardinality function equal to the number of pixels in a set.

$$IOU(L, P) = \frac{card(L \cap P)}{card(L \cup P)} \quad (2.14)$$

This allows us to calculate the AP and mAP for different threshold values of IOU, with a mAP@0.5, for example, being the mAP for an IOU threshold of 0.5. However, the higher the threshold, the closer the prediction needs to be to the ground truth, and therefore, the harder it is for a model to score highly. The COCO detection challenge [138] uses an AP (and mAP) metric which are averaged over multiple IOU values between 0.5 and 0.95, as shown below:

$$A = \{0.5, 0.55, \dots, 0.95\}$$

$$mAP@0.5:0.95 = \frac{1}{10} \sum_{a \in A} mAP@a \quad (2.15)$$

Finally, for studies involving random forests (RFs), Out Of Bag (OOB) accuracy may be used as a measure of how well the model works on the training dataset, and as such is somewhat analogous to validation accuracy. RFs are constructed of multiple decision trees, with each decision tree generated from a random sampling of

the training data. The OOB data refers to all samples which were *not* included in the random sampling, and the OOB accuracy of the decision tree refers to its accuracy on the OOB data. The OOB accuracy of the RF as a whole can then be calculated by aggregating (e.g., using the mean of) the OOB accuracies of its individual decision trees. In cases where OOB accuracy is discussed, the generalization (GE) accuracy refers to the accuracy of the RF on a test dataset, not included in any of the decision trees during training.

2.3.5 Methods Specific to Crop Fields

There has been an abundance of research carried out on the localization of weeds in crop fields, with crops including sugar beet [4, 25, 113, 146, 151, 162, 192, 206, 234], soybean [46, 194], carrot [85, 116, 117, 124, 157, 232], rice [3, 101–103, 112, 114, 153, 258], maize [7, 107, 182, 254], and grains [128, 271, 281]. Crop fields offer advantages over complex environments like pastures, including the position of crop plants in lines, and the ability to easily segment green plants from the brown soil background using vegetation indices. These advantages have been leveraged in a variety of ways, which will be discussed in this section.

Perhaps the most pervasive of techniques used in crop field studies, is taking advantage of the clear colour differences between green plants and background soil. This allows vegetation to be easily segmented from background using a vegetation index and thresholding technique [50, 124]. A common choice for this is NDVI [202] and Otsu thresholding [178]. In the task of distinguishing crop from weed, this simplifies the detection process by reducing the number of pixel which need to be classified [157], with the greatest effect being realized in the case where plants are small, and the majority of pixels belong to the background class, an example of which is the CWFID dataset [85].

The thresholding approach to segmenting vegetation from background can also be used as the basis of an Object-Based Image Analysis (OBIA) process, in which contiguous sections (or connected components) of plant pixels are treated as individual objects, and are segmented from the image for analysis. Classification can be carried out easily in cases where an object corresponds to a single plant (either crop or weed), but is complicated where there is overlap between multiple plants [58]. In this case, a further step is required to break down the mixed object into regions of crop and weed [148].

Lottes et al. [148] use a cascading approach in their OBIA study of weed localization in sugar beet fields using UAV image capture. The first stage is to extract the individual objects, and classify them using conventional machine learning techniques. In the case where the classifier returns a very positive result for either crop or weed, then no further processing is done. In the case where the result is ambiguous, the object is considered to be a mixture of crop and weed, and a patch-based

approach to feature extraction and segmentation is used (here each patch is 10×10 pixels in size).

As mentioned in Section 2.3.2, infrared data can be used as a basis for plant segmentation, due to the high reflectivity of plants to infrared light, causing plants to appear very bright when images are captured in this spectrum. On the other hand, infrared data can be useful in weed localization, even when it is not used as part of an explicit thresholding technique. For example, in a study on a dataset gathered in sugar beet fields using both visible and NIR spectra, Wang et al. [251] found that the segmentation CNN models performed significantly better when NIR data was included in the input, with a pixel-wise mIOU of 78.82%, compared to 72.01% for the RGB colour space. These results could further be improved to 87.13% when image enhancing techniques were used prior to input to the model. Interestingly, these improvements were not realised when OBIA was performed on the same dataset. In this case, it was found that RGB without any image enhancement performed best, with a mean accuracy of 96.06%.

Thresholding and connected component extraction may work well in cases where the plants are small, with few overlapping components, but in cases where crop plants and weeds are more mature, with little to no distinct soil boundaries between them, another approach is required. In the study of dos Santos Ferreira et al. [46] on grass and broadleaf weeds growing in soy bean fields, a superpixel-based segmentation technique is used. In this case each of the superpixels can be isolated and used for feature extraction and classification.

A common process to detect crop rows that is encountered in the literature is the use of Hough transforms [92]. Hough transforms are a CV technique used to detect imperfect shapes in images. This is achieved by considering a parameter space of all potential shapes, and searching for local maxima within that space, with respect to some measure of fitness. For example, if we consider an image of a crop field, then the crop lines can be found by searching for lines which contain the most number of green pixels.

Being able to detect crop lines in this manner allows us to perform self-supervised learning, eliminating the need to manually annotate images, and greatly reducing the amount of human effort required to produce our trained model. For example, Wendel and Underwood [254] working with HSI, used a Hough transform to find the centre of crop lines in a corn field. The training data was automatically generated in the following way. Firstly, CC extraction of vegetation was performed using NDVI and Otsu thresholding. Any vegetation pixel that fell directly on the Hough transform was treated as crop, while any CC which was wholly disconnected from the crop line was treated as entirely composed of weed pixels. When trained on the automatically generated dataset, an SVM classifier achieved a mean F1-score of 0.87, compared to 0.92 for the dataset that was labelled by hand. Interestingly, the reduction in performance when trained on the automatic datasets was caused mainly by a greater number of weed plants being incorrectly labelled as crop.

In another, similar study, Bah et al. [10] used a Hough transform to detect crop lines in fields of beans and spinach, using UAV imagery. Superpixels were calculated using the SLIC algorithm [2], and those which intersected the Hough lines were used to define crop rows. Those which did not intersect the Hough lines, but which were connected to them via regions of vegetation were defined as potential weeds, while those which were disconnected were defined as weeds. When tested using ResNet18 [86], the unsupervised approach yielded a ROC AUC of 88.73% and 94.34% for the bean and spinach fields respectively, as compared to 94.84% and 95.70% using manually labelled data. One possible reason why unsupervised labelling performed worse for the bean field is because there were fewer weeds present, meaning that potential weeds were included in the dataset as weed samples, as compared to the spinach field, in which only superpixels that were wholly disconnected from the crop lines were counted as weeds.

An alternative approach to the use of Hough transforms is a CNN which has been specially designed for the task of detecting crop rows. This is the approach used by Adhikari et al. [3] in their study on wild millet growing in paddy fields. Semantic graphics, in which a sketch of an abstract concept, such as a crop line, is used as the segmentation target, is used in this study. The advantage of this is that such a sketch is easy for a human to produce, even in cases where high weed concentrations confuse the distinction between row and intra-row spaces.

The last technique of note in crop fields is the use of crop signalling technology, as described by Raja et al. [191]. In this study, drinking straws marked with fluorescent dye are placed with tomato and lettuce plants during greenhouse cultivation of seedlings, and transplanted along with the plant into the field. This fluorescent dye can then be visualized through the use of UV illumination, allowing a robotic weeder to easily detect crop plants, and avoid accidental damage to them during weed control.

2.3.6 Methods used in Pastures and Other Complex Environments

Many of the methods that were discussed in Section 2.3.5 are not applicable to pastures and other complex environments, due to the more haphazard arrangement of plants, and the fact that backgrounds may be similar in colour to weeds or other plants of interest. The studies that are discussed in this section will mostly be concerned with the localization of weeds from ground level, or via low-altitude UAV, but some studies on remote sensing, either using high-altitude UAV or satellite image capture, will be discussed.

There are a number of studies that have been carried out in Europe on the localization of dockleaf (*Rumex obtusifolius*) in pastures, including classification [119, 226, 249, 272], patch-based classification [17] and segmentation [39, 123] studies. This is a relevant task, because dockleaf provides limited nutritional value to livestock, and cannot be controlled by blanket spraying. This is because the herbicides that are

effective against dockleaf also kill clover, which is beneficial to the pasture, due to its ability to fix nitrogen in the soil, reducing the need for fertilizer [226]. It is also a challenging task, due to the similarity in colour between weed and background, as well as the high level of occlusion between plants that is observed [36, 212, 224]. In spite of this, it is possible to achieve reasonable accuracies of 73 - 83% on classification of dockleaf and stinging nettle (*Urtica dioica*), even when the problem is reduced to the classification of small image patches, corresponding to 46 - 106 mm squares [17]. Another study on images of grassland containing dockleaf used a pre-trained VGG-16 model, in order to obtain an F1-score of 0.787 when using a sliding window approach [123].

In another study [20], a UNet architecture was employed to segment spiny pigweed within bermudagrass pastures in Florida, using RGB imagery captured by a UAV flying at moderately high altitude. The dataset comprised images from twelve paddocks, which ranged from weed-free to varying levels of infestation. The UNet model successfully detected approximately 80% of the weeds, enabling up to a 76% reduction in herbicide usage and significantly lowering the cost of weed control.

In Australia, research has been carried out on weed localization problems in rangelands using ground based [22, 23, 54, 174, 175], UAV based [80, 104] and satellite imagery [198, 220], pastures [208–211], and conservation areas [212]. Another study used images captured in a greenhouse of Parthenium Weed (*Parthenium hysterophorus* L.), which is a significant weed in both agricultural and non-agricultural lands [36].

The DeepWeeds dataset [174], introduced in Section 2.3.3, was designed to evaluate the ability of models to generalize across diverse field conditions, which is an essential requirement for the success of robotic weed control in agriculture. Two deep learning models, Inception-v3 [238] and ResNet-50 [86] were evaluated on this dataset, achieving classification accuracies of 95.1% and 95.7% respectively. Meanwhile, ResNet-50 achieved an inference time of 53.4 ms on an NVIDIA Jetson TX2, making it suitable for real-time applications.

Another study [22] looked at the feasibility of classifying Harrisia cactus (*Eriocereus martini*) growing in a rangeland environment in South Queensland using ResNet50 [86] and Inception-V3 [238]. The data collection was limited to two locations, during a time of significant drought, meaning that the models produced may not generalize well to different locations and growing conditions. Of interest in this study was the use of class activation maps [275], which allowed the researchers to see which parts of the image contributed the most to a positive or negative result. In their study, ResNet50 performed best, with an accuracy of 98.14%, compared to 96.50% for Inception-v3.

In a study on the localization of Parthenium weed in a greenhouse setting [36], two separate strategies were considered. Parthenium is an invasive species of national significance in Australia. In the first strategy, YOLOv4 [18] acting on RGB images was used as a bounding box detector, achieving an overall accuracy of 95%,

and a classification accuracy of 86% when weed plants were separated into flowering and non-flowering classes. A separate experiment was conducted using hyperspectral imaging (HSI) data comprising 274 spectral bands to perform per-pixel classification of three weed classes at different growth stages (seedling, rosette, and flowering) as well as grass, concrete, and soil classes. The classification was performed using XGBoost [30], an advanced ensemble tree-boosting algorithm. This approach achieved a per-pixel accuracy of 99%.

In another study [54] images of serrated tussock, Chilean needle grass and native grasses were collected in ACT, Australia using hyperspectral cameras. When LBP and HOG features were extracted and classified using an SVM, the researchers achieved an accuracy of 80.9%. Of particular challenge was the classification of Chilean needle grass, which only attained an accuracy of 56.3%, likely due to similar appearance to Serrated Tussock. These features were shown to be superior to Gabor filters, which are a form of wavelet-based textural feature, in this test setting. The Gabor-based classification technique only attained an overall accuracy of 67.3%, with just 43.8% for Chilean needle grass.

Hung et al. [104] conducted experiments on three separate datasets of invasive weed species in Australia, including water hyacinth, tropical soda apple and serrated tussock, which were collected by UAV. It was found that altitude of image capture was crucial for good results, with lower heights ranging from 5 to 10 metres being better. Moreover GSD and classification window size were also shown to be significant factors in the interpretation of results. A sparse autoencoder, trained on images from the CIFAR-10 benchmark dataset [121] was used to generate a bank of filters, which were used as feature extractors. K-means clustering was applied to the pooled feature responses, and a linear classifier used to separate the data into positive and negative classes. In the best case, this method generated F1-scores of 92.9%, 94.3% and 72.2% for serrated tussock, water hyacinth and tropical soda apple, respectively.

In another study [198], very high resolution satellite imagery was utilized to monitor the occurrence of Gamba grass in northern Australia. Gamba grass is an invasive species which has the potential to increase the severity of grass fires. The images were segmented using a fully convolutional network called ResUNet-a [44]. This model is similar to UNet, but uses residual blocks and PSP pooling blocks to assist in the training of very deep models, and to incorporate multi-scale contextual information in feature maps. Based on network complexity and window size, this method achieved a best IOU of 53.5%, indicating that it was at least partially successful in detecting Gamba grass infestation. The authors note that the model was better at identifying larger patches of Gamba grass, which are more important from an environmental and fire safety perspective.

Shendryk et al. [220] also conducted a study on Gamba grass in northern Australia using hyperspectral satellite imagery. In their research, they extracted Grey-Level Co-Occurrence Matrix (GLCM) and spectral features from the imagery and

applied an XGBoost algorithm to classify small image patches into positive or negative classes. When all features were combined, this approach achieved a mean accuracy of 91.2% across both classes, indicating that the model was effective for the detection of Gamba grass in this setting.

2.3.7 Conventional Machine Learning Methods

A range of conventional machine learning methods have been used in weed localization studies in farmlands. Several categories of feature have been used, including morphological, spectral, textural and spatial, which have been introduced in Section 2.2.3. Several conventional classifiers used in weed localization studies, including SVMs, RFs and ANNs have been introduced in Section 2.2.4

One comprehensive comparison study on the use of conventional machine learning techniques for weed localization in pastures was carried out by Binch and Fox [17] on the classification of small patches of images containing background pasture elements, dockleaf (*Rumex*) or stinging nettle (*Urtica*). The study aimed to minimize experimenter bias, with researchers remaining neutral with respect to the methods used. By utilizing a single standardized dataset and avoiding parameter tuning based on the test set, the study ensured a more objective and controlled evaluation of the methods. This study found that the SVM was the best classifier of those considered, and when coupled with LBP features, resulted in an accuracy of 82.88%, which could be slightly improved to 83.40% when incorporated into an ensemble of multiple classifiers.

LBP features were also used by Le et al. [126, 127, 129] in their research on the segmentation of canola, corn and radish seedlings. In one study [127], segmentation prior to feature extraction was employed, in order to remove irrelevant, background noise. Parameter optimization demonstrated that the number of neighbours and radius used for LBP feature extraction, was an important consideration, and that the use of multiple parameter combinations lead to improved results, with a best classification accuracy of 91.85%.

This result was negatively impacted by the difficulty in distinguishing canola and radish plants, which share a similar leaf shape, but have a distinct leaf margin morphology. For this reason, another study was conducted [126], in which leaf contour masks were used to focus on the extraction of features coming from the leaf margin (i.e., the pixels surrounding the interface between plant and background), allowing for an improved accuracy of 98.63%. This method was called k-FLBPCM.

In another study, morphological features were used to classify four weed species growing in corn fields in the USA [180]. Of the features considered, it was found that hollowness, solidity and convex area were the most important. A comparison of a K-Nearest Neighbour (kNN), an SVM and a RF was carried out. There was little difference between these three methods, with best F1-scores ranging from 0.898 for

the SVM to 0.905 for the RF. However, the RF was able to achieve this result using fewer features, making it the best candidate for use in this test case scenario.

Another feature that is commonly used in weed localization studies is the histogram of oriented gradients (HOG) [38], which looks at the orientation of light and dark pixels in localized patches of the image. A rotationally invariant version of HOG features were used by Olsen et al. [175] in their study on the detection of *Lantana* (*Lantana camara*) growing in a complex field environment. Choice of image preprocessing technique, including Gaussian smoothing and gamma correcting, as well as HOG hyperparameter optimization, such as the choice of block normalization, signed vs unsigned orientations, number of orientation bins and block overlaps was carried out on a separate dataset. Scale invariance was achieved by fusing features extracted from windows of varying sizes. Classification was performed in two stages, with the first stage predicting the class of multiscale image windows drawn from regions of interest in the image. In the second stage, a decision algorithm determined whether the target species was present in the entire image by evaluating the counts of windows predicted as positive or negative. Out of a pool of different classifiers, including an artificial neural network (ANN), two SVM variants and linear regression (LR), it was found that the ANN and LR performed best for stages one and two respectively, resulting in an accuracy of 86.07%.

Haralick features, derived from the grey level co-occurrence matrix (GLCM) are another textural feature used in weed localization studies. These features were utilised in combination with LBP and HOG features in a study conducted by Zhang et al. [270] on weeds growing in lettuce fields. Classifiers tested included an SVM, a kNN, and an RF, and a genetic algorithm (GA) was used for hyperparameter optimization. The best results were achieved using an SVM, with an F1 score of 94.98%.

In the preceding paragraphs, varying choice of features and classifiers have been discussed. Another important design choice is which channels to use as input to the feature extractors. A simple grey-scale image contains one channel (the intensity of each pixel), whereas colour images typically contain three channels whose meanings vary based on the chosen color space. Hyperspectral imagery (HSI) involves multiple channels, with each channel corresponding to a separate wavelength band. Channels can be further enhanced with additional information, including textural details and 3D data from sources such as LiDAR.

An example study which incorporated additional textural and 3D information channels was conducted by Yuba et al. [264] on the detection by UAV of *Pennisetum alopecuroides* growing in pastures in Japan. Images were segmented into superpixels using SLIC, and statistical features, including the median, standard deviation, minimum and maximum of each channel across the superpixel, were classified using an RF. Based on studies of two subplots of a field, their models achieved best OOB and GE accuracies of 96.6 - 99.2% and 96.2 - 100.0% respectively, when tested on images captured at 28 m.

One example study using HSI was carried out by Li et al. [135] on the detection

of grass and broadleaf weed species present in New Zealand pastures, and was conducted on images captured in a greenhouse setting. Although the HSI contained 235 separate wavelengths, these were collapsed into three channels using PCA. As with the study presented in the preceding paragraph, the SLIC algorithm was used to segment the image into superpixels, and spectral and spatial features were extracted from each superpixel. Classification was carried out using a number of different models, including SVMs and ANNs, with the ANNs achieving the best accuracies ranging from 81.6% to 89.1%.

A comparison study of various colour spaces, including greyscale, RGB, HSV, and Y'CbCr was conducted by Sadgrove et al. [209] using images of bull thistle (*Cirsium vulgare*, also called spear thistle) captured by a downward facing camera at a height of 2 m. In contrast to most other studies, in which features are extracted from the image, raw pixel values were used as inputs to the classifier, which was a variant of the ELM called the CF-ELM (discussed in Section 2.2.4). The choice of color space was determined to be critical for developing a high-performance model. The Y'CbCr color space achieved an accuracy of 84%, outperforming greyscale by 15% and surpassing the second-best choice, RGB, by 9%.

The preceding study utilized images of fixed resolution, and the use of raw pixel intensities as input to the ELM resulted in a very large number of hidden layer weights, and high numerical complexity. In order to produce a bounding box model which can detect the locations of thistle of varying sizes growing within a larger image, multiple candidate boxes must be checked individually, and the model must perform quickly for each candidate, if it is to be deployed in real time.

For this reason, additional research was conducted [208, 210], in which a summed area table (SAT) was utilized to generate multiple images of a uniform, low resolution, corresponding to bounding boxes of varying sizes, which were then input into the CF-ELM for classification. This study utilized an ensemble approach, called the multiple expert CF-ELM (MEC-ELM), in which each model was trained on a different subset of the training images, and a consensus was reached by voting. This ensemble model performed favourably on the detection of bull thistle growing in pastures, attaining a precision of 95% and recall of 89%.

Continuing with an ensemble approach, Sadgrove et al. [211] went on to develop the Segmented Colour Feature ELM (SCF-ELM). The SCF-ELM breaks down an image into small rectangular patches, which are mapped to different CF-ELMs based on k-means clustering, and classified separately. The mapping ignores the location of the image patch within the overall image, meaning that the SCF-ELM should be invariant to changes in orientation. The classification results are combined using a weighted voting system, with patches that are closer to the centre of the image being given a higher weight. In a comparison study with other conventional machine learning models, including other ELM and SVM variants, the SCF-ELM achieved the best results, with an accuracy of 90.6% for bull thistle, and 89.5% for horehound (*Marrubium vulgare*).

In other research, Zamani and Baleghi [266] considered the task of classifying weeds growing in paddy fields, using a combination of visible light and thermal imagery. The study utilized morphological, spectral, textural and thermal features, extracted from plant instances as a whole. The optimal set of features was determined using a genetic algorithm approach. The best results were achieved using an ELM, with a classification accuracy of 98.08%. One of the limitations of this study was that individual plants were manually extracted from the images by humans, meaning that the presented method was not a completed weed detection pipeline.

2.3.8 Deep Learning Methods

For studies on weed localization using deep learning methods, there are a number of important design choices, including whether to utilize pre-existing or novel models. In the latter case, it is often considered desirable to develop a model with a small number of parameters [1, 114, 157]. If a pre-existing model is to be used, then another consideration is whether to apply transfer learning, by incorporating model weights that have been pre-trained on a large dataset such as ImageNet.

The choice of model will partially be dictated by whether the task at hand is a classification, detection, segmentation or instance segmentation problem. In the case of classification tasks, pre-existing models can be used as is, after replacement of the classification layer. For detection and segmentation tasks, pre-existing models can also form the backbone of the network, with a suitable decoder arm design being left to the researchers conducting the study.

Although a lot of studies utilize pre-existing architectures, either as is, or with minimal modifications, these models have in general been designed with different tasks in mind, such as general object classification or segmentation challenges such as ImageNet [205] and COCO [138], or for use in medical imaging [51]. As such they may not be optimized for agricultural applications, in which the content of the images vary greatly from those originally considered by the model developers. For this reason, some studies introduce novel network structures, in an attempt to improve model performance, either in terms of accuracy, or real-time considerations.

An example of this is the work of Adhikari et al. [3] on segmentation and detection challenges, in which a novel architecture called the extended skip network (ESNet) was compared to U-Net [200], DeepLabV3 [29], FCN-8s [143], and Faster-RCNN [197]. The ESNet, is similar to a downsized U-Net, but includes extended skip connections between the encoder and decoder arms. These extended skips were inspired by the “inception” modules of Szegedy et al. [236], and consist of multi-scale filter banks, which are processed in parallel, followed by further convolutional layers. In addition to this, large convolutional kernels are included at the bottom of the network, effectively increasing the receptive field. When tested on a crop line segmentation challenge, the ESNet attained a mIOU of 62.73%, compared to FCN-8s, which was the second best performing model, with a mIOU of 60.35%.

The ESNet was two to three times slower than the other models, but was still capable of processing 11 FPS, making it suitable for real-time applications.

Banu and Deivalakshmi [13] also modified the skip connections of UNet, through the incorporation of an attention layer, in order to improve the model's ability to focus on and highlight the most important or relevant features of the feature map. In the downsampling stages, max pooling was replaced with wavelet pooling, which employs 2D Haar wavelets to better preserve important image details while reducing spatial resolution. When tested on the CWFID dataset [85], their model, called AWUNet outperformed pre-existing architectures, achieving an IOU score of 94.81%. An ablation study demonstrated that both the wavelet pooling and attention mechanisms were required in order to achieve this result.

Liu et al. [140] also utilized an attention mechanism called the Convolutional Block Attention Module (CBAM) [255] in their study on weeds growing in rice fields in China. The CBAM is a two-stage module which first determines which channels, and then which locations in the feature map to focus on, and can be incorporated into almost any convolutional architecture, such as the residual modules of ResNets [86]. The CBAM, in combination with a semi-supervised labelling approach, in which the model's predictions are used to assign pseudo-labels to unannotated sample, allowed for an improvement in classification accuracy of 16.5 to 19.7%.

Another study [281] utilized a VGG-16 model which was truncated before the fourth pooling layer, resulting in a more shallow network. The reduction in pooling steps, however results in a reduction in receptive field of the network, the effects of which were negated through the use of dilated convolutions [261] in later convolutional layers. The modified VGGNet was slightly faster, both in terms of training time and inference time, compared to VGG-16, and had a comparable accuracy (98.15% vs 98.13% for the unmodified VGG-16). Of the other models considered, ResNet50 was the best, with an accuracy of 98.57%, but had an inference time that was almost twice as long. For the segmentation task, a modified U-Net was proposed, with an encoder arm that was identical to the original U-Net, but a simplified decoder arm, containing fewer layers, and an additional skip connection from the original input image to the last stage of the decoder. This modified U-Net performed better than the original, with an IoU of 88.98% compared to 86.38%, and was also better than FCN and SegNet, which achieved IoU scores of 80.23% and 80.78% respectively.

In some cases, two (or more) separate models are used, either as an ensemble, or with different models being used for different tasks. For example, Skovsen et al. [224] utilized two FCN-8s [143] architectures in their study on the segmentation of pastoral images containing grass, clover and various weed species. In this study, there were two clover species present, and one FCN-8s was responsible for distinguishing between them, with the other model separating the broader classes.

By comparison, Potena et al. [184] utilized a cascade of two light-weight CNNs on their study of weeds growing in sugar beet fields. In this case, the first network,

called sNet, was responsible for segmenting plant pixels from soil pixels, while the second network, called cNet, further classified the plant pixels as either crop or weed. In this study, the network was called once for every pixel to be classified, using a square centred around the pixel as input. Although the use of a conservative NDVI threshold could reduce the number of pixels requiring classification, the cascade was too slow for real-time segmentation of images, and therefore the authors proposed an object-based approach. In this case, connected components of plant pixels were classified by a majority voting on a random sampling of pixels. This OBIA approach led to a mean accuracy of 98% and 97.2% for different network configurations which could segment an image in 2 and 1 second respectively. As with most OBIA approaches, this network would not function well in conditions with significant plant overlap.

A similar approach was used by Moazzam et al. [163] in their study on weeds growing in tobacco fields. Here, a CNN was first used to segment images into plant/soil classes. Those pixels corresponding to the soil class were then set to black, and the second CNN used to further segment the plant pixels into crop/weed classes. In contrast to the study by Potena et al., this study utilised variations of U-Net, in which skip connections are used to inject fine-grained feature information back into the feature map in the upsampling decoder stage. The best results were achieved when a light-weight U-Net was utilized in the first section, and a heavy U-Net with a VGG-16 backbone used in the second section. In this case, a mIOU of 0.92 was achieved, in comparison to 0.79 for the case where segmentation was carried out by a single VGG-16 based U-Net.

In another study [58], a modified U-Net was used to segment plant pixels from the background soil. The RGB images were augmented with several channels corresponding to vegetation indices. OBIA was then applied to each connected component of plant pixels, by extracting a square patch around the object, and classifying with a VGG-16 model pretrained on ImageNet. The model was tested on data coming from sugar beet, carrot and sunflower fields, with a mIOU of 0.80, 0.86 and 0.92 respectively.

Khan et al. [114] used an architecture consisting of four downsized U-Net [200] models, which they named the cascading encoder-decoder (CED-Net), in their study on the semantic segmentation problem. In this case, the network considered input images at two different scales, resulting in a “coarse to fine” segmentation, with two of the models responsible for weed segmentation, while the other two carry out crop segmentation. CED-Net was evaluated against U-Net, SegNet [9], FCN-8s [143], and DeepLabv3 [29] across three publicly available segmentation datasets. CED-Net achieved the highest performance, surpassing the second-best model by an F1 score margin ranging from 0.52% to 2.41%. In a detection dataset of weeds growing in rice fields, the performance of CED-Net was also superior to other models, with a mAP over recall of 49.13%, compared to 42.80% for the second placed model. In addition to outperforming other models, CED-Net had significantly fewer parameters.

However, time data is not presented, so it is not clear whether this resulted in faster training and inference times.

One study that used an ensemble of CNNs was conducted by McCool et al. [157] using the CWFID dataset [85]. The CNNs were light-weight, and used either singly or in groups of 2 to 8, in order to produce a patch-based segmentation of the dataset. As with the study of Potena et al. [184] discussed above, the use of a vegetation index reduced the number of pixels needing to be classified, by easily removing regions of the image containing background soil. The ensemble approach achieved an accuracy of 89.7 to 90.7%. By comparison, Inception-v3 [238] had an accuracy of 93.9%, but was too slow for use in real-time applications.

Using a novel ensemble approach, Asad et al. [8] trained multiple SegNet models on datasets featuring different crop and weed species, and then employed a meta-UNet model to combine their outputs. This meta-model utilizes spatial awareness and multi-scale feature learning to effectively integrate the concatenated predictions from the base models into a refined final output. The meta-UNet model achieves up to a 12% improvement in mIOU for Canola and a 4.5% increase in IOU for Kochia compared to single-task models, demonstrating enhanced segmentation performance under diverse field conditions.

One of the most informative aspects of research involves the comparison of various techniques, which may include different model selections, either predefined or custom-built. In one comparison study [247], VGG [222], ResNet [86], Inception-ResNet [237], MobileNet [94] and NASNet [280], were tested on the Plant Seedling Dataset [72], and a custom dataset, comprising 208,477 images of 21 different weed species. The best performing models were ResNet for the Plant Seedling Dataset, and NASNet for the custom dataset, with F1-scores of 0.956 and 0.964 respectively. When combining the outputs of all five models, these scores could be improved to 0.971 and 0.979 respectively.

Andrea et al. [7] performed a comparison study of LeNet, AlexNet, and the cNet and sNet of Potena et al. [184], described earlier in this section. In this case however, sNet and cNet were used as stand-alone networks, and can therefore be considered as light-weight alternatives to AlexNet. Images of maize or weed were first segmented using ExG to remove the non-plant pixels, and then classified by the networks. Of the networks tested, cNet performed best with an accuracy of 97.3% in spite of the fact that it only contained 1/12th the number of parameters of AlexNet.

Ramirez et al. [192] also compared a range of segmentation models, including U-Net, SegNet and DeepLabv3 using UAV images of sugar beet fields with weeds. The dataset contained images with a total of 7 channels, being taken both from visible and near-infrared spectra, as well as including several composite channels, such as NDVI. In this study, DeepLabv3 performed best, achieving a mean F1-score of 0.84, and mean ROC AUC score of 0.88, which was 0.08 and 0.09 better than the second-placed model. The authors suggest that the dilated convolutions of varying scales in DeepLabv3 allowed it to perform better than the other models. However,

DeepLabv3 was the most computationally expensive model, suggesting that U-Net could be a valid choice in the case where computational resources were limited.

In other research, Hashemi-Beni et al. [84] conducted a comparison study of different segmentation models using the CWFID dataset [85], and a sugarcane dataset that was captured by UAV [165]. The best performing networks for the CWFID dataset were DeepLab-v3+ [28] and FCN-8s [143] at 84.3% and 81.1% accuracy respectively. For the sugarcane dataset, FCN-8s achieved an accuracy of 76.6% followed by U-Net [200] at 74.0%. The authors suggest that the poorer performances on the sugarcane dataset were likely due to the resolution of the images, which had a GSD of 5cm, due to the height at which the images were captured (125 to 200 metres). SegNet [9] was the worst performing network for both datasets, although it is not clear why this was the case.

Kamath et al. [110] considered PSPNet, SegNet and U-Net in their comparison study on the detection of broad-leaf and sedge weeds growing in paddy fields. Images were first segmented to remove background water and soil from green vegetation. It was found that PSPNet and U-Net performed better with a ResNet-50 backbone, whereas SegNet performed better with a VGG-16 backbone. The networks were initialized with weights prelearned on the ADE20K dataset [276], which is a scene-parsing dataset with 27.5k images. PSPNet achieved a mIOU of 0.624 for a four-class dataset (background, paddy, broad-leaf weed and sedge), which was better than SegNet and UNet, with mIOUs of 0.513 and 0.319 respectively.

In other studies, Gupta et al. [77] compared several U-Net models on an eggplant dataset, attaining an F1-score of 96.8% when Inception-ResNetV2 [237] was used as the backbone. Meanwhile, Shahi et al. [216] compared a range of different segmentation models, built with various backbone networks, and found that UNet combined with EfficientNetB0 [239] performed best, with an F1-score of 88.2% when tested on their multi-species weed dataset. EfficientNets are a family of CNNs, in which depth, width and resolution are scaled uniformly to produce a model with the desired computational complexity and efficiency.

For the task of bounding box detection, a comparison study on different faster R-CNN models was conducted by Le et al. [128], using images of weeds growing in barley fields in Western Australia. The study utilized different backbone models for the faster R-CNN, including ResNet-50 [86], ResNet-101 [86], Inception-V2 [238] and Inception ResNet-V2 [237]. The dataset used contained considerable leaf overlap, and a large number of bounding boxes per images, complicating the issue of producing a robust model. The ResNet-V2 backbone performed best with a mAP@0.5:0.95 of 0.289, and an mAP@0.5 of 0.555. The model was also suitable for real-time detection, with an inference time of 0.38 seconds per image.

By comparison Zhuang et al. [279] compared a range of both bounding box detection and classification models in their study on Florida pusley growing in pastures. Although the detection models failed to achieve good results, with F1 scores ≤ 0.54 at an IOU threshold of 0.5, the classification networks performed well, with

F1 scores ≥ 0.97 .

A number of recent studies have explored the performance of bounding box detection models for weed localization, often evaluating different YOLO variants, RetinaNet, and Faster R-CNN. For instance, Almalky and Ahmed [6] investigated the detection of various growth stages of forking larkspur and reported the best performance from a ResNet-101-based RetinaNet, achieving a mAP@0.5:0.95 of 49.99%. Similarly, Fatima et al. [57] compared YOLOv5 and SSD-ResNet50 on weed detection in okra and gourd fields, finding that YOLOv5 significantly outperformed SSD with a mAP@0.5 of 0.88 versus 0.53.

Broad comparative analyses have also emerged. Dang et al. [40] created a dataset containing 12 weed species in cotton fields and evaluated 25 different YOLO models, with mAP@0.5:0.95 performance ranging from 68.18% for YOLOv3-tiny to 89.72% for Scaled-YOLOv4. In another large-scale comparison, Rahman et al. [187] benchmarked 13 pretrained detectors on a dataset of weeds growing in cotton fields, identifying RetinaNet [137] as the most accurate (mAP@0.5 of 80.0%) and YOLOv5n [108] as the fastest (17 ms inference time with mAP@0.5 of 76.6%). Gallo et al. [64] compared the performance of YOLOv7 [252] and other YOLO models across multiple datasets, achieving mAP@0.5 values between 48.0 and 67.5%. Likewise, Sportelli et al. [227] compared several YOLO models on the detection of weeds in turfgrass, finding that YOLOv8l [109] achieved outstanding performance with a mAP@0.5 of 0.9795 and mAP@0.5:0.95 of 0.8123.

Further examining the boundaries of training and model tuning, Zoubek et al. [282] systematically evaluated 39 configurations of YOLOv5, YOLOR, and YOLOv7 under various training setups, ultimately identifying YOLOv5-x (batch size 4, learning rate 0.01) as the top performer. Similarly, Zhao et al. [24] introduced LettWd-YOLOv8l, an enhanced YOLOv8-based detector incorporating Global Attention Mechanism [141] and Coordinate Attention [91]. Their model achieved top-tier results, including an F1-score of 99.5%, outperforming not only standard YOLOv8 variants but also the more recent YOLOv10, both in accuracy and robustness.

Another option when studying weed detection is to train models on a composite dataset with images coming from a range of different sources. One example of this is the work of Hasan et al. [83], in which four different datasets, including the DeepWeeds [174], and Soybean datasets [46] were combined into a single big dataset, containing 20 different crop and weed species. To circumvent the issue of an unbalanced dataset, with training images per class ranging from 32 to 4425, data augmentation was applied to minority classes, resulting in the same number of training images for each class. A range of different CNN models, including VGG-16, ResNet-50, Inception-V3, Inception-ResNet-v2 and MobileNet-v2 were tested on the combined dataset, with ResNet-50 performing best, with an accuracy 98.06%. However, there was very little difference in performance between models, and all of them achieved accuracies of >96%.

One of the issues with constructing a composite dataset is that an individual

species may only be present in images coming from a single source, and the network could learn to recognise irrelevant features from the images, such as illumination source or background soil colour and texture, and associate that with the species. For instance, in the study mentioned in the previous paragraph, the Soybean dataset included image patches that were segmented using the superpixel method, with the superpixels bordered by black pixels. In an attempt to ameliorate such issues, one study [48], which utilized a composite dataset of six smaller datasets containing a total of 22 different plant species, removed all non-plant pixels from the images prior to input into the network. Using a custom, light-weight CNN, the authors achieved a test accuracy of 86.2%.

The majority of studies discussed in this section have used cross-entropy loss as the objective function during training, and while this may be considered to be a “default” option in many computer vision studies, there are situations in which this loss function may not perform as favourably as other choices. An example of this is in the case of very unbalanced datasets, in which one class (often the negative class) is over-represented in the training dataset. One option is to incorporate a separate weighting for different classes in the cross-entropy loss. This works by allocating a higher weight to samples coming from a class with fewer training examples, and was the approach taken in one of the studies by Skovsen et al. [225] on the GrassClover dataset.

Another option is to utilize an entirely different loss function in the training process. This is the approach taken by Schori et al. [39] in their study on the segmentation of dock (*Rumex* spp.) in pastures. The network was a modified U-Net architecture, using a VGG-19 encoder arm, and a lightweight decoder arm with fewer filters per convolutional layer. The custom loss function, which is based on the Dice coefficient, is provided in equation 2.16, and was combined with the binary cross-entropy loss, which has been defined previously in equation 2.6, to produce the objective function. Although the authors do not explain their choice of loss function, it was likely used to combat the effect of an unbalanced dataset, with the majority of pixels being in the negative class, in this case.

$$L_{Dice} = -\frac{1}{N} \cdot \frac{2 \cdot \sum_{i=1}^N \hat{Y}_i \cdot Y_i}{\sum_{i=1}^N \hat{Y}_i + Y_i} \quad (2.16)$$

$$L = L_{Dice} + L_{BCE}$$

This model achieved an F1-score (averaged over all validation images) of 0.84 ± 0.12 . The authors also demonstrated that fewer dock patches were missed when a larger number of images were included in the training set.

A similar loss function was used by Rist et al. [198] in their study on satellite

imagery of Gamba grass in Northern Australia. Gamba grass is an introduced, invasive species which may increase the intensity of bushfires. The authors use a model called ResUNet-a [44], which is based on U-Net, but with residual connections and Pyramid scene parsing pooling layers. The authors demonstrate that this method is effective at identifying larger patches of Gamba grass, with a best IoU score of 0.535 in the binary segmentation task.

An alternative to using a custom loss function is to introduce a loss layer at an internal point within the network, which is referred to as deep supervision. This technique is used in a variant of U-Net called U-Net++ [277], which has been used in weed localization studies [56]. Deep supervision was also utilized by Picon et al. [182] in their study of grass and broadleaf weeds growing in maize fields in Central and Western Europe. The model used was a PSPNet with a ResNet50 backbone. The PSPNet was modified by incorporating a classification branch after feature extraction to generate a classification score for each region of the image, which is equivalent to patch-based segmentation, discussed in Section 2.2.7. The dual-purpose classification/segmentation model performed better than the base PSPNet, achieving an F1-score of 47.97% compared to 45.33% for the base PSPNet.

Valente et al. [249] used AlexNet pretrained on ImageNet to classify images that were captured by an UAV at an elevation of 10m of dock growing in grassland. When trained on images in which the dock plants were centred, the model achieved a validation F1-score of 95.4%. However, this decreased drastically to 74.1% and 35.6% when the validation set contained images in which the dock was off centre, or partially cut off respectively. These validation scores could be improved respectively to 89.9% and 87.5% when the training set also contained off-centre or cut-off images. In the case where a sliding window approach with sufficient overlap between frames is used, however, it is likely that the model trained solely on centred dock plants would be sufficient.

Sharpe et al. [218] performed a comparison study of VGG-16, GoogLeNet, and DetectNet pretrained on ImageNet in their study on Carolina Geranium growing in strawberry fields. Several datasets were captured during this study, and notably validation sets were taken at different locations to training sets. Although the first two of these models are for classification, DetectNet, as the name suggests, is a detection model, which produces bounding box predictions. Two detection systems were considered for DetectNet, in which the target was chosen to be either canopies (being composed of one or more overlapping plants), or individual leaves. The classification networks both performed poorly at the task, mostly due to the number of false negatives. When large images were split into four subimages, these results improved somewhat, with validation F1-scores of 0.77 and 0.62 respectively for VGG-16 and GoogLeNet. By comparison DetectNet performed far better, in particular when leaf-based detection was considered, achieving a validation F1 score of 0.97.

The same authors performed a study of goosegrass growing in both strawberry

and tomato fields [219]. In this case, YOLOv3-tiny was chosen as the detection network, which was trained on both whole plant, and leaf-blade detection tasks. As with the study discussed in the preceding paragraph, a smaller detection target, being the leaf blade, resulted in better performance, with an F1 score of 0.85 and 0.65 for the strawberry and tomato datasets respectively.

In a similar study, Kong et al. [118] demonstrated that labeling individual leaves resulted in improved performance in cases where there was significant overlap between different species of plants in their study using a dataset of four crop plants at the seedling stage. Using YOLOv5 [108], that was augmented with a transformer mechanism, in order to extract global feature information, the authors obtained a mAP@0.5 of 85.9%.

2.3.9 Hybrid Methods

So far we have looked at conventional machine learning models, such as SVMs and RFs, as well as CNNs. In this section, we will consider hybrid models, which utilize components of both conventional and deep learning methods. In some cases, a CNN may be decapitated, by removing the final layer or layers, and used, instead as a feature extractor, with the output features passed to a conventional classifier. As mentioned in Section 2.2.7, this approach was used for the original R-CNN, with an SVM producing the final classification score for each of the region proposals.

For weed localization, Rasappan et al. [193] utilize a Faster RCNN model for region proposal, and feature extraction, and an ELM to classify the proposed objects into crop or weed classes. The Faster RCNN model extracts features at multiple scales to ensure that small plants are not overlooked due to excessive downsampling of the feature map. Their method is tested on a publicly available dataset [233], and found to offer superior performance compared to a range of other conventional and deep learning methods.

Espejo-Garcia et al. [50] also used a hybrid approach in their study on images collected from multiple tomato and cotton fields across Greece. The authors employed pre-trained models like DenseNet and VGGNet, which were fine-tuned for the specific task of crop and weed identification. To further enhance the model's performance and prevent overfitting, they replaced the final classification layers of these networks with conventional classifiers such as SVMs, resulting in consistently high accuracy and robust results.

Kounalakis et al. [119] use a similar approach on their study of dockleaf growing in pastures. In this case, pre-trained CNNs are used as is, without further fine tuning, making the method suitable for cases where datasets are limited, and fine-tuning is likely to result in significant overfitting.

By comparison, Farooq et al. [55] combined the outputs of several mid and high-level layers from a lightweight CNN model with superpixel-based LBP features, to produce a rich feature representation that was classified using an SVM. The study

demonstrated that this model achieved superior results compared to its individual components, including the CNN, the multi-level CNN fusion, and the superpixel-based LBP method.

In another study [104], a sparse autoencoder was first trained on the CIFAR10 dataset [121] in order to learn useful features. These features were then applied to images of several weed species, and pooling used to reduce the dimensionality of the final feature vector. K-means clustering was then applied to the pooled feature vectors, in order to separate images into weed and non weed classes. When tested on a serrated tussock dataset, the model achieved a best F1-score of 92.9%.

2.3.10 Transfer Learning

Another point to be considered is the use of transfer learning, in which the parameters of a model which have been learned on a separate task are used as the starting point of training for a different task. This approach is often used for CNN models. In the context of transfer learning, parameters are initialized with weights pretrained on an extensive dataset, such as ImageNet [205]. Only those layers which are task-specific need to be initialized randomly. In the simple case of a classification model, this might involve taking a preexisting model (e.g., VGG-16 or ResNet50) trained on ImageNet, removing the final classification layer (with 1000 neurons corresponding to the 1000 classes in ImageNet), and replacing it with a new classification layer with randomly initialized weights (and a number of neurons equal to the number of classes in the task at hand). In more complicated cases, such as is the case with segmentation models, the pretrained model may correspond only to the encoder section of the model, in which case the entire decoder arm would need to be initialized randomly.

One key benefit of using transfer learning is that a model trained on a large, complex dataset (such as ImageNet) learns highly general and effective features, that can lead to superior results when applied to a variety of downstream tasks [217]. In addition, transfer learning can reduce the time required for training in the specialized (downstream) task, and also reduce the number of training samples required to produce a robust model which does not suffer from overfitting [105]. These “transferred” weights can either be fixed during training, or fine-tuned.

In one study on the use of transfer learning either with or without finetuning, VGG-16, ResNet-50, Inception-V3, Inception-ResNet-v2 and MobileNet-v2 were tested on four publicly available datasets [83]. The networks were initialized using weights pretrained on ImageNet, with the final fully connected layers removed and replaced with two fully connected layers containing 1024 neurons each, and a classification layer. In the case of VGG-16, a global average pooling layer was also inserted after the final convolutional layer. Those layers with pretrained weights were either frozen during training (transfer learning approach), or allowed to be trained along with the rest of the network (fine-tuning approach).

Interestingly, across all datasets, the transfer learning approach only worked well for VGG-16, with the other models performing very poorly, unless fine-tuning was allowed. The authors suggest that this may have been caused by overfitting, although it is unclear why this issue should have been ameliorated by allowing the pretrained weights to be fine-tuned, or why it should have been far less of an issue for VGG-16. In addition, the authors also suggest that the generally poor results for transfer learning without fine-tuning may be due to the fact that the images present in the ImageNet dataset are quite different to the images present in crop and weed datasets. The best results for the DeepWeeds [174] dataset were the fine-tuned versions of ResNet-50 and VGG-16, with a testing accuracy of 93.78% and 93.36% respectively.

Smith et al. [226], in their study on the detection of dockleaf in pastures, explored the use of transfer learning, and the ability to train a model given limited application-specific data. In particular, they demonstrated that a Mobilenet [94] model pretrained on ImageNet was able to achieve a validation accuracy of >80% when trained with as few as 25 images, whereas a 26-layer ResNet [86] trained from scratch was only slightly better than a random choice. Given sufficient training samples, however, the ResNet performed better than Transfer learning-based MobileNet. Of note is that the convolutional layers of the MobileNet were frozen during training. It is not clear whether allowing the learning of these layers (instead of just the task-specific fully connected layers), would have allowed the MobileNet to maintain its superiority, even when trained on a large enough sample size. The best accuracy achieved was 95.6% for ResNet, and 93.2% for MobileNet.

Gao et al. [66] utilized a combination of transfer learning and synthetic image generation to train their tiny YOLOv3 model [195] for detection of bindweed growing among sugar beet. Their model attained a mAP@0.5 of 0.829 with an inference time of 6.5ms, making it suitable for most real-time applications. This is in comparison to a mAP@0.5 of 0.751 when training was carried out without synthetic images.

Other examples of transfer learning in the literature include the use of weights trained on ImageNet in the segmentation of weeds growing in rice fields based on images capture by UAV [101]. Suh et al. [234] also used models pretrained on ImageNet in their study on volunteer potatoes growing in sugar beet fields, with a VGG-19 models achieving an accuracy of 98.7%.

2.3.11 Data Augmentation Techniques

One of the problems associated with training CNNs, and in particular those with a large number of parameters, is the tendency of such models to overfit to the training data, especially when this data is limited [174]. One of the techniques used to ameliorate this problem is data augmentation, in which transformations are applied to the training images, in order to generate new training data. A range of different

augmentation techniques have been applied in the field of weed localization studies, including spatial transformations [3, 61, 83, 174, 226], such as image flipping, rotation, translation, zooming and cropping, photometric transformations such as brightness and contrast shifting [3, 61, 174], and the application of noise [83, 214], including Gaussian noise, Gaussian blurring and salt-and-pepper noise. Other techniques include cutouts [42], in which part of the image is removed, and the blending of two (or more) images, to create a new image [265, 269].

In some studies, a single augmentation technique is applied, for example in [273], in which random horizontal and vertical flips are used. Others use just a few spatial transformations such as rotations and cropping [55], cropping and flipping [46], or rotations, flipping and zooming [58]. In some studies, a mixture of spatial and photometric techniques [174, 182], or spatial and blurring techniques [167] are used.

In one study [21], which used a range of spatial augmentation strategies, it was found that performance could be improved by dividing high-resolution images into smaller patches, each of which was treated as a separate training sample, and augmented separately.

In another study [189] on the detection of weed species growing in crop fields, a range of different augmentation techniques were compared, including spatial transformations, cutouts, photometric transformations and the application of noise. For the bounding-box task used in this study, it was found that augmentations applied at the bounding-box level, including flips and rotations yielded the best performance, as opposed to augmentation regimes which were applied to the image as a whole.

In addition to spatial and photometric augmentation techniques, Skovsen et al. [225] incorporated a style-transfer approach in their dataset of grass, clover, and weeds growing in pastures. This technique transformed images captured in dry conditions to mimic those taken during or after rain or morning dew, where water-covered leaves result in a greatly different leaf texture, and appearance.

Although studies in weed localization have considered a range of different augmentation techniques, these have largely been limited to spatial and photometric transformations, the application of noise and blurring, and random cutouts. Other techniques, such as CutMix [265] and MixUp [269], in which two images are blended together, to produce a new training image, have not been considered.

2.3.12 Generalization of Models

One important question to ask when considering a model for use in weed detection is how well the model generalizes to growing conditions not seen during training. These conditions include such things as lighting, background soil and pasture elements, growth stage of plants, and moisture content of the environment, which may result in different plant colours. Some datasets (for example [46, 174]) draw training, testing and validation sets from the same pool of images, meaning that even though the images may have been collected under varying field conditions, the models are

never tested on entirely new conditions. This section will discuss studies in which the test images are taken at a distinct time and/or place to the training images.

In one study [149], data was collected from different sugarbeet fields and at different times in Germany, with different growing conditions, including plant size, and background soil colour and texture. To assist in generalization, image preprocessing is used, including a Gaussian kernel for noise removal, and standardization of colour channels to the range of $[-1, 1]$, with a mean of 0. The authors demonstrate that the use of spatial information, specifically, the arrangement of crops in lines, allows for a model with better generalization capacities. This kind of spatial information, however, cannot be used in pastures, where plants are not arranged in rows.

By contrast, Yu et al. [263] studied the classification of dandelion, ground ivy and spotted spurge growing in perennial rye grass lawns and golf courses in north America, using VGG, AlexNet and GoogleNet. Two separate test sets were utilized in this study which were captured at geographically different locations to the training sets. VGGNet performed best, reaching F1-scores of 0.928 to 0.997 for different test sets and weed species, followed by AlexNet. By comparison, GoogleNet performed very poorly, due to a large number of false positives (low precision), even though the recall was very good. In the same study, the authors used DetectNet in a single species dataset containing dandelion, to achieve F1-scores ≥ 0.9843 .

In another study [126] using two dataset consisting of canola and wild radish grown in a laboratory setting with or without barley, the ability of models to generalize to different plant growth stages was explored. Models considered were VGG-16, Inception-v3, and a novel classifier called k-FLBPCM. The k-FLBPCM uses a combination of LBPs and contour masks as a feature extractor, in order to distinguish between plants with similar leaf morphology. The authors demonstrate that the k-FLBPCM performs better than the CNN models when trained on larger plants, and tested on smaller plants, suggesting that the novel classifier had better scale invariance.

By comparison Moazzam et al. [164], determined the generalization capability of their W-shaped network [163] to across multiple tobacco and sesame fields, containing a diverse range of weeds. Although some degradation of performance was observed when tested on a fields not used during training, their model was still able to achieve a mIOU of 0.92 compared 0.97 when training and testing data came from the same field.

2.3.13 Models with Multiple Outputs

The majority of studies discussed so far in Section 2.3 utilize models which have a single output, which is a classification score or map, showing whether the image contains a weed, and/or where in the image the weeds are located. However, some

studies also consider alternative tasks, such as biomass estimation or nutrient content [166, 224]. In these cases, it is common for the study to have two entirely separate networks, one which classifies or segments the images into weed, crop, soil, and so forth, and the other which attempts to estimate the other tasks. In spite of this, there are a very few studies on network architectures with two outputs. For example, Picon et al. [182] developed a model which produced both a dense segmentation map, and a coarser patch-based classification score, demonstrating that the dual-task network performed better than vanilla segmentation models. In a sense, this model performs the same task (weed localization) twice, but at different scales.

One example where two distinct tasks are considered, is the use of CNNs to simultaneously generate both a plant segmentation map, as well as a stem localization map. These models are useful in cases where alternative treatment options are available, which are either applied to the entire plant, such as selective herbicide spraying of weeds, or application of fertilizer to a crop plant, or only applied to the stem, such as the mechanical removal of weeds using a stamping tool [150]. These models are typically fully convolutional networks, that are composed of a single encoder, with two decoder arms that generate segmentation maps for each of the tasks [150, 273].

The first such model that we are aware of was introduced by Lottes et al. [150] (hereafter Lottes' model) for the task of joint plant and stem detection in sugar beet fields. This model utilized a slim fully convolutional architecture composed of dense blocks, in which new feature channels are added to the feature map in a sequential way using densely connected convolutional layers [98]. The use of a joint encoder for both tasks, not only reduces the number of parameters and FLOPs of the model (as compared to using two separate models), but was also shown to produce superior feature maps, allowing for an improvement in mAP of up to 5.4% and 2.4% for the stem and plant segmentation tasks respectively.

A similar model was introduced by Zhang et al. [273] (hereafter Zhang's model) for use on the segmentation and stem localization of cauliflower growing in fields with heavy weed infestation. As with Lottes' model, Zhang's model contained a single encoder, but had multiple decoder arms for each task, each operating at a different scale, and producing its own segmentation map, which were fused to generate the final output. Adjacent intra-task decoder arms shared feature maps bidirectionally, and inter-task communication from the segmentation task to the stem localization task was used, allowing the stem decoder arms to focus on areas of the image which contained cauliflower plants. This model outperformed Lottes' model on the Cauliflower dataset, in particular with respect to the stem task, with a successful detection rate of 97.8% compared to 91.2% for Lottes' model. Inter-task feature exchange was demonstrated to be critical for high performance, with the successful detection rate dropping to 88.0% when a separate model was used for each task.

2.3.14 Comparison Studies of Conventional and Deep Methods

We have discussed a range of studies in Section 2.3 that use various conventional and deep learning techniques. In this section, we will now have a look at studies which have compared deep learning methods directly with conventional methods, using a range of different testing environments and datasets. These studies have the potential to give us an understanding of which of the choices: deep vs conventional, lead to more favourable outcomes in weed localization studies in agricultural land.

One comprehensive study of various conventional and deep learning techniques was carried out by dos Santos Ferreira et al. [46], in the paper in which they introduced the soybean dataset, which has already been discussed in Section 2.3.3. This dataset contains images captured by UAV in a soybean plantation in Brazil. The images have been segmented using the superpixel technique, with superpixels that largely correspond to one of four classes (soil, soybean, grass weed, and broadleaf weed) included in the dataset. In terms of features, GLCM, HOG, LBP and statistical colour features were included in the study, and RFs, SVMs and AdaBoost-C4.5 [62] classifiers were used, and compared with the deep learning AlexNet architecture. The AdaBoost-C4.5 classifier is an ensemble learning method that combines the C4.5 decision tree algorithm with the AdaBoost boosting technique to iteratively improve the model's accuracy by adjusting the weights of misclassified instances and creating a strong classifier from a series of weaker decision trees. AlexNet achieved an accuracy of 99.5% on the soybean dataset, compared to 96 - 98.2% for the conventional models.

In another comparison study, Huang et al. [103] compared an OBIA approach with deep learning, in order to segment weeds and crop plants in UAV images taken in rice fields in China. For the OBIA approach, k-means segmentation was applied to segment the image into different objects, and LBP features were extracted from these objects and sent to an ANN, SVM, or RF for classification. By comparison, the deep learning approach utilized an FCN architecture with several backbone networks. The Deep learning method performed better in this study, with a mIOU of 80.2% compared to 66.6% for the OBIA method, whilst simultaneously being over 7 times faster, with an inference speed of 326.8 ms.

By contrast, Le et al. [126] compared their contour-enhanced LBP-based method, called k-FLBPCM, which has been discussed in Section 2.3.7 with several CNN models, including VGG-16 and ResNet, on images of canola and radish plants growing in both a laboratory and a field setting. In the field setting, the k-FLBPCM technique was comparable to CNN methods, with accuracies better than all of the CNN models tested, except for VGG-16. When training and testing data were taken from different growth stages, however, the k-FLBPCM method was superior to the CNN models, attaining accuracies of 0.968 - 0.993 compared to 0.625 - 0.947 for the CNN models.

Zhang et al. [272] performed a comparison study of several conventional machine learning methods and a CNN model in their study of dockleaf growing in pastures. For the conventional methods, LBPs were chosen for the feature extraction, and classification was carried out by a number of different methods, including several SVMs and ensemble techniques. Of the conventional methods, the quadratic-SVM performed best, with an accuracy of 0.894, when LBP histograms were extracted from each of the three channels in the RGB images. In comparison, the custom CNN performed far better, attaining an accuracy of 0.969.

As has been mentioned in Section 2.3.9, it is possible to use a CNN as a feature extractor by removing the classification layer, and replacing it with any of the range of conventional classifiers. In a study by Yan et al. [258], an AlexNet CNN, which had been pretrained on a large dataset was used as a feature extractor, and compared with a number of different hand crafted features, on a dataset of images acquired in a paddy field in China. When classification was carried out using an SVM, the CNN features enabled an accuracy of 0.945 as compared to the hand crafted features, which achieved an accuracy of 0.878, even when a fusion of several different features was utilized. This suggested that the CNN-learned features were more expressive and robust than the hand-crafted features.

In other studies, the AlexNet CNN architecture achieved an accuracy of 0.924 compared with 0.862 for a LBP/RF-based method, when tested on UAV images of weeds growing in a crop field of Chinese cabbage [176]. Osorio et al. [177] compared a HOG/SVM based model with YOLOv3 and Mask R-CNN, finding that both CNN-based approaches achieved an F1-score of 0.94, compared to 0.88 for the conventional method, when tested on images captured in lettuce fields. Sarvini et al. [214] conducted a study on a dataset containing several crop and weed species, and using a range of different classification methods. Of the techniques considered, a lightweight CNN performed best, obtaining a validation accuracy of 0.984 compared to 0.916 and 0.920 for an SVM and shallow ANN-based approach respectively.

2.3.15 Summary

In this section, the agricultural weed localization studies available in the literature have been reviewed. These studies can be categorized based on several factors: the method of image capture, the location of image capture, the models used for classification, the specific localization tasks addressed, whether different methods were compared, whether images were collected from multiple locations, and whether the models were designed with multiple outputs for handling multiple tasks simultaneously.

The majority of studies discussed were carried out in crop fields, in which the arrangement of crop plants in rows may assist with the localization tasks, through the use of such techniques as Hough transforms. In addition, if the images were captured in the early stage of crop growth, with both crop and weed plants being

small and with minimal overlap, then a thresholding approach combined with object based image analysis (OBIA) can be used to simplify the task into classifying connected components of plant pixels.

In more complex environments, including pastures and crop fields with high weed pressure, OBIA fails due to the significant overlap between plants. In these cases, per-image classification, superpixel classification and dense segmentation may be used, depending on the level of localization required by the study.

The models used in the literature roughly fall into two categories, being conventional machine learning models, and deep learning models. In recent years, it is likely that deep learning models, and in particular CNNs, have been used for the majority of weed localization studies. Some studies use a hybrid approach, such as by utilizing the convolutional layers of a CNN to extract features, which are then classified using a conventional machine learning model.

There are multiple confounding factors when comparing the results of two different studies, including differences in datasets used, weeds (and crop species) considered, localization task performed, and metrics used. For this reason, studies which compare the efficacy of multiple machine learning models over one or more datasets, are of particular interest. However, even in these cases, different studies may conclude that different models are superior, and it is clear that no one model has been demonstrated to be best for all weed localization use cases.

2.4 Conclusions

Weed localization through the use of computer vision is one of the critical steps in the task of developing a fully automated site specific weed management (SSWM) system. In this system, cameras capture images of the ground, including soil, target weed species, and background crop or pasture plants, and Computer Vision is used to determine where in the image weeds of interest are located. This information can then be used to determine when to activate the herbicide spray nozzles, or other treatment systems, in order to provide the treatment only where required, thereby reducing herbicide use.

Localization studies vary in the precision of localization required, and range from classification tasks, in which a single class is assigned to an entire image, to dense segmentation tasks, in which each pixel in the image is assigned its own class. Intermediate localization techniques include bounding box, superpixel, and patch-based detection.

Computer vision, as a subset of machine learning broadly encompasses two categories, being conventional and deep learning methods. In conventional machine learning, hand-crafted features, including LBP, HOG, GLCM and colour features are extracted from the image, and then classified using a range of different classification algorithms, such as SVMs, RFs and shallow neural networks. By comparison, deep learning, including convolutional neural networks (CNNs), which are a specialized

type of neural network, can automatically learn meaningful feature representations during training, thereby circumventing the need to manually select which features to incorporate in the model. A number of hybrid approaches also exist, which combine aspects of both conventional and deep learning. An example of a hybrid approach is using the early layers of a CNN as a feature extractor, followed by a conventional classifier that operates on these extracted features. In general, CNNs take longer to train, and typically require GPUs in order to be achieve acceptable training and inference times. However, the use of transfer learning, in which a CNN is initialized with weights pretrained on another (typically very general) dataset can reduce the time needed for training, and can also improve the quality of the final model.

Extreme Learning Machines (ELMs) are a type of single hidden layer feedforward neural network (SLFN) in which the hidden layer weights are randomly initialized, and the final layer weights are computed using the Moore-Penrose pseudoinverse. Various versions of ELMs exist, which have been designed to produce superior performance on various test cases, including those where the hidden layer weights are tuned to the training data, those with partially connected hidden layers, and deep networks constructed using stacked ELM-based autoencoders. Additionally, several ensemble-based ELMs have been designed. ELMs are faster to train than conventional neural networks, which require the iterative process of back propagation in order to learn their weights. Although some work has been carried out on the use of ELMs in weed localization studies, their use is very restricted in this space, and more research is required, in particular in cases where precise and accurate location of target plants is required.

In cases where a comparison of different methods has been considered, it is common for deep learning techniques to produce superior results compared to conventional machine learning techniques, although there are exceptions to this rule. In cases where different CNN models have been compared, there is no consistently best model, with some performing better in some test settings, but worse in others.

There are numerous studies on the use of CNNs in weed localization, including those that utilized standard, often pretrained, networks which have been developed for more general classification tasks, including VGGNets, ResNets and Inception, and bounding box detection tasks, such as RCNN, YOLO, and variations thereof. For dense segmentation tasks, several architectures are used, including FCN, UNet, SegNet, and DeepLab models. For bounding box and segmentation models, it is common for a backbone classification model to be used as the encoder, resulting in a model whose earlier layers are already trained to produce features that are very general in nature, and that may be useful for a variety of downstream tasks, including weed localization. This reduces the number of iterations of training required to produce optimized weights.

Based on the literature reviewed in this chapter, it would appear that the majority of work carried out on weed localization studies have focused on crop fields,

while fewer studies have considered pastures. This is in spite of the fact that in Australia, alone, the economic impact of weeds on grazing land production has been estimated to be \$2.4 billion, annually. The environments encountered in crop fields are often simpler than pastures, due to the position of crop plants in neat rows, and the ease with which plants can be segmented from background soil regions using simple vegetation indices. This allows for several techniques which are applicable in crop fields, but not in pastures. These techniques include Hough transforms for the detection of crop rows, and OBIA techniques, in the case where individual plants have minimal overlap, as is often the case in early growth stages. Such techniques cannot be transferred to pastoral environments, as the position of plants is random, and there is likely to be significant overlap between plants, making it difficult to separate different plant due to similarity in colour and texture.

While there are numerous studies on the localization of dockleaf in European pastures, research on the localization of weed species in Australian pastures and other complex environments is less common. The Deep Weed dataset is extensive for species significant in northern Australia but does not include species frequently found in Southeast Australia, such as serrated tussock, horehound, and thistle. Although these species have been studied, the focus has mainly been on conventional machine learning techniques using images from a single location and time. Additionally, these studies have not considered segmentation techniques to accurately pinpoint the locations of weeds.

For building a robust weed detector, it is essential that models can generalize to growing conditions or new locations not included in the training data. However, commonly used datasets, such as Deep Weed and the Soybean dataset, combine all available images from various locations and time points, and then randomly split these into training, validation, and test sets. While some studies have tested models on entirely new data not seen during training, such studies are rare. Therefore, further research is needed to explore the generalization ability of models when faced with new data, including different lighting conditions and other variations.

Another emerging field in weed localization is the use of multi-task models, which have two or more outputs corresponding to different tasks. For example, joint plant and stem detection systems have been developed for early-stage crop fields. These models enable more varied treatment options, such as applying herbicide to the entire plant or using a mechanical stamping tool or jet of steam specifically on the stem, effectively destroying the plant without relying on chemicals.

Overall, there is a substantial body of research into the use of computer vision for the task of weed localization in farmland, and there are a number of comprehensive datasets available to assist with the consideration of different machine learning approaches. However, fewer studies focus on pastures and other complex environments, in spite of the high financial cost of weeds in grazing lands in Australia. This highlights the need for more research in this area, including innovative CNN architectures and novel ELM methods, in order to improve SSWM outcomes, resulting in

better outcomes for farmers and food production.

Chapter 3

Developing an Extreme Learning Machine Based Approach to Weed Segmentation in Pastures

3.1 Introduction

This Paper, titled “Developing an Extreme Learning Machine Based Approach to Weed Segmentation in Pastures” [59], is linked to Research Question 1, which asks whether a shallow network can be used to segment weed species in Australian pastures. It also examines whether such a network can be applied in realtime, as posited in Research Question 3.

The choice of weeds was largely motivated by the target species identified by the CRC, including serrated tussock, thistle species, and horehound, as well as a dataset of Buthurst burr which was included opportunistically, and provided further evidence of the ability of such networks to segment a wide variety of weed species, and growing conditions.

One of the limitations of this study was that most datasets consisted of images taken at the same location and time. This means that Research Question 6, which relates to the generalization of networks to diverse growing conditions not seen during training, was not addressed. Furthermore, because the weed species were tested separately, the question of a multispecies detector, as outlined in Research Question 5 was also not considered.

The images were high resolution (2016×2400 pixels), but the dataset is small, containing between 20 and 141 images based on the weed species in question, making the problem suitable for shallow networks, which in general are less prone to overfitting than deeper networks with more parameters [15].

Based on previous work [17, 46, 54], a number of feature extraction techniques were considered, including local binary patterns [173], histograms of oriented gradients [38], colour features, and combinations thereof.

In terms of real-time applications, feature extraction proved to be the most time-intensive component, with times ranging between 0.080 and 0.848 seconds for a single feature. It should be noted, however, that feature extraction was carried out

on the CPU (and in some cases without multithreading), meaning that it should be possible to reduce these times by a considerable factor with a suitable GPU implementation.

An overview of the datasets used in this study is presented in table 3.1.

Image Resolution	Annotation Level	Dataset	Number of Images	Negative	Positive
2400×2016	16×16 pixel	Bathurst Burr	20	0.73	0.27
		Horehound	46	0.73	0.27
		Thistle	141	0.76	0.24
		Tussock	63	0.66	0.34

TABLE 3.1: Dataset summary including number of images and class representation. Image resolution and annotation level are consistent across all datasets.

3.2 Published Paper

This paper was published in Smart Agricultural Technology, which is ranked by SJR ¹ as a Q1 journal in Agricultural and Biological Sciences (miscellaneous) and Computer Science (miscellaneous) for 2023. As of 7th July 2026, this paper had 11 citations on Google Scholar.

¹<https://www.scimagojr.com/journalsearch.php?q=21101111783&tip=sid&clean=0>



Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Developing an extreme learning machine based approach to weed segmentation in pastures[☆]

Jonathan Ford^{*}, Edmund Sadgrove, David Paul

School of Science and Technology, University of New England, Armidale, Australia

ARTICLE INFO

Editor: Stephen Symons

Keywords:

Extreme learning machine
Semantic segmentation
Weed detection
Computer vision
Precision agriculture
Feature extraction

ABSTRACT

Effective weed management in pastures is critical for maintaining the productivity of grazing land. Autonomous ground vehicles (AGVs) are increasingly being considered for weed localization and treatment in agricultural land. Weeds, however, can be difficult to distinguish from background plants, due to similarities in colour, shape and texture. While deep learning approaches can be used to solve the localization issue, they are computationally expensive, and require a large volume of training images in order to combat overfitting. In this paper we present a novel Extreme Learning Machine based network for segmenting weeds from the background pasture. The proposed method utilizes a combination of LBP, HOG and colour features, and is tested on four small datasets, achieving a high mean Intersection over Union of 87.1, 79.5, 81.6 and 87.6 for Bathurst burr, horehound, thistle and serrated tussock respectively.

1. Introduction

The world population is projected to reach 9.7 billion by 2050 [1]. With the continued growth in population, it is becoming increasingly important to maximize agricultural yield using the resources available, including arable land, chemicals, and manpower. Precision agriculture, driven by the artificial intelligence revolution, is an essential component in the solution to ensuring that world agriculture maintains sufficient output to feed a growing population [44]. One important application of artificial intelligence in agriculture is the use of computer vision to detect and classify weeds in crop fields and pastures [27]. Effective weed detection will allow the selective spraying of undesirable plants, while avoiding damage to other plants in the field, and reducing the amount of herbicide used, which reduces both cost and the potential harmful effects of chemicals on human health and the environment [27].

The majority of weed detection studies found in the literature involve crop fields, including sugarbeet [39,30,29], soy [14,47], lettuce [38], carrot [25,22,32], and grain crops [26,49]. In such an environment, the segmentation of vegetation from background soil and crop stubble can be carried out using a vegetation index, such as Excess

Green (ExG). Such segmented vegetation then needs to be analysed to determine whether it is part of the crop plant, or a weed.

Fewer studies have been carried out on the detection of weeds in pastures. However, there has been considerable research on the task of locating dockleaf in pastures in Europe [5,45,51,24,11], with an accuracy of 83.4% achieved using conventional machine learning [5], and 95.6% [45] to 96.9% [51] for deep learning. It should be noted that the above studies each used different datasets, making it difficult to make a meaningful comparison between them. In Australia, one study looked at the detection of serrated tussock, a weed of national interest, on a dataset collected in farmland in northern NSW, using unsupervised feature learning, k-means clustering and a linear classifier to achieve an accuracy of 92.9% [20]. In another study a fusion of several features was passed to a SVM to classify images of serrated tussock, Chilean needlegrass and native grass clumps from south east Australia, with a best accuracy of 80.9% achieved using a combination of LBP and HOG features [16].

In other work, Olsen et al. [37] developed an extensive dataset comprising images of eight weed species of national significance from North-East Australia, and produced a baseline accuracy of 95.1% and 95.7% using Inception-v3 [46] and ResNet-50 [18] deep learning mod-

[☆] The first author was supported by a Commonwealth of Australia RTP scholarship, with a top-up scholarship from the "Novel Autonomous Robotic Weed Control to Maximise Agricultural Productivity" CRC. The authors would like to thank the NSW Department of Primary Industries and Agent Oriented Software for providing the dataset.

^{*} Corresponding author.

E-mail addresses: jford3@myune.edu.au (J. Ford), esadgro2@une.edu.au (E. Sadgrove), dpaul4@une.edu.au (D. Paul).

<https://doi.org/10.1016/j.atech.2023.100288>

Received 28 March 2023; Received in revised form 13 July 2023; Accepted 14 July 2023

Available online 25 July 2023

2772-3755/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

els respectively. These results were further improved by Hu et al. [19] who used a graph-based deep learning network to achieve an accuracy of 98.1%. On the other hand, Calvert et al. achieved an accuracy of 98.1% on a dataset of *Harrisia cactus* growing in dry Eastern Australian rangelands, using ResNet-50 [8]. They went on to test their robotic spot spraying vehicle, AutoWeed, in the field, achieving a weed knock-down rate of 95.0% to 96.3% compared to a rate of 98.3% that was achieved using manual application of herbicide [9].

In contrast, Sadgrove et al. [40–43] utilised a shallow ANN with a partially connected hidden layer, which was trained using a variation of the Extreme Learning Machine algorithm (ELM, discussed in the Methods section) called the Colour Feature ELM (CF-ELM), to classify images of thistle and horehound. These models were able to obtain a precision of 82% - 98% and recall of 84% - 94% for thistle, and a precision of 84% and recall of 97% for horehound. In other agricultural studies, Maimaitijiang et al. [31] used an ELM to estimate biochemical and biophysical parameters, such as chlorophyll content, nitrogen content and biomass from aerial images, and Aqel et al. [4] use an ELM to classify four different plant leaf diseases.

The ELM has also been used in a variety of computer vision problems aside from weed detection. In one study [2], a multilayer, ELM-based autoencoder was used to segment hyperspectral satellite imagery into land-use categories. Dixit and Hedge [13], also working with hyperspectral satellite imagery, developed an ELM model to classify vegetation cover as tree, shrub or grassland. Dewi et al. [12] used an ELM model to classify leaves of different varieties of Patchouli. Turkoglu and Hanbay [50], similarly used an ELM model to classify leaves of different species of plant, testing their approach on multiple publicly available datasets.

In spite of the studies discussed in the previous paragraphs, the use of ELMs for weed classification in agriculture remains largely unexplored, and those studies that have been done have avoided the more complex problem of segmentation, in which the specific location of weeds within an image needs to be determined. In order to explore this space further, we designed and implemented an ELM-based model which inputs hand crafted image features, and outputs a fine-grained weed heat-map. So that the model can handle large, and arbitrarily sized images, the model is passed, like a convolutional filter, across the feature cube extracted from the image. This generates multiple heat-maps for different image regions, which are then combined into a single segmentation result for the entire image. Using the ELM as a convolutional filter results in a significant reduction in the computational cost of feature extraction, as the extraction need only be completed in a single pass (for the entire image), as opposed to separately for multiple, overlapping frames, as would be required in the sliding window approach. This model has been developed to be part of the detect-and-spray pipeline of an autonomous ground vehicle, allowing site-specific management of weeds in pastures, thereby reducing herbicide use and the manpower required to control weeds.

The main contributions of this paper are:

- 1) Use of a novel ELM-based network to produce a fine-grained segmentation of several datasets containing weeds of interest in south-east Australian pastures.
- 2) Presentation of the results of a rigorous comparison of different LBP and HOG hyperparameters, used on several weed datasets collected from southeast Australia.

The remainder of the paper will be organised as follows. Section 2 provides a brief review of related research. Section 3 describes the approach used in this study. Section 4 gives the results of the study. Section 5 provides an explanation of how this study fits into the broader research landscape. Finally section 6 provides some possible future directions leading on from this work.

2. Related work

In this section, we will discuss previous studies on weed detection in crop-lands and pastures, focusing on studies which have used Local Binary Patterns (LBPs), Histogram of Oriented Gradients (HOG) and colour features. We will also present a review of studies involving the fine-grained segmentation of pictures into regions of crops and weeds.

2.1. Feature extraction

LBPs have been at the core of many conventional classifiers used for the classification of weeds in crop fields [28,14,35,26] and pastures [5,51,16]. While some studies either do not specify their choice of hyperparameters or use default values,¹ there are a few cases where a more extensive comparison of different LBP descriptors has been carried out, as summarized in Table 1. In one study [30], the LBP features were extracted from multiple colour bands including a vegetation index (NDVI), and the gradient and double gradient of each band was also considered. In another study [16], LBP features were extracted from a total of 27 bands from both the near infrared and visible spectrum, and concatenated to form the final feature vector. On the other hand, two studies considered multiple combinations of radius and number of neighbours in their analysis [5,35].

HOG features are also commonly used in plant classification algorithms [38,14,16,23]. The above cited papers, however, do not specify their choice of hyperparameters for HOG features, such as the number of orientation bins, or whether signed or unsigned orientation is used. Based on the original HOG paper of Dalal and Triggs [10], performance of HOG features for pedestrian identification improves as the number of orientation bins is added, up to a total of roughly 9 bins for unsigned orientation, however the optimal number of bins for a weed classification problem may be different to this.

A range of colour features has been used in previous studies on weed classification and segmentation in agriculture. These features include the colour histogram [3], as well as statistical features such as the mean, standard deviation, minimum, maximum, range, median, skewness, kurtosis and entropy, either applied to different colour channels [14], or a mixture of colour channels and vegetation indices [30,21].

2.2. Image segmentation

The majority of studies on the fine-grained segmentation of agricultural images use convolutional neural networks, such as SegNet [6], fully convolutional networks [34,29], or custom encoder/decoder networks [22,33,32,39].

The use of conventional machine learning in the segmentation of agricultural imagery is uncommon. However, in one study, images of sugarbeet seedlings and weeds were broken up into key-points, corresponding to 10×10 pixel squares, and each keypoint was classified, based on features extracted from the 80×80 pixel neighbourhood, and passed to a Random Forest [28]. The same authors went on to consider an object-based approach, where an object corresponds to an individual connected component of plant pixels, and a combination of the object and key-point based approaches [30]. Where the connected component corresponds to a single class, the object-based approach can be used for classification, but if the connected component consists of both classes, the object-based method will return an intermediate result (i.e. part way between crop and weed). In the latter case, the key-point based approach can then be used to further segment the component into regions consisting of crop plants, and regions consisting of weeds.

The method discussed in the preceding paragraph, may work well in crop fields, where plants can easily be separated from background

¹ Here we take default values to be 8 neighbours and a radius of 1.

Table 1

LBP hyperparameters (r = radius, n = number of neighbours) and colour channels (Grey = greyscale; NIR = near infrared; R, G and B from the RGB colour space; NDVI = normalized difference vegetation index; H, S and L from the HSL colour space where the value for R is replaced by NDVI) used in previous studies on weed detection.

Study	Hyperparameters	Colour Channel	Other
Dockleaf in pasture [5]	$r = 1 \dots 16$ $n = 2 \dots 30$	Grey	
Canola, radish and barley [35]	$r = 1, n = 8$ $r = 2, n = 16$ $r = 3, n = 24$	Grey	Upright, and Uniform
Tussock and needlegrass [16]	$r = 2, n = 8$	16 visible and 11 NIR bands	
Sugarbeet [30]	$r = 1, n = 8$	NDVI, G, B, H, S and L	Also uses the gradient and Laplacian of each channel

Table 2

List of hyperparameters considered for features used in this study.

Feature	Hyperparameter	Range of Values	MATLAB argument
HOG	Number of Bins*	4,6,9,12,15	NumBins
HOG	Signed Orientation	T/F	UseSignedOrientation
LBP	Number of Neighbours	8, 16, 24**	NumNeighbors
LBP	Radius	1, 2, 3, 4, 6, 8, 10	Radius
LBP	Rotational Invariance	T/F	Upright

* When signed orientation is used, the number of bins is effectively doubled.

** The following combinations of radius (r) and number of neighbours (n) were not considered: ($r = 1, n = 16$), ($r = 1, n = 24$), ($r = 2, n = 24$).

Table 3

List of colour spaces/indices used in this study.

Name	Description	Formula
Grey	Greyscale	$\frac{1}{3}(R + G + B)$
ExG	Excess Green	$2 \times G - R - B$
ExR	Excess Red	$1.4 \times R - G$
ExGR	Excess Green minus Excess Red	$ExG - ExR$

soil using a vegetation index, such as NDVI. In the more complex environment encountered in pastures, where we have green weeds against a backdrop of green pasture elements (the so called green on green problem), another approach is needed.

3. Methods

All coding was implemented in MATLAB using built in functions wherever possible, including `extractHOGFeatures` for HOG features, and `extractLBPFeatures` for LBP features.² The above MATLAB functions allow for a number of different customizable options. Accordingly, tests were carried out to determine the appropriate hyperparameters to use. The list of all hyperparameters tested in this work is given in Table 2. In addition to this, we used a range of vegetation indices as well as grayscale to produce a single colour channel image to be inputted into the feature extraction functions, as shown in Table 3. All tests were carried out on one of two Centos 7 systems with 192 Gb of memory and 32 Intel 2.5 Ghz CPU (hyperthreaded) cores.

² Documentation for these functions can be found at <https://au.mathworks.com/help/vision/ref/extracthogfeatures.html> and <https://au.mathworks.com/help/vision/ref/extractlbpfeatures.html> respectively.

3.1. Local binary patterns

LBP features [36] are a texture analysis method in which the intensity of a pixel is compared with its neighbours. Each neighbour is either classified as darker or brighter than the central pixel, meaning that the algorithm generates one bit of information per neighbour, as shown in equations (1) and (2).

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (1)$$

$$LBP(g^*) = \sum_{p=0}^{P-1} s(g_p - g^*) 2^p \quad (2)$$

Here, $s(x)$ is a sign function, $LBP(g^*)$ is the LBP descriptor for pixel g^* , P is the number of neighbours, and g is the grey level of the pixel. The image can then be broken down into rectangular blocks, and an LBP histogram (LBPH) calculated for each block. If P is set to 8, then the LBPH will have 256 bins, one for each of the possible descriptor values. In MATLAB, the `extractLBPFeatures` method automatically combines all patterns that are not “uniform”, as described by [36], into a single bin, meaning that the number of bins will be $P * (P - 1) + 3$, or $P + 2$ depending on whether rotational invariance is used. For this study, both rotationally invariant and rotationally variant features were considered, and a range of different values for the number of neighbours, and the radius of the neighbours were tested. Fig. 1 gives an example of LBP features extracted from image patches of an image taken from the tussock dataset.

3.2. Histogram of oriented gradients

HOG descriptors were originally designed as a solution to the problem of detecting pedestrians in images [10]. In HOG feature extraction, the image is divided into overlapping blocks of cells, and the occurrence of gradient (or edge) orientation is calculated. The orientation is classified into equal bins from 0° to 180° for unsigned, and 0° to 360° for

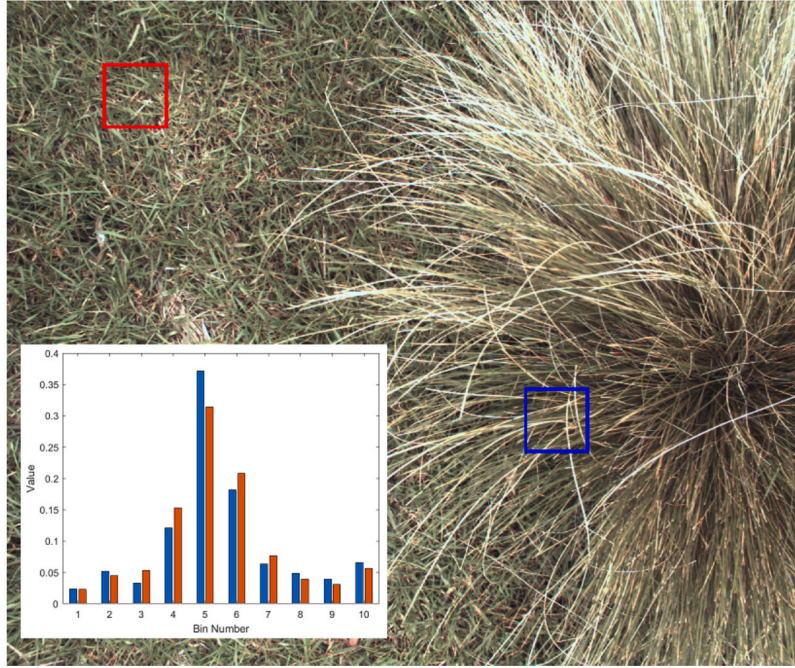


Fig. 1. Example LBP features extracted from image patches containing tussock (blue), and background pasture (red). Hyperparameters used were radius = 1, number of neighbours = 8, rotational invariance = True.

signed orientation. The number of orientation bins can be customized in MATLAB, and so this study considered a range of possible bin numbers, as well as testing both signed and unsigned orientation.

3.3. Colour features

In addition to the textural features discussed in the previous two subsections, colour features were also considered, including the mean, standard deviation, minimum, maximum, range, median, skewness and kurtosis of each channel, using either the RGB or YCbCr colour spaces.

3.4. Datasets

RGB Images were collected by the NSW Department of Primary Industries (DPI) from pastures in a range of locations across south east Australia. Four separate datasets were chosen for testing as shown in Table 4. Images were taken using a Mako G507 camera mounted on an unmanned ground vehicle, and facing straight downwards. The camera was at a height of 1200 mm. Images were annotated under supervision of experts from DPI, by drawing a polygon around the weeds of interest, using the VGG image annotator [15]. This provided a good compromise between the accuracy of the annotation, and the time taken to annotate each image. Care was taken to ensure that there was no overlap between any two images in a dataset, in order to avoid polluting the test set with data from the training set.

For the Bathurst burr, horehound and thistle datasets, tests were carried out using 5 × 10 fold cross-validation. For the tussock dataset, there were sufficient images to produce 10 entirely independent train/test split, with each image used exactly once, either in a single training set or a single test set. Statistical significance was determined using a corrected repeated k-fold cv test [7] for the Bathurst Burr, Horehound and Thistle datasets. For the Tussock dataset, the train/test sets were entirely independent, so a standard t-test was used. During each iteration of cross validation, the train/test splits were identical for each tested feature, meaning that a two-tailed, paired t-value was used to determine significance, with $\alpha = 0.05$, and 49 (or 9 for the Tussock dataset) degrees of freedom.

For all datasets, stratification was used when determining the splits. Specifically, each image was placed in one of two pseudo-classes, based on the percentage coverage of weed in that image. For composite datasets, with multiple sub-sets taken at different locations and/or times, each sub-set was split into pseudo-classes separately, yielding $2 \times N$ pseudo-classes, where N is the number of sub-sets in the dataset.

In order to see if a fusion of features could improve on the use of a single feature, a set of features was generated for each dataset, consisting of the two colour features, the best LBP feature for each choice of radius, and the best HOG feature for each choice of number of bins, giving a total of 14 features. Starting with the best individual feature, as determined by the individual feature tests, additional features were added incrementally, choosing the feature that produced the best results during each iteration.

3.5. Extreme learning machine

The Extreme Learning Machine (ELM) is a fast learning algorithm proposed by Huang and Seiw [17] for a single hidden layer feed forward neural network. In the ELM, the hidden layer weights are randomly assigned, and remain unchanged during the training process. The hidden layer outputs $\mathbf{H}$, can be found using equation (3), where $\mathbf{W}(\mathbf{b})$ is the hidden layer weights matrix (biases vector), and g is the activation function.

$$\mathbf{H} = g(\mathbf{W}\mathbf{X} + \mathbf{b}) \quad (3)$$

The final layer weights $\hat{\beta}$ are then calculated using the closed form solution given in equation (4), where $\dagger$ is the Moore-Penrose pseudo-inverse.

$$\hat{\beta} = \mathbf{H}^\dagger \mathbf{T} \quad (4)$$

While the ELM algorithm described in the paragraph above uses random hidden layer weights, further studies indicated that hidden layer weights which had been computed from the training data in some way, such as the constrained ELM [52] and computed input weights [48], produced superior results. In this study, we use the constrained differ-

Table 4
List of datasets used in this study.

Dataset	Weed Species	Location	Date of Collection
Bathurst Burr	Bathurst Burr (<i>Xanthium spinosum</i>)	Ferndale (34.636S 147.833E)	4-2-2022
Horehound	Horehound (<i>Marrubium vulgare</i>)	Yellangelo (34.861S 149.169E)	13-10-2021 23-11-2021
Thistle	Scottish Thistle (<i>Onopordum acanthium</i>)	Jugiong (34.860S 148.297E)	19-10-2021
	Saffron Thistle (<i>Carthamus lanatus</i>)	Binalong (34.690S 148.644E)	22-11-2021
		Yellangelo (34.861S 149.168E)	23-11-2021
		Clarkefield (37.480S 144.765E)	20-01-2022
Tussock	Serrated Tussock (<i>Nassella trichotoma</i>)	Yellangelo (34.861S 149.169E)	13-10-2021

Table 5
Syntax and operations used in this paper.

Syntax and methods	Explanation
*	Convolution
[X, x]	Join matrix with vector in horizontal direction
X[:, k]	Submatrix of X with all rows, and only the kth column(s).
$\odot$	Element-wise (Hadamard) division
sample ₊	Random sampling of positive and negative pair
init	Method for initializing weights
flatten	Operation for converting matrix into vector
unflatten	Operation for converting vector into square matrix
rand	Operation for generating a random number

ence (CD-weights) method of [52], and the sigmoid activation function, which is shown in equation (5).

$$g(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

3.6. Proposed model

Conceptually, it is beneficial for us to treat both the feature extractor and the ELM classifier as convolutional kernels. The proposed workflow is given below in a “procedural” format. In MATLAB, however, most steps can be simplified using matrix (or tensor) manipulations. We start with the following components:

- F_f : the convolutional kernel containing the feature extractor, parameterized by the feature extractor f
- $E_{W,b,\hat{\beta}}$: the convolutional kernel containing the ELM classifier, parameterized by the weights and biases of the ELM.
- M : the operation for unflattening and combining the output from $E_{W,b,\hat{\beta}}$

We also have a set of input images and corresponding ground truth weed heatmaps: $(I, L) \in \mathcal{I}$, and a training and validation set, $\mathcal{I}_t$ and $\mathcal{I}_v$, where $\mathcal{I}_t \cup \mathcal{I}_v = \mathcal{I}$ and $\mathcal{I}_t \cap \mathcal{I}_v = \emptyset$. An explanation of the syntax used in this section is provided in Table 5.

To train the ELM, we first calculate the input and target matrices, X and T , using Algorithm 1.

Algorithm 1 Generate training matrices.

Require: $\mathcal{I}_t$
Require: F_f

```

1:  $X \leftarrow \mathbf{0}_{0 \times 0}$ 
2:  $T \leftarrow \mathbf{0}_{0 \times 0}$ 
3: for  $(I, L) \in \mathcal{I}_t$  do
4:    $J = I * F_f$ 
5:   for  $J_p \in J$  do
6:      $L_p \leftarrow$  corresponding  $\hat{L} \in L$ 
7:      $X \leftarrow [X, \text{flatten}(J_p)]$ 
8:      $T \leftarrow [T, \text{flatten}(L_p)]$ 
9:   end for
10: end for
11: return  $(X, T)$ 
```

Now we can calculate the parameters of the ELM using Algorithm 2. With respect to hidden layer parameter initialization, each sample in X is considered positive (respectively negative), if its corresponding target in T is composed entirely of 1s (respectively 0s). If the target is composed of a mixture of 1s and 0s, then it is ignored. To combat imbalance in the dataset, negative samples (as defined in the preceding sentence) are randomly removed from the training matrices, with a probability of p equal to 0.3, 0.3, 0.7 and 0 for the Bathurst Burr, Horehound, Thistle and Tussock datasets respectively.

Algorithm 2 Generate ELM weights.

Require: X
Require: T
Require: $p \in [0, 1]$

```

1:  $W \leftarrow \mathbf{0}_{0 \times 0}$ 
2:  $b \leftarrow \mathbf{0}_{0 \times 0}$ 
3: for  $i \leftarrow 1$  to  $m$  do
4:    $(x^+, i^+), (x^-, i^-) \leftarrow \text{sample}_+^+(X, T)$ 
5:    $(w, b) \leftarrow \text{init}((x^+, i^+), (x^-, i^-))$ 
6:    $W \leftarrow [W, w]$ 
7:    $b \leftarrow [b, b]$ 
8: end for
9: keep  $\leftarrow [t \neq 0 \vee \text{rand}(0, 1) > p] \text{ for } t \in T$ 
10:  $\hat{X} \leftarrow X[:, \text{keep}]$ 
11:  $\hat{T} \leftarrow T[:, \text{keep}]$ 
12:  $H \leftarrow g(\hat{X}W + b)$ 
13:  $\hat{\beta} \leftarrow H^+ \hat{T}$ 
14: return  $(W, b, \hat{\beta})$ 
```

Using the trained ELM, we can calculate the predicted heatmap for an image, using Algorithm 3. The unflattening operation M is given in Algorithm 4.

Table 6

All Datasets: Mean IoU for colour features. The best result for each dataset is in **bold**. Results shown are the mean $\pm$ standard deviation, for a 5×10 fold cross validation (Bathurst burr, horehound and thistle datasets), or for a set of 10 entirely independent train/test splits (tussock). Results which are not significantly different from the best result in the column are labelled with a † .

Colour Channel	Dataset			
	Bathurst Burr	Horehound	Thistle	Tussock
RGB	74.9 $\pm$ 9.9	70.8 $\pm$ 6.5†	71.6 $\pm$ 2.5	78.4 $\pm$ 5.2 †
YCbCr	79.5 $\pm$ 7.7†	69.6 $\pm$ 6.4 †	74.3 $\pm$ 3.5†	79.4 $\pm$ 4.1†

Algorithm 3 Calculate predicted heatmap.

Require: I
Require: F_f
Require: $E_{w,b,\beta}$
1: $J = I * F_f$
2: $K = J * E_{w,b,\beta}$
3: $Y = M(K)$
4: **return** Y

Algorithm 4 Unflatten and combine output of ELM.

Require: K
1: $Y \leftarrow 0$
2: $C \leftarrow 0$
3: **for** $k_p \in K$ **do**
4: $k_p \leftarrow \text{unflatten}(k_p)$
5: $Y_p \leftarrow \text{corresponding } \hat{Y} \in Y$
6: $C_p \leftarrow \text{corresponding } \hat{C} \in C$
7: $Y_p \leftarrow Y_p + k_p$
8: $C_p \leftarrow C_p + 1$
9: $Y \leftarrow Y \oslash C$
10: **return** Y
11: **end for**

In this study, all images were of size (with height dimension first) $2016 \times 2400 \times 3$. The feature extraction kernel used a size of 48×48 , and stride of 48×48 . For HOG features, the cell size was set at 24×24 , meaning that each block consisted of 2×2 cells. The ELM kernel used a size of 4×4 and stride of 1×1 , corresponding to an image patch size of 192×192 and 48×48 respectively. The heatmap was downsampled by a factor of 16, meaning that the ELM outputted a 12×12 flattened heatmap vector. If a by-pixel segmentation was required, then the heatmap could further be upsampled, for example, using bilinear interpolation. However, for the purpose of determining when to switch herbicide spray nozzles on and off, a by-pixel segmentation is unnecessary, because the nozzles will have a limited accuracy. For this reason, we leave the heatmap downsampled in this study.

4. Results

The efficacy of individual features was first tested, according to the methodology in Section 3. The results are presented in Table 6 for colour features, Tables 7, 9, 11 and 13 for LBP features and Tables 8, 10, 12 and 14 for HOG features. For these tests, the number of hidden layer neurons was varied from 100 to 1000, with only the best result shown for each feature. In most, but not all cases, this corresponded to a hidden layer size of 1000. Extensions of the figures listed in this section are provided in the supplemental material, including the standard deviation of all results, and the number of neurons which produced the best result for each feature.

Out of the two colour spaces used for colour feature extraction, YCbCr performed best for three datasets (Bathurst Burr, Thistle and Tussock), and was not significantly different from RGB in the other dataset (Horehound). For LBP features, the best choice of colour channel was dependent on the dataset. For Bathurst Burr, ExGR performed

best for all choices of radius, except for $r = 1$. By contrast the best results for all choices of radius for horehound were achieved using the grey colour channel. For tussock and thistle, the grey and ExG colour channels produced the best results, depending on choice of radius. The choice of colour channel was also important for HOG features, with the best results for thistle and horehound being ExG, and for tussock being grey. By comparison, the best results for Bathurst burr was achieved using grey or ExGR, depending on the choice of number of bins. In terms of the choice of number of bins, 4 bins almost always produced inferior results, whereas there was often no statistically significant different between the choice of bin numbers ranging from 6 to 15.

It should be pointed out that there was considerable variance in the results for individual features, in particular for the Bathurst burr and horehound datasets (SD as high as 12.6% and 9.2% respectively). This is likely due to the fact that there were only a small number of images in these datasets, and some images were considerably harder than others to segment, due to plants being run over and flattened by the AGV, or heavily occluded by other vegetation. The use of identical train/test splits for each tested feature, however, allowed us to still demonstrate statistically significant results for some features, because the individual features tended to be good or bad at the same train/test splits.

The best result by dataset and feature type (colour, LBP or HOG) is presented in Table 15. As can be seen, LBP produced the best features for three out of four datasets, while colour features were best for the thistle dataset. By comparison, HOG features performed much worse for three out of four datasets (e.g., up to 22.2% worse than the best LBP feature, in the case of the Bathurst burr dataset). For the tussock dataset, the performance of the best HOG feature was only 7.6% worse than the best LBP feature. We suggest that the reason that HOG features performed better for this dataset is because the tussock plants are arranged with many blades of grass aligned in a rough radial pattern around the centre of the plant, resulting in a more pronounced feature value for particular bins, as compared to the more haphazard arrangement of grass blades in the background pasture.

Fig. 2 displays the results of the feature fusion tests. For these tests, the number of hidden layer neurons was varied from 1000 to 3000, with only the best result shown for each feature fusion. In most, but not all cases, this corresponded to a hidden layer size of 3000. As can be seen, the results improved as features were added, but plateaued quickly, and even started to go back down as the fusion became larger. The improvements achieved through the fusion, as compared to the best individual feature, varied from 4.1% for the tussock dataset, to 7.3% for the thistle dataset. Table 16 gives the order of features added to the fusion for each dataset. As can be seen, the best results were achieved using a combination of LBP, HOG and colour features for the Bathurst burr and horehound datasets. By comparison, the best result for the thistle and tussock datasets did not include any HOG or colour features respectively. The extraction of multiple features from an image can be computationally expensive, and so the smallest fusion which is not significantly different from the best result can be seen as a good candidate for a compromise between accuracy and speed. For Bathurst burr, horehound and thistle, this fusion consisted of two LBP features and one colour feature, whereas for tussock, it consisted of three LBP

Table 7

Bathurst Burr Dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Feature parameters are given in format ch-r-nn, where ch is the colour channel or index, r is rotational [v]ariant or [i]nvariant, and nn is the number of neighbours. The best result for each set of three rows (which corresponds to fixing the colour channel and rotational invariance) is in **bold**, for each column (which corresponds to fixing the radius) is underlined, and overall is *italicized*. Results shown are the mean for a 5 × 10 fold cross validation. Results which are not significantly different from the best result in the set of three rows (column) are labelled with a * (*).

Parameters	Radius						
	1	2	3	4	6	8	10
grey-i-8	63.6 [†]	67.1 [†]	64.2	64.9	65.9 [†]	63.4	61.5
grey-i-16	-	70.7 [†]	70.4	71.6 [†]	72.3 ^{*†}	69.7 [†]	65.9 [†]
grey-i-24	-	-	72.3 [†]	74.2^{*†}	73.8 ^{*†}	72.5 ^{*†}	68.1 [†]
grey-v-8	64.8 [†]	70.4 ^{*†}	70.0	69.4 [*]	66.7 [†]	63.2	61.5
grey-v-16	-	71.7 ^{*†}	72.2[*]	70.5 ^{*†}	64.8	60.2	58.9
grey-v-24	-	-	69.6	70.6 ^{*†}	64.9	61.9	58.7
exg-i-8	70.9^{*†}	69.5 ^{*†}	68.2 [*]	67.5 ^{*†}	63.4 [†]	63.6 [†]	64.4 [†]
exg-i-16	-	68.7 ^{*†}	67.2	67.7 [†]	66.3 [†]	66.5 [†]	66.8 [†]
exg-i-24	-	-	67.0	67.5 [†]	66.6 [†]	67.2 [†]	66.4 [†]
exg-v-8	72.3^{*†}	66.8 [†]	64.9	66.4 [†]	62.0	61.0 [†]	61.3 [†]
exg-v-16	-	68.2 [†]	62.0	62.8	60.6	60.1	59.9
exg-v-24	-	-	61.8	64.4	60.4	60.3 [†]	59.6
exr-i-8	67.1 ^{*†}	60.8	62.9	68.4 [*]	64.7	60.6	55.3
exr-i-16	-	62.1	67.9	70.2 [*]	67.6 [†]	66.4 [†]	60.9
exr-i-24	-	-	68.8 [*]	72.0^{*†}	68.4 [†]	67.7 [†]	63.1 [†]
exr-v-8	67.2 [†]	65.9	62.7	72.1^{*†}	65.3	61.0	56.5
exr-v-16	-	65.2	67.2	69.9 [*]	67.1 [†]	68.2 [†]	60.0
exr-v-24	-	-	65.7	67.3	69.6 ^{*†}	65.1 [†]	64.7 [†]
xgr-i-8	67.9 [†]	68.7	68.2	71.9	66.3	63.1	57.9
xgr-i-16	-	74.1 [†]	80.2 ^{*†}	76.8 [†]	73.2	73.4 [†]	69.8
xgr-i-24	-	-	81.6^{*†}	78.7[†]	77.2[†]	73.0 [†]	72.9[†]
xgr-v-8	69.4 [†]	72.2	70.1	78.4 ^{*†}	70.6	66.1	59.4
xgr-v-16	-	76.1[†]	80.0^{*†}	75.9	72.8	73.4 [†]	69.3
xgr-v-24	-	-	79.7 [*]	77.5 [†]	75.4	72.2 [†]	71.5 [†]

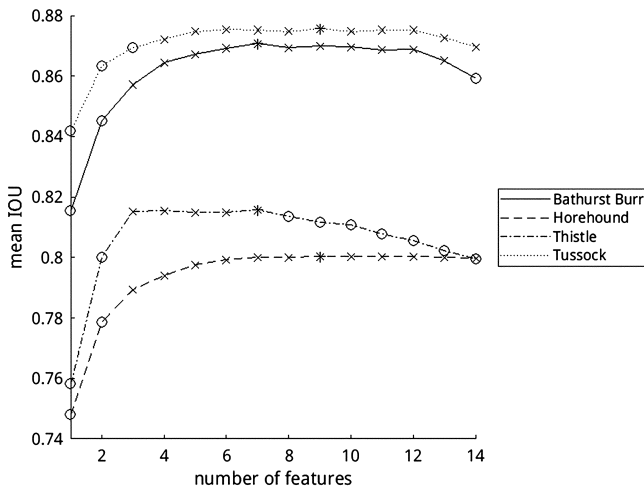


Fig. 2. Results for a fusion of features for all datasets. The data-point marked with a '*' is the best result for that dataset. Datapoints marked with an 'o' (x) are (are not) significantly different from the best result for that dataset ($\alpha = 0.05$). Best result for Bathurst burr was 87.1 ± 3.9 , for horehound was 79.5 ± 4.8 , for thistle was 81.6 ± 2.1 and for tussock was 87.6 ± 1.9 .

features, and one HOG feature. Fig. 3 gives example outputs of our model. In these images, the feature fusions which produced the best results for each dataset have been used. The model was able to segment some images very well, but other images posed significant challenges due to heavy occlusion (in the case of horehound), or the presence of many small weed plants against complex background elements (in the case of thistle).

Depending on the choice of hyperparameters, feature extraction for LBP feature took between 0.266 s and 0.848 s, for HOG features between 0.080 s and 0.090 s, and for colour features roughly 0.367 s. Based on CPU usage, it would appear that the `extractLBPFeatures` function does not support parallelization, whereas the `extractHOGFeatures` function does, which could explain the difference in time taken for these methods. Excluding feature extraction, the time taken to process an image by the network and generate a mask varied depending on the dimensionality of the feature used, and the number of hidden layer neurons in the model. Times for a model with 100 neurons ranged from 0.001 s to 0.014 s, and for a model with 1000 neurons ranged from 0.002 s to 0.015 s. This means that the majority of the time was spent on feature extraction. In principle, it should be possible to speed up feature extraction through the development of a GPU implementation.

5. Discussion

This paper has presented a novel method of using a single hidden layer feedforward neural network trained with the ELM algorithm to produce a fine-grained segmentation of an image. The method has been tested on four datasets of pastoral lands in southeast Australia, and has demonstrated good performance, with a best mIOU of 79.5% to 87.6% achieved depending on the dataset. The horehound and thistle datasets proved to be the most difficult to segment. A likely explanation for this is that some plants in the horehound set were heavily occluded by other plants, whereas some of the thistle plants were small, and difficult to distinguish from the background. LBP features performed best for three out of four datasets. This strong performance compared to other feature extraction methods is consistent with the findings of other studies where multiple features are compared (see for example

Table 8

Bathurst Burr Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. The best result for each row is in **bold**, for each column is underlined, and overall is *italicized*. Results shown are the mean for a 5×10 fold cross validation. Results which are not significantly different from the best result in the row (column) are labelled with a * ([†]).

Parameters	Number of Bins				
	4	6	9	12	15
grey-u	<u>58.9</u> [†]	58.0 [†]	57.9 [†]	56.3 [†]	55.8 [†]
grey-s	53.8	<u>60.0</u> [†]	<u>59.8</u> [†]	59.2 [†]	58.2 [†]
exg-u	41.2	<u>46.3</u> *	46.3*	45.6*	45.1*
exg-s	36.5	43.8*	<u>44.3</u> *	44.0*	43.7*
exr-u	42.9	47.6	<u>51.1</u> *	50.8*	50.5*
exr-s	40.0	46.1*	49.1*	50.2*	<u>50.6</u> *
xgr-u	51.6 [†]	57.9 [†]	59.5 [†]	<u>59.7</u> [†]	59.4 [†]
xgr-s	47.4	56.4 [†]	59.1 [†]	59.6 [†]	<u>59.9</u> [†]

Table 9

Horehound Dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Table has been annotated in the same way as Table 7.

Parameters	Radius						
	1	2	3	4	6	8	10
grey-i-8	<u>65.1</u> [†]	67.0 [†]	68.2 [†]	67.7	66.3	66.4	67.4*
grey-i-16	-	<u>68.5</u> [†]	70.4 [†]	69.1	70.0 [†]	69.6 [†]	69.4 [†]
grey-i-24	-	-	70.3 [†]	<u>70.9</u> [†]	70.3 [†]	<u>69.8</u> [†]	<u>69.5</u> [†]
grey-v-8	64.9 [†]	67.7 [†]	68.4 [†]	67.7	66.0	66.2	68.0 [†]
grey-v-16	-	67.9 [†]	<u>70.4</u> [†]	70.1 [†]	<u>70.8</u> [†]	69.2 [†]	67.9 [†]
grey-v-24	-	-	69.2*	70.6 [†]	70.1 [†]	68.5 [†]	66.5
exg-i-8	60.3 [†]	61.0*	61.5*	61.5*	62.4*	61.8*	59.0*
exg-i-16	-	60.5*	62.3 [†]	<u>62.6</u> *	61.7*	61.6*	60.6*
exg-i-24	-	-	61.6 [†]	61.3*	61.3*	59.9*	57.6*
exg-v-8	58.3*	60.3*	59.2*	59.0*	<u>60.4</u> *	60.2*	59.5*
exg-v-16	-	56.1*	56.5*	56.7*	55.7*	55.5*	54.0
exg-v-24	-	-	56.3*	55.2*	55.8*	55.9*	52.1
exr-i-8	<u>60.2</u> [†]	59.6*	58.4*	59.1*	56.9*	55.4*	56.6*
exr-i-16	-	59.3*	59.1*	59.6*	58.1*	56.7*	57.0*
exr-i-24	-	-	58.4*	58.1*	56.4*	56.5*	59.9*
exr-v-8	<u>59.5</u> [†]	50.5	52.2	53.5	54.9*	54.7*	55.9*
exr-v-16	-	51.7	53.7	53.8	54.7	53.6	53.5
exr-v-24	-	-	52.4	52.2	53.6	56.0*	57.5*
xgr-i-8	63.1 [†]	60.2	63.0 [†]	63.0	62.9 [†]	61.1	60.7
xgr-i-16	-	61.5	64.7 [†]	64.1 [†]	64.7 [†]	64.5 [†]	64.2 [†]
xgr-i-24	-	-	64.1 [†]	<u>67.2</u> [†]	65.2 [†]	64.1 [†]	64.1 [†]
xgr-v-8	<u>63.5</u> [†]	58.7*	59.7*	60.6*	61.1*	60.8*	59.7
xgr-v-16	-	56.1	62.5 [†]	60.5*	63.3 [†]	62.1*	60.8*
xgr-v-24	-	-	60.7*	63.4 [†]	62.3 [†]	62.9 [†]	60.7*

Table 10

Horehound Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. Table has been annotated in the same way as Table 8.

Parameters	Number of Bins				
	4	6	9	12	15
grey-u	48.8 [†]	49.5*	49.8 [†]	50.1 [†]	<u>50.4</u> [†]
grey-s	52.8 [†]	<u>56.3</u> [†]	<u>55.4</u> [†]	54.1 [†]	53.1 [†]
exg-u	<u>53.0</u> [†]	54.1 [†]	54.9 [†]	<u>55.2</u> [†]	<u>55.0</u> [†]
exg-s	45.2	54.2 [†]	54.4 [†]	<u>54.4</u> [†]	54.3 [†]
exr-u	44.9	48.2*	48.2*	48.0 [†]	<u>49.0</u> [†]
exr-s	45.9	<u>49.7</u> [†]	49.0*	48.2 [†]	47.6 [†]
xgr-u	51.0 [†]	51.1 [†]	51.6 [†]	51.8 [†]	<u>51.8</u> [†]
xgr-s	46.4	<u>54.6</u> [†]	53.3 [†]	52.8 [†]	52.5 [†]

Table 11

Thistle Dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Table has been annotated in the same way as Table 7.

Parameters	Radius						
	1	2	3	4	6	8	10
grey-i-8	66.3	67.8	69.1 [†]	68.7	68.4	68.7	68.6
grey-i-16	-	70.0 [†]	70.5 [†]	71.0	70.9 ^{*,†}	70.1	69.2
grey-i-24	-	-	70.7 [†]	71.6^{*,†}	71.0^{*,†}	70.4 [†]	69.5[†]
grey-v-8	65.6	66.2	66.3	66.8	67.6	68.3[*]	68.2 [*]
grey-v-16	-	64.7	64.7	66.1	67.8 [*]	68.0 [*]	67.7
grey-v-24	-	-	63.8	65.4	67.3	67.7 [*]	67.4
exg-i-8	71.8^{*,†}	71.2^{*,†}	70.7 [†]	68.0 [†]	65.2	65.4	64.4
exg-i-16	-	71.1 ^{*,†}	71.0^{*,†}	67.9 [†]	66.4	64.8	64.4
exg-i-24	-	-	70.8 [†]	68.5 [†]	66.2	65.4	64.1
exg-v-8	71.7^{*,†}	70.0	69.1	67.8 [†]	65.8	65.7	64.9
exg-v-16	-	69.3	67.8	66.4	64.5	62.8	62.7
exg-v-24	-	-	66.6	64.8	63.0	61.4	60.6
exr-i-8	65.4 [*]	64.1	63.5	65.3 [*]	59.7	59.7	59.4
exr-i-16	-	65.7 [*]	66.3[*]	65.5 [*]	65.4 [*]	64.0	60.5
exr-i-24	-	-	65.9 [*]	66.1 [*]	63.9	63.7	62.2
exr-v-8	68.6[*]	63.9	62.6	64.9	60.0	61.3	60.6
exr-v-16	-	62.5	63.0	64.2	62.2	60.3	58.4
exr-v-24	-	-	62.8	62.4	60.6	59.5	57.9
xgr-i-8	66.5	66.4	67.2	67.7 [*]	61.2	61.6	61.7
xgr-i-16	-	67.5 [*]	68.3^{*,†}	67.8 [*]	68.0 [*]	67.6 [*]	64.9
xgr-i-24	-	-	67.9 [†]	66.1	67.8 [*]	66.9 [*]	65.3
xgr-v-8	68.5[*]	66.3	66.5	67.4 [*]	61.3	62.5	61.4
xgr-v-16	-	66.1	65.8	64.4	63.7	63.4	60.6
xgr-v-24	-	-	65.2	63.3	63.2	62.4	60.2

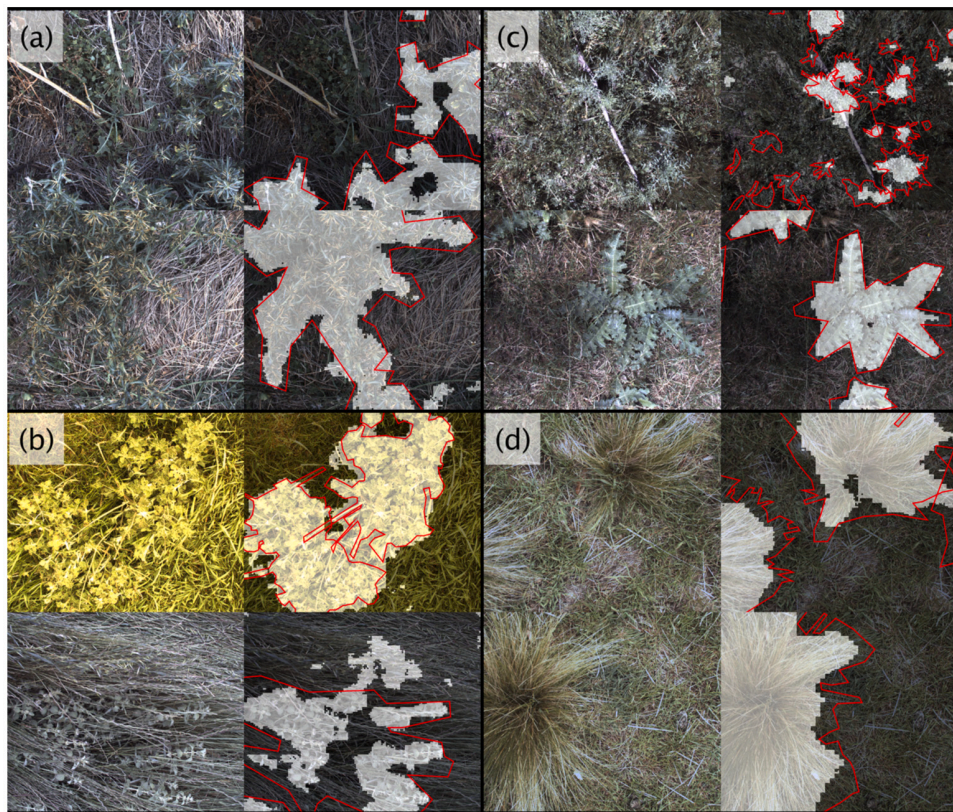


Fig. 3. Example outputs of model for (a) Bathurst burr, (b) horehound, (c) thistle and (d) tussock datasets. Polygon annotations drawn by a human annotator are shown in red, and regions classified as weed by the model are shown in white.

[5,16,14]). Colour features also performed well, being the best feature for one dataset, and not significantly different from the best feature for another. By comparison, HOG features performed surprisingly poorly on all datasets tested.

Appropriate optimization of hyperparameters also proved to be important in designing the feature extractors. For LBP features, the best choice of hyperparameters leads to an improvement in mIOU of 4.8% to 18.4% over the default choice (we take this to be $r = 1$, $n = 8$, and

Table 12

Thistle Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. Table has been annotated in the same way as Table 8.

Parameters	Number of Bins				
	4	6	9	12	15
grey-u	46.3 [*]	46.9[*]	45.5	45.0	44.4
grey-s	<u>53.4[†]</u>	57.6[†]	<u>57.1[†]</u>	<u>55.8[†]</u>	<u>54.9[†]</u>
exg-u	38.3	51.6	52.5 [*]	52.3 [*]	53.0[†]
exg-s	40.7	45.2	50.4 [*]	50.8[*]	50.3
exr-u	38.5	45.4	50.0	51.1[*]	50.0
exr-s	47.0	49.7	50.1	50.1	52.3[*]
xgr-u	41.7	46.2	48.1 [*]	48.6[*]	48.2
xgr-s	50.3	53.5 [*]	54.9 [*]	54.9[†]	54.6 [†]

Table 13

Tussock dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Table has been annotated in the same way as Table 7.

Parameters	Radius						
	1	2	3	4	6	8	10
grey-i-8	75.5	75.6	76.6	78.8	79.6	76.1	70.1
grey-i-16	-	80.0	81.1	<u>83.3[†]</u>	83.1 [†]	81.3 [†]	79.7 [†]
grey-i-24	-	-	82.3 [†]	83.2 [†]	83.4[†]	81.4 [†]	79.9 [†]
grey-v-8	<u>82.4[†]</u>	83.1[†]	82.4 [*]	82.3 [†]	81.2	77.5	72.9
grey-v-16	-	82.6 [†]	82.0	82.6 [†]	81.7	80.1 [†]	79.2
grey-v-24	-	-	82.6 [†]	82.0	81.4	79.6	78.7
exg-i-8	76.8	79.7 [*]	78.8	79.6	80.5	79.0	72.3
exg-i-16	-	79.5 [*]	78.4	80.4	81.3 [*]	81.6[†]	<u>80.9[†]</u>
exg-i-24	-	-	79.3	80.5	81.4 [*]	80.9	80.6 [†]
exg-v-8	81.3 [†]	<u>83.4[†]</u>	83.4[†]	82.8 [†]	81.6	79.2	74.4
exg-v-16	-	82.4	81.8	81.7 [†]	81.7	81.1 [†]	80.3 [†]
exg-v-24	-	-	80.8	81.1	81.0	80.8 [†]	80.2 [†]
exr-i-8	75.2 [*]	70.9	69.7	69.1	73.8	68.4	60.1
exr-i-16	-	73.9	75.2 [*]	73.7	76.1[*]	74.5	74.2
exr-i-24	-	-	73.0	75.6 [*]	74.7	74.5	73.9
exr-v-8	79.5[*]	76.3	74.2	74.6	75.9	71.8	63.6
exr-v-16	-	75.9	76.4	77.2	76.1	73.2	71.7
exr-v-24	-	-	75.3	75.6	75.0	73.4	71.6
xgr-i-8	75.7	72.6	72.1	75.8	78.3	76.7	70.0
xgr-i-16	-	75.5	76.1	79.8[*]	78.4	77.0	77.8
xgr-i-24	-	-	75.6	78.1	78.2	77.7	75.4
xgr-v-8	81.1[†]	79.7	80.1 [*]	81.1 [*]	80.3 [*]	78.3	74.0
xgr-v-16	-	75.4	78.7	79.7 [*]	78.2	77.8	77.4
xgr-v-24	-	-	76.9	78.7	77.1	75.4	73.9

Table 14

Tussock Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. Table has been annotated in the same way as Table 8.

Parameters	Number of Bins				
	4	6	9	12	15
grey-u	<u>75.4[†]</u>	75.8[†]	<u>75.5[†]</u>	<u>75.2[†]</u>	<u>75.1[†]</u>
grey-s	61.8	75.3[*]	75.2 [†]	74.8	74.5
exg-u	68.8	73.0	73.6[*]	73.2	72.7
exg-s	51.9	71.8	73.4[*]	73.1	72.6
exr-u	63.1	69.3	71.1	71.6[*]	71.2 [*]
exr-s	43.4	63.7	67.8	69.7[*]	69.7 [*]
xgr-u	70.8	73.7	74.5[†]	74.4 [†]	74.1 [†]
xgr-s	51.7	72.1	73.7[*]	73.4	73.1

grey channel), depending on the dataset. For HOG features, the difference in results for varying combinations of hyperparameters was not so pronounced, with the choice of colour channel being a far more important consideration. It was found that the use of 4 bins was almost always worse than other choices, except for an unsigned feature extractor working on the grey colour channel. There was often

no significant difference between other choices of bins (i.e. 6 up to 15).

It is interesting to note that different colour channels were most effective for different datasets, which suggests that, if a multi-class segmenter was required, then features may need to be extracted from multiple channels/indices to produce acceptable results.

Table 15

All Datasets: Comparison of best feature for each feature type (colour, LBP and HOG). The best result for each dataset is in **bold**. Results shown are the mean $\pm$ standard deviation, for a 5×10 fold cross validation (Bathurst burr, horehound and thistle datasets), or for a set of 10 entirely independent train/test splits (tussock). Results which are not significantly different from the best result in the column are labelled with a $\dagger$.

Feature Type (Best)	Dataset			
	Bathurst Burr	Horehound	Thistle	Tussock
Colour	79.5 $\pm$ 7.7 $\dagger$	70.8 $\pm$ 6.5	74.3 $\pm$ 3.5$\dagger$	79.4 $\pm$ 4.1
LBP	81.6 $\pm$ 7.7$\dagger$	74.6 $\pm$ 5.3$\dagger$	71.8 $\pm$ 3.3	83.4 $\pm$ 3.5$\dagger$
HOG	59.4 $\pm$ 6.6	58.8 $\pm$ 7.8	57.6 $\pm$ 3.6	75.8 $\pm$ 2.1

Table 16

All Datasets: order in which the different features were added to the feature fusion. At every iteration, the features were added to the existing fusion, one at a time, and the one which produced the best average mIOU was selected. The features are labelled as LBP- r for LBP features, where r is the radius, and HOG- n for HOG features, where n is the number of bins. RGB and yCbCr are the two colour features. The feature whose addition produced the best result for each dataset is labeled with a $*$, and the feature whose addition produced the smallest fusion whose result was not significantly different from the best result is labelled with a $\dagger$. This table should be read as follows: the feature fusion for a dataset, consisting of n individual features was made up of the features given in rows 1, ..., n in the column for that dataset.

Feature Order	Bathurst Burr	Horehound	Thistle	Tussock
1	LBP-3	LBP-4	yCbCr	LBP-3
2	yCbCr	LBP-10	LBP-4	LBP-1
3	LBP-4 $\dagger$	RGB $\dagger$	LBP-3 $\dagger$	LBP-6
4	HOG-9	LBP-1	LBP-1	HOG-15 $\dagger$
5	LBP-10	LBP-3	RGB	LBP-4
6	RGB	LBP-8	LBP-2	LBP-8
7	HOG-6 $*$	HOG-4	LBP-6 $*$	LBP-2
8	HOG-4	yCbCr $*$	LBP-8	HOG-6
9	HOG-12	HOG-9	HOG-4	HOG-4 $*$
10	LBP-8	HOG-12	HOG-6	HOG-12
11	LBP-6	LBP-2	LBP-10	HOG-9
12	HOG-15	HOG-15	HOG-9	LBP-10
13	LBP-1	HOG-6	HOG-12	yCbCr
14	LBP-2	LBP-6	HOG-15	RGB

6. Conclusion

This work demonstrates the feasibility of using a shallow ANN trained by the ELM algorithm, and working on manually extracted features, to produce a fine-grained segmentation of agricultural images. Future work, however, should determine how attractive this approach is compared to other methods, such as the use of CNNs, which are the most common choice for segmentation problems.

This study also demonstrated that the choice of colour channel (or index) was important, and had a significant impact on the usefulness of extracted features, as measured using the mIOU metric. The use of a feature fusion was an effective method of improving results, resulting in an increase of 4.2 - 7.3% compared to the best individual feature, depending on the dataset considered. Future research could involve testing more colour channels/indices, or looking at novel ways of combining multiple features into a more effective model.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.atech.2023.100288>.

References

- [1] Growing at a slower pace, world population is expected to reach 9.7 billion in 2050 and could peak at nearly 11 billion around 2100, <https://www.un.org/development/desa/en/news/population/world-population-prospects-2019.html>.
- [2] Muhammad Ahmad, Adil Mehmood Khan, Manuel Mazzara, Salvatore Distefano, Multi-layer extreme learning machine-based autoencoder for hyperspectral image classification, 2019.
- [3] Mansoor Alam, Muhammad Shahab Alam, Muhammad Roman, Muhammad Tufail, Muhammad Umer Khan, Muhammad Tahir Khan, Real-time machine-learning based crop/weed detection and classification for variable-rate spraying in precision agriculture, in: 2020 7th International Conference on Electrical and Electronics Engineering (ICEEE), IEEE, 2020, pp. 273–280.
- [4] Darah Aqel, Shadi Al-Zubi, Ala Mughaid, Yaser Jararweh, Correction to: extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture, Clust. Comput. 25 (3) (jun 2022) 2303.

- [5] A. Binch, C.W. Fox, Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland, *Comput. Electron. Agric.* 140 (2017) 123–138.
- [6] Petra Bosilj, Erchan Aptoula, Tom Duckett, Grzegorz Cielniak, Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture, *J. Field Robot.* 37 (1) (2020) 7–19.
- [7] Remco R. Bouckaert, Eibe Frank, Evaluating the replicability of significance tests for comparing learning algorithms, in: *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 3–12.
- [8] Brendan Calvert, Alex Olsen, Bronson Philippa, Mostafa Rahimi Azghadi, AutoWeed: detecting *Harrisia cactus* in the Goondiwindi region for selective spot-spraying, in: *Proceedings of the 1st Queensland Pest Animal and Weed Symposium*, Page 52, Weed Society of Queensland Pty. Ltd., Toowoomba, QLD, Australia, 2019.
- [9] Brendan Calvert, Alex Olsen, James Whinney, Mostafa Rahimi Azghadi, Robotic spot spraying of *Harrisia cactus* (*Harrisia martinii*) in grazing pastures of the Australian rangelands, *Plants* 10 (10) (2021) 2054.
- [10] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE.
- [11] Schori Damian, Anken Thomas, Seatovic Dejan, Using fully convolutional networks for rumex obtusifolius segmentation, a preliminary report, in: *2019 International Symposium ELMAR*, IEEE, 2019, pp. 119–122.
- [12] Candra Dewi, Wayan Firdaus Mahmudy, Rio Arifando, Yoke Kusuma Arbawa, Beryl Labique Ahmadi, Improve performance of extreme learning machine in classification of patchouli varieties with imbalanced class, in: *Proceedings of the 5th International Conference on Sustainable Information Engineering and Technology*, ACM, New York, NY, USA, nov 2020, pp. 16–22.
- [13] Anita Dixit, Nagaratna Hedge, B. Eswara Reddy, Vegetation type classification system using pseudo Zernike moments and ELM, in: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPSCSI)*, IEEE, sep 2017, pp. 2502–2508.
- [14] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, Marcelo Theophilo Folhes, Weed detection in soybean crops using ConvNets, *Comput. Electron. Agric.* 143 (2017) 314–324.
- [15] Abhishek Dutta, Andrew Zisserman, The VIA annotation software for images, audio and video, in: *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, ACM, New York, NY, USA, 2019.
- [16] Adnan Farooq, Xiuping Jia, Jun Zhou, Texture and shape features for grass weed classification using hyperspectral remote sensing images, in: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2019, pp. 7208–7211.
- [17] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, IEEE, 2004.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.
- [19] Kun Hu, Guy Coleman, Shan Zeng, Zhiyong Wang, Michael Walsh, Graph weeds net: a graph-based deep learning method for weed recognition, *Comput. Electron. Agric.* 174 (2020) 105520.
- [20] Calvin Hung, Zhe Xu, Salah Sukkarieh, Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a UAV, *Remote Sens.* 6 (12) (2014) 12037–12054.
- [21] Wajahat Kazmi, Francisco Garcia-Ruiz, Jon Nielsen, Jesper Rasmussen, Hans Jørgen Andersen, Exploiting affine invariant regions and leaf edge shapes for weed detection, *Comput. Electron. Agric.* 118 (2015) 290–299.
- [22] Abbas Khan, Talha Ilyas, Muhammad Umraiz, Zubaer Ibna Mannan, Hyongsuk Kim, CED-Net: crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture, *Electronics* 9 (10) (2020) 1602.
- [23] Tsampikos Kounalakis, Georgios A. Triantafyllidis, Lazaros Nalpantidis, Image-based recognition framework for robotic weed control systems, *Multimed. Tools Appl.* 77 (8) (2018) 9567–9594.
- [24] Tsampikos Kounalakis, Georgios A. Triantafyllidis, Lazaros Nalpantidis, Deep learning-based visual recognition of rumex for robotic precision farming, *Comput. Electron. Agric.* 165 (2019) 104973.
- [25] Petre Lameski, Eftim Zdravevski, Vladimir Trajkovik, Andrea Kulakov, Weed detection dataset with RGB images taken under variable light conditions, in: *ICT Innovations 2017*, Springer International Publishing, Cham, 2017, pp. 112–119.
- [26] Vi Le Nguyen Thanh, Selam Ahderom, Kamal Alameh, Performances of the LBP based algorithm over CNN models for detecting crops and weeds with similar morphologies, *Sensors* 20 (8) (2020) 2193.
- [27] Bo Liu, Ryan Bruch, Weed detection for selective spraying: a review, *Curr. Robot. Rep.* 1 (1) (2020) 19–26.
- [28] P. Lottes, M. Hoferlin, S. Sander, M. Muter, P. Schulze, Lammers C. Stachniss, An effective classification system for separating sugar beets and weeds for precision farming applications, in: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 5157–5163.
- [29] Philipp Lottes, Jens Behley, Andres Milioto, Cyrill Stachniss, Fully convolutional networks with sequential information for robust crop and weed detection in precision farming, *IEEE Robot. Autom. Lett.* 3 (4) (2018) 2870–2877.
- [30] Philipp Lottes, Markus Hörferlin, Slawomir Sander, Cyrill Stachniss, Effective vision-based classification for separating sugar beets and weeds for precision farming, *J. Field Robot.* 34 (6) (2017) 1160–1178.
- [31] Maitiniyazi Maimaitijiang, Abduwasit Ghulam, Paheding Sidike, Sean Hartling, Matthew Maimaitiyiming, Kyle Peterson, Ethan Shavers, Jack Fishman, Jim Peterson, Suhas Kadam, Joel Burken, Felix Fritsch, Unmanned Aerial System (UAS)-based phenotyping of soybean using multi-sensor data fusion and extreme learning machine, *ISPRS J. Photogramm. Remote Sens.* 134 (dec 2017) 43–58.
- [32] Chris McCool, Tristan Perez, Ben Upcroft, Mixtures of lightweight deep convolutional neural networks: applied to agricultural robotics, *IEEE Robot. Autom. Lett.* 2 (3) (2017) 1344–1351.
- [33] Andres Milioto, Philipp Lottes, Cyrill Stachniss, Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2229–2235.
- [34] Anders Krogh Mortensen, Søren Skovsen, Henrik Karstoft, Rene Gislum, The oil radish growth dataset for semantic segmentation and yield estimation, in: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2019, pp. 2703–2710.
- [35] Vi Nguyen Thanh Le, Benjamin Apopei, Kamal Alameh, Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods, *Inf. Process. Agric.* 6 (1) (2019) 116–131.
- [36] T. Ojala, M. Pietikainen, T. Maenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (7) (2002) 971–987.
- [37] Alex Olsen, Dmitry A. Konovalov, Bronson Philippa, Peter Ridd, Jake C. Wood, Jamie Johns, Wesley Banks, Benjamin Girgenti, Owen Kenny, James Whinney, Brendan Calvert, Mostafa Rahimi Azghadi, Ronald D. White, DeepWeeds: a multiclass weed species image dataset for deep learning, *Sci. Rep.* 9 (1) (2019).
- [38] Kavar Osorio, Andrés Puerto, Cesar Pedraza, David Jamaica, Leonardo Rodríguez, A deep learning approach for weed detection in lettuce crops using multispectral images, *AgriEngineering* 2 (3) (2020) 471–488.
- [39] Ciro Potena, Daniele Nardi, Alberto Pretto, Fast and accurate crop and weed identification with summarized train sets for precision agriculture, in: *Intelligent Autonomous Systems 14*, Springer International Publishing, Cham, 2017, pp. 105–121.
- [40] Edmund Sadgrove, Greg Falzon, David Miron, David Lamb, Fast object detection in pastoral landscapes using a Colour Feature Extreme Learning Machine, *Comput. Electron. Agric.* 139 (2017).
- [41] Edmund Sadgrove, Greg Falzon, David Miron, David Lamb, Fast object detection in pastoral landscapes using a multiple expert colour feature extreme learning machine, in: *Asian-Australasian Conference on Precision Pastures and Livestock Farming*, 2017.
- [42] Edmund Sadgrove, Greg Falzon, David Miron, David Lamb, Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (MEC-ELM), *Comput. Ind.* 98 (2018).
- [43] Edmund Sadgrove, Greg Falzon, David Miron, David Lamb, The segmented colour feature extreme learning machine: applications in agricultural robotics, *Agronomy* 11 (11) (2021) 2290.
- [44] Abhinav Sharma, Arpit Jain, Prateek Gupta, Vinay Chowdary, Machine learning applications for precision agriculture: a comprehensive review, *IEEE Access* 9 (2021) 4843–4873.
- [45] Lyndon N. Smith, Arlo Byrne, Mark F. Hansen, Wenhao Zhang, Melvyn L. Smith, Weed classification in grasslands using convolutional neural networks, in: Michael E. Zelinski, Tarek M. Taha, Jonathan Howe, Abdul A. Awwal, Khan M. Iftikharuddin (Eds.), *Applications of Machine Learning*, SPIE, 2019, p. 42.
- [46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the inception architecture for computer vision, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.
- [47] JingLei Tang, ZhiGuang Zhang, Dong Wang, Jing Xin, LiJun He, Research on weeds identification based on K-means feature learning, *Soft Comput.* 22 (22) (2018) 7649–7658.
- [48] Jonathan Tapson, Philip de Chazal, André van Schaik, Explicit Computation of Input Weights in Extreme Learning Machines, *Proceedings of ELM-2014*, vol. 1, Springer International Publishing, Cham, 2015, pp. 41–49.
- [49] Vi Nguyen Thanh Le, Giang Truong, Kamal Alameh, Detecting weeds from crops under complex field environments based on faster RCNN, in: *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, IEEE, 2021, pp. 350–355.
- [50] Muammer Turkoglu, Davut Hanbay, Leaf-based plant species recognition based on improved local binary pattern and extreme learning machine, *Phys. A, Stat. Mech. Appl.* 527 (2019) 121297.
- [51] Wenhao Zhang, Mark F. Hansen, Timothy N. Volonakis, Melvyn Smith, Lyndon Smith, Jim Wilson, Graham Ralston, Laurence Broadbent, Glynn Wright, Broad-leaf weed detection in pasture, in: *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, 2018, pp. 101–105.
- [52] Wentao Zhu, Jun Miao, Laiyun Qing, Constrained extreme learning machines: a study on classification cases, <https://arxiv.org/abs/1501.06115>, 2015.

3.3 Supplementary Material

The supplementary material to the published paper is provided here.

Colour Channel	Dataset			
	Bathurst Burr	Horehound	Thistle	Tussock
RGB	74.9±9.9	70.8±6.5[†]	71.6±2.5	78.4±5.2 [†]
YCbCr	79.5±7.7[†]	69.6±6.4 [†]	74.3±3.5[†]	79.4±4.1[†]

Table S1: All Datasets: Mean IoU for colour features. The best result for each dataset is in **bold**. Results shown are the mean ±standard deviation, for a 5 × 10 fold cross validation (Bathurst burr, horehound and thistle datasets), or for a set of 10 entirely independent train/test splits (tussock). Results which are not significantly different from the best result in the column are labeled with a [†].

Colour Channel		Number of Bins				
		4	6	9	12	15
Grey	Unsigned	58.8±8.0^{*†}	58.0±7.6 ^{*†}	57.7±7.1 ^{*†}	56.3±6.9 [†]	55.7±7.2 [†]
	Signed	52.9±6.6	59.4±6.6^{*†}	59.3±6.3 ^{*†}	58.7±6.0 ^{*†}	57.7±6.3 ^{*†}
ExG	Unsigned	40.7±4.5	45.9±7.0 [*]	46.0±6.6[*]	45.8±6.6 [*]	45.2±6.4 [*]
	Signed	36.5±3.7 ¹	44.1±6.6 [*]	44.4±6.4[*]	43.7±5.5 [*]	43.7±5.6 [*]
ExR	Unsigned	42.9±6.4	46.8±7.2	50.5±7.5[*]	50.3±7.5 [*]	49.8±7.4 [*]
	Signed	39.8±5.3	45.6±7.5 [*]	48.4±8.0 [*]	49.4±8.2 [*]	49.7±8.5[*]
ExGR	Unsigned	50.9±9.8	57.0±11.2 ^{*†}	58.7±9.8 ^{*†}	58.9±9.7^{*†}	58.9±10.2 ^{*†}
	Signed	46.7±9.4	55.2±9.3 [†]	58.2±9.7 ^{*†}	58.9±9.9 ^{*†}	59.1±9.8^{*†}

Table S2: Bathurst Burr Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. The best result for each row is in **bold**, for each column is underlined, and overall is *italicized*. Results shown are the mean ±standard deviation, for a 5 × 10 fold cross validation. Results which are not significantly different from the best result in the row (column) are labeled with a * ([†]). All results presented are for a hidden layer size of 1000, except where marked by ¹, ³ and ⁶, which correspond to a hidden layer size of 100, 300 and 600 respectively.

		Number of Bins				
Colour Channel	Orientation	4	6	9	12	15
Grey	Unsigned	41.8±3.7	45.2±3.2	46.3±3.3	46.6±3.2*	46.8±3.1*
	Signed	48.8±5.9	51.6±5.9*	48.3±4.5	46.5±3.6	45.6±3.3
ExG	Unsigned	54.3±7.4 [†]	57.7±7.9 [†]	58.7±7.8* [†]	58.8±7.8*[†]	58.5±7.8* [†]
	Signed	36.9±2.6	56.0±7.2	57.1±7.6*	57.1±7.5*	57.0±7.7*
ExR	Unsigned	37.1±2.7	42.3±4.6	45.0±5.0	45.2±5.0	46.3±5.1*
	Signed	40.2±4.1	42.9±4.9*	45.1±4.7*	45.8±4.5*	45.6±4.6*
ExGR	Unsigned	49.7±7.4	51.8±8.3	52.7±8.3*	52.6±8.2*	52.5±8.1*
	Signed	39.4±3.3	51.6±6.2*	52.7±7.6*	52.4±7.5*	52.1±7.7*

Table S3: Horehound Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. Table has been annotated in the same way as table S2.

		Number of Bins				
Colour Channel	Orientation	4	6	9	12	15
Grey	Unsigned	46.3±2.6*	46.9±2.3*	45.5±2.2	45.0±2.1	44.4±2.1
	Signed	53.4±3.6 [†]	57.6±3.6*[†]	57.1±3.4* [†]	55.8±3.0 [†]	54.9±2.9 [†]
ExG	Unsigned	38.3±1.7	51.6±3.2	52.5±3.3*	52.3±3.0*	53.0±3.2*[†]
	Signed	40.7±1.9	45.2±2.7	50.4±3.1*	50.8±3.0*	50.3±2.9
ExR	Unsigned	38.5±1.7	45.4±2.2	50.0±2.8	51.1±2.9*	50.0±3.0
	Signed	47.0±3.9	49.7±4.1	50.1±3.3	50.1±3.2	52.3±3.1*
ExGR	Unsigned	41.7±2.0	46.2±2.3	48.1±2.4*	48.6±2.6*	48.2±2.7
	Signed	50.3±4.1	53.5±3.7*	54.9±3.2*	54.9±3.2*[†]	54.6±3.2* [†]

Table S4: Thistle Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. Table has been annotated in the same way as table S2.

Colour Channel	Orientation	Number of Bins				
		4	6	9	12	15
Grey	Unsigned	75.4±2.1 [†]	75.8±2.1^{*†}	75.5±2.0 [†]	75.2±1.9 [†]	75.1±2.0 [†]
	Signed	61.8±2.6	75.3±1.9[*]	75.2±2.0 ^{*†}	74.8±1.9	74.5±1.9
ExG	Unsigned	68.8±2.2	73.0±2.1	73.6±2.3[*]	73.2±2.4	72.7±2.2
	Signed	51.9±2.7	71.8±1.9	73.4±2.4[*]	73.1±2.3	72.6±2.3
ExR	Unsigned	63.1±3.0	69.3±3.4	71.1±3.3	71.6±3.4[*]	71.2±3.3 [*]
	Signed	43.4±2.8	63.7±2.7	67.8±3.0	69.7±3.1[*]	69.7±3.2 [*]
ExGR	Unsigned	70.8±2.9	73.7±2.8	74.5±2.9^{*†}	74.4±3.0 ^{*†}	74.1±2.9 [†]
	Signed	51.7±2.6	72.1±3.0	73.7±3.0[*]	73.4±2.9	73.1±3.0

Table S5: Tussock Dataset: Mean IoU for HOG features for various choices of colour channel, signed orientation and number of bins. Table has been annotated in the same way as table S2.

Feature Type (Best)	Dataset			
	Bathurst Burr	Horehound	Thistle	Tussock
Colour	79.5±7.7 [†]	70.8±6.5	74.3±3.5[†]	79.4±4.1
LBP	81.6±7.7[†]	74.6±5.3[†]	71.8±3.3	83.4±3.5[†]
HOG	59.4±6.6	58.8±7.8	57.6±3.6	75.8±2.1

Table S6: All Datasets: Comparison of best feature for each feature type (colour, LBP and HOG). The best result for each dataset is in **bold**. Results shown are the mean ±standard deviation, for a 5 × 10 fold cross validation (Bathurst burr, horehound and thistle datasets), or for a set of 10 entirely independent train/test splits (tussock). Results which are not significantly different from the best result in the column are labeled with a [†].

Colour Channel	Rotationally Invariant	Number of Neighbours	Radius						
			1	2	3	4	6	8	10
Grey	Y	8	63.2±8.1	66.6±11.2	64.0±12.6	64.9±12.2	65.2±10.3	62.7±9.2	60.8±8.6
		16	-	70.3±9.8 [†]	70.6±11.7	71.4±10.5	71.8±9.2 [†]	69.3±9.0 [†]	65.6±9.2
		24	-	-	72.1±11.1	74.4±10.7^{*†}	73.5±9.3 ^{*†}	72.1±9.3 ^{*†}	67.7±9.2 [†]
	N	8	64.6±8.6 [†]	70.2±9.2 [*]	69.6±9.5	69.3±9.8	66.3±9.2	62.6±8.9	60.6±8.1
		16	-	71.8±9.5 ^{*†}	72.1±10.1[*]	70.7±10.1 [*]	64.8±10.5	59.8±10.2	57.9±9.2
		24	-	-	70.1±10.3	70.3±10.7	65.0±11.2	61.5±10.5	57.6±10.2
ExG	Y	8	71.3±11.4^{*†}	69.9±11.5 ^{*†}	68.5±11.7 [*]	68.1±11.8 [*]	64.2±11.6	64.0±11.2	64.8±10.3
		16	-	69.3±11.5 ^{*†}	67.8±11.4 [*]	68.3±10.9 [*]	67.1±10.9	67.2±10.9 [†]	67.4±10.3 [†]
		24	-	-	67.9±11.3 [*]	68.0±11.7	67.2±10.9	67.8±10.8 [†]	66.8±10.2
	N	8	73.1±11.8^{*†}	68.1±11.0	66.4±11.7	67.5±11.5	63.1±10.8	61.8±10.2	62.2±10.7
		16	-	68.7±11.5	63.6±11.2	63.9±11.7	61.8±11.1	60.7±10.3	60.7±10.1
		24	-	-	63.0±11.9	65.6±11.6	61.2±11.1	61.1±10.4	60.8±10.5
ExR	Y	8	67.5±6.2 ^{*†}	60.6±9.1	62.5±9.4	67.8±9.6	64.7±10.2	60.4±9.9	55.1±9.5
		16	-	62.2±7.9	67.5±8.4	70.0±9.5 ^{*6}	67.8±10.2	66.2±10.4 ⁶	60.6±9.8
		24	-	-	68.4±6.9	71.8±8.0[*]	68.3±10.0 ^{*6}	67.1±9.8 ³	62.9±9.7
	N	8	67.5±9.0 [†]	66.2±8.0	62.1±9.7	71.7±7.9[*]	65.2±9.7	60.9±8.7	56.2±8.7
		16	-	65.6±6.6	67.0±8.5	69.7±8.3 [*]	67.2±10.2	67.7±9.8	59.4±9.1
		24	-	-	65.8±8.4	66.9±8.9	69.0±9.6 [*]	64.7±10.2	64.1±9.6
ExGR	Y	8	67.6±8.8 [†]	68.9±10.8	68.3±9.6	71.8±9.9	66.2±11.9	63.4±10.1	57.7±9.9
		16	-	74.2±8.6 [†]	80.1±8.3 ^{*†}	77.0±10.4 [†]	73.3±10.6	73.2±9.8 ^{†3}	70.1±8.6
		24	-	-	81.6±7.7^{*†}	78.8±8.9 [†]	77.1±11.1 ^{†6}	73.4±11.4 [†]	72.9±9.6 ^{†6}
	N	8	69.9±10.6 [†]	72.2±9.5	70.3±9.4	78.4±6.6 ^{*†}	70.6±10.9	65.7±10.3	59.2±9.9
		16	-	76.4±8.6 [†]	79.6±8.4[*]	76.3±9.2	72.9±11.0	73.0±9.9 [†]	69.0±9.2
		24	-	-	79.5±9.2 [*]	77.5±9.9 ^{*†}	75.3±11.3	72.1±11.6 [†]	71.5±10.5 [†]

Table S7: Bathurst Burr Dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. The best result for each set of three rows (which corresponds to fixing the colour channel and rotational invariance) is in **bold**, for each column (which corresponds to fixing the radius) is underlined, and overall is *italicized*. Results shown are the mean $\pm$ standard deviation, for a 5×10 fold cross validation. Results which are not significantly different from the best result in the set of three rows (column) are labeled with a * ([†]). All results presented are for a hidden layer size of 1000, except where marked by ¹, ³ and ⁶, which correspond to a hidden layer size of 100, 300 and 600 respectively.

Colour Channel	Rotationally Invariant	Number of Neighbours	Radius						
			1	2	3	4	6	8	10
Grey	Y	8	$69.8 \pm 6.5^\dagger$	71.4 ± 6.6	72.2 ± 6.8	70.7 ± 6.6	69.4 ± 6.0	70.4 ± 5.2	71.4 ± 4.9
		16	-	$73.1 \pm 5.8^\dagger$	$74.2 \pm 5.8^{*\dagger}$	73.2 ± 5.9	$73.4 \pm 5.7^\dagger$	$73.7 \pm 4.9^{*\dagger}$	$73.4 \pm 4.6^{*\dagger}$
		24	-	-	$74.3 \pm 5.5^{*\dagger}$	$74.6 \pm 5.3^{*\dagger}$	$73.6 \pm 5.5^{*\dagger}$	$73.6 \pm 4.9^{*\dagger}$	$73.5 \pm 4.4^{*\dagger}$
	N	8	$68.9 \pm 6.6^\dagger$	$71.1 \pm 6.3^\dagger$	$72.0 \pm 6.0^*$	70.4 ± 6.2	69.1 ± 5.8	70.1 ± 5.0	$71.1 \pm 4.8^*$
		16	-	$71.8 \pm 5.8^\dagger$	$73.3 \pm 5.6^{*\dagger}$	$72.3 \pm 6.1^*$	$72.8 \pm 5.7^{*\dagger}$	$72.4 \pm 5.1^*$	$71.3 \pm 4.9^*$
		24	-	-	72.5 ± 5.9	$72.9 \pm 6.0^*$	$72.1 \pm 6.0^*$	$71.5 \pm 5.3^*$	69.9 ± 4.9
ExG	Y	8	$66.7 \pm 8.4^{*\dagger}$	$66.5 \pm 8.4^*$	$67.0 \pm 8.6^*$	$65.8 \pm 8.0^*$	64.5 ± 7.3	63.6 ± 6.7	62.2 ± 6.5
		16	-	$66.8 \pm 8.9^*$	$67.4 \pm 8.5^*$	$66.8 \pm 8.1^*$	$65.7 \pm 7.5^*$	$64.7 \pm 7.1^*$	63.3 ± 7.0
		24	-	-	$67.5 \pm 8.8^*$	66.0 ± 8.2	$65.9 \pm 7.7^*$	64.7 ± 7.5	62.6 ± 7.2
	N	8	$65.9 \pm 8.1^*$	$65.4 \pm 8.0^*$	$64.5 \pm 8.6^*$	63.7 ± 7.9	62.9 ± 7.0	$62.6 \pm 6.3^*$	$62.0 \pm 6.6^*$
		16	-	63.8 ± 8.9	63.4 ± 9.1	62.6 ± 8.1	60.4 ± 6.7	57.6 ± 5.8	55.9 ± 6.1
		24	-	-	63.1 ± 9.2	61.8 ± 8.9	60.5 ± 6.4	59.6 ± 6.5	55.2 ± 6.1
ExR	Y	8	$64.3 \pm 6.5^*$	$62.1 \pm 7.1^*$	$61.9 \pm 7.1^*$	59.5 ± 6.6	54.5 ± 7.1	53.3 ± 6.6	56.0 ± 6.3
		16	-	61.7 ± 7.3	$63.0 \pm 7.1^*$	60.8 ± 7.3	60.8 ± 6.0	57.8 ± 5.4	57.4 ± 5.7
		24	-	-	$62.1 \pm 7.8^*$	61.0 ± 7.3	58.5 ± 5.8	59.7 ± 4.9	$62.9 \pm 5.4^*$
	N	8	$66.3 \pm 6.1^*$	56.5 ± 7.3	57.2 ± 6.8	57.3 ± 6.9	54.2 ± 5.9	54.0 ± 6.7	56.6 ± 6.3
		16	-	58.0 ± 6.9	60.3 ± 6.9	57.9 ± 6.1	58.5 ± 4.6	56.7 ± 5.0	55.3 ± 5.6
		24	-	-	58.0 ± 6.9	57.4 ± 6.8	57.2 ± 5.4	58.4 ± 4.9	60.3 ± 4.5
ExGR	Y	8	$67.2 \pm 6.8^\dagger$	66.5 ± 7.7	67.1 ± 7.4	64.8 ± 7.3	65.0 ± 7.5	63.3 ± 6.3	62.6 ± 5.6
		16	-	67.9 ± 7.5	$70.4 \pm 7.1^*$	68.5 ± 7.6	68.9 ± 7.5	67.9 ± 7.0	66.8 ± 6.4
		24	-	-	$70.2 \pm 6.9^*$	$70.7 \pm 7.4^*$	68.3 ± 6.9	68.2 ± 6.6	68.0 ± 6.3
	N	8	$67.9 \pm 6.5^{*\dagger}$	$65.6 \pm 7.9^*$	65.7 ± 7.2	63.9 ± 6.5	64.2 ± 5.8	63.0 ± 5.2	61.8 ± 5.3
		16	-	63.6 ± 8.6	$68.1 \pm 6.9^*$	65.6 ± 7.1	$66.9 \pm 6.3^*$	65.5 ± 5.9	63.1 ± 5.7
		24	-	-	66.7 ± 7.1	$67.9 \pm 7.4^*$	$65.5 \pm 6.5^*$	65.6 ± 6.1	63.8 ± 5.7

Table S8: Horehound Dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Table has been annotated in the same way as table S7.

Colour Channel	Rotationally Invariant	Number of Neighbours	Radius						
			1	2	3	4	6	8	10
Grey	Y	8	66.3±3.7	67.8±4.1	69.1±3.9 [†]	68.7±3.8	68.4±3.7	68.7±3.6	68.6±3.6
		16	-	70.0±4.0 [†]	70.5±3.9 [†]	71.0±3.8	70.9±3.5 ^{*†}	70.1±3.4	69.2±3.6
		24	-	-	70.7±3.8 [†]	71.6±3.8^{*†}	71.0±3.6 ^{*†}	70.4±3.5 [†]	69.5±3.6 [†]
	N	8	65.6±3.7	66.2±3.7	66.3±3.7	66.8±3.7	67.6±3.7	68.3±3.5[*]	68.2±3.5 [*]
		16	-	64.7±3.5	64.7±3.8	66.1±3.6	67.8±3.5 [*]	68.0±3.3 [*]	67.7±3.3
		24	-	-	63.8±3.7	65.4±3.6	67.3±3.5	67.7±3.3 [*]	67.4±3.3
ExG	Y	8	71.8±3.9^{*†}	71.2±3.6 ^{*†}	70.7±3.5 [†]	68.0±4.7 [†]	65.2±5.2	65.4±5.1	64.4±5.2
		16	-	71.1±3.6 ^{*†}	71.0±3.7 ^{*†}	67.9±4.7 [†]	66.4±4.9	64.8±5.2	64.4±5.3
		24	-	-	70.8±3.8 [†]	68.5±4.6 [†]	66.2±4.9	65.4±4.9	64.1±5.2
	N	8	71.7±3.1^{*†}	70.0±3.6	69.1±3.9	67.8±4.7 [†]	65.8±5.0	65.7±5.1	64.9±5.2
		16	-	69.3±3.7	67.8±4.2	66.4±4.9	64.5±5.0	62.8±5.2	62.7±5.2
		24	-	-	66.6±4.5	64.8±5.0	63.0±5.1	61.4±5.0	60.6±5.2
ExR	Y	8	65.4±3.7 [*]	64.1±3.5	63.5±3.9	65.3±3.6 [*]	59.7±3.6	59.7±3.6	59.4±3.5
		16	-	65.7±3.5 [*]	66.3±4.1[*]	65.5±3.8 [*]	65.4±3.8 [*]	64.0±3.5	60.5±3.5
		24	-	-	65.9±4.0 [*]	66.1±3.7 [*]	63.9±3.7	63.7±3.5	62.2±3.5
	N	8	68.6±3.3[*]	63.9±3.5	62.6±3.4	64.9±3.3	60.0±3.7	61.3±3.7	60.6±3.5
		16	-	62.5±3.2	63.0±3.2	64.2±3.6	62.2±3.6	60.3±3.7	58.4±3.5
		24	-	-	62.8±3.2	62.4±3.4	60.6±3.7	59.5±3.4	57.9±3.3
ExGR	Y	8	66.5±3.4	66.4±4.1	67.2±4.2	67.7±3.8 [*]	61.2±3.1	61.6±3.0	61.7±3.1
		16	-	67.5±4.4 [*]	68.3±4.4^{*†}	67.8±3.9 [*]	68.0±3.6 [*]	67.6±3.7 [*]	64.9±3.4
		24	-	-	67.9±4.4 [†]	66.1±3.7	67.8±3.6 [*]	66.9±3.6 [*]	65.3±3.4
	N	8	68.5±3.5[*]	66.3±4.0	66.5±4.0	67.4±4.0 [*]	61.3±3.0	62.5±3.3	61.4±3.3
		16	-	66.1±4.0	65.8±4.2	64.4±3.8	63.7±3.6	63.4±3.4	60.6±3.3
		24	-	-	65.2±4.1	63.3±3.8	63.2±3.5	62.4±3.2	60.2±3.0

Table S9: Thistle Dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Table has been annotated in the same way as table S7.

Colour Channel	Rotationally Invariant	Number of Neighbours	Radius						
			1	2	3	4	6	8	10
Grey	Y	8	75.5±5.0	75.6±4.6	76.6±4.7	78.8±4.3	79.6±4.6	76.1±3.8	70.1±3.1
		16	-	80.0±4.4	81.1±4.3	83.3±4.2 ^{*†}	83.1±4.2 ^{*†}	81.3±4.7 [†]	79.7±4.7 [†]
		24	-	-	82.3±4.4 ^{*†}	83.2±4.2 ^{*†}	83.4±4.1^{*†}	81.4±4.7 [†]	79.9±4.9 [†]
	N	8	82.4±4.0 [†]	83.1±3.5^{*†}	82.4±3.0 [*]	82.3±3.2 ^{*†}	81.2±3.7	77.5±3.0	72.9±2.5
		16	-	82.6±3.6 [†]	82.0±3.3	82.6±3.5 ^{*†}	81.7±4.2	80.1±4.3 [†]	79.2±3.7
		24	-	-	82.6±3.3 ^{*†}	82.0±3.6	81.4±4.3	79.6±4.3	78.7±4.0
ExG	Y	8	76.8±3.6	79.7±3.6 [*]	78.8±4.0	79.6±4.7 [†]	80.5±4.0	79.0±3.9	72.3±4.6
		16	-	79.5±3.3 [*]	78.4±4.1	80.4±4.3	81.3±4.0 [*]	81.6±4.1^{*†}	80.9±4.0 [†]
		24	-	-	79.3±3.8	80.5±4.3	81.4±3.8 [*]	80.9±4.4 [†]	80.6±4.4 [†]
	N	8	81.3±2.9 [†]	83.4±2.9 ^{*†}	83.4±3.5^{*†}	82.8±4.0 [†]	81.6±3.8	79.2±4.0	74.4±3.4
		16	-	82.4±3.3	81.8±3.7	81.7±4.3 [†]	81.7±4.3	81.1±4.3 [†]	80.3±4.2 [†]
		24	-	-	80.8±4.0	81.1±4.1	81.0±4.0	80.8±4.4 [†]	80.2±4.5 [†]
ExR	Y	8	75.2±3.9 [*]	70.9±6.0	69.7±5.1	69.1±4.0	73.8±4.6	68.4±4.4	60.1±2.7
		16	-	73.9±4.4	75.2±5.0 [*]	73.7±5.1	76.1±5.3[*]	74.5±5.2	74.2±4.7
		24	-	-	73.0±5.3	75.6±4.3 [*]	74.7±5.4	74.5±5.8	73.9±4.8
	N	8	79.5±4.4[*]	76.3±5.2	74.2±5.3	74.6±4.6	75.9±4.3	71.8±3.8	63.6±2.7
		16	-	75.9±5.3	76.4±5.4	77.2±5.5	76.1±5.1	73.2±4.8	71.7±4.2
		24	-	-	75.3±4.8	75.6±4.8	75.0±5.1	73.4±5.0	71.6±4.8
ExGR	Y	8	75.7±3.9	72.6±3.8	72.1±4.4	75.8±5.0	78.3±4.2	76.7±4.1	70.0±3.8
		16	-	75.5±4.0	76.1±4.3	79.8±4.5[*]	78.4±4.3	77.0±4.8	77.8±4.9
		24	-	-	75.6±4.6	78.1±4.4	78.2±4.1	77.7±4.6	75.4±5.1
	N	8	81.1±3.9^{*†}	79.7±2.9	80.1±3.0 [*]	81.1±3.8 [*]	80.3±3.7 [*]	78.3±3.7	74.0±2.8
		16	-	75.4±4.2	78.7±3.9	79.7±4.2 [*]	78.2±4.5	77.8±4.3	77.4±4.4
		24	-	-	76.9±4.6	78.7±4.3	77.1±4.3	75.4±4.6	73.9±4.9

Table S10: Tussock dataset: Mean IoU for LBP features for various choices of colour channel, rotational invariance, number of neighbours, and radius. Table has been annotated in the same way as table S7.

Higher Degree Research Thesis by Publication
University of New England

STATEMENT OF AUTHORS' CONTRIBUTION

We, the Research Master/PhD candidate and the candidate’s Principal Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate’s contribution as indicated in the *Statement of Originality*.

	Author’s Name (please print clearly)	% of contribution
Candidate	Jonathan Ford	33
Other Authors	Edmund Sadgrove	33
	David Paul	33

Name of Candidate:

Jonathan Ford

Name/title of Principal Supervisor:

Dr Edmund Sadgrove



13th October 2025

Date



14/10/25

Date

Candidate

Principal Supervisor

**Higher Degree Research Thesis by Publication
University of New England**

STATEMENT OF ORIGINALITY

We, the Research Master/PhD candidate and the candidate's Principal Supervisor, certify that the following text, figures and diagrams are the candidate's original work.

Type of work	Page number/s
Journal Paper	79-90
Supplementary Material	92-98

Name of Candidate: Jonathan Ford

Name/title of Principal Supervisor: Dr Edmund Sadgrove



Candidate

13th October 2025

Date



Principal Supervisor

14/10/25

Date

Chapter 4

Joint Plant-Spraypoint Detector with ConvNeXt Modules and HistMatch Normalization

4.1 Introduction

This paper, titled “Joint Plant-Spraypoint Detector with ConvNeXt Modules and HistMatch Normalization” [61] presents a study on the question of whether a joint plant and spraypoint detector can be developed for use in Australian pastures (Research Question 4). In addition to this, it also considered Research Questions 2, 3 and 6, which respectively relate to the use of CNNs for segmentation in Australian pastures (including modern architectures), the development of real-time networks, and the development of networks that generalize well to field conditions not seen during training.

This paper was inspired by the studies of Lottes et al. [150] and Zhang et al. [273], which looked at the development of joint plant and stem localization networks in early stage crop fields. The purpose of such networks is that they allow for more varied treatment options. For example, in a crop field, some weeds may be sprayed with herbicide, requiring the entire plant to be identified, whereas other weeds may be destroyed using mechanical means, such as a stamping tool, or jet of steam, which is only applied to the stem.

Likewise, in pastures, some herbicides, such as glyphosate, are applied to the entire plant, as they are taken up mostly by the leaves of the plant. On the other hand, other herbicides, such as flupropanate, are mostly absorbed by the root system, and so it makes sense to apply them only to the centre of the plant, or spraypoint, where the roots are more densely located.

In addition to developing a joint plant and spraypoint detector, this paper also set out to determine how well such models generalize to growing conditions not seen during training (Research Question 6). This is a relevant question, as it is difficult to capture the entire range of conditions in the training dataset, including background pasture elements, soil appearance and moisture content, but it is important to produce a model which can respond to these different environmental factors during

deployment.

For this reason, three separate datasets of serrated tussock, captured over a period of ten months at Yelangelo, NSW were utilized, with two datasets (A and C) used in training and validation, and the third dataset (B) reserved for testing. As shown in Figure 3(c) of the published paper, some models were frequently confused by variations in environmental conditions, often incorrectly classifying most or all of an image as positive, even when the target weed was only partially present.

At the time that this research was conducted, there were not sufficient annotated datasets for other target species, and so Research Question 5, concerning a multi-species detector was not considered.

An overview of the datasets used in this study is presented in table 4.1.

Image Resolution	Annotation Level	Dataset	Number of Images	Class Representation		Spraypoint Count
				Negative	Positive	
768×576	1×1 pixel	A	411	0.67	0.33	286
		B	295	0.55	0.45	224
		C	556	0.75	0.25	378

TABLE 4.1: Dataset summary including number of images and class representation. Image resolution and annotation level are consistent across all datasets.

4.2 Published Paper

This paper was published in Precision Agriculture, which is ranked by SJR¹ as a Q1 journal in Agricultural and Biological Sciences (miscellaneous) for 2024. As of 7th July 2026, this paper had 4 citations on Google Scholar.

¹<https://www.scimagojr.com/journalsearch.php?q=86245&tip=sid>

Precision Agriculture (2025) 26:24
<https://doi.org/10.1007/s11119-024-10208-y>



Joint plant-spraypoint detector with ConvNeXt modules and HistMatch normalization

Jonathan Ford¹ · Edmund Sadgrove¹ · David Paul¹

Accepted: 17 November 2024 / Published online: 22 January 2025
© The Author(s) 2025

Abstract

Context Serrated tussock (*Nassella trichotoma*) is a weed of national significance in Australia which offers little to no nutritional value to livestock, and has the potential to reduce carrying capacity and agricultural return of infested pastures.

Aims The aim of this study was to adapt existing Convolutional Neural Networks (CNNs) for plant segmentation and spraypoint detection in the challenging environments of pastures.

Methods CNNs that were designed for joint plant and stem segmentation in crop fields were repurposed for dual-task applications in pastures. Given the poor performance of these models in complex pasture environments, a new model drawing inspiration from the recently proposed ConvNeXt was developed, tested for its effectiveness on unseen field data, and enhanced with a novel normalization technique, called HistMatch.

Key results Experimentation demonstrated that unlike pre-existing models, which were designed for the simpler environments encountered in early-stage crop fields, our model was able to generalize well to growing conditions not seen during training, achieving 0.807 mIoU and 0.796 F1-score for the plant and spraypoint tasks respectively. This is in comparison to pre-existing models, which achieved 0.270 - 0.454 mIoU and 0.073 - 0.496 F1-score for the same tasks. These results were further improved to 0.854 mIoU and 0.806 F1-score using HistMatch normalization. In spite of greater model complexity, our model had a inference time of 15.7 ms which was comparable to pre-existing models, and suitable for real-time applications.

Conclusion Models with greater complexity are required for the relatively complex environments encountered in pastures, but this greater complexity need not come at the expense of real time capability. HistMatch normalization can improve model accuracy, and is particularly effective in cases where models are struggling to generalize well to testing conditions that vary significantly from those seen during training.

Implications and impacts The successful adaptation and improvement of CNNs for weed management in pastures could significantly reduce the reliance on blanket herbicide application. HistMatch normalization could also be considered for other agricultural applications, including weed management and disease detection in crop fields and orchards.

Keywords Convolutional Neural Networks · ConvNeXt · Computer vision · Site-specific weed management · Precision agriculture.

Extended author information available on the last page of the article

Introduction

The use of computer vision in agriculture is a promising area of research, which offers, among other things, the potential for effective detection and treatment of weeds. Site-specific weed management (SSWM) is the process of detecting weeds in a field, and applying an appropriate treatment such as herbicide only where required. This reduces the use of chemicals, which can be expensive and have the potential to cause harm to both people and the broader natural environment (Liu & Bruch, 2020). Through automatic weed management, human effort is also reduced (Osorio et al., 2020). Serrated tussock (*Nassella trichotoma*, hereafter tussock) is a weed of national significance in Australia, which offers little to no nutritional value to livestock, and has the ability to significantly reduce the carrying capacity of infested land (Campbell, 1998). It was estimated in 2018 that the annual cost of weeds, including control costs and the cost of lost production in the Australian livestock industry was \$AUD2,409 million (\$USD1,583¹) (McLeod, 2018). The cost of control for tussock has been estimated to be well in excess of \$AUD100 (\$USD66) per hectare of infested land (Johnson et al., 2022).

Computer vision is a field of artificial intelligence that enables computers to interpret, analyze, and understand visual information from the world, such as images or videos. Advanced techniques in computer vision, include neural networks, such as convolutional neural networks (CNNs), as well as other methods, including Gaussian process regression. Such techniques have been applied to the detection of weeds in pastures, including dockleaf (Binch & Fox, 2017; Smith et al., 2019; Zhang et al., 2018; Kounalakis et al., 2019; Schori et al., 2019), thistle species (Sadgrove et al., 2017, 2017, 2018), horehound (Sadgrove et al., 2021), and grass species (Farooq et al., 2019). Other agricultural applications include plant diseases detection (Saleem et al., 2019; Li et al., 2021), yield prediction (Van Klompenburg et al., 2020; Muruganantham et al., 2022), and price forecasting (Jin & Xu, 2024, 2024).

Studies on the detection of tussock in the literature are uncommon, and typically include hand crafted feature extraction, such as local binary patterns (Ojala et al., 2002) and histograms of oriented gradients (Dalal & Triggs, 2005). In one study (Farooq et al., 2019), hyperspectral images of tussock and other grass species were used to develop a support vector machine (Cortes & Vapnik, 1995) based classification framework, with an accuracy of 80.90%. In another study (Ford et al., 2023), utilising a shallow network trained using the extreme learning machine algorithm (Guang-Bin et al., 2004), images containing tussock and background pasture elements were segmented, with a mean Intersection over Union (mIoU) of 87.6%.

The use of multipurpose networks, to simultaneously carry out two (or more) tasks is one interesting area of active research in agricultural computer vision. An example is the use of joint plant segmentation and stem localization networks for use in crop fields (Lottes et al., 2018; Zhang et al., 2020). Such models are Convolutional Neural Networks (CNNs), built using the typical encoder/decoder structure of segmentation models, but with two decoder arms, one for each task. This “joint” encoder structure reduces the computational complexity of the network as a whole (as compared to a separate encoder for each task), and has been demonstrated to produce superior feature maps, allowing for more robust results, especially for the stem localization task (Lottes et al., 2018; Zhang et al., 2020).

¹Based on the average exchange rate for the 12 months to June 2024

The advantage of having a joint plant and stem localization network is that it allows for the application of more varied treatments, including herbicide, physical removal of weeds, or fertilization of crop plants. Specifically, while herbicide and fertilizer may be applied to the whole plant, physical removal is often only applied to the stem, using methods such as a metal stamping tool, or jet of steam (Lottes et al., 2018; Kounalakis et al., 2018). In pastures, herbicide may either be applied to the whole plant, if uptake is through the leaves, or the centre of the plant (hereafter the spraypoint), if uptake is through the root system. The latter is true for Flupropanate which is commonly used to treat tussock, due to its ability to remain active in the ground for long periods of time².

To the best of our knowledge, no studies have been conducted on the development of a combined plant and spraypoint detector specifically for use in pastures. To address this gap, we undertook a controlled study in this area. In order to minimize complicating factors, all data were collected at a single location, with images captured over a period of nine months, encompassing conditions with varying levels of drought stress. This research paves the way for further exploration of multipurpose detectors in pastures, enabling more diverse treatment options and ultimately leading to improved outcomes for farmers. In addition to pre-existing models, a new model was developed, based on elements of ConvNeXt (Liu et al., 2022), which was considered due to its superior performance on complex image tasks such as ImageNet (Russakovsky et al., 2015).

Although previous work utilized the arrangement of crop plants in lines in order to produce models which generalized well to new field conditions (Lottes et al., 2020), this technique is not available in pastures, where no such uniform arrangement of plants is present. For this reason, and in order to assist with model generalization, a new normalization technique was developed, which we call HistMatch normalization. This technique generates a more uniform probability distribution for network input compared to pre-existing methods.

The major contributions of the paper are:

1. Development of a joint plant and spraypoint detector for use in pastures, based on pre-existing models (Lottes et al., 2018; Zhang et al., 2020), as well as a new design utilising elements of ConvNeXt (Liu et al., 2022). To the best of our knowledge, this is the first example of this kind of multipurpose detector designed specifically for use in pastures.
2. Testing of the generalization ability of models to new datasets with different growing conditions.
3. Development of a new normalization method, called HistMatch normalization, for use in cases where the pixel intensity distributions of test images may vary significantly from those seen during training. The remainder of the paper is structured as follows. Section 2 provides the approach taken in this study, including pre-existing models, our model, the dataset used, implementation details, normalization techniques used, and experiments conducted. Section 3 presents the results of this study, while Sect. 4 provides an analysis of these results. Finally Sect. 5 provides concluding remarks, and identifies some possible future directions.

²Information on the treatment of tussock can be found at <https://weeds.dpi.nsw.gov.au/Weeds/serratedtussock>

Methods

Pre-existing models

For the dual plant segmentation and stem detection task, two pre-existing studies were found. In the first, by Lottes et al. (Lottes et al., 2018), a model with a single encoder, and two decoders was developed and tested on images captured in early stage sugar beet fields, achieving a mean average precision (Everingham et al., 2010) of 0.753–0.792 for the stem task, and 0.838–0.873 for the plant task. In the second study by Zhang et al. (Zhang et al., 2020), an alternative network structure was developed and tested on images of cauliflower seedlings growing in fields with very high weed presence, achieving an F-measure score of 0.974 for the plant segmentation task, and detection rate of 0.978 for the stem task.

The primary building block of Lottes' model (Lottes et al., 2018) is the dense block, which is composed of L densely connected 3×3 convolutional layers, each of which outputs G feature maps, where G is called the growth rate. The dense block also makes use of bottleneck layers, which are implemented by a point-wise convolution, in order to reduce computational complexity. Downsampling operations are implemented using 5×5 convolutional layers, with a stride of 2, and in total, there are three downsampling operations in the encoder. Upsampling operations in the decoder arms are implemented using transpose convolutions (Zeiler et al., 2010) with a kernel size of 2×2 and a stride of 2. Skip connections are implemented using concatenation along the filter axis. An overview of the dense block, and the macro structure of Lottes' model is given in supplementary figure S1.

Zhang's model utilizes an encoder, which consists of multiple inverted residual modules (IRMs) as used in MobileNet V2 (Sandler et al., 2018). IRMs utilize depth-wise and point-wise convolutions to reduce computational complexity while maintaining model accuracy. Point-wise convolutions are convolutions with a kernel size of 1×1 , whereas depth-wise convolutions are a form of convolution in which each input feature is convolved separately with its own corresponding filter. Downsampling is achieved by using a stride of 2 in the depth-wise convolution of the first IRM in each stage. At the macro scale, different decoder arms branch off the backbone at different resolutions for both the segmentation and the stem tasks. There is bidirectional feature map exchange between adjacent intra-task decoder arms in order to improve feature representation. There is also a cross-task feature fusion linking the coarsest segmentation decoder to each of the stem decoders. This forces the stem decoder to focus on those sections of the image in which the plant is located, while ignoring irrelevant background details. Each decoder arm produces its own detection mask, which is fused into a single mask for each of the two tasks (plant segmentation and stem localization). In addition, deep supervision is used to directly train each output of the decoder arms. An overview of the IRM, as well as the macro structure of Zhang's model is given in supplementary figure S2.

New model

Minimizing the number of parameters in a model is often seen as a desirable design choice (see for example (McCool et al., 2017; Khan et al., 2020; Abdalla et al., 2019)). In this study, however, focus was instead on the development of a model which had training and inference time that were similar to (or better than) pre-existing models found in the literature. These

are often more important considerations, because many iterations of training are required in order to optimize hyperparameter choices, and inference time is a critical factor for any model that is to be deployed in real time.

For our model, taking inspiration from Lottes' and Zhang's models, a joint encoder with two decoders for the spraypoint and plant was used. Starting with Zhang's model as a base, elements of ConvNeXt (Liu et al., 2022) were incorporated into our network structure, as listed below:

1. "Patchify" stem: which is an initial convolutional layer with a kernel size and stride of 4.
2. ConvNeXt modules: consisting of an inverted bottleneck design, a large kernel size for the depthwise convolutions, to improve spatial awareness, and fewer activation and normalization layers, which slightly reduces computational complexity. In addition, the depth-wise convolution is brought to the front, and the expansion rate is reduced to 4, which further reduces parameters and FLOPs. The ConvNeXt module is shown in Fig. 1a.
3. An adjustable learning rate scheduler with a linear warmup followed by cosine decay.
4. Adjusted stage ratio: with four ConvNeXt modules in the first and third stages, and twelve in the second stage. Other ConvNeXt elements, such as GELU activation (Hendrycks & Gimpel, 2016), Layer Normalization (Ba et al., 2016), and Layer Scale (Touvron et al., 2021) were not included in our design, because they were found to produce consistently worse results during preliminary testing. The stage ratio, which include a larger number of modules at the $16\times$ downsampling level, is a feature of the ConvNext model (Liu et al., 2022), and allows for a larger number of filters, while reducing the overall FLOPs of the network.

The use of the patchify stem also reduces the computational complexity of the model, by greatly reducing the resolution of the feature maps quickly, and allowing for a larger number of filters. For this reason, we also elected to maintain Zhang's downsampling technique, which is applied to the first module of each stage. Another result of the patchify stem was that the final output of our encoder was at a resolution downsampled by a factor of $32\times$ as opposed to the factor of $16\times$ for Zhang's model. The number of filters used in the patchify stem, and stages one to three were 64, 64, 128 and 128 respectively. The number of filters in the decoder for both the spray point and plant segmentation task were 64. Our network used only a single $32\times$ downsampled side output for the spraypoint task, and two side outputs at $32\times$ and $8\times$ downsampling for the plant segmentation task. The final result was a model which had far more parameters than either Lottes' or Zhang's model, but which had a comparable inference time, and could be trained roughly three times faster. The macrostructure of our model is show in Fig. 1b.

Dataset

The images used in this study were collected at Yelangello, NSW (lat. -34.863 , long. 149.169) on three separate occasions spaced over 9 months (Oct-21, Apr-22, Jul-22). The growing conditions were diverse, both in terms of drought stress, as well as the length of the background pasture elements, as can be seen in the example images given in Fig. 3. Images

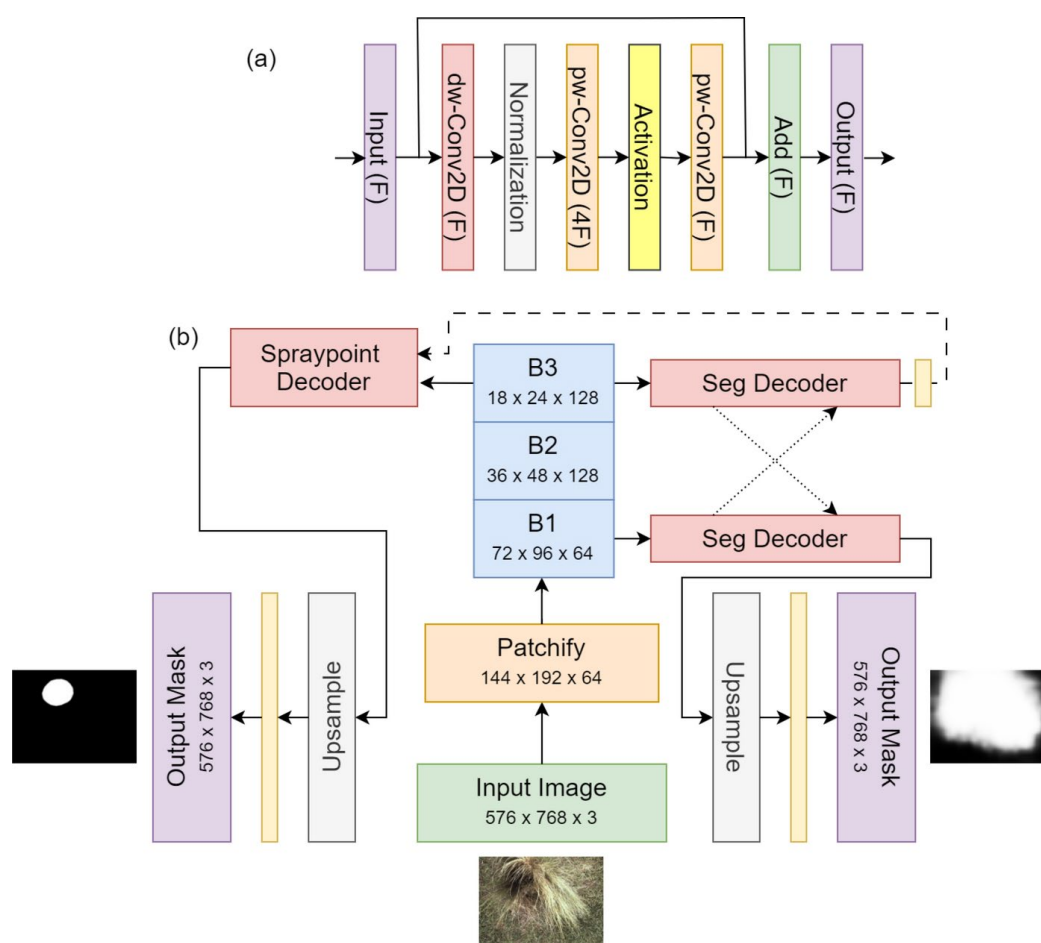


Fig. 1 Our model. **(a)** A ConvNeXt block, with a modified bottleneck design and fewer activation and normalization layers (which are explicitly shown here). **(b)** Macro structure showing the input image (green), patchify stem (orange), backbone (blue), decoders (red) and the output feature maps (purple). Loss layers are shown in yellow, and include losses for deep supervision (small rectangles), as well as for the final outputs of the model (large rectangles). Arrows display the flow of information through the model. In particular, the cross-task feature fusion is given with a dashed line, and the bidirectional feature map exchanges are given with a dotted line

were annotated using the VGG image annotator (Dutta & Zisserman, 2019), which allowed for a compromise between annotation accuracy and effort required per image. Three classes were used, being the negative, positive, and hard class, with the latter being for regions of the image where it was not clear whether or not tussock was present. The left section of Fig. 2 gives a breakdown of the number of images coming from the separate collection time for each of the various train, validation and test sets used in this study, while the right section shows the sets that were used in each of the experiments in this study.

Images were resized to 768×576 pixels for all models, which corresponded to a ground sampling distance of $\sim 1\text{mm/pixel}$. Data augmentation, consisting of random rotations, zooms, vertical and horizontal flips, and contrast and brightness shifts were used for all experiments. However, because Tensorflow utilizes the CPU for all dataset manipulations (including augmentations, in our case), it was found that this became a significant computational bottleneck for training. For this reason, a bank of augmented images were produced

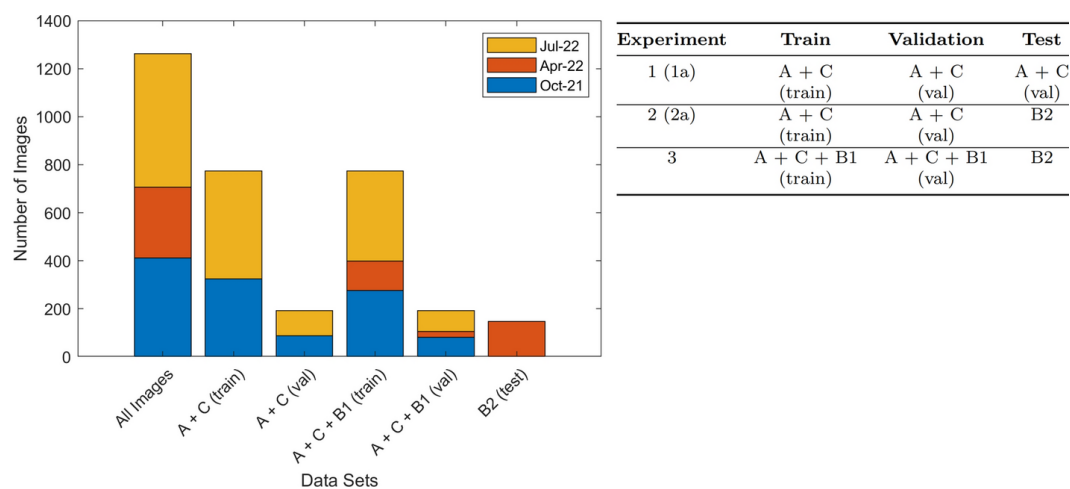


Fig. 2 The datasets used in this study. On the left is a breakdown of the number of images coming from each of the three collection dates. On the right is a table showing which datasets were used for training, validation and testing in each of the experiments

prior to training to be used by all models, which increased the size of the training set from 774 to 92,880 images.

Implementation details

All coding was carried out in Python with Keras/Tensorflow version 2.13.1. For Lottes' model, no source code could be found online, so it was recreated from scratch, based on the original paper (Lottes et al., 2018). The code for Zhang's model, downloaded using the link provided in their paper (Zhang et al., 2020), was in Tensorflow. For the sake of consistency, this code was converted into Keras, and modified so it could work with a multiclass problem.

All models were trained for a total of 350 epochs. The batch size was set to 16 for Lottes' and Zhang's model, as this was the maximum size possible without running into memory errors. For our model, batch sizes between 16 and 128 were tested, and a size of 64 was selected, as this performed slightly better during validation.

All models were trained using AdamW³, with an optimal learning rate and weight decay calculated greedily via line search. The learning rate was decayed by a factor of 10 at epoch 150 for Zhang's model, and at epochs 50 and 250 for Lottes' model. For our model, a linear warmup of 100 epochs, followed by cosine decay was used. Training was carried out on an HPC cluster with 16 1-core 2.79 GHz Intel Xeon (Icelake) processors, 256 GB RAM, and a 40 GB NVIDIA A100-PCIE GPU. Inference time was calculated using an XPS 8950 Desktop running Linux, with a 10 GB NVIDIA GeForce RTX 3080 GPU.

Weights were initialized using the default Glorot uniform (Glorot & Bengio, 2010) initializer for Zhang's and our models. For Lottes' model, according to the instructions provided in the original paper (Lottes et al., 2018) He initialization (He et al., 2015) was used. Keras provides both He normal and He uniform weight initializers, the latter of which was selected, because it performed slightly better during validation.

³Adam and AdamW are identical in Keras, save for the default value for weight decay.

Stem-localization is a task that results in highly imbalanced classes, due to the fact that only a tiny proportion of the image pixels will correspond to a plant stem. For this reason, both Lottes' and Zhang's models employ an IOU loss for this task, which is designed to be more stable in problems with high class imbalance. For the task of plant segmentation, a more conventional cross-entropy loss is used. Although the original IOU loss paper by Rahman and Wang (Rahman & Wang, 2016) only considers the binary classification task, the problem generalizes to the multiclass problem by calculating the average loss across all classes. The formula for the IOU loss, $\mathcal{L}_{IOU}$, is given in Eq. 1, where V is the set of pixels in the training batch, $y^{(c)}$ is the ground truth label for class c , $\hat{y}^{(c)}$ is the predicted label for class c , and k is the number of classes.

$$\begin{aligned} I^{(c)} &= \sum_{v \in V} y_v^{(c)} \hat{y}_v^{(c)} \\ U^{(c)} &= \sum_{v \in V} y_v^{(c)} + \hat{y}_v^{(c)} - I^{(c)} \\ \mathcal{L}_{IOU} &= 1 - \frac{1}{k} \sum_{i=1}^k \frac{I^{(c_i)}}{U^{(c_i)}} \end{aligned} \quad (1)$$

Cross entropy loss was used for the plant segmentation task, while IOU loss averaged over the positive and negative classes was used for the stem task. Each task was assigned equal weight. Loss was calculated at the end of each epoch, and the weights that produced the best validation loss were restored at the end of training.

In order to remove noise, and small spurious detections from the spraypoint segmenter, postprocessing, including Gaussian smoothing and removal of small connected components was applied, with the optimal parameters of Gaussian sigma, and minimum component size determined during validation using grid search.

Training time was measured as the average of five runs. The inference time, which is the time taken to pass a single image through the network, was measured as the average for 1000 separate passes through the model. This is reported as the step time when calling the Keras Model.evaluate method with a batch size of 1.

Normalization techniques

Several different data preprocessing (or normalization) techniques were used. In every case each channel was normalized separately, across either a single image (per-image), or an entire batch of images (per-batch). In the first method, the data was rescaled to the interval $[-0.5, 0.5]$. We called this method "default" normalization, as it was the technique used in Lottes' and Zhang's papers. The formula for default normalization is presented in Eq. 2, where I_{max} and I_{min} are the maximum and minimum pixel values, and λ is a small constant to prevent division by zero:

$$\hat{I} = \frac{I - I_{min}}{\max(I_{max} - I_{min}, \lambda)} - 0.5 \quad (2)$$

In the second method, which we called “standard normal” or “naïve” normalization, the data was centered around 0, by subtracting the mean, and dividing by the standard deviation. This method will transform the data into a standard normal distribution ($\mathcal{N}$), but only if the data were normally distributed to start with. This method is presented in Eq. 3, where $\mu(I)$ and $\sigma(I)$ are the mean and standard deviations of the image pixel values:

$$\hat{I} = \frac{I - \mu(I)}{\max(\sigma(I), \lambda)} \quad (3)$$

For the third method, which we called HistMatch normalization, we need first to introduce the technique of image histogram matching. This is the process whereby the image histogram⁴ of one image is transformed to the histogram of another image. An example of histogram matching is provided on the right of Fig. 4. Using this concept, we “histogram match” the image data to $\mathcal{N}$.

This histogram matching is shown pictorially for a single channel in Fig. 4. The preprocessing function, $g : [0, 1] \rightarrow [-\infty, \infty]$ ⁵ can be found by comparing the cumulative distribution functions for the image histogram (denoted F_I), and $\mathcal{N}$ (denoted Φ). Starting on the x-axis for a given $x \in [0, 1]$, we draw a vertical line to F_I , then a horizontal line to Φ , and finally a vertical line back to the x-axis to find $g(x)$. The normalization step is then performed by broadcasting g across all pixels in the image. As can be seen in Fig. 4, the naïve approach (dotted cyan) does not match $\mathcal{N}$ very closely, while the HistMatch approach (dotted green) does, except where (in this case) overexposure causes a loss of intensity information.

The algorithm for HistMatch normalization is presented in Algorithm 1. Φ refers to the cumulative distribution function of a standard normal distribution, and count is a function that returns the count of pixels for which a condition is true. P is the set of all possible pixel values, $\hat{n}$ and $\hat{y}$ are hyperparameters which determine those values, $Y \subset [-\hat{y}, \hat{y}]$, of cardinality $\hat{n} + 1$, for which we calculate Φ , and I is the input image of size $h \times w$. Where applicable, functions are broadcast over a vector or matrix. On line 1, we calculate the cumulative histogram, h for the input image I , and on line 2, we calculate the midpoints between adjacent entries of h , in order to account for the discrete distribution of pixel values. In lines 3 and 4, we define the function g , which maps every pixel value $p \in P$ to the point $y \in Y$, which minimizes the absolute difference between $h(p)$ and $\Phi(y)$. Finally, in line 5, this function, g is broadcast over the input image. A Python implementation of HistMatch normalization is presented at the end of the supplementary material.

⁴The image histogram is a histogram of the proportion of image pixels corresponding to each intensity level

⁵For the purpose of the figure, we have pre-normalized the image pixels to the range $[0, 1]$, however in practice this step would be unnecessary, and g would be defined over $\{0, 1, \dots, 255\}$

Algorithm 1 HistMatch normalization

Require: $\mathbf{P} = \{0, 1, \dots, 255\}$
Require: $\hat{n} \in \mathbb{N}$
Require: $\hat{y} \in \mathbb{R}^+$
Require: $\mathbf{Y} = \{-\hat{y} + \frac{2n\hat{y}}{\hat{n}} | n \in \{0, 1, \dots, \hat{n}\}\}$
Require: $\mathbf{I} \in \mathbf{P}^{h \times w}$

- 1: $\mathbf{h} \leftarrow [\frac{1}{hw} \text{count}(I \leq p)]_{p \in \{-1\} \cup \mathbf{P}}$
- 2: $\mathbf{h} \leftarrow [\frac{1}{2}(\mathbf{h}_p + \mathbf{h}_{p-1})]_{p \in \mathbf{P}}$
- 3: $A : \mathbf{P} \times \mathbf{Y} \rightarrow \mathbb{R}_0^+, A(p, y) = |\mathbf{h}_p - \Phi(y)|$
- 4: $g : \mathbf{P} \rightarrow \mathbf{Y}, g(p) = \arg \min_y A(p, y)$
- 5: **return** $g(\mathbf{I})$

Algorithm 1 HistMatch normalization**Experiments**

Initially, three experiments were conducted. In the first experiment, datasets A and C were combined (hereafter dataset A + C) and used in a 80%/20% train/validation split, in order to calculate suitable hyperparameters for the models, and provide baseline metrics against which to measure the other experiments. In the second experiment, dataset A + C was again used for training and validation, and half of dataset B (hereafter dataset B2) was used for calculating test statistics. This experiment was conducted in order to see how well the models generalized to images taken with growing conditions that they had not seen during training. In the third experiment, images in dataset A + C were randomly selected and replaced with the other half of dataset B (hereafter called B1), yielding dataset A + C + B1. This dataset was used for training and validation, while B2 was again used for testing. This experiment was conducted to produce a baseline result for the B2 dataset in which the models had been partially trained with images taken at the same time, and for comparison with experiment two. A summary of the datasets used for each experiment is provided in the right section of Fig. 2. For experiments 1, 2 and 3, “default” per-batch normalization was used.

Experiments 1a and 2a were also conducted as repetitions of experiments 1 and 2 respectively, in which all six normalization techniques (i.e. default, naïve or HistMatch applied either per-image or per-batch) were compared.

Finally, the models were evaluated using the publicly available Cauliflower dataset from the study by Zhang et al. (Zhang et al., 2020). Two different versions of our model were used for this dataset, with the first being identical to the version used in the previous experiments, and the second, which we call ours-4s having an additional lateral decoder arm for each task. The structure of the ours-4s model is given in the supplementary material (figure S3).

The version of the dataset we downloaded did not include stem annotations. To address this, we created our own stem annotations by first marking the stem location as a point using the VGG image annotator (Dutta & Zisserman, 2019), and then applying a Gaussian filter at that location with $\sigma = 7$ for Lottes’ and Zhang’s models, and $\sigma = 15$ for our model. In both cases, the brightness of the generated annotations was adjusted so that the center pixel, corresponding to the stem location, had a value of 255. We opted for a higher σ value for our

model to provide a larger target, accommodating the fact that our model produces an output downsampled by a factor of 32 (or 8 for ours-4s model) for the stem task.

For Lottes' and Zhang's models, we used the same learning rate and augmentation strategy as in the original study (Zhang et al., 2020). Specifically, the training dataset was increased by a factor of four using horizontal and/or vertical flips. For our model, the optimal learning rate of 0.001 was found using a line search, and the augmentation strategy was expanded by incorporating zooms and rotations in addition to flips. Default, per image normalization was used. For the stem task, the IOU-loss (positive class only) was used for Lottes' and Zhang's model, and the IOU-loss (positive and negative classes) used for our model. For the plant segmentation task, crossentropy loss was used for all models.

For experiments using our dataset, the plant segmentation task was evaluated using the per-pixel mean intersection over union (IoU) across positive and negative classes. For the spraypoint task, per-object precision, recall, and F1 score were employed. A spraypoint detection was considered positive if its centroid was within a radius of 50 pixels (equivalent to 50 mm) from the nearest unassigned ground truth centroid. All metrics presented include the mean and standard deviation across five separate runs.

For experiments using Zhang's dataset, we followed the same metrics as those used in the original study by Zhang et al. (Zhang et al., 2020). Specifically, the F1 score was used for the plant segmentation task. For the stem task, since each image in this dataset contained exactly one stem, the geometric mean of the average activation was used to determine the predicted stem location. The successful detection rate (SDR), defined as the ratio of successful detections to the total number of test images, and the average Euclidean distance (ED) between the predicted and ground truth stem locations (in pixels) for successful detections were reported. A detection was considered successful if it was within 17.5 pixels of the ground truth location.

Results

Table 1 shows the optimal hyperparameter as well as the number of parameters, FLOPs, training time and inference time of each of the three models, as determined in experiment 1, according to the methodology presented in Sect. 2.

Tables 2 and 3 provide the results for experiments 1, 2 and 3, for the plant and spraypoint tasks respectively, showing the mean and standard deviation of the metrics used for five

Table 1 List of hyperparameters for training and post-processing, and specifications for Lottes', Zhang's and our models for experiments on our dataset

Hyperparameter/ Specification	Lottes	Zhang	Our
Batch Size	16	16	64
Weight Decay	0.01	0.01	0.1
EMA	Yes	No	No
Learning Rate	0.003	0.003	0.001
Minimum Size (CC)	2000	2000	4000
Gaussian Sigma	10	10	0
Parameters	59,045	153,263	2,643,779
GFLOPs	10.0	11.3	4.9
Training Time (s)	14,475	15,809	5,064
Inference Time (ms)	14.8	21.5	15.7

Table 2 Results of experiments 1, 2 and 3 for the plant task

Experiment	Model	IoU(-)	IoU(+)	mIoU
One	Lottes	0.871 ± 0.013	0.606 ± 0.043	0.739 ± 0.027
	Zhang	0.953 ± 0.000	0.865 ± 0.001	0.909 ± 0.001
	Our	0.957 ± 0.001	0.874 ± 0.003	0.915 ± 0.002
Two	Lottes	0.089 ± 0.096	0.451 ± 0.025	0.270 ± 0.055
	Zhang	0.332 ± 0.225	0.576 ± 0.083	0.454 ± 0.154
	Our	0.829 ± 0.016	0.785 ± 0.029	0.807 ± 0.022
Three	Lottes	0.646 ± 0.039	0.559 ± 0.028	0.602 ± 0.026
	Zhang	0.879 ± 0.011	0.870 ± 0.010	0.875 ± 0.011
	Our	0.885 ± 0.004	0.873 ± 0.003	0.879 ± 0.003

Results are given as mean $\pm$ standard deviation of five runs, and the best result for each experiment is given in bold

Table 3 Results of experiments 1, 2 and 3 for the stem task

Experiment	Model	Precision	Recall	F1-Score
One	Lottes	0.467 ± 0.069	0.358 ± 0.011	0.404 ± 0.031
	Zhang	0.858 ± 0.023	0.900 ± 0.030	0.878 ± 0.005
	Our	0.875 ± 0.014	0.900 ± 0.022	0.887 ± 0.015
Two	Lottes	0.049 ± 0.017	0.143 ± 0.062	0.073 ± 0.026
	Zhang	0.482 ± 0.267	0.620 ± 0.092	0.496 ± 0.141
	Our	0.941 ± 0.020	0.693 ± 0.061	0.796 ± 0.041
Three	Lottes	0.518 ± 0.149	0.195 ± 0.101	0.266 ± 0.097
	Zhang	0.834 ± 0.042	0.676 ± 0.065	0.745 ± 0.041
	Our	0.931 ± 0.016	0.698 ± 0.012	0.797 ± 0.011

Results are given as mean $\pm$ standard deviation of five runs, and the best result for each experiment is given in bold

Table 4 Results with HistMatch (image) normalization for experiment 2a, in which models were tested on growing conditions not seen during training, given as mean $\pm$ standard deviation of five runs

Model	Plant	Spraypoint		
	mIoU	Precision	Recall	F1
Lottes	0.703 ± 0.052	0.634 ± 0.052	0.572 ± 0.040	0.599 ± 0.019
Zhang	0.821 ± 0.054	0.735 ± 0.043	0.732 ± 0.039	0.733 ± 0.037
Our	0.854 ± 0.006	0.843 ± 0.023	0.772 ± 0.034	0.806 ± 0.028

Best results for each metric in bold

iterations of each model. In addition, supplementary figure S4 shows the confusion matrices for all models for experiments 1, 2 and 3, for the plant segmentation task.

Figure 5 shows the results for experiments 1a and 2a, in which different normalization methods were tested. The results for HistMatch (image) normalization for each model for experiment 2a is provided in Table 4. A table of all results for experiments 1a and 2a is provided in the supplementary material (tables S1 and S2).

Table 5 presents the results from experiments conducted using Zhang's dataset. These results include the findings reported by Zhang et al. (Zhang et al., 2020) for both Lottes' and

Table 5 Results for experimentation on Zhang's dataset, both as reported by Zhang et al. (Zhang et al., 2020), and as carried out in this study.

Study	Model	Plant		Stem			
		F1		SDR		ED (pixels)	
(Zhang et al., 2020)	Lottes	0.975		0.912		2.62	
	Zhang	0.974		0.978		2.56	
This study	Lottes	0.956	±0.002	0.931	±0.010	3.85	±0.09
	Zhang	0.974	±0.001	0.995	±0.004	2.21	±0.07
	Ours	0.968	±0.0002	0.997	±0.000	2.45	±0.04
	Ours-4s	0.975	±0.0002	0.996	±0.001	2.38	±0.02

For our study, the results are given as the mean ± standard deviation of five runs. Best results for each metric is in bold

Zhang's models, as well as the outcomes for the four models considered in our study, which include Lottes' and Zhang's models, in addition to our two models.

Discussion

This paper presented a study on a joint plant and spraypoint detector which was tested on images of tussock taken in south-east Australia. Lottes' and Zhang's model were used as is, and an alternative model was proposed for testing, which utilised several elements of ConvNeXt, including a patchify stem, ConvNeXt modules, and a learning rate schedule with linear warmup followed by Cosine decay. Our model performed best in all of the metrics tested, as show in in Tables 2 and 3, although it was only slightly better than Zhang's model in the first experiment, in which dataset A + C was used for both training and validation. When tested on growing condition which were not included in the training set (experiment 2/2a), our model was able to achieve metrics of 0.8 – 0.85 in both tasks, as show in Table 4.

For the plant segmentation task, it was difficult to separate each tussock plant from its neighbours, due to poorly defined boundaries. Moreover, the tussock plants were often large, and not centered, with only part of the plant appearing in an image. This made it impractical to use an object-based approach in the interpretation of results. The mIoU metric gives a balanced, single score, which takes into account how well both the positive and negative classes were detected. False negative regions may lead to inadequate, or missing treatment of a target plant, whereas false positive regions may lead to herbicide overuse, and damage to the background pasture.

By comparison, for the spraypoint task, each spraypoint formed a clearly defined unit, and for this reason per-object metrics were used. The recall is the proportion of target plants that were detected by the model, whereas the precision is the proportion of the model's detections which corresponded to actual (ground truth) target plants. The F1 score combines both these metrics into a single score, allowing for easier comparison of models.

For the stem detection task, the IOU loss averaged over the positive and negative classes proved to be a better choice than the alternative approach, in which only the positive class is considered when calculating the loss. This latter approach is faithful to the original work of Rahman and Wang (Rahman & Wang, 2016). However, in our case, we found that this

occasionally led to instability in training, during which the model would learn to predict almost all of the pixels as positive⁶.

Lottes' model performed significantly worse than the other two models as show in Tables 2 and 3, even in experiment one, in which the training and validation sets were drawn from the same dataset. One possible reason for this may be due to the fact that it had fewer downsampling steps, and hence a lower effective receptive field size⁷, coupled with the fact that many tussock plants were large, with some covering most of an image. Another possible reason is that the number of parameters was insufficient to capture the more complex nature of a pasture, as compared to the comparatively simpler environment in an early stage crop field, with small (mostly) non-overlapping green plants against a soil background, as was the case in the original study by Lottes et al. (Lottes et al., 2018).

Zhang's model performed well on the first experiment, but did not generalize well to field conditions not seen in training. This was largely due to these models predicting most pixels as positive, as seen in supplementary figure S4. Again, this may be because the model has relatively few parameters, which cannot learn sufficiently robust features to be useful in the general case. This result highlights the need to test models on a dataset which is wholly separate from the training dataset, as without this, our conclusion would likely have been that the performance of Zhang's model was almost as good as our new model.

In the original paper by Zhang et al. (2020), the authors tested their model on images containing a single plant, and used the spacial average of the activation map as the stem location, suggesting that connected component analysis be used if multiple plants were present in a frame. However, in our study, multiple fragmented detections were produced by Zhang's model which required post-processing, in order to achieve acceptable results. The postprocessing for each model included applying a Gaussian blur to the image in order to join close, disjoint regions, followed by the removal of small detections. In this case, while Lottes' and Zhang's model required both of these steps, our model only required the removal of small detections, as shown in Table 1.

Experiment 3 incorporated half of dataset B (B1) into the training set, which allowed us to gain an understanding of whether dataset B2 was inherently easier or harder than dataset A + C. The results of our and Zhang's models demonstrated that for the plant segmentation task, dataset B2 was not significantly harder, but for the spraypoint detection task it was, especially with respect to the number of false negatives (and hence a significantly lower recall). Closer examination of dataset B2 revealed that a lot of the missed spraypoints were on the edge of the image (for example, see rows (c) and (d) in Fig. 3), and in fact, in this location, it was often difficult for the annotator to decide whether to include it as a ground-truth annotation, or place it in the "hard" class. Interestingly, there was no significant difference between experiments two and three for our model in the spraypoint task, indicating that the model was already robust in detecting spraypoints, even when tested on images with growing conditions it had not been trained on.

When comparing the results of experiments two and three, it is evident that our model was the only one able to generalize effectively. The reduction in performance metrics for the

⁶Due to the way that Rahman and Wang's loss is implemented, a model which classifies all pixels as positive has a loss of $1 - p$, where p is the proportion of pixels that are in the positive class. By contrast, a model which classifies all pixels as negative has a loss of 1. This means that when the model is struggling to distinguish the two classes, it is likely to err on the side of classifying most or all of the pixels as positive.

⁷For a study on effective receptive field, the reader is referred to (Luo et al., 2016)

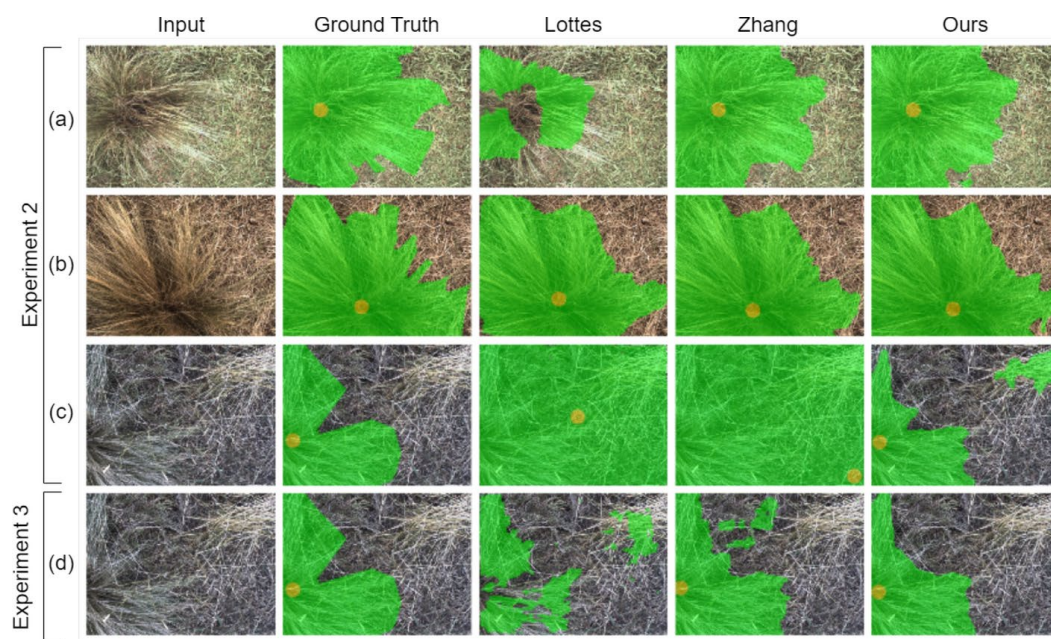


Fig. 3 Example images and segmentation results for Lottes', Zhang's and our models, for experiments 1 to 3. Images are from the Oct-21 (a), Jul-22 (b) and Apr-22 (c, d) sets. Images in row (c) show an example where models were tested on field conditions not seen during training. In this situation Lottes' and Zhang's models often classified most or all pixels as positive. Green positive (plant), yellow positive (spraypoint centroid). Row (d) shows the same image as in row (c), but where images from the Apr-22 dataset were included during training. In this case, there was typically little difference in performance between Zhang's and our model, whereas Lottes' model often continued to struggle

plant and spraypoint tasks was just 0.072 and 0.001, respectively, when the test set included growing conditions not seen during training. In contrast, Zhang's model showed a decline of 0.421 and 0.249 for the same tasks, while Lottes' model exhibited reductions of 0.332 and 0.193.

Our model contained a patchify stem, which resulted in a rapid reduction in the spatial resolution of the feature map in the early stages of our model. Taken in conjunction with a reduction in the expansion rate and fewer activation and normalization layers in our ConvNeXt modules, this allowed us to use more filter banks, while having fewer FLOPs than both Lottes' and Zhang's model. The result was a model with 17 times as many parameters as Zhang's model, but which was faster at both inference and training ($1.4\times$ and $3\times$ respectively). Also of interest is the fact that Lottes' model was marginally faster than our model at inference ($1.1\times$), in spite of the fact that it required $2\times$ as many FLOPs. This demonstrated that neither parameters nor FLOPs are necessarily useful indicators for predicting model speed.

During the interpretation of results for experiment two as shown in Tables 2 and 3, in which the models' ability to generalize was tested, it was found that Lottes' and Zhang's models sometimes performed poorly, predicting most pixels as positive for the plant segmentation task. To investigate why this was the case, we experimented with histogram matching techniques, in which the histogram of one image is transformed to match the histogram of a second image. In this way, we could transform the histogram of an image in the test set to match the histogram of an image in the training set, in order to make the test image more closely resemble the kinds of image seen during training. With this technique, we got

ambiguous results, with models sometimes performing better and sometimes performing worse than without histogram matching.

One possible way to overcome differences in image histogram distributions between separate datasets could be to adopt an additional step during data augmentation, in which images are transformed to a randomly generated image histogram. However, to adopt this, one would need to find a probability distribution for the image histograms, which closely matched the kinds of images likely to be encountered by the model, and find suitable bounds for the parameters of this distribution. Instead, we realised that it would be easier to include a preprocessing step (applied to an image immediately prior to input into the model), which transformed the image histogram of the input image to a pre-chosen probability distribution, the most obvious choice being $\mathcal{N}$, and this was the technique which we used in this study. As discussed in Sect. 2.5, the naïve approach to this problem does not produce a distribution that closely matched $\mathcal{N}$, which lead us to develop our own HistMatch normalization.

Experiments 1a and 2a, were conducted to determine whether the choice of normalization method was important, with the results given in Fig. 5. Paying particular attention to experiment 2a, in which models were tested on growing conditions not seen during training, it was found that HistMatch normalization applied to each image separately was the best choice by a considerable margin for Lottes' model for all metrics tested. For Zhang's and our models, the differences in normalization methods were not so pronounced, with HistMatch normalization providing the best results in three out of four metrics for Zhang's model (compared to one for naïve), and two out of four metrics for our model (compared to one each for default and naïve). HistMatch normalization simplifies the input data distribution by transforming it into a standard normal distribution, as illustrated in Fig. 4. This approach may offer the greatest benefit to simpler models with fewer parameters, which may struggle to manage the underlying complexities in the original data distribution (Fig. 5).

Finally, experiments were conducted on Zhang's dataset to evaluate how well our model generalized to the tasks of plant segmentation and stem detection in crop fields. In addition to the original model used in previous experiments, an adjusted model, referred to as Ours-4s, was also tested. This modified version included an additional decoder for each task, resulting in smaller upsampling steps of $4\times$ for segmentation and $8\times$ for stem detection, compared to $8\times$ and $32\times$, respectively, in the original model. These adjustments

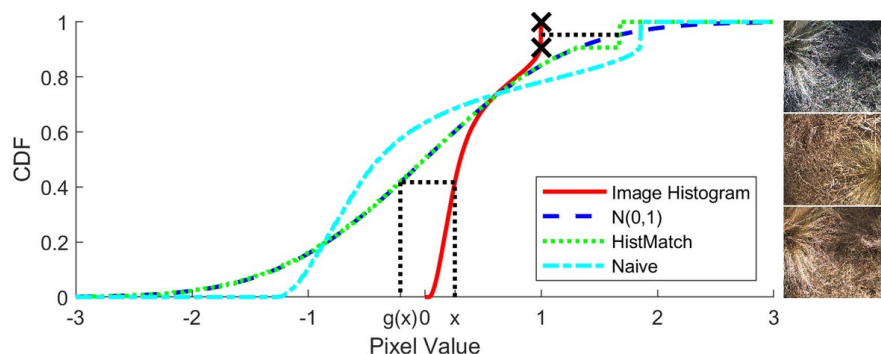


Fig. 4 Demonstration of HistMatch normalization, in which the image pixels are transformed to $\mathcal{N}$. To account for the discrete distribution of pixel intensities, the midpoint between adjacent points on F_I (dark crosses) is used when calculating g . On right of figure, is an example of histogram matching to a reference image. From top to bottom, original image, reference image, original image histogram matched to reference image

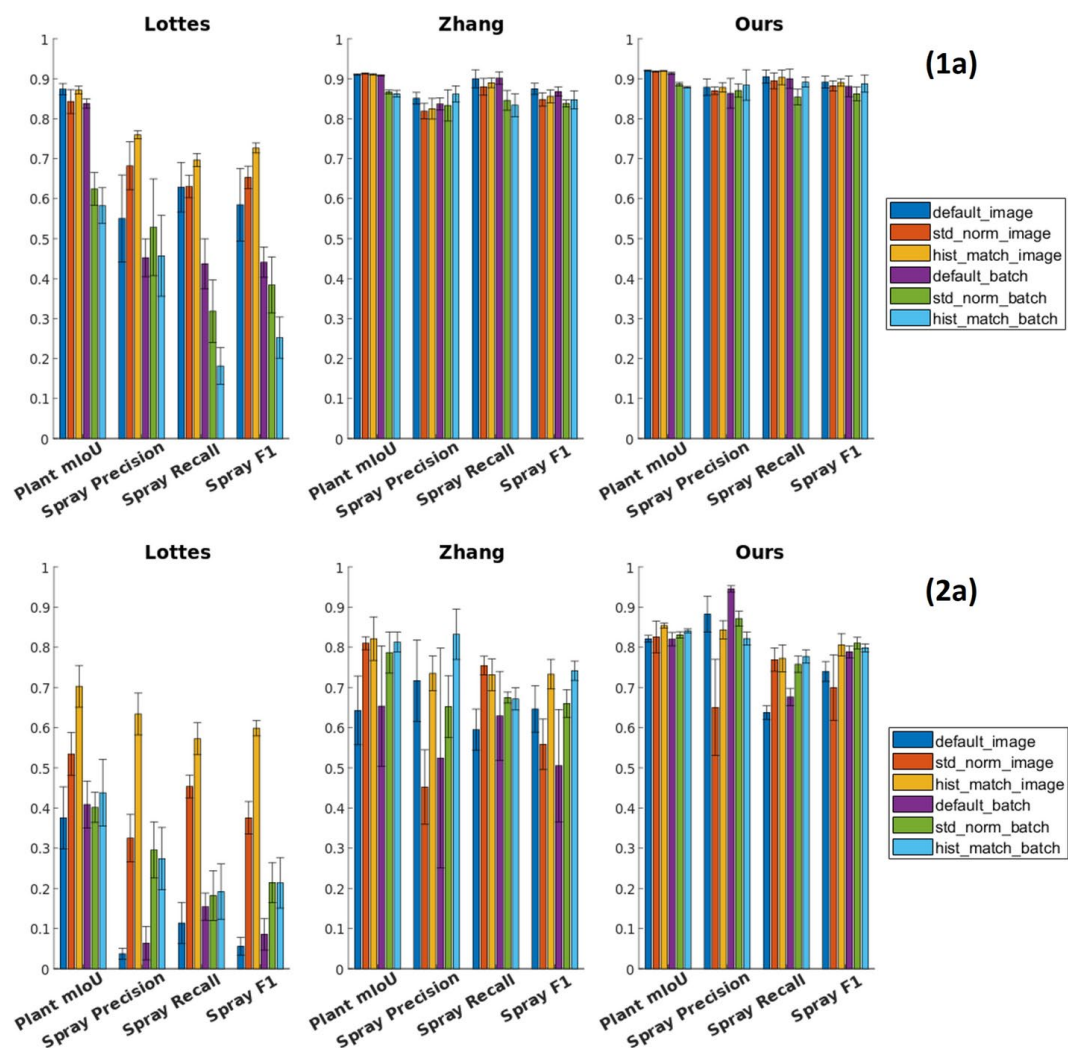


Fig. 5 Experiment 1a (top) and 2a (bottom): plant mIoU, and spraypoint precision, recall and F1-score for different normalization methods. In experiment 1a, the same dataset was divided into a training and a validation set. In experiment 2a, the test images were drawn from a different dataset to the training images (different growing conditions)

were expected to enhance the segmentation of fine plant edges and improve the accuracy of detecting small features such as the plant stem. The Ours-4s model was designed as an intermediate between the large upsampling stages of our original model and Zhang's model, which does not require final upsampling, as its classification layer operates at the full image resolution.

As shown in Table 5, Ours-4s model performed best for the plant segmentation task, achieving an F1 score of 0.975, which was equivalent to the results reported for Lottes' model by Zhang et al. (Zhang et al., 2020). For the stem detection task, our model achieved the highest SDR of 0.997, followed closely by Ours-4s at 0.996 and Zhang's model at 0.995. In terms of ED, which is the distance between the predicted stem and the ground truth stem, Zhang's model performed best with an ED of 2.21 pixels, compared to 2.38 for Ours-4s and 2.45 for our model.

The stem detection metrics for Zhang's and Lottes' models in our study were significantly better than those reported by Zhang et al. (2020), except for the ED for Lottes' model. This is likely due to the fact that we produced our own annotations for the stem task. In contrast, for the segmentation task, Lottes' model performed worse in our study compared to Zhang et al.'s reported results, suggesting there may have been implementation differences between the two studies. Unfortunately, we were unable to find an online version of Lottes' model for direct comparison with our implementation. Although we made every effort to follow the original paper (Lottes et al., 2018) as closely as possible, it is likely that some discrepancies between the two implementations contributed to the performance differences.

Our models demonstrated performance comparable to or slightly better than Zhang's model on the plant F1 and stem SDR metrics, though they performed somewhat worse on the ED metric. Notably, this was achieved despite our models employing final bilinear upsampling steps of $8\times$ for the plant task and $32\times$ for the stem task (or $4\times$ and $8\times$ for Ours-4s model). In contrast, Zhang's model generates output masks at full resolution, eliminating the need for upsampling. Despite these large upsampling steps, our models were able to locate the stem with an average error of 2.45 pixels (or 2.38 pixels for Ours-4s model). While Zhang's model slightly outperformed ours in terms of the ED metric, it came at a significantly higher computational cost. The two full-resolution decoder arms in Zhang's model, one for each task, account for 6.1 GFLOPs, which is more than half of the total computational cost of 11.3 GFLOPs. This highlights the importance of carefully considering model design, particularly as to whether the task at hand requires the additional output resolution and whether the corresponding computational overhead is justified. Minimizing computation at full resolution can help optimize efficiency with minimal or no performance loss.

Conclusion and future directions

This carefully controlled study has demonstrated the feasibility of a joint plant and spraypoint detector for use in pastures, paving the way for further research into multi-task models in pastoral environments. It was observed that dual-task models designed for early-stage crop fields struggled to perform well in pastures, especially when tested on growing conditions not represented in the training set. In contrast, our model significantly outperformed other models in this setting on both the segmentation and spraypoint tasks, achieving at least 30% better metrics. The challenges faced by other models were mitigated through the use of HistMatch normalization, suggesting that this technique may be particularly effective in enhancing model generalization in environments where robust performance is difficult to achieve.

To advance the development of multi-task networks in pasture environments, further data collection is required. Specifically, creating a dataset that includes multiple long-blade grass species, in addition to tussock, such as African lovegrass (*Eragrostis curvula*) and Chilean needle grass (*Nassella neesiana*), both of which are significant weeds in Australia, would be instrumental in demonstrating the effectiveness of such networks in distinguishing between different target species. Unfortunately, to our knowledge, no such multi-task dataset currently exists.

Furthermore, the integration of additional tasks, such as biomass or nutrient content estimation, which are typically handled by separate networks from the segmentation network (Skovsen et al., 2019; Mortensen et al., 2019) could further expand the potential of multi-task networks in pastures. However, this would necessitate additional network design, as these tasks involve regression rather than classification.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11119-024-10208-y>.

Acknowledgements The first author was supported by a Commonwealth of Australia RTP scholarship, with a top-up scholarship from the “Novel Autonomous Robotic Weed Control to Maximise Agricultural Productivity” CRC (CRCPIX000099). The authors would like to thank the NSW Department of Primary Industries and Agent Oriented Software for providing the dataset.

Author contributions All authors contributed equally to this paper.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data availability The data used in this study will be made available on request by email to the first author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., & He, Y. (2019). Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. *Computers and Electronics in Agriculture*, 167, 105091. <https://doi.org/10.1016/j.compag.2019.105091>
- Ba, J.L., Kiros, J.R., Hinton, G.E. (2016). Layer normalization. arXiv preprint [arXiv:1607.06450](https://arxiv.org/abs/1607.06450)
- Binch, A., & Fox, C. W. (2017). Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland. *Computers and Electronics in Agriculture*, 140, 123–138. <https://doi.org/10.1016/j.compag.2017.05.018>
- Campbell, M. (1998). Biological and ecological impact of serrated tussock (nassella trichotoma (nees) arech.) on pastures in australia. *Plant Protection Quarterly*, 13, 80–85.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Dalal, N., Triggs, B. (2005). Histograms of Oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. IEEE, San Diego, California . <https://doi.org/10.1109/CVPR.2005.177>
- Dutta, A., Zisserman, A. (2019). The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM international conference on multimedia*. MM '19. ACM, New York, NY, & USA. <https://doi.org/10.1145/3343031.3350535>

- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 303–338. <https://doi.org/10.1007/s11263-009-0275-4>
- Farooq, A., Jia, X., Zhou, J. (2019). Texture and shape features for grass weed classification using hyperspectral remote sensing images. In *IGARSS 2019 - 2019 IEEE international geoscience and remote sensing symposium* (pp. 7208–7211). IEEE, Yokohama, Japan. <https://doi.org/10.1109/IGARSS.2019.8900132>
- Ford, J., Sadgrove, E., & Paul, D. (2023). Developing an extreme learning machine based approach to weed segmentation in pastures. *Smart Agricultural Technology*, 5, 100288. <https://doi.org/10.1016/j.atech.2023.100288>
- Glorot, X., Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings
- He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034). <https://doi.org/10.1109/ICCV.2015.123>
- Hendrycks, D., Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint [arXiv:1606.08415](https://arxiv.org/abs/1606.08415)
- Huang, G.-B., Zhu, Q.-Y., Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*. IEEE, Budapest, Hungary. <https://doi.org/10.1109/IJCNN.2004.1380068>
- Jin, B., Xu, X. (2024). Wholesale price forecasts of green grams using the neural network. *Asian Journal of Economics and Banking*
- Jin, B., & Xu, X. (2024). Forecasting wholesale prices of yellow corn through the gaussian process regression. *Neural Computing and Applications*, 36(15), 8693–8710.
- Johnson, S.B., Baker, S.J., Davidson, B., Harris, C.A., Sinclair, K., Wu, H., Bajwa, A.A. (2022). Making a difference to invasive grasses management on the ground. In *2022 Australasian weeds conference* (vol. 1, p. 316).
- Khan, A., Ilyas, T., Umraiz, M., Mannan, Z. I., & Kim, H. (2020). CED-Net: Crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture. *Electronics*, 9(10), 1602. <https://doi.org/10.3390/electronics9101602>
- Kounalakis, T., Triantafyllidis, G. A., & Nalpantidis, L. (2019). Deep learning-based visual recognition of rumex for robotic precision farming. *Computers and Electronics in Agriculture*, 165, 104973. <https://doi.org/10.1016/j.compag.2019.104973>
- Kounalakis, T., Triantafyllidis, G. A., & Nalpantidis, L. (2018). Image-based recognition framework for robotic weed control systems. *Multimedia Tools and Applications*, 77(8), 9567–9594. <https://doi.org/10.1007/s11042-017-5337-y>
- Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning-a review. *IEEE Access*, 9, 56683–56698.
- Liu, B., & Bruch, R. (2020). Weed detection for selective spraying: A review. *Current Robotics Reports*, 1(1), 19–26. <https://doi.org/10.1007/s43154-020-00001-w>
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11976–11986). <https://doi.org/10.1109/CVPR52688.2022.01167>
- Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2020). Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming. *Journal of Field Robotics*, 37(1), 20–34.
- Lottes, P., Behley, J., Chebrolu, N., Milioto, A., Stachniss, C. (2018) Joint stem detection and crop-weed classification for plant-specific treatment in precision farming. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 8233–8238). <https://doi.org/10.1109/IROS.2018.8593678>
- Luo, W., Li, Y., Urtasun, R., Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. *Advances in Neural Information Processing Systems* 29
- McCool, C., Perez, T., & Upcroft, B. (2017). Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics. *IEEE Robotics and Automation Letters*, 2(3), 1344–1351. <https://doi.org/10.1109/LRA.2017.2667039>
- McLeod, R. (2018). Annual costs of weeds in australia. esys development pty limited. Centre for Invasive Species Solutions, Canberra
- Mortensen, A.K., Skovsen, S., Karstoft, H., Gislum, R. (2019) The oil radish growth dataset for semantic segmentation and yield estimation. In *2019 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 2703–2710). IEEE. <https://doi.org/10.1109/CVPRW.2019.00328>. <https://ieeexplore.ieee.org/document/9025602/>

- Muruganantham, P., Wibowo, S., Grandhi, S., Samrat, N. H., & Islam, N. (2022). A systematic literature review on crop yield prediction with deep learning and remote sensing. *Remote Sensing*, 14(9), 1990.
- Ojala, T., Pietikainen, M., & Maenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971–987. <https://doi.org/10.1109/TPAMI.2002.1017623>
- Osorio, K., Puerto, A., Pedraza, C., Jamaica, D., & Rodríguez, L. (2020). A deep learning approach for weed detection in lettuce crops using multispectral images. *AgriEngineering*, 2(3), 471–488. <https://doi.org/10.3390/agriengineering2030032>
- Rahman, M.A., Wang, Y. (2016) Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing* (pp. 234–244). https://doi.org/10.1007/978-3-319-50835-1_22. Springer
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*. <https://doi.org/10.1007/s11263-015-0816-y>
- Sadgrove, E., Falzon, G., Miron, D., & Lamb, D. (2017). Fast object detection in pastoral landscapes using a Colour Feature Extreme Learning Machine. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2017.05.017>
- Sadgrove, E., Falzon, G., Miron, D., & Lamb, D. (2018). Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (MEC-ELM). *Computers in Industry*. <https://doi.org/10.1016/j.compind.2018.03.014>
- Sadgrove, E., Falzon, G., Miron, D., & Lamb, D. (2021). The segmented colour feature extreme learning machine: Applications in agricultural robotics. *Agronomy*, 11(11), 2290. <https://doi.org/10.3390/agronomy11112290>
- Sadgrove, E., Falzon, G., Miron, D., Lamb, D. (2017). Fast object detection in pastoral landscapes using a multiple expert colour feature extreme learning machine. *Asian-Australasian Conference on Precision Pastures and Livestock Farming* 1
- Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants*, 8(11), 468.
- Schori, D., Anken, T., Seatovic, D. (2019). Using Fully Convolutional Networks for Rumex Obtusifolius Segmentation, a Preliminary Report. In *2019 International Symposium ELMAR* (pp. 119–122). IEEE, Zadar, Croatia. <https://doi.org/10.1109/ELMAR.2019.8918914>
- Smith, L. N., Byrne, A., Hansen, M. F., Zhang, W., Smith, M. L. (2019). Weed classification in grasslands using convolutional neural networks. In M. E. Zelinski, T. M. Taha, & J. Howe, A. A. Awwal, & K.M. Iftekharruddin (Eds.), *Applications of machine learning* (p. 42). SPIE, San Diego, California) <https://doi.org/10.1117/12.2530092>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520). <https://doi.org/10.1109/CVPR.2018.00474>
- Skovsen, S., Dyrmann, M., Mortensen, A.K., Laursen, M.S., Gislum, R., Eriksen, J., Farkhani, S., Karstoft, H., Jorgensen, R.N. (2019). The grassclover image dataset for semantic and hierarchical species understanding in agriculture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 0–0).
- Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H. (2021). Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 32–42).
- Van Klompenburg, T., Kassahun, A., & Catal, C. (2020). Crop yield prediction using machine learning: A systematic literature review. *Computers and electronics in agriculture*, 177, 105709.
- Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R. (2010). Deconvolutional networks. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 2528–2535). IEEE. <https://doi.org/10.1109/CVPR.2010.5539957>
- Zhang, W., Hansen, M. F., Volonakis, T. N., Smith, M., Smith, L., Wilson, J., Ralston, G., Broadbent, L., Wright, G. (2018). Broad-leaf weed detection in pasture. In *2018 IEEE 3rd international conference on image, vision and computing (ICIVC)* (pp. 101–105). IEEE, Chongqing, China. <https://doi.org/10.1109/ICIVC.2018.8492831>
- Zhang, X., Li, N., Ge, L., Xia, X., Ding, N. (2020). A unified model for real-time crop recognition and stem localization exploiting cross-task feature fusion. In *2020 IEEE international conference on real-time computing and robotics (RCAR)* (pp. 327–332). <https://doi.org/10.1109/RCAR49640.2020.9303270>

Authors and Affiliations

Jonathan Ford¹  · Edmund Sadgrove¹ · David Paul¹

✉ Jonathan Ford
jford3@myune.edu.au

Edmund Sadgrove
esadgro2@une.edu.au

David Paul
dpaul4@une.edu.au

¹ School of Science and Technology, University of New England, Armidale, NSW
2351, Australia

4.3 Supplementary Material

The supplementary material to the published paper is provided here.

Joint Plant-Spraypoint Detector with ConvNeXt Modules and HistMatch Normalization

Supplementary Material

Authors:

Ford, J.

Sadgrove, E.

Paul, D.

Figure S1: Lottes' model: (a) A dense block containing a total of three convolutional layers. Each layer adds four filters to the feature map ($L=3$, $G=4$). **(b)** Macro structure showing the input image (green), the encoder (blue), the decoders (red) and the output feature maps (purple). Arrows display the flow of information through the model. Image resolutions presented are those given in the original paper.

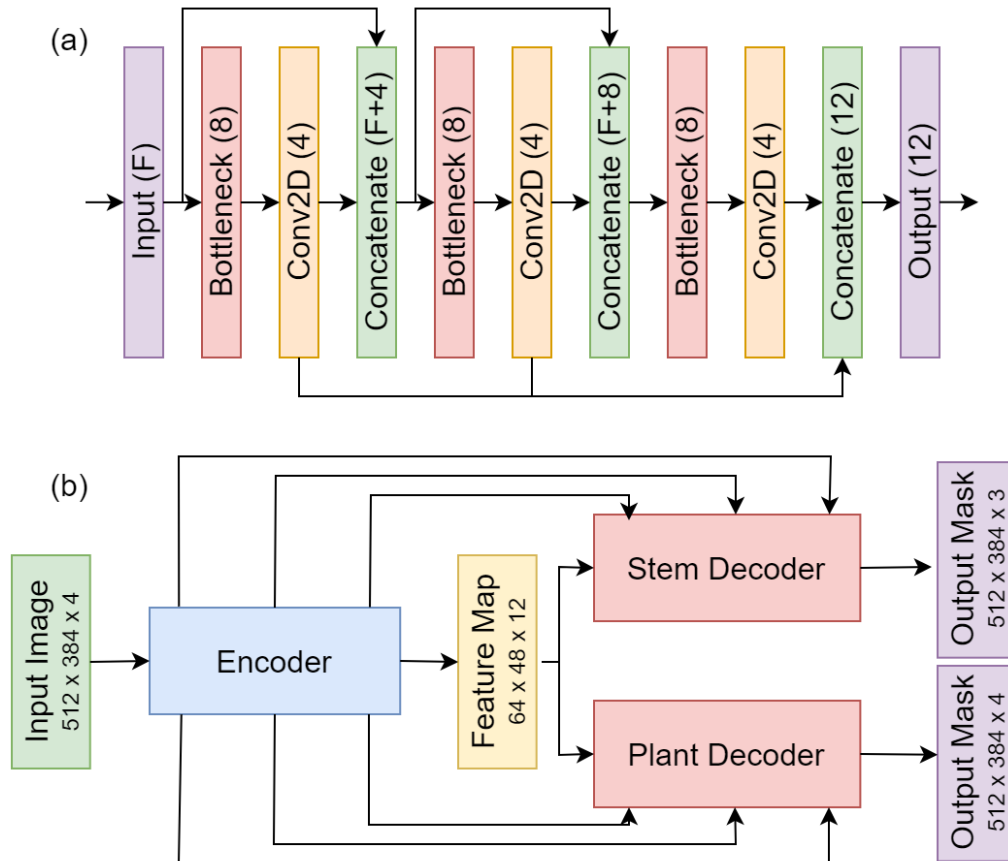


Figure S2: Zhang's model: (a) An IRM which employs two point-wise convolutions (orange), with a depth-wise convolution (red) in between. In the IRM, normalization occurs after each convolution, and activation occurs after the first point-wise convolution and the depth-wise convolution. (b) Macro structure showing the input image (green), the backbone (blue), the decoders (red) and the output feature maps (purple). Loss layers are shown in yellow, and include losses for deep supervision (small rectangles), as well as for the final outputs of the model (large rectangles). Arrows display the flow of information through the model. In particular, the cross-task feature fusion is given with a dashed line, and the bidirectional feature map exchanges are given with a dotted line. Image resolutions presented are those given in the original paper.

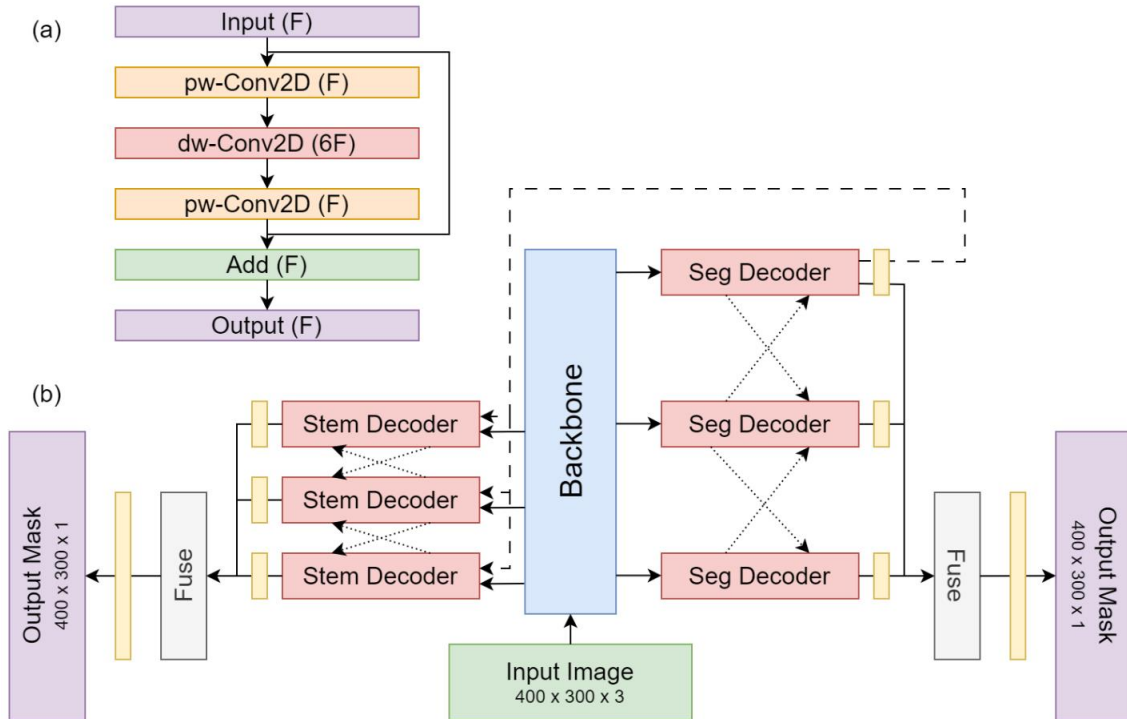


Figure S3: Ours-4s model: The 4s version of our model, which has one additional decoder for each tasks, which is situated at the 8x downsampling level for the spraypoint (or stem) task, and 4x downsampling level for the segmentation task. Components show are input image (green), patchify stem (orange), backbone (blue), decoders (red) and the output feature maps (purple). Loss layers are shown in yellow, and include losses for deep supervision (small rectangles), as well as for the final outputs of the model (large rectangles). Arrows display the flow of information through the model. In particular, the cross-task feature fusion is given with a dashed line, and the bidirectional feature map exchanges are given with a dotted line.

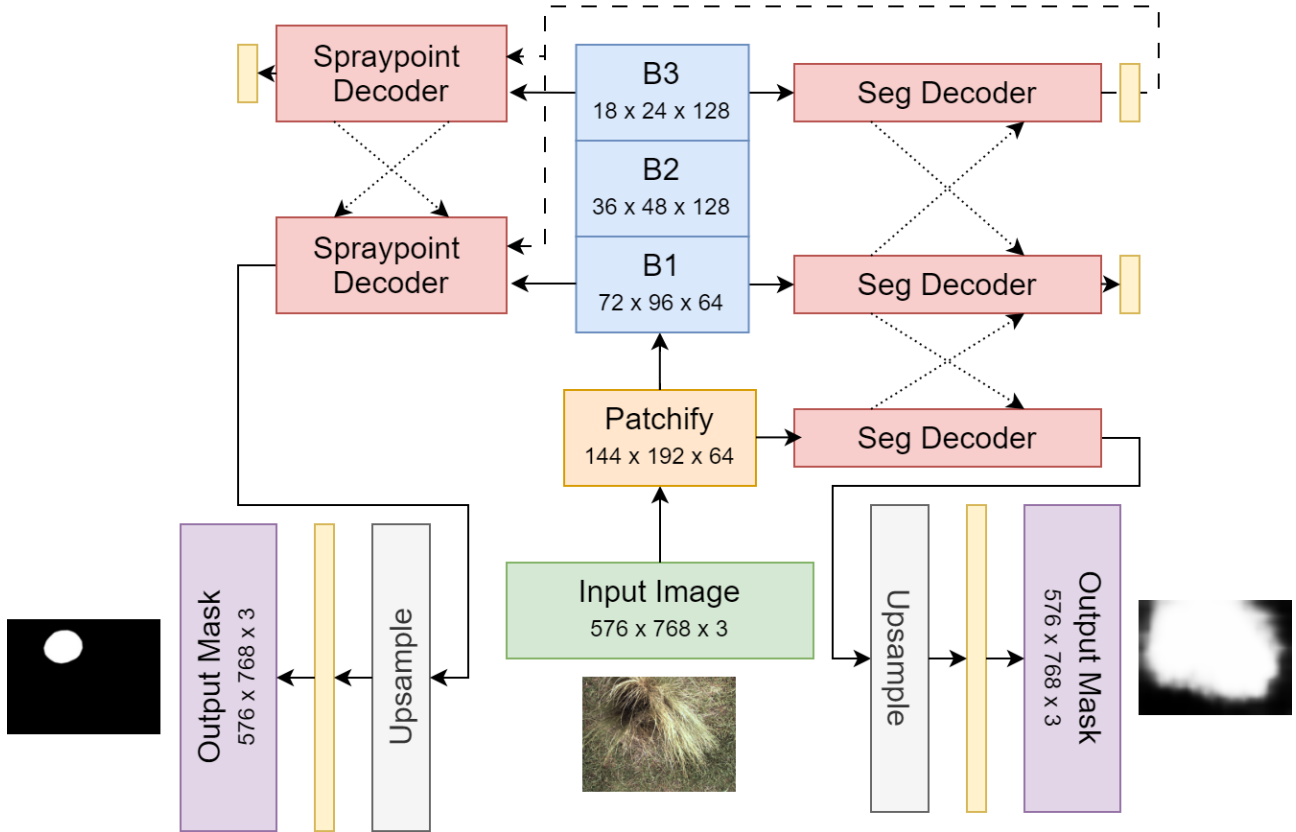


Figure S4: Normalized confusion matrices for the plant segmentation task for all models, for experiments 1, 2 and 3.

		Lottes		Zhang		Ours	
Experiment	1	Predicted		Predicted		Predicted	
		Neg		Neg		Neg	
		Actual	Pos	Actual	Pos	Actual	Pos
		Neg	96.7	Neg	97.6	Neg	97.8
		Pos	33.4	Pos	7.3	Pos	6.9
			66.6		92.7		93.1
Experiment	2	Predicted		Predicted		Predicted	
		Neg		Neg		Neg	
		Actual	Pos	Actual	Pos	Actual	Pos
		Neg	9.8	Neg	33.7	Neg	96.2
		Pos	9.0	Pos	1.0	Pos	18.2
			90.2		99.0		81.8
Experiment	3	Predicted		Predicted		Predicted	
		Neg		Neg		Neg	
		Actual	Pos	Actual	Pos	Actual	Pos
		Neg	84.3	Neg	91.8	Neg	93.6
		Pos	34.2	Pos	5.0	Pos	6.4
			65.8		95.0		93.6

Table S1: Experiment 1a metrics:

Task	Model	Normalization	mIoU								
			mean	std							
Plant	Lottes	default	0.874	0.014							
		Image Norm	std norm	0.843	0.030						
		hist match	0.871	0.010							
		Batch Norm	default	0.838	0.012						
		std norm	0.624	0.041							
		hist match	0.583	0.045							
	Zhang	Image Norm	default	0.911	0.002						
		std norm	0.913	0.001							
		hist match	0.911	0.002							
		Batch Norm	default	0.909	0.001						
		std norm	0.867	0.005							
		hist match	0.862	0.008							
	Ours4	Image Norm	default	0.920	0.001						
		std norm	0.918	0.002							
		hist match	0.920	0.001							
		Batch Norm	default	0.913	0.003	Precision		Recall		F1	
		std norm	0.886	0.004	mean	std	mean	std	mean	std	
		hist match	0.879	0.002							
Lottes	Image Norm	default	0.646	0.021	0.550	0.109	0.629	0.062	0.585	0.091	
	std norm	0.656	0.007	0.682	0.060	0.631	0.028	0.653	0.028		
	hist match	0.695	0.002	0.760	0.010	0.697	0.016	0.727	0.012		
	Batch Norm	default	0.603	0.013	0.452	0.047	0.437	0.062	0.441	0.038	
	std norm	0.570	0.017	0.529	0.121	0.319	0.078	0.384	0.070		
	hist match	0.543	0.017	0.457	0.101	0.181	0.046	0.252	0.051		
Spray	Zhang	Image Norm	default	0.800	0.005	0.851	0.015	0.900	0.022	0.875	0.014
		std norm	0.792	0.008	0.819	0.020	0.880	0.021	0.848	0.017	
		hist match	0.792	0.007	0.825	0.026	0.890	0.012	0.856	0.016	
		Batch Norm	default	0.797	0.006	0.837	0.015	0.902	0.016	0.868	0.011
		std norm	0.787	0.003	0.833	0.039	0.846	0.025	0.838	0.009	
		hist match	0.778	0.007	0.862	0.020	0.834	0.029	0.847	0.022	
	Ours4	Image Norm	default	0.830	0.003	0.879	0.020	0.905	0.016	0.892	0.015
		std norm	0.822	0.006	0.870	0.009	0.895	0.020	0.882	0.012	
		hist match	0.822	0.002	0.878	0.012	0.903	0.018	0.891	0.009	
		Batch Norm	default	0.823	0.004	0.863	0.037	0.900	0.024	0.881	0.026
		std norm	0.812	0.005	0.871	0.017	0.854	0.020	0.862	0.017	
		hist match	0.815	0.006	0.884	0.038	0.892	0.012	0.887	0.021	

Table S2: Experiment 2a metrics:

Task	Model	Normalization	mIoU								
			mean	std	Precision		Recall		F1		
Plant	Lottes	Image	default	0.376	0.077						
		Norm	std norm	0.534	0.054						
			hist match	0.703	0.052						
			Batch	default	0.409	0.058					
		Norm	std norm	0.402	0.037						
			hist match	0.438	0.083						
	Zhang		Image	default	0.643	0.085					
		Norm	std norm	0.810	0.016						
			hist match	0.821	0.054						
			Batch	default	0.654	0.150					
		Norm	std norm	0.787	0.051						
			hist match	0.813	0.025						
	Ours4		Image	default	0.821	0.008					
		Norm	std norm	0.826	0.039						
			hist match	0.854	0.006						
			Batch	default	0.820	0.016					
		Norm	std norm	0.831	0.008						
			hist match	0.841	0.005						
Lottes	Image		default	0.307	0.096	0.037	0.014	0.114	0.051	0.056	0.022
	Norm	std norm	0.577	0.010	0.325	0.059	0.454	0.028	0.376	0.041	
		hist match	0.633	0.021	0.634	0.052	0.572	0.040	0.599	0.019	
		Batch	default	0.429	0.085	0.064	0.041	0.154	0.034	0.086	0.039
	Norm	std norm	0.522	0.011	0.296	0.070	0.182	0.062	0.214	0.050	
		hist match	0.528	0.016	0.274	0.078	0.192	0.069	0.214	0.063	
Zhang		Image	default	0.690	0.026	0.717	0.101	0.595	0.051	0.646	0.058
	Norm	std norm	0.685	0.019	0.452	0.093	0.754	0.023	0.559	0.063	
		hist match	0.734	0.010	0.735	0.043	0.732	0.039	0.733	0.037	
		Batch	default	0.667	0.047	0.524	0.274	0.629	0.110	0.505	0.140
	Norm	std norm	0.702	0.010	0.652	0.077	0.675	0.014	0.660	0.034	
		hist match	0.700	0.011	0.832	0.063	0.672	0.027	0.741	0.024	
Ours4		Image	default	0.747	0.005	0.882	0.045	0.637	0.017	0.740	0.025
	Norm	std norm	0.739	0.037	0.650	0.120	0.769	0.029	0.700	0.081	
		hist match	0.776	0.008	0.843	0.023	0.772	0.034	0.806	0.028	
		Batch	default	0.752	0.006	0.946	0.008	0.676	0.021	0.788	0.015
	Norm	std norm	0.783	0.004	0.871	0.019	0.758	0.020	0.810	0.014	
		hist match	0.776	0.002	0.822	0.016	0.777	0.017	0.799	0.010	

HistMatch Normalization in Python (numpy):

```

import numpy as np
from math import erf

# CDF for N(0,1)
def phi(x):
    return (1.0 + erf(x / 2.0**0.5))/2.0

# The number of points for which to calculate phi
N = 301

# [-3,3] will capture 99.7% of a standard normal distribution
yval = np.linspace(-3,3,N,endpoint=True)
ncdf = np.tile(np.array(list(map(phi,yval))).reshape(1,-1), (256,1))

# perform HistMatch normalization on an image.
def hist_match_norm(I):
    # Assume greyscale. Pixel values from 0 to 255
    # Normalize each channel separately for colour image

    # Cumulative histogram (normalized)
    h1 = np.sum(I.reshape(-1,1) <= range(256),0)
    h1 = h1/h1[-1]

    # Calculate midpoints
    h1 = (np.pad(h1, (1,0))[:-1] + h1)/2

    # Reshape for comparison
    h1 = h1.reshape(-1,1)

    # Find closest match between h1 and ncdf
    A = np.abs(h1-ncdf)
    f = yval[np.argmin(A,axis=1)]

    # apply function to I and return
    return f[I]

if __name__ == "__main__":
    # Display normalization graphs of test image (red channel)
    import PIL
    import matplotlib.pyplot as plt
    from matplotlib import gridspec

    I = np.array(PIL.Image.open("test.png"))
    I = I[:, :, 0]

    J = (I - np.mean(I))/np.std(I)

    K = hist_match_norm(I)

    def snpdf(x):
        return (1 / np.sqrt(2*np.pi)) * np.exp(-0.5*x**2)

    x = np.linspace(-3,3,101,endpoint=True)
    p = snpdf(x)

```

```
fig = plt.figure(figsize=(6,8))
gs = gridspec.GridSpec(3,1,height_ratios=[4,1,1])

ax = [plt.subplot(g) for g in gs]

ax[0].imshow(I,cmap='gray')
ax[0].set_title('Image')
ax[0].set_axis_off()

ax[1].hist(J.flatten(),bins=50,density=True,alpha=0.6)
ax[1].plot(x,p,'k',linewidth=2)
ax[1].set_title('Naive')

ax[2].hist(K.flatten(),bins=50,density=True,alpha=0.6)
ax[2].plot(x,p,'k',linewidth=2)
ax[2].set_title('HistMatch')

plt.tight_layout()
plt.show()
```

Higher Degree Research Thesis by Publication
University of New England

STATEMENT OF AUTHORS' CONTRIBUTION

We, the Research Master/PhD candidate and the candidate's Principal Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated in the *Statement of Originality*.

	Author's Name (please print clearly)	% of contribution
Candidate	Jonathan Ford	33
Other Authors	Edmund Sadgrove	33
	David Paul	33

Name of Candidate: Jonathan Ford

Name/title of Principal Supervisor: Dr Edmund Sadgrove



Candidate

13th October 2025

Date



Principal Supervisor

14/10/25

Date

Higher Degree Research Thesis by Publication
University of New England

STATEMENT OF ORIGINALITY

We, the Research Master/PhD candidate and the candidate's Principal Supervisor, certify that the following text, figures and diagrams are the candidate's original work.

Type of work	Page number/s
Journal Paper	103-124
Supplementary Material	126-134

Name of Candidate: Jonathan Ford

Name/title of Principal Supervisor: Dr Edmund Sadgrove



Candidate

13th October 2025

Date



Principal Supervisor

14/10/25

Date

Chapter 5

Dual-Task Multi-Species Network Design and Training Using Advanced Augmentation Techniques

5.1 Introduction

This paper, titled “Dual-Task Multi-Species Network Design and Training Using Advanced Augmentation Techniques” [60] further develops the joint plant and spraypoint detector presented in the previous chapter, by evaluating it on an expanded dataset containing multiple weed species. In addition, it investigates the use of advanced, sample-blending augmentation techniques, which have not previously been applied in weed localization studies. Consequently, beyond the research questions explored in the preceding paper, this study also addresses Research Question 5, related to multi-species detection, and Research Question 7, concerning the effectiveness of advanced augmentation techniques.

In addition to serrated tussock, the expanded dataset used in this study also included variegated thistle, horehound, and nettle, providing valuable data on the use of CNNs on a multi-species dataset (Research Question 5). While spraypoint annotations were available only for serrated tussock, this design allowed the study to focus on evaluating the spraypoint detector’s ability to avoid false positives when exposed to a broader diversity of weed and non-weed pasture species. Furthermore, because some weed species were underrepresented in the dataset, the study also tested the robustness of the models when trained on unbalanced data—a common challenge in real-world weed detection scenarios. As such, while distinguishing between different long-bladed grass species was outside the scope of this work, the expanded dataset still provided a valuable test of the detectors’ generalization and reliability under practical conditions.

Although sample-blending augmentation techniques have been widely explored

in the broader computer vision literature, they have not yet been tested in the context of weed localization. To address this gap, this study evaluated two such techniques, comparing them against a standard zoom-and-rotate augmentation commonly used in weed localization studies. The results demonstrate the potential of sample-blending methods for this field and highlight their ease of implementation, paving the way for their broader adoption and further investigation in future research (Research Question 7).

As shown in Figure 8 (columns 3 and 5) of the published paper, models trained using zoom-and-rotate augmentation often failed to generalize robustly, frequently misclassifying large regions of an image. These errors typically involved assigning the wrong weed species to a region or incorrectly classifying regions as either positive or negative. Regarding sample-blending augmentation, the different models often produced highly similar output maps, with the main difference being incorrect spraypoint classification (Figure 8, column 4, row 1 of the published paper).

One of the novel ideas explored in this study is the use of reduced-size output segmentation heatmaps. This approach was partially inspired by the observation that a large proportion (over 50%) of the FLOP budget in some models was consumed by the final convolutional layers, where the network operated on full-resolution feature maps. By removing these later layers and instead introducing a simple upsampling layer when full-resolution output is required, the model’s FLOP budget—and potentially its inference time—can be reduced, while still maintaining sufficient resolution to enable effective control of spray nozzles for site-specific weed management.

Although a variety of spatial context and attention modules were explored during this study, their performance often fell short of the baseline model. Nevertheless, the results of these experiments are included in Appendix A for completeness and to support future research in this area.

An overview of the datasets used in this study is presented in table 5.1.

Dataset	Images	Class Representation					Spraypoints Count
		Neg.	Horehound	Nettle	Tussock	Var. Thistle	
Jug-Aug22	240	0.53	0.06	0.13	–	0.38	–
Jug-Oct22	400	0.76	–	0.01	–	0.23	–
Wal-Dec22	300	0.83	0.17	–	–	–	–
Wal-May23	237	0.73	0.14	0.11	–	0.01	–
Yel-Oct21	300	0.67	–	–	0.33	–	212
Yel-Apr22	295	0.54	–	–	0.46	–	223
Yel-Jul22	300	0.71	–	–	0.29	–	192
<i>Image resolution: 768 × 576</i>		<i>Annotation level: 16 × 16 pixels</i>					

TABLE 5.1: Dataset summary showing image counts, class representation, and spraypoint counts. Image resolution and annotation level are common to all datasets.

5.2 Published Paper

This paper was published in *Smart Agricultural Technology*, which is ranked by SJR¹ as a Q1 journal in Agricultural and Biological Sciences (miscellaneous) and Computer Science (miscellaneous) for 2023.

¹<https://www.scimagojr.com/journalsearch.php?q=21101111783&tip=sid&clean=0>



Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technologyDual-task multi-species network design and training using advanced augmentation techniques[☆]Jonathan Ford^{*}, Edmund Sadgrove, David Paul^{ID}

School of Science and Technology, University of New England, Armidale, Australia

ARTICLE INFO

Keywords:

Convolutional neural networks
Computer vision
Site-specific weed management
Precision agriculture
CutMix augmentation

ABSTRACT

Accurate weed localization in pasture environments is essential for effective site-specific weed management (SSWM). This study evaluates two dual-task convolutional neural networks (CNNs) - a truncated ConvNeXt (tCN) and a truncated UniStemNet (tUSN) - for simultaneous plant segmentation and spraypoint detection across four weed species in south-east Australian pastures. To assess model performance, two testing regimes were used: a pooled test, where all images were randomly split for training and validation, and condition-invariance tests, in which models were trained and evaluated on entirely separate groups of images collected under different environmental conditions. Four data augmentation strategies were compared, including advanced image blending techniques CutMix and MixUp, which had not previously been explored in this domain. In the pooled test, the tUSN model achieved a mean Intersection over Union (mIU) of up to 0.897 for plant segmentation, while the tCN model reached an F1-score of 0.880 for spraypoint detection. Under condition-invariance testing, CutMix augmentation provided the most consistent generalization, with mIU values up to 0.897 for plant segmentation, and F1-scores up to 0.954 for spraypoint detection. HistMatch normalization assisted with model generalization in weaker augmentation setups. These findings demonstrate that advanced augmentation and normalization strategies are critical for achieving robust CNN-based weed detection across diverse real-world field conditions.

1. Introduction

Efficient and accurate weed management is a critical challenge in modern agriculture, particularly in extensive pasture systems where manual weed control is impractical. Site specific weed management (SSWM) aims to target undesirable pasture species while leaving other sections of the pasture untreated, thereby reducing chemical use and limiting damage to surrounding vegetation [12]. However, the success of such systems relies on the ability to detect, localize, and discriminate between weed species under diverse environmental conditions [8].

Convolutional neural networks (CNNs) are commonly used in weed localization studies [14,3,18,5], but their large number of parameters makes them prone to overfitting, leading to reduced performance when tested on unseen samples. Although data augmentation techniques can be used to mitigate this issue [14], to date, most augmentation strategies employed in weed localization studies have been limited to combinations of spatial transformations [21,6,1], pixel intensity shifts [16] and

the addition of random noise or blurring [7]. By comparison, sample mixing techniques, where two or more samples are combined to create a new one, have not yet been investigated. Such techniques, including CutMix [19] and MixUp [20] may offer promising avenues for developing more robust and diverse augmentation strategies.

Previous work has explored dual-task computer vision models for weed management, particularly where varied treatment methods, such as herbicide application or mechanical removal are required [10,11,21,6]. These models typically use convolutional neural networks (CNNs), and take an input image of pasture or crop field and output multiple heatmaps, one showing weed (or crop) locations, and another indicating the centre of each plant. While foliar herbicides or fertilizers are generally applied across the entire plant, treatments like root-absorbed herbicides or mechanical removal target the plant's centre, which may be referred to as the spraypoint or stem [10,6].

Ford et al. [6] demonstrated the feasibility of real-time, pixel-level weed detection and spraypoint localization in complex pasture environ-

[☆] The first author was supported by a Commonwealth of Australia RTP scholarship, with a top-up scholarship from the "Novel Autonomous Robotic Weed Control to Maximise Agricultural Productivity" (Commonwealth Grant CRCPIX000099) CRC. The authors would like to thank the NSW Department of Primary Industries and Agent Oriented Software for providing the dataset.

^{*} Corresponding author.

E-mail addresses: jford3@myune.edu.au (J. Ford), esadgro2@une.edu.au (E. Sadgrove), dpaul4@une.edu.au (D. Paul).

<https://doi.org/10.1016/j.atech.2025.101594>

Received 29 July 2025; Received in revised form 15 October 2025; Accepted 31 October 2025

ments, using a dual-task CNN. Their model was a ConvNeXt-inspired architecture with separate decoders for each task, which outperformed prior approaches, including models by Lottes et al. [10] and Zhang et al. [21], across multiple experimental conditions. To improve generalization under variable imaging conditions, a histogram-based normalization method (HistMatch) was introduced, which significantly enhanced performance, particularly for models with lower representational capacity.

However, the above research was limited to a dataset containing a single weed species, serrated tussock (*Nassella trichotoma*, hereafter tussock). This left untested whether the models were sufficiently robust to handle the situation where multiple weed species were present, as is commonly the case in pastures. For this reason, this study was conducted to further test the abilities and limitations of the models in a more complex pastoral setting. Specifically, the following research questions were addressed:

- How well do the CNN models handle multispecies detection in pastures, including underrepresented weeds?
- How robust are the models to unseen growing conditions during testing?
- Can the spraypoint detector maintain accuracy with additional species present?
- Do advanced augmentation techniques improve segmentation performance in pasture settings?

2. Related work

2.1. Dual-task CNN models

Two key studies have proposed architectures for the dual task of plant segmentation and stem detection. Lottes et al. [10] designed a model with a shared encoder and two separate decoders. The encoder is built using dense blocks composed of multiple densely connected 3×3 convolutional layers, with each layer outputting a fixed number of feature maps (the growth rate). Bottleneck layers using 1×1 convolutions help reduce computational load, and downsampling is achieved via 5×5 convolutions with stride 2. The decoder arms use transpose convolutions (2×2 , stride 2) for upsampling, and skip connections are implemented via filter-axis concatenation.

In contrast, Zhang et al. [21] constructed a model based on MobileNetV2's inverted residual modules (IRMs), which combine depth-wise and point-wise convolutions to improve efficiency while preserving accuracy. The encoder contains multiple stages of IRMs, with downsampling achieved by stride-2 depth-wise convolutions. The architecture features task-specific decoder branches that split off from different encoder resolutions. These decoder arms include intra-task connections to enhance feature flow and a cross-task fusion link from the segmentation decoder to the stem decoder, ensuring that stem localization is guided by plant structure. Each decoder arm produces an independent output, and deep supervision is applied at all outputs to strengthen training.

The model described in the preceding paragraph was further refined by Ford et al. [6] through simplification of the decoder arms, an increase in parameter count, and the incorporation of ConvNeXt-style architectural elements [9], including a "patchify" stem, ConvNeXt modules, and adjusted stage ratios. This updated model was evaluated on a joint plant segmentation and spraypoint localization task using a dataset containing serrated tussock, with ground-level images collected from south-east Australian pastures.

2.2. Data augmentation in weed localization studies

Data augmentation strategies artificially expand the size of a training dataset by generating new images from existing ones through a variety of random image transformations. This is particularly important when training CNNs with a large number of parameters, which are prone to overfitting [14]. In the field of weed classification, augmentation techniques are typically limited, and include spatial transformations such

as random horizontal and vertical flips [21,13], zooms, rotations, and shears [7,6,1,14,18]. Other common approaches include adjusting pixel intensity and/or contrast [6,1,14], and adding random noise or blurring [7,16]. More specialized techniques have also been explored, such as dividing images into $n \times n$ patches and augmenting each patch independently [2], or applying style-transfer methods to simulate environmental effects like dew on plants [17].

3. Methods

3.1. Dataset

The dataset consisted of images containing four weed species: serrated tussock, horehound, nettle, and variegated thistle. The tussock images were the same as those used in [6]. The images were captured at three distinct locations — Jugiong (lat. -34.862, long. 148.319), Wal-lacetown (-34.947, 147.488), and Yelangelo (-34.860, 149.168) — over a 20-month period from October 2021 to May 2023, spanning multiple collection dates. All images were resized to 768×576 pixels, and the corresponding segmentation maps were downsampled by a factor of 16, resulting in a final output size of 48×36 pixels. The downsampled output greatly decreases the computational complexity of the models, while maintaining sufficient resolution to allow for effective activation of herbicide spray nozzles.

Not all weed species were present at every location or collection date. The proportion of image patches covered by each species (and the negative class), as well as the number of images across the different locations and collection dates is presented in Fig. 1.

Two output segmentation maps were generated: one for plant segmentation and another for spraypoint segmentation. The spraypoint represents the centre of the plant, where root-absorbed herbicides are applied to maximize uptake and effectiveness. Spraypoints were labelled only for serrated tussock, as the other weed species are typically managed using foliar herbicides that are applied to the entire plant.

3.2. CNN models

Two models were evaluated, based on those referred to as Zhang's model and our model in [6]. Zhang's model is the UniStemNet model by Zhang et al. [21], which was adapted for the spraypoint detection task. To align with the downsampled output segmentation maps, both models were truncated by removing the decoder arms responsible for processing higher-resolution features.

For Zhang's models, this means that there is a single decoder arm for each task, at a downsampling factor of 16. This has removed the bidirectional feature map exchanges that operated between adjacent intra-task decoder arms. However, the cross-task feature fusion that operates from the plant segmentation decoder to the spraypoint decoder has been retained.

For our model, both tasks have a single decoder arm, at the $32 \times$ downsampled scale for the spraypoint task, and at the $16 \times$ downsampled scale for the plant task. For the plant task, a U-Net style connection is used to concatenate upsampled feature maps from the $32 \times$ downsampled scale to the $16 \times$ downsampled feature maps from the encoder arm. This is in contrast to the bidirectional feature map exchanges that were present in the original model. As is the case for Zhang's model, the cross-task feature fusion has been retained. The number of layers in the encoder has also been halved, so that there are 2, 6 and 2 layers at downsampling scales of 8, 16 and 32 respectively. These models will be called the truncated UniStemNet (tUSN) and truncated ConvNeXt (tCN) models in the remainder of the paper. An overview of the structure of both models used in this paper is presented in Fig. 2.

3.3. Augmentation strategies

In order to evaluate the effectiveness of different augmentation strategies, including sample-mixing techniques, four different augmen-

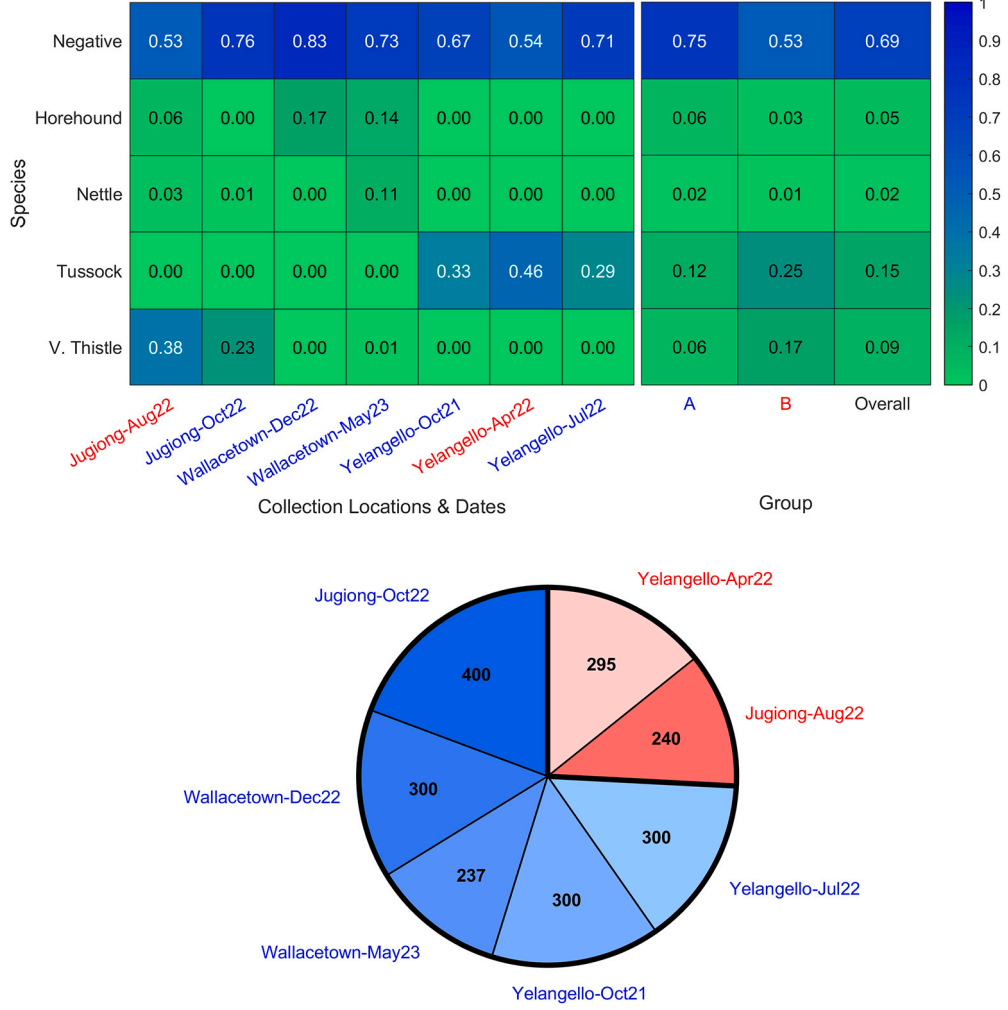


Fig. 1. The proportion of image patches covered by each species (and the negative class) and the number of images across different locations and collection dates. The images have been separated into groups A (blue), and B (red).

tation schemes were tested in this study, being Zoom/Rotate, CutMix [19], MixUp [20] and both CutMix and MixUp. For Zoom/Rotate augmentation, images were randomly zoomed with limits of -20% to 10% and rotated with limits of $\pm \frac{\pi}{8}$.

CutMix blends two images by replacing specific regions of one image with patches from another, effectively combining their content. This enhances regularization and encourages the model to learn to identify objects even when key discriminative features are partially missing. While vanilla CutMix randomly selects a rectangular region for replacement, in this study, a free-form mask was generated by applying a Gaussian filter, with $\sigma = 2$, to a randomly generated mask, followed by binarization. The threshold for binarization was set to the median pixel value of the mask prior to binarization, ensuring that each image contributed an equal area to the augmented image. The mask was chosen to have dimension 48×36 , matching the size of the output masks. The process of mask generation is shown on the right side of Fig. 3.

By contrast, MixUp generates new samples by blending two images at the pixel level, which is achieved using a weighted average of their pixel values. This technique encourages smoother decision boundaries and reduces overfitting. In this study, the coefficient for the weight factor was drawn from a beta distribution with $\alpha = \beta = 2$, as this worked well in preliminary tests. In all forms of augmentation, random horizontal and vertical flips were also applied. Examples of the different augmentation strategies are provided in Fig. 3.

3.4. Testing

An initial pooled test was conducted in which all images were combined into a single dataset, from which an 80%/20% training/validation split was generated. This setup was used to optimize the learning rate and to evaluate the different augmentation strategies under conditions where the training and validation data shared the same growing conditions.

To evaluate how well the models generalized to previously unseen conditions, two condition-invariance tests were conducted. The training data were first divided into two groups, A and B, as indicated by text colour in Fig. 1, where blue and red represent groups A and B, respectively. In Invariance Test 1, models were trained and validated on Group A (using an 80%/20% split), and tested on Group B. In Invariance Test 2, the setup was reversed, with Group B used for training and validation, and Group A for testing.

3.5. Training parameters

Training was carried out in Keras/TensorFlow as per [6]. Learning rates from 0.0001 to 0.01 were tested in the pooled test. For the condition-invariance tests, a fixed learning rate of 0.001 was used, as this was found to be the most consistent in the pooled test. For the pooled test, 300 epochs of training were used, whereas for the condition-invariance tests, 700 epochs were used to ensure that the training converged to a maximum. For the spraypoint task, due to extreme class

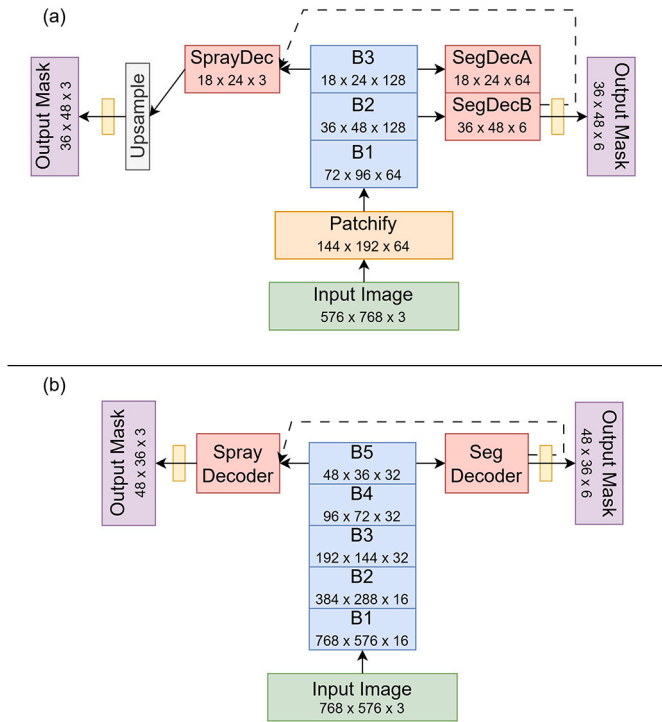


Fig. 2. The truncated ConvNeXt (a) and truncated UniStemNet (b) models used in this paper. Colours represent different network components: green for the input layer, orange for the patchify stem, blue for the encoder, red for the decoder, grey for upsampling, yellow for the loss layer, and purple for the output layer. Arrows indicate the flow of information, while dashed arrows represent cross-task feature fusion.

imbalance, IOU loss was used [15]. For the segmentation task, cross entropy loss was used. Other hyperparameters were set to the values used in [6], without additional optimization. In all cases, the model weights corresponding to the best performance on the validation data were restored at the end of training. Normalization was carried out on a per-image basis. For the pooled test, standard normal normalization was used. For the condition-invariance tests, default, standard normal, and HistMatch normalization techniques were compared. Further information about these normalization techniques can be found in [6].

3.6. Post-processing and metrics

To convert the model-generated heatmaps for the spraypoint task into individual predictions, the following steps were applied. First, a threshold value of 0.5 was used to binarize the heatmap. The resulting binary mask was then processed to identify connected components, corresponding to potential detections. To reduce noise and eliminate small, spurious detections, a minimum size threshold was applied to each component, with values of 2, 4, 6, 8, and 10 pixels tested.

The centroid of each connected component was used to determine the positions of both true and predicted spraypoints. These spraypoints were then matched based on proximity, with a distance threshold of 3.125 (equivalent to 50 pixels in the input image, which is the same threshold used in [6]) used to classify a predicted spraypoint as a true positive. Although this value was not rigorously selected, it is based on an estimation of the precision of the herbicide spray nozzles, and the spread of the central root system of the tussock plants. Each true spraypoint was paired with at most one predicted spraypoint, and vice versa. True spraypoints that were not matched within the required distance were counted as false negatives, while unmatched predicted spraypoints were counted as false positives.

For the plant segmentation task, the mean Intersection over Union (mIU) was computed on a per-patch basis. For the spraypoint task, evaluation was performed using the per-object F1 score.

4. Results

The mIU for the plant segmentation task, and the F1-score for the spraypoint task for the pooled test are presented in Fig. 4. For the tCN model, connected components with a size of at least 4 output pixels were counted as valid detections, whereas for the tUSN model, the threshold was set to 10 pixels, as these were the values that led to the best results (in the pooled test). The average improvement in F1 score (across all augmentation techniques) when small connected components were ignored was 0.001 for the tCN model, compared to 0.049 for the tUSN model.

The results for the condition-invariance tests are presented in Fig. 5 for test one, and Fig. 6 for test two. Each figure presents four heatmaps: the top row corresponds to the tCN model, and the bottom row to the tUSN model. The left column displays results for the plant segmentation task using the mIU metric, while the right column shows results for the spraypoint task using the F1 score. Within each heatmap, rows represent the four different augmentation strategies, and columns represent the three different normalization techniques. The colour scale is non-linear and was chosen to accentuate differences between closely ranked metric scores, particularly at the higher end of the scale. The same colour scale is used for both figures, as well as for Fig. 4, to ensure consistency and comparability.

In test one, CutMix augmentation with default preprocessing performed best for both models and tasks (Fig. 5, second row, left column of each heatmap). The tCN model performed better than the tUSN model for both tasks, with an improvement in metrics of 0.012 and 0.047 for the plant and spraypoint tasks respectively. In test two, CutMix augmentation with default preprocessing achieved the best performance for both models for the plant task (Fig. 6, left-hand heatmaps). For the spraypoint detection task, Both augmentation with HistMatch preprocessing performed best for the tCN model, while Both augmentation with default preprocessing was the top performer for the tUSN model (Fig. 6, right-hand heatmaps). In terms of model comparison, the tUSN model slightly outperformed the tCN model in plant segmentation, with a marginal improvement of 0.001. For the spraypoint detection task, the tCN model performed better, showing a more substantial improvement of 0.015.

Fig. 7 presents the normalized confusion matrix for the plant segmentation task, comparing both models on the combined test using CutMix augmentation. In both cases, confusion between weed species was minimal, with the majority of the mIU error caused by weeds being misclassified as negative.

Fig. 8 presents example outputs from condition invariance test one, using default normalization. The outputs are shown for both the Zoom/Rotate and CutMix augmentation strategies, selected to highlight differences between the two models as well as the effects of each augmentation approach. As observed (columns 3 and 5), models trained with Zoom/Rotate augmentation often produced large regions with incorrect class labels in the plant segmentation task and frequently failed to detect spraypoints.

5. Discussion and conclusion

This study aimed to evaluate the effectiveness of CNN models for the detection and segmentation of multiple weed species in South-East Australian pastures, with four species represented in the dataset. The selected CNNs were dual-task models capable of simultaneously generating a segmentation mask of the target weed and identifying the plant centre, which corresponds to the root zone and is therefore the optimal spraypoint for root-absorbed herbicides, such as Flupropanate. Spraypoint annotations were provided only for serrated tussock, however, which is commonly treated with Flupropanate. The study also placed particular emphasis on augmentation strategies, especially advanced image blending techniques, which have not previously been explored in the context of weed localization.

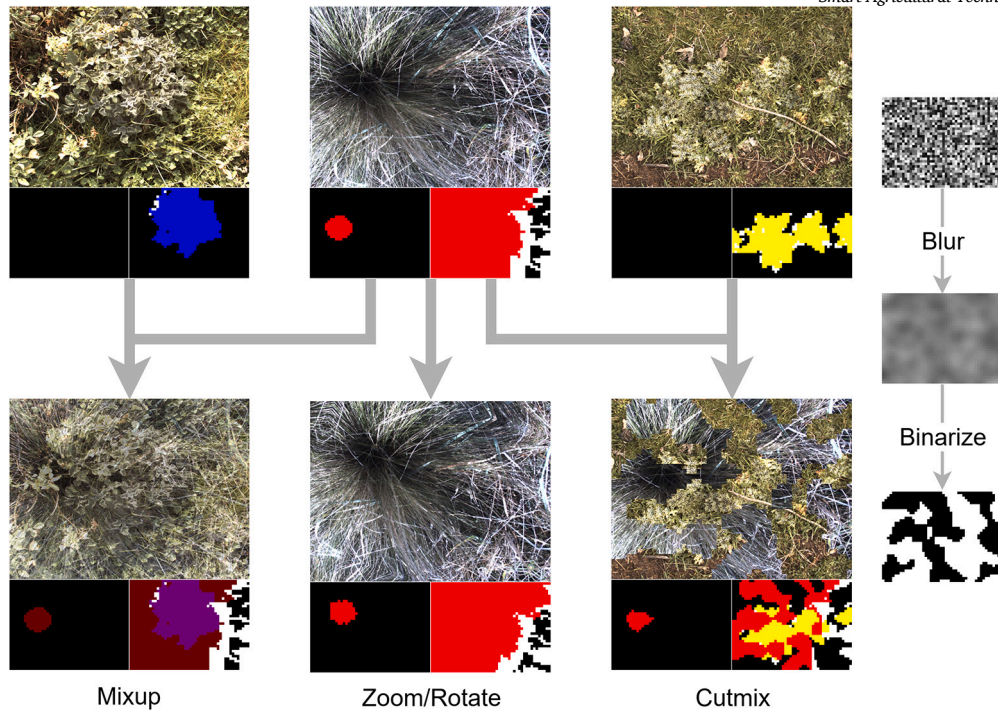


Fig. 3. Examples of the augmentation methods used. The top row shows original samples from the dataset, while the bottom row shows corresponding augmented samples. The masks are shown below the image with the spraypoint mask on the left, and the segmentation mask on the right. The method of mask generation for CutMix is visualized on the right side of the figure.

Augmentation Strategies	Model/Task			
	tCN - Plant mIU	tCN - Spraypoint F1	tUSN - Plant mIU	tUSN - Spraypoint F1
	ZR	CutMix	MixUp	Both
	0.861 ±0.002	0.885 ±0.008	0.876 ±0.002	0.804 ±0.019
Augmentation Strategies	Model/Task			
	tCN - Plant mIU	tCN - Spraypoint F1	tUSN - Plant mIU	tUSN - Spraypoint F1
	ZR	CutMix	MixUp	Both
	0.880 ±0.001	0.854 ±0.011	0.897 ±0.002	0.784 ±0.019
Augmentation Strategies	Model/Task			
	tCN - Plant mIU	tCN - Spraypoint F1	tUSN - Plant mIU	tUSN - Spraypoint F1
	ZR	CutMix	MixUp	Both
	0.851 ±0.002	0.874 ±0.018	0.868 ±0.004	0.835 ±0.010
Augmentation Strategies	Model/Task			
	tCN - Plant mIU	tCN - Spraypoint F1	tUSN - Plant mIU	tUSN - Spraypoint F1
	ZR	CutMix	MixUp	Both
	0.868 ±0.001	0.881 ±0.009	0.891 ±0.002	0.815 ±0.008

Fig. 4. Results are presented for the pooled test, where all images were combined into a single dataset from which training and testing sets were sampled. Reported values represent the mean $\pm$ standard deviation across seven test iterations. Metrics include the mean Intersection over Union (IoU) for the segmentation task and the F1 score for the spraypoint detection task, evaluated for both the truncated ConvNeXt (tCN) and truncated UniStemNet (tUSN) models.

When images from all collection locations and dates were pooled (Fig. 4), both models performed well on the multispecies plant segmentation task, achieving mIU scores between 0.851 and 0.897. On average, the tUSN model outperformed the tCN model by 0.018 across different augmentation strategies. For the spraypoint detection task, F1-scores ranged from 0.784 to 0.880, with the tCN model performing 0.064 better than the tUSN model on average. These results indicate that both models were effective in a multispecies context and that the spraypoint detector remained robust even with an increased number of plant species present.

The dataset used in this study was highly imbalanced, with the negative class covering 69% of the total image area, while individual weed

species accounted for between 2% (nettle) and 15% (tussock), as illustrated in Fig. 1. As shown in the example confusion matrices for both models (Fig. 7), weed species with lower representation (horehound and nettle) tended to exhibit reduced classification accuracy. Nevertheless, overall accuracy remained relatively high, with all species achieving scores of at least 0.90. These results highlight the robustness of the selected CNN models, even when faced with significant class imbalance in the dataset.

When the dataset was split based on collection locations and dates, robust model performance could still be achieved, with mIU scores of up to 0.897 for the plant segmentation task, and F1-scores as high as 0.962 for the spraypoint detection task (Figs. 5 and 6). These results demonstrate that the models could generalize well to unseen growing conditions, even in a multispecies context.

However, in this case, the choice of augmentation strategies, and normalization techniques became critically important, with model performance being significantly impacted in the worst cases. In most cases, Zoom/Rotate augmentation performed poorly in the condition invariance tests, with mIU scores dropping as low as 0.468 for plant segmentation and F1-scores as low as 0.160 for spraypoint detection. This poor performance is highlighted in Fig. 8. While these results could be improved somewhat using HistMatch normalization, they were still significantly worse than those using other augmentation techniques.

Overall, CutMix proved to be the most consistent augmentation strategy for condition invariance, irrespective of model, test, task or normalization technique. While CutMix is generally considered to be better adapted to localization tasks, as compared to classification tasks, in which MixUp is superior [4], it is notable that this distinction only became apparent in the condition invariance experiments conducted in this study. The strength of this augmentation strategy, compared to others, may stem from both the approach itself and the specific method of mask generation, which was tailored to reflect the common scenario in pastures where plant overlap leads to partial occlusion of target species.

With respect to normalization strategies, it was found that HistMatch normalization greatly improved results in cases where the models were not robust enough to adapt to changes in growing conditions, as was the

tCN - Plant Task mIU					tCN - Spraypoint Task F1				
Augmentation Strategies	ZR	0.510 ±0.070	0.652 ±0.021	0.737 ±0.009	Augmentation Strategies	ZR	0.370 ±0.075	0.778 ±0.021	0.819 ±0.018
	CutMix	0.883 ±0.002	0.867 ±0.003	0.869 ±0.003		CutMix	0.933 ±0.009	0.889 ±0.016	0.910 ±0.013
	MixUp	0.760 ±0.010	0.752 ±0.006	0.732 ±0.018		MixUp	0.822 ±0.011	0.836 ±0.015	0.846 ±0.012
	Both	0.857 ±0.004	0.836 ±0.005	0.825 ±0.007		Both	0.914 ±0.008	0.880 ±0.013	0.879 ±0.010
	Default StdNorm HistMatch					Default StdNorm HistMatch			
Normalization Techniques					Normalization Techniques				

tUSN - Plant Task mIU					tUSN - Spraypoint Task F1				
Augmentation Strategies	ZR	0.468 ±0.052	0.632 ±0.016	0.727 ±0.006	Augmentation Strategies	ZR	0.160 ±0.116	0.729 ±0.038	0.797 ±0.021
	CutMix	0.871 ±0.004	0.850 ±0.004	0.849 ±0.007		CutMix	0.886 ±0.015	0.843 ±0.016	0.869 ±0.013
	MixUp	0.740 ±0.016	0.736 ±0.015	0.725 ±0.011		MixUp	0.770 ±0.037	0.773 ±0.011	0.826 ±0.020
	Both	0.852 ±0.005	0.818 ±0.009	0.809 ±0.006		Both	0.871 ±0.013	0.815 ±0.018	0.839 ±0.017
	Default StdNorm HistMatch					Default StdNorm HistMatch			
Normalization Techniques					Normalization Techniques				

Fig. 5. Results are presented for invariance tests 1, in which group A was used for training and validation, and group B used for testing. Reported values are the same as those presented in Fig. 4, evaluated for default, standard norm and HistMatch normalization, as discussed in [6].

tCN - Plant Task mIU					tCN - Spraypoint Task F1				
Augmentation Strategies	ZR	0.515 ±0.015	0.528 ±0.005	0.569 ±0.014	ZR	0.955 ±0.004	0.958 ±0.008	0.961 ±0.007	
	CutMix	0.896 ±0.002	0.876 ±0.005	0.872 ±0.004	CutMix	0.954 ±0.007	0.957 ±0.004	0.954 ±0.005	
	MixUp	0.708 ±0.024	0.744 ±0.008	0.722 ±0.008	MixUp	0.942 ±0.004	0.953 ±0.009	0.957 ±0.009	
	Both	0.878 ±0.002	0.847 ±0.006	0.849 ±0.004	Both	0.956 ±0.005	0.961 ±0.008	0.962 ±0.008	
	Default			StdNorm	HistMatch	Default			StdNorm
Normalization Techniques					Normalization Techniques				

tUSN - Plant Task mIU					tUSN - Spraypoint Task F1				
Augmentation Strategies	ZR	0.503 ±0.014	0.532 ±0.009	0.572 ±0.009	ZR	0.915 ±0.007	0.916 ±0.001	0.918 ±0.011	
	CutMix	0.897 ±0.002	0.877 ±0.005	0.868 ±0.003	CutMix	0.923 ±0.009	0.935 ±0.010	0.941 ±0.012	
	MixUp	0.672 ±0.008	0.719 ±0.007	0.698 ±0.010	MixUp	0.932 ±0.011	0.937 ±0.009	0.935 ±0.003	
	Both	0.885 ±0.004	0.860 ±0.003	0.852 ±0.004	Both	0.947 ±0.006	0.941 ±0.008	0.946 ±0.005	
	Default			StdNorm	HistMatch	Default			StdNorm
Normalization Techniques					Normalization Techniques				

Fig. 6. Results are presented for invariance test 2, in which group B was used for training and validation, and group A used for testing. Reported values are the same as those presented in Fig. 5.

case when Zoom/Rotate augmentation was used. In the case of CutMix, where models already performed strongly with default normalization, applying HistMatch led to reduced performance, which may be due to a loss of image descriptiveness or due to confusion introduced during training by enforcing a standard pixel intensity distribution. Finally, where model performance was strongest, the choice of normalization strategy appeared to have no discernible effect. This is exemplified in the right-hand column of Fig. 6, showing the spraypoint results for invariance test two.

In the pooled test, the parameter-rich ConvNeXt-style tCN model outperformed the tUSN model on the spraypoint task, achieving F1-score improvements ranging from 0.039 to 0.081 depending on the augmentation strategy. Conversely, the tUSN model delivered superior results on the segmentation task, with mIU gains between 0.015 and 0.023. In the condition invariance tests—particularly focusing on the top-performing CutMix augmentation, the tCN model outperformed the tUSN model by an average of 0.008 in mIU for segmentation and 0.033 in F1-score for spraypoint detection.

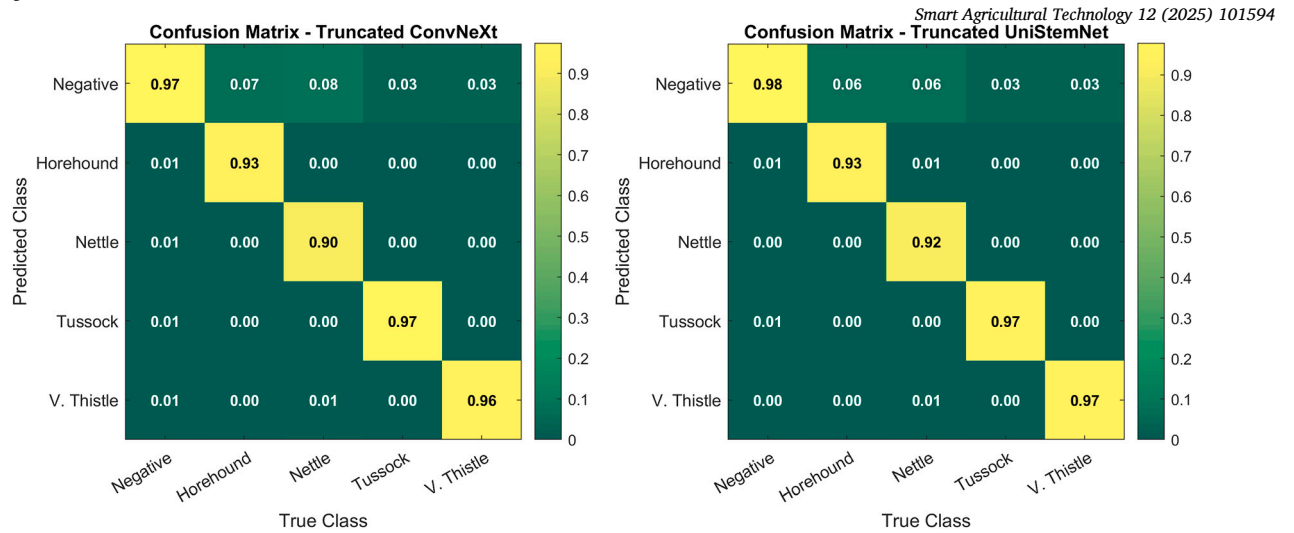


Fig. 7. Normalized confusion matrices for the plant segmentation task using the truncated ConvNeXt and truncated UniStemNet models. Results are shown for the pooled test with CutMix augmentation.

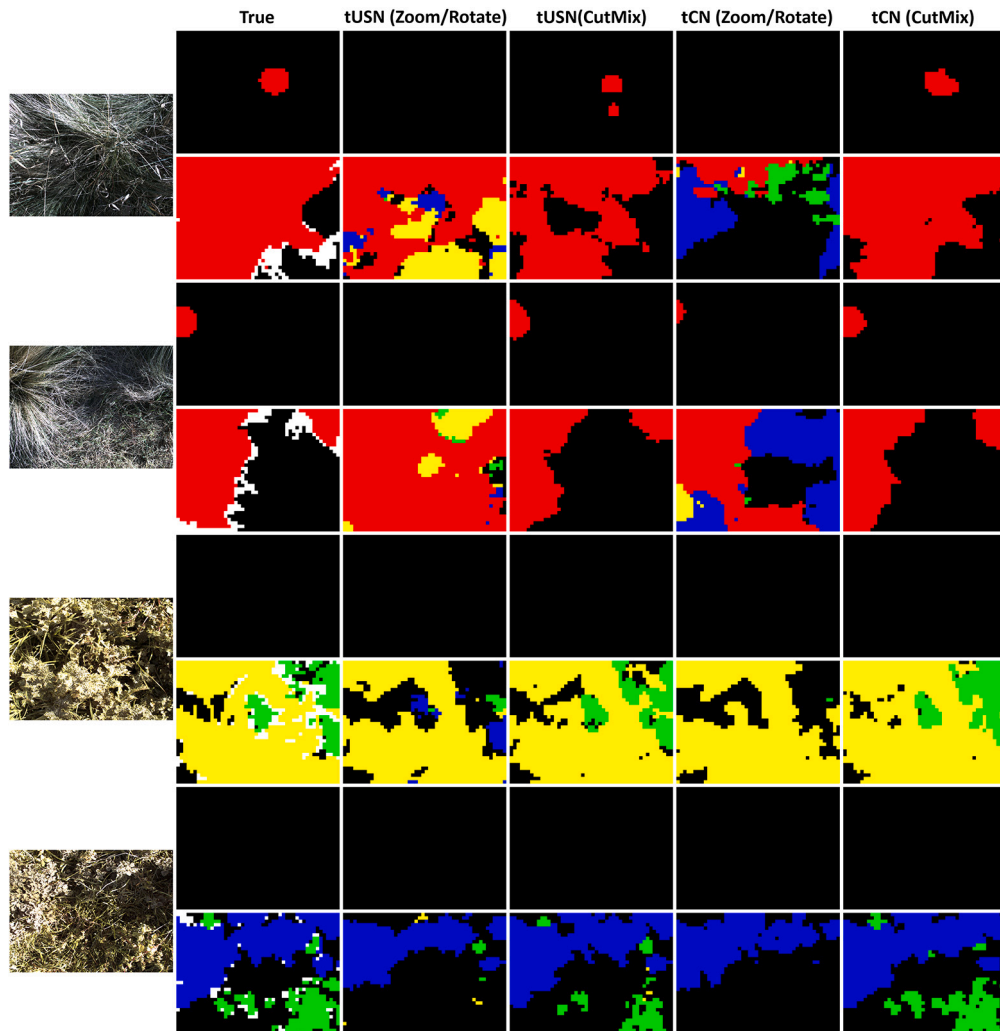


Fig. 8. Example outputs from condition invariance test one using default normalization. Left to right: input image, ground truth masks, and predicted masks under different model and augmentation settings. For each output, the spraypoint mask is shown above the plant segmentation mask. Colour legend—Black: Negative, Red: Tussock, Blue: Horehound, Yellow: Variegated Thistle, Green: Nettle, White: Difficult.

The choice of normalization technique has a significant impact on model performance, especially when evaluating condition invariance.

However, the optimal normalization method appears to depend on the specific training regimen, highlighting the need for further research

to better understand the factors that influence normalization effectiveness.

In the context of weed detection in Australian pastures, this study has demonstrated the potential of CNN models. However, further research is necessary to assess their effectiveness across a wider range of weed species that better represent the diversity of these environments. In particular, future work is needed to evaluate the models' ability to distinguish between morphologically similar long-bladed grass species, which may be either invasive weeds or beneficial native grasses.

This study further highlights the effectiveness of joint plant and spraypoint CNN detectors in multi-species environments. However, achieving robust performance under varying growing conditions requires careful consideration of normalization methods and data augmentation strategies. In this context, CutMix augmentation proved particularly useful, outperforming simpler approaches such as zooms and rotations. As a result, future weed segmentation studies, and in particular those involving test images from diverse locations or time points, should consider using CutMix augmentation during training. Moreover, while this study focused on joint plant segmentation and spraypoint detection, the augmentation strategies developed are equally applicable to single-task problems, which are more commonly addressed in the literature.

CRedit authorship contribution statement

Jonathan Ford: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Edmund Sadgrove:** Writing – review & editing, Supervision, Methodology, Conceptualization. **David Paul:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used chatGPT in order to improve the readability of the text. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] Shyam Prasad Adhikari, Heechan Yang, Hyongsuk Kim, Learning semantic graphics using convolutional encoder-decoder network for autonomous weeding in paddy, *Front. Plant Sci.* 10 (2019).
- [2] Anderson Brilhador, Matheus Gutoski, Leandro Takeshi Hattori, Andrei de Souza Inácio, André Eugênio Lazzaretto, Heitor Silvério Lopes, Classification of weeds and crops at the pixel-level using convolutional neural networks and data augmentation, in: 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI), IEEE, 2019, pp. 1–6.
- [3] Brendan Calvert, Alex Olsen, Bronson Philippa, Mostafa Rahimi Azghadi, Autoweed: detecting harrisia cactus in the goondiwindi region for selective spot-spraying, in: Proceedings of the 1st Queensland Pest Animal and Weed Symposium, Weed Society of Queensland Pty. Ltd., Toowoomba, QLD, Australia, 2019, p. 52.
- [4] Chengtai Cao, Fan Zhou, Yurou Dai, Jianping Wang, Kunpeng Zhang, A survey of mix-based data augmentation: taxonomy, methods, applications, and explainability, *ACM Comput. Surv.* 57 (2) (2024) 1–38.
- [5] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, Marcelo Theophilo Folhes, Weed detection in soybean crops using ConvNets, *Comput. Electron. Agric.* 143 (2017) 314–324.
- [6] Jonathan Ford, Edmund Sadgrove, David Paul, Joint plant-spraypoint detector with convnet modules and histmatch normalization, *Precis. Agric.* 26 (1) (2025) 24.
- [7] A.S.M. Mahmudul Hasan, Ferdous Sohel, Dean Diepeveen, Hamid Laga, Michael G.K. Jones, Weed recognition using deep learning techniques on class-imbalanced imagery, *Crop Pasture Sci.* (2022) 628–644.
- [8] Kun Hu, Zhiyong Wang, Guy Coleman, Asher Bender, Tingting Yao, Shan Zeng, Dezheng Song, Arnold Schumann, Michael Walsh, Deep learning techniques for in-crop weed recognition in large-scale grain production systems: a review, *Precis. Agric.* 25 (1) (2024) 1–29.
- [9] Zhuang Liu, Hanzhi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.
- [10] Philipp Lottes, Jens Behley, Nived Chebrolu, Andres Milioto, Cyril Stachniss, Joint stem detection and crop-weed classification for plant-specific treatment in precision farming, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 8233–8238.
- [11] Philipp Lottes, Jens Behley, Nived Chebrolu, Andres Milioto, Cyril Stachniss, Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming, *J. Field Robot.* 37 (1) (2020) 20–34.
- [12] Wouter H. Maes, Kathy Steppe, Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture, *Trends Plant Sci.* 24 (2) (2019) 152–164.
- [13] S. Imran Moazzam, Umar S. Khan, Waqar S. Qureshi, Tahir Nawaz, Faraz Kunwar, Towards automated weed detection through two-stage semantic segmentation of tobacco and weed pixels in aerial imagery, *Smart Agric. Technol.* 4 (2023) 100142.
- [14] Alex Olsen, Dmitry A. Konovalov, Bronson Philippa, Peter Ridd, Jake C. Wood, Jamie Johns, Wesley Banks, Benjamin Girenti, Owen Kenny, James Whinney, Brendan Calvert, Mostafa Rahimi Azghadi, Ronald D. White, DeepWeeds: a multiclass weed species image dataset for deep learning, *Sci. Rep.* 9 (1) (2019) 2058.
- [15] Md Atiqur Rahman, Yang Wang, Optimizing intersection-over-union in deep neural networks for image segmentation, in: International Symposium on Visual Computing, Springer, 2016, pp. 234–244.
- [16] T. Sarvini, T. Sneha, G.S. Sukanya Gowthami, S. Sushmitha, et al., Performance comparison of weed detection algorithms, in: 2019 International Conference on Communication and Signal Processing (ICCSP), IEEE, 2019, pp. 843–847.
- [17] Søren Kelstrup Skovsen, Morten Stigaard Laursen, Rebekka Kjeldgaard Kristensen, Jim Rasmussen, Mads Dyrman, Jørgen Eriksen, René Gislum, Rasmus Nyholm Jørgensen, Henrik Karstoft, Robust species distribution mapping of crop mixtures using color images and convolutional neural networks, *Sensors* 21 (1) (2020) 175.
- [18] Lyndon N. Smith, Arlo Byrne, Mark F. Hansen, Wenhao Zhang, Melvyn L. Smith, Weed classification in grasslands using convolutional neural networks, in: Michael E. Zelinski, Tarek M. Taha, Jonathan Howe, Abdul A. Awwal, Khan M. Iftekharuddin (Eds.), Applications of Machine Learning, SPIE, San Diego, California, 2019, p. 42.
- [19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, Youngjoon Yoo, Cutmix: regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6023–6032.
- [20] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz, mixup: beyond empirical risk minimization, in: Proceedings of the Sixth International Conference on Learning Representations, 2018.
- [21] Xiaoguang Zhang, Nan Li, Luzhen Ge, Xuan Xia, Ning Ding, A unified model for real-time crop recognition and stem localization exploiting cross-task feature fusion, in: 2020 IEEE International Conference on Real-Time Computing and Robotics (RCAR), 2020, pp. 327–332.

Higher Degree Research Thesis by Publication
University of New England

STATEMENT OF AUTHORS' CONTRIBUTION

We, the Research Master/PhD candidate and the candidate's Principal Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated in the *Statement of Originality*.

	Author's Name (please print clearly)	% of contribution
Candidate	Jonathan Ford	33
Other Authors	Edmund Sadgrove	33
	David Paul	33

Name of Candidate: Jonathan Ford

Name/title of Principal Supervisor: Dr Edmund Sadgrove



Candidate

13th October 2025

Date



Principal Supervisor

14/10/25

Date

Higher Degree Research Thesis by Publication

University of New England

STATEMENT OF ORIGINALITY

We, the Research Master/PhD candidate and the candidate’s Principal Supervisor, certify that the following text, figures and diagrams are the candidate’s original work.

Type of work	Page number/s
Journal Paper	140-147

Name of Candidate:

Jonathan Ford

Name/title of Principal Supervisor:

Dr Edmund Sadgrove



13th October 2025

Candidate

Date



14/10/25

Principal Supervisor

Date

Chapter 6

Discussion and Conclusion

6.1 Introduction

In the preceding chapters, three studies have been presented, which have expanded the understanding of machine learning techniques for the segmentation of weeds in complex environments in Southeast Australian pastures. In these studies, several weeds of interest were selected, including serrated tussock (*Nassella trichotoma*), Scottish thistle (*Onopordum acanthium*), saffron thistle (*Carthamus lanatus*), variegated thistle (*Silybum marianum*), white horehound (*Marrubium vulgare*), Bathurst burr (*Xanthium spinosum*) and stinging nettle (*Urtica incisa*).

Compared to crop fields, pastures often present a more complex environment, characterized by multiple overlapping randomly arranged plants with similar colors and textures. This renders some techniques commonly used in crop fields, such as Hough transformations and Object-Based Image Analysis (OBIA) methods, less effective. OBIA techniques often rely on identifying connected plant components segmented from the background using simple thresholding, which may not perform well in such cluttered environments. While segmentation studies in pastures are rare [20, 36, 224], they offer the advantage of providing more precise weed localization. This allows herbicide spray nozzles to be activated with greater accuracy, potentially reducing herbicide use and minimizing collateral damage to the surrounding pasture.

The remainder of this chapter is structured as follows. First, a summary of the work undertaken is presented. Next, each of the research questions posed in Chapter 1 is revisited, considering the extent to which they have been answered and highlighting potential directions for future research. Finally, the chapter concludes by drawing together the main findings, reflecting on their broader implications, and outlining the overall contributions of this thesis.

6.2 Research Summary

In Chapter 3, a pipeline for the patch-based segmentation of images using an Extreme Learning Machine (ELM) as a convolutional filter across larger images was developed. This study focused primarily on Research Question 1, concerning the

use of shallow networks in weed segmentation studies in pastures, and to a lesser extent Research Question 3, concerning real-time applications of such technologies. Building on previous studies, this chapter demonstrates the potential of ELM-based segmentation for precise weed localization, critical for activating herbicide spray nozzles at appropriate times. Such precision has the potential to reduce herbicide usage, lower weed control costs, and minimize environmental and health impacts.

This pipeline was tested on four datasets containing four weed species commonly found in Southeast Australian pastures. Results showed good generalizability across different species. Although Research Question 5 was not addressed directly, this study demonstrated that the proposed models can be effective across multiple species, provided that only one species is present at each location.

A key innovation was the use of fine patch-based segmentation, which is in contrast to previous studies, which focused on classification [54, 209, 211] or bounding box detection [208, 210]. Since weeds are rarely rectangular and often overlap, bounding boxes may be of limited value in this context. By comparison, a sufficiently high-resolution heatmap, as used in this study, enables herbicide to be applied across all areas where weeds are present, without the need to delineate the boundaries of individual plants. Although the $16\times$ downsampled output could theoretically be upsampled to a dense segmentation map, it is unlikely that any practical SSWM treatment options in pastures require this level of precision¹.

This study compared the performance of a range of features, including LBP, HOG and colour features (defined in Section 2.2.3), being computed over a range of colour channels, vegetation indices, or in the case of the colour features, colour spaces. A key finding was that precise hyperparameter optimization, especially for LBP features, significantly improved the mean Intersection over Union (mIOU) by 4.8% to 18.4%. While HOG features showed less pronounced gains from optimization, selecting the optimal colour space for colour features could enhance results by up to 4.6%.

As mentioned in Chapter 1, for the purpose of this thesis, a real-time application is deemed to be anything that can be completed in 164 ms or less. For the ELM-based segmentation strategy, the majority of time taken in processing an image was the feature extraction step, in which LBP features took from 266 to 848 ms, HOG features took from 80 to 90 ms, and colour features took 367 ms. By comparison, the ELM network itself had an inference time of 1 to 15 ms, depending on input feature and number of hidden layer neurons. This demonstrated that the majority of time taken in processing an image was the feature extraction step, and of the features considered, only the HOG features were fast enough to present a viable solution for real-time applications.

This means that some additional work would be required in order to render the other techniques (LBP and colour features) suitable for real-time use. Below is a list

¹However, it could be argued that mechanical weed destruction methods, such as the use of stamping tools targeting the stems of small weed plants in crop fields, may require this level of precision.

of possible solutions to the issue of real-time applications:

1. **Reduce Image Resolution:** by reducing the resolution of images prior to feature extraction, the time taken can be reduced.
2. **Implement parallelization:** based on the CPU usage reported by the Linux `top` command, the Matlab `extractLBPFeatures` function, which was used in this study, does not support parallelization. The use of parallelization should reduce the time taken to extract features.
3. **Implement GPU support:** All Matlab feature extraction methods utilized in this study ran on the CPU. As with other image processing techniques, it is likely that significant time savings could be achieved through the use of a GPU implementation.

The simplest method for reducing inference time would be to reduce the image resolution, as this would reduce the dimensionality of the data required to be processed by the feature extraction method. It is possible that Matlab's parallel computing toolbox may also provide support for the use of parallel processing or GPU use for the techniques used in this study. Alternatively, a custom parallel implementation could be considered.

In summary, Chapter 3 demonstrated the effectiveness of shallow networks for weed detection (Research Question 1) across a range of species in Southeast Australia. While a multi-species detector (Research Question 5) was not developed, individual detectors achieved good results across a range of target species. However, in order to achieve real-time performance (Research Question 3), image resolution would need to be reduced, or development of parallel or GPU implementations considered.

In Chapter 4, a dual-task CNN was developed for plant and spraypoint detection, effectively addressing Research Question 2, concerning the use of CNN models in Southeast Australian pastures, and Research Question 4, concerning dual-task models for simultaneous plant segmentation and spraypoint detection. The dataset chosen for this study contained images of serrated tussock, captured at a single location over a period of nine months.

The plant task is responsible for generating a mask of the entire weed plant of interest, while the spraypoint task is responsible for detecting the centre of the plant. The spraypoint detection is particularly useful for herbicides that are absorbed primarily through the weed's root system, making it more efficient to target the center of the plant, where the roots are most concentrated. This model incorporated components from ConvNeXt, a recently developed architecture known for its superior performance in general datasets, thereby further addressing Research Question 2, concerning the use of modern architectural elements.

Unlike previous dual-task networks [150, 273] designed for joint plant and stem detection in early-stage crop fields, our model was relatively heavy, with approximately 2.6 million parameters, compared to the 59 to 153 thousand parameters in

existing models. Despite its size, the model maintained a fast inference time of 15.7 ms, comparable to the 14.8 to 21.5 ms range of previous models, making it well-suited for real-time image processing (Research Question 3), particularly in scenarios where the imaging vehicle is also responsible for spraying herbicide.

Additionally, our model required approximately half the number of FLOPs compared to previous models, with 4.9 GFLOPs versus 10.0 to 11.3 GFLOPs. Although some studies use a low parameter or FLOP count as a proxy for real-time processing efficiency, our findings demonstrated that these factors do not necessarily result in faster inference times. This suggests that future studies should prioritize direct measurements of inference time for more accurate performance evaluation.

Another notable finding of this study was that strong performance achieved when training and validation data were drawn from a single pooled dataset did not necessarily translate to good results on a separate test set, where images were captured under different growing conditions (Research Question 6). This led to a substantial drop in performance for pre-existing models, with the plant segmentation task showing an mIOU decrease of up to 0.421 and the spraypoint detection task experiencing an F1-score drop of up to 0.249. In contrast, the ConvNeXt-based model proved more resilient to changes in growing conditions, with only a 0.072 reduction in mIOU for the plant task and a minimal 0.001 decrease in F1-score for the spraypoint task, highlighting its greater robustness, when faced with unseen growing conditions.

To assess whether the performance of previous models could be improved when tested under unseen growing conditions, several normalization techniques were evaluated. The goal of normalization is to standardize the input data to ensure consistency across samples. One such technique developed in this study, HistMatch normalization, transforms each input image to follow a standard normal distribution with a mean of 0 and a standard deviation of 1. This contrasts with the naïve approach, defined by the formula $J = \frac{I - \mu_I}{\sigma_I}$, which also centers and scales the data but does not produce a standard normal distribution².

In the test study, HistMatch normalization led to improvements of up to 0.433 over default normalization, highlighting its potential value for future work. It was especially beneficial in scenarios where the model previously struggled, suggesting that variations in lighting and contrast between training and test sets were hindering model performance. By reducing these discrepancies, HistMatch normalization enabled the model to focus more effectively on the discriminative features necessary to distinguish between target and non-target plant species.

The study presented in Chapter 5 further tested the effectiveness and robustness of a dual plant segmentation and spraypoint detection model in the case where multiple weed species are present, across multiple locations (Research Question 5), as well as looking at advanced sample-mixing techniques of data augmentation (Research Question 7).

²Unless the data were already normally distributed.

In order to address Research Question 5, concerning the detection of multiple weed species, a dataset containing four common weeds from Southeast Australia, was constructed. Images were taken at three distinct locations, over a period of nineteen months, allowing for images with a range of different growing conditions, and background species present. Despite class imbalance, the models performed well across all species, maintaining high classification accuracy (≥ 0.90) and reliable generalization. This expands the potential applicability of the approach to more complex, mixed-species field conditions.

To address Research Question 6, which concerns the ability of models to generalize to field conditions not encountered during training, the dataset was divided into two groups based on the date and location of image collection. Two tests were conducted: in the first, one group was used for training and validation while the other was used for testing; in the second, the roles of the groups were reversed. Results showed that, particularly with CutMix augmentation, models generalized extremely well, achieving plant segmentation mIU values up to 0.894 and spraypoint F1-scores as high as 0.962.

Finally, the study explored the use of advanced data augmentation techniques not previously applied in weed localization research (Research Question 7). In particular, CutMix and MixUp, which are two sample-mixing augmentation strategies, were tested against more conventional spatial augmentations. CutMix proved to have the most consistent performance across tests, models, and tasks, especially in condition-invariance tests, where it improved generalization and reduced prediction errors. These results suggest that advanced augmentation techniques offer a promising avenue for further improving model reliability in uncontrolled field settings.

The studies presented in this thesis encompass both conventional and deep learning techniques and address a range of weed species commonly found in these settings. The model architectures were informed by recent advances, incorporating elements of ConvNeXt into designs inspired by earlier plant and stem detection models in the literature [150, 273].

These models demonstrated strong performance on the selected datasets, achieving excellent results even under previously unseen growing conditions. This is a significant outcome, as other models, despite performing well during validation, often struggled when faced with such variability. This highlights the importance of using a separate hold-out test set, consisting of images collected at a different date or location, when evaluating model performance, and suggests that caution should be applied when interpreting studies that do not incorporate this step.

Of equal importance is the finding that model parameter counts or FLOPs do not necessarily correlate with actual inference times. This indicates that models with fewer parameters or FLOPs should not be assumed to perform faster in real-time settings. Instead, empirical benchmarking of inference times is essential to substantiate claims regarding a model's suitability for real-time applications.

Preliminary testing of additional techniques, including spatial context, and attention, presented in Appendix A yielded disappointing results, with their inclusion having little to no impact or even degrading performance. While these approaches were largely outside the scope of this study, this initial analysis suggests limited promise in this context.

Taken as a whole, the research presented in this thesis demonstrates that model and feature choice are critical determinants of performance, particularly in the context of condition-invariance testing. It should be noted that even modest improvements in mIoU in the range of 1–5% correspond to more consistent pixel-level weed delineation, especially along object boundaries. In a precision spraying context, this translates to fewer missed weed regions and reduced over-spray into surrounding pasture, potentially lowering herbicide usage while improving targeting efficiency. Although such numerical gains may appear modest in isolation, small improvements in segmentation accuracy can have amplified downstream effects in automated spraying systems due to their cumulative impact across large spatial areas.

To further advance the use of AI models in Southeast Australian pastures, additional data collection is needed to increase the diversity of weed species and growing conditions represented in the training set. Future work should also assess the model's ability, particularly the spraypoint detector, to distinguish between similarly shaped, long-bladed grass species, which may include both invasive weeds and beneficial native grasses.

Accurate and rapid weed detection is a critical component of autonomous site-specific weed management (SSWM), which holds promise for treating weeds more efficiently while reducing herbicide use, lowering labor requirements, and enhancing productivity. The studies presented in this thesis provide a solid foundation for real-time weed segmentation and spraypoint detection in Australian pastoral environments, paving the way for future research in this field.

6.3 Can a shallow network be used to segment weeds in pastures in SE Australia?

Research Question 1 explores the use of shallow networks for segmenting weeds in Southeast Australian pastures. This was examined in Chapter 3, where a Single Hidden Layer Feed-Forward Network (SLFN) was trained using the ELM algorithm. A variety of weed species and feature types were considered, enabling a rigorous comparison across approaches. The results showed that while the effectiveness of different approaches varied across weed species, HOG features generally performed worse than both LBP and colour features, underscoring the importance of evaluating multiple feature types when addressing these kinds of problems.

This work established a strong foundation for applying shallow network approaches to segmentation tasks in Southeast Australian pastures. Looking ahead,

these findings open the door to further exploration of other research questions posed in this thesis. For example, investigating whether shallow networks can maintain reliable performance in multi-species contexts (Research Question 5), or demonstrating their adaptability when faced with field conditions not encountered during training (Research Question 6). Such future directions would not only test the robustness of shallow methods, but also highlight their potential for broader application in complex, real-world scenarios.

6.4 Can CNNs be used to segment weeds in pastures in SE Australia – focusing on modern architectures?

Research Question 2 examines the use of modern CNN architectures for weed segmentation in Southeast Australian pastures. This was addressed in Chapters 4 and 5, where a ConvNeXt-inspired architecture [142] was developed to generate models capable of segmenting target weed species in images collected across multiple locations and different collection dates.

The studies demonstrated that this model was highly effective at distinguishing between weed and non-weed species. It showed improved robustness to confounding factors, such as testing on growing conditions not seen during training, compared to pre-existing models, while maintaining comparable inference times despite having a larger number of parameters.

Moving forward, additional data collection is clearly needed, particularly for other long-blade grass species such as African lovegrass (*Eragrostis curvula*), Chilean needlegrass (*Nassella neesiana*), and native grasses (*Poa* and *Austrostipa* species, among others). This will help determine whether the developed models can reliably distinguish between different long-bladed grass species.

6.5 Can segmentation be done in real time?

Research Question 3 concerns the use of models in a real-time context, which was defined in Chapter 1 as 164 ms or below. This was an important consideration for both the shallow networks and the CNNs considered in this work. For the shallow ELM network presented in Chapter 3, the networks considered exhibited very fast inference times (1-15 ms). Feature extraction, however, proved to be the main bottleneck.

The fastest feature was HOG (80–90 ms), comfortably below the real-time threshold; however, it was also the worst performing feature given the experimental setup used and dataset. LBP (266–848 ms) and colour features (367 ms) were shown to be unsuitable for real-time performance under the same conditions. Therefore, additional work would be needed to achieve a real-time segmentation network using these features. The simplest approach would be to reduce image resolution, though

testing is required to ensure this does not negatively impact accuracy. Other potential strategies include implementing parallelization or utilizing GPUs to speed up feature extraction.

For the CNNs evaluated in Chapters 4 and 5, the answer is more straightforward. The reduced image resolution, and GPU implementation through TensorFlow used in these studies enabled inference times ranging from 14.8 to 21.5 ms, well below the 164 ms real-time threshold. This demonstrates that these models can reliably achieve real-time segmentation, comprehensively addressing the research question.

In addition to exploring GPU-based approaches for manual feature extraction, future research could evaluate the performance of the tested models on low-power edge devices, with particular attention to whether they remain capable of real-time detection in the absence of modern high-performance GPUs. Conversely, further investigation into more computationally demanding models—provided they still meet the 164 ms real-time threshold—could reveal whether increasing parameter counts yields improvements beyond the results presented in this thesis.

6.6 Can we produce a combined plant/spraypoint detector?

Research Question 4 addresses the development of joint plant segmentation and spraypoint detection models, which share a single backbone or encoder but employ two separate decoders—one for each task. While similar architectures have been applied to plant and stem detection in early-stage crop fields, the use of dual-task networks for this specific application had not previously been considered. This approach is valuable because it allows the model to learn shared feature representations for both tasks, potentially improving efficiency and performance while reducing computational cost compared to training separate networks for each task.

The studies presented in Chapters 4 and 5 examined the use of a joint network in both single-species and multi-species contexts, drawing on a diverse dataset spanning multiple collection dates and locations. These findings indicate that, although research on such models is still in its early stages, there is a strong foundation for applying dual or multi-task networks in agriculture, particularly within pasture systems.

As outlined earlier in this chapter, future research should extend the models to encompass multiple long-bladed grass species, in order to assess the ability of these networks to discriminate between morphologically similar species. Furthermore, subsequent work could broaden the scope of tasks addressed by the model to include biomass estimation or nutrient assessment of the photographed pasture.

6.7 Can we produce a multi-species detector?

Research Question 5 concerns the testing of models trained to detect multiple weed species simultaneously, reflecting the conditions expected in real-world environments. While the work in Chapter 3 considered several different weed species, each model was trained and evaluated on only a single species or a defined group of species. A true multi-species detector was not investigated until Chapter 5, which examined the use of CNNs to develop a joint plant segmentation and spray-point detection model.

The findings from Chapter 5 were highly encouraging. Despite challenges such as class imbalance within the dataset, the developed models achieved strong performance across all included weed species, with classification accuracy of at least 0.90, while also demonstrating reliable generalization capabilities. These outcomes significantly broaden the potential applicability of the approach to complex, real-world field conditions, where multiple weed species commonly occur simultaneously.

That said, there remains a clear need to broaden the range of weeds represented in the training data, thereby enhancing the networks' ability to discriminate between species. Achieving this will require substantial additional data collection across diverse locations and time points. Such efforts would support the development of models applicable to a wider variety of weed species. Alternatively, given the short inference time of the current models, a different strategy might involve training separate models for individual weed species and then deploying only the models relevant to a given field, running them either in parallel or in sequence, while maintaining a suitable inference time for real-time applications.

6.8 How well do segmentation models generalize to fields or conditions that they have not been trained on?

Research Question 6 examines whether models can maintain strong performance when applied to data collected under conditions that differ from those used in training, such as variations in location or growing stage. This question was addressed in both Chapters 4 and 5.

The findings revealed that high accuracy on a test set captured concurrently with the training set did not necessarily translate into strong performance on datasets obtained at different times or from different locations. Factors such as normalization methods and augmentation strategies were shown to have a substantial influence on model robustness and generalization, highlighting their importance in developing models suitable for real-world deployment.

It should be noted that this study focused on differences in growing conditions and, as such, did not investigate model generalization to other forms of domain shift, such as changes in illumination or phenological growth stages. Consequently,

further research is required to evaluate the adaptability of the proposed models to these and other domain shifts.

Although a newly developed normalization method, HistMatch normalization, was shown to improve performance in some cases, these gains were typically observed when models were struggling significantly with unseen data. In situations where models already demonstrated sufficient robustness, HistMatch normalization often led to a slight reduction in accuracy. This suggests that its effectiveness may be limited to specific scenarios. Nonetheless, the development of new normalization strategies remains a promising direction for future research aimed at optimizing model performance.

6.9 Can advanced data augmentation techniques improve results?

Research Question 7 addresses the question of whether advanced data augmentation techniques can improve the results of weed detection models. This question was considered in Chapter 5, in which sample mixing augmentation strategies such as CutMix were considered.

The results showed that augmentation strategy is a critical factor in model performance, particularly when models are tested under new growing conditions or at unseen locations. Traditional augmentation methods, such as simple Zoom/Rotate, were found to be inadequate, leading to marked performance drops (mIU scores as low as 0.468 for plant segmentation and F1-scores as low as 0.160 for spraypoint detection). By comparison, CutMix, particularly with its tailored free-form mask generation, proved to be the most consistent and effective augmentation strategy, across all models and normalization strategies used.

These findings highlight the importance of selecting robust augmentation techniques in weed detection research. In particular, they suggest that future studies involving diverse datasets should consider the use of sample-mixing approaches such as CutMix during training to improve generalization and reliability.

6.10 Broader Implications

The studies highlighted several issues that may be of relevance to the broader computer vision community, either in agriculture or in other fields of computer vision in unstructured, complex environments. Specifically, there is nothing inherent to the field of weed localization which appears to explain why these issues should only be confined to studies on weed localization.

First, while low parameter counts or FLOPs are often used as proxies for fast inference, the findings indicate that neither metric is consistently reliable. Specifically, models with substantially more parameters can sometimes exhibit faster inference

times than smaller models, and models with similar FLOP counts may still differ in execution time. Consequently, future studies should measure inference time directly when evaluating a model's capacity for real-time performance.

Second, a model may perform well on a validation set drawn from the same data as its training set, yet exhibit poor performance when exposed to different growing conditions or locations. In such cases, factors such as augmentation strategy or normalization technique can significantly influence the model's ability to generalize. Therefore, results from studies that do not include a hold-out set collected from different times or locations should be interpreted with caution.

While it is unclear whether this issue affects all models, those pretrained on large datasets such as ImageNet, having already learned general and robust convolutional filters, may be less prone to overfitting to a specific dataset. Similarly, models pretrained on data collected under diverse growing conditions may adapt more effectively to new environments during testing. However, these hypotheses requires validation through future research.

Finally, building on the previous discussion, the choice of normalization technique and augmentation strategy can substantially influence a model's generalization ability. In Chapter 5, a free-form mask was used for CutMix augmentation, resulting in irregular regions of each image being incorporated into the augmented image. This approach was chosen to better emulate the irregular shapes of plants observed in the testing datasets. Consequently, other computer vision applications in which regions of interest exhibit similarly irregular shapes may also benefit from this, or a comparable, mask generation process.

More generally, in order to develop models that perform robustly across diverse real-world scenarios—where factors such as lighting or the presence of unfamiliar plant species may reduce accuracy—it is recommended to explore a range of normalization and augmentation approaches. Further research is also needed to understand why certain normalization techniques outperform others in specific contexts.

6.11 Conclusion

The studies presented in this thesis have investigated both shallow and deep neural networks for the segmentation of weeds in pastures across Southeast Australia, providing a strong foundation for future research in this area. Across the three core studies, a variety of network architectures were developed and evaluated on multiple weed species, addressing both single-species and multi-species scenarios.

The shallow networks examined in this research demonstrated promising performance across a range of weed species. It was observed that both the selection of features and the choice of hyperparameters were critical to achieving high model performance, with local binary patterns (LBP) and color features proving particularly effective. Additionally, the choice of color space or vegetation index used for feature extraction was found to significantly affect results. Future work could extend

these studies by developing multi-species models and evaluating their generalization across different locations and growing conditions.

The deep networks, implemented as convolutional neural networks (CNNs), were subjected to more rigorous evaluation, including the development of multi-species detectors and assessment of generalization to new growing conditions and locations. The models demonstrated high effectiveness in both single-species and multi-species detection, achieving accuracies above 0.90 for each species in the multi-species dataset despite substantial class imbalance. In addition, the models achieved inference speeds more than seven times faster than the threshold set for real-time applications, enabling their practical deployment in mobile settings such as AGV-based weed treatment systems.

The CNNs evaluated in this study were also designed with two outputs, enabling them to detect both the entire plant and its centre. This design reflects the fact that the choice of herbicide influences whether it should be applied to the whole plant for foliar uptake or targeted at the centre of the plant for uptake via the roots. The promising performance of these dual-task models suggests that CNNs could be well suited to weed management systems requiring multiple herbicide application strategies.

Several of the issues identified in this work, including the ability of models to generalize to new datasets with differences in lighting, colour, or plant species, as well as the effects of normalization techniques and augmentation strategies, are also likely to be relevant to other computer vision applications involving complex, unstructured environments. This work demonstrated that these considerations can have a substantial impact on model performance. Consequently, future computer vision research should carefully evaluate the training techniques incorporated into model development and, where possible, assess generalization using a hold-out test dataset collected at a different time, location, or under different environmental conditions.

Taken together, the findings of this thesis demonstrate both the potential and the challenges of applying neural networks to pasture-based weed segmentation problems, with broader implications for computer vision in complex, unstructured environments. By addressing issues of model design, training optimization, generalization, and real-time performance, this work establishes a foundation upon which future research can build toward the development of reliable systems capable of deployment in real agricultural settings, thereby supporting more sustainable and efficient farming practices.

Appendix A

Exploring Context and Attention in CNN Model Design

A.1 Introduction

During the course of the research outlined in Chapter 5 on the use of dual-task plant segmentation and spraypoint detection CNNs for use in a multi-species environment, a range of model modifications were considered for the base truncated ConvNeXt model (tCN), which incorporated the use of either spatial context, or attention mechanisms. Although these modifications did not improve results, and for the plant segmentation task actually made them worse, the results are provided here for the reader's consideration.

A.2 Methodology

Spatial context, introduced in Section 2.2.7, refers to methods that incorporate information from a broader spatial area—often the entire image—into the feature map at each location. This enables the model to consider global context when determining the class at each point in the image. For this study, two spatial context modules were selected: the Pyramid Pooling Module (PPM) [274] and the Atrous Spatial Pyramid Pooling (ASPP) module [26].

Although both modules inject spatial context into local feature representations, they do so through different mechanisms. The PPM aggregates features at multiple spatial scales using pooling operations, followed by convolutional layers. The resulting feature maps are upsampled and concatenated with the original features. In contrast, ASPP employs dilated convolutions with varying dilation rates to capture multi-scale context directly. Further details on these modules are provided in Section 2.2.7.

By comparison, attention mechanisms, introduced in Section 2.2.6, are designed to guide the model's focus, by identifying which information is important and should be emphasized, and which should be ignored or suppressed [255]. In this study, the Convolutional Block Attention Module (CBAM) [255] was selected to explore the

role of attention. This module combines both channel and spatial attention mechanisms to identify which features, and which spatial regions of the feature map, should be emphasized or suppressed.

Figure A.1 illustrates the insertion points of the various modules within the base tCN model. Although all modules are shown within a single diagram, it is important to emphasize that only one modification was applied per variant tested in this study. As shown in Figure A.1(b, top), the PPM and ASPP modules were inserted at the end of the backbone, prior to passing the feature maps to the decoder arms. In contrast, CBAM modules were evaluated in two distinct locations: within the residual blocks of the backbone (Figure A.1(a); note that the backbone is labeled B1/B2/B3 in (b)), and between the two segmentation decoder arms, before upsampling the feature map from SegDecA (Figure A.1(b, top-right)). These configurations are referred to as E-CBAM (encoder-CBAM) and S-CBAM (segmentation-CBAM), respectively. Additionally, two reduction ratios — 4 and 16, as defined in [255] — were tested, resulting in four CBAM variants: E-CBAM(4), E-CBAM(16), S-CBAM(4), and S-CBAM(16).

The pooled dataset, Zoom/Rotate augmentation, and standard normal normalization, as defined in the methods section of Chapter 5, were used. Other training parameters, and metrics were identical to those used in the same chapter, except that a fixed learning rate of 0.01 was used. Statistical analysis was performed using a one-way independent-measures ANOVA, followed by Tukey’s Honest Significant Difference (HSD) post hoc test to identify which variants differed significantly from one another.

A.3 Results

Table A.1 provides the results of this experiment, with the mean Intersection over Union (mIU) metric for the plant segmentation task shown on the left, and the F1-score metric for the spraypoint detection task shown on the right. As can be seen, the base model (without any modifications) performed best for the plant segmentation task, with a mIU of 0.861 ± 0.002 , while the E-CBAM(16) variant performed best for the spraypoint detection task, with an F1-score of 0.902 ± 0.017 .

TABLE A.1: Performance of model variants on the pooled dataset.

Model	Plant (mIU)	Spraypoint (F1)
Base	0.861 ± 0.002	0.885 ± 0.008
ASPP	0.857 ± 0.002	0.881 ± 0.013
PPM	0.857 ± 0.004	0.879 ± 0.008
E-CBAM(4)	0.850 ± 0.004	0.892 ± 0.015
E-CBAM(16)	0.848 ± 0.004	0.902 ± 0.017
S-CBAM(4)	0.857 ± 0.003	0.894 ± 0.012
S-CBAM(16)	0.858 ± 0.002	0.885 ± 0.006

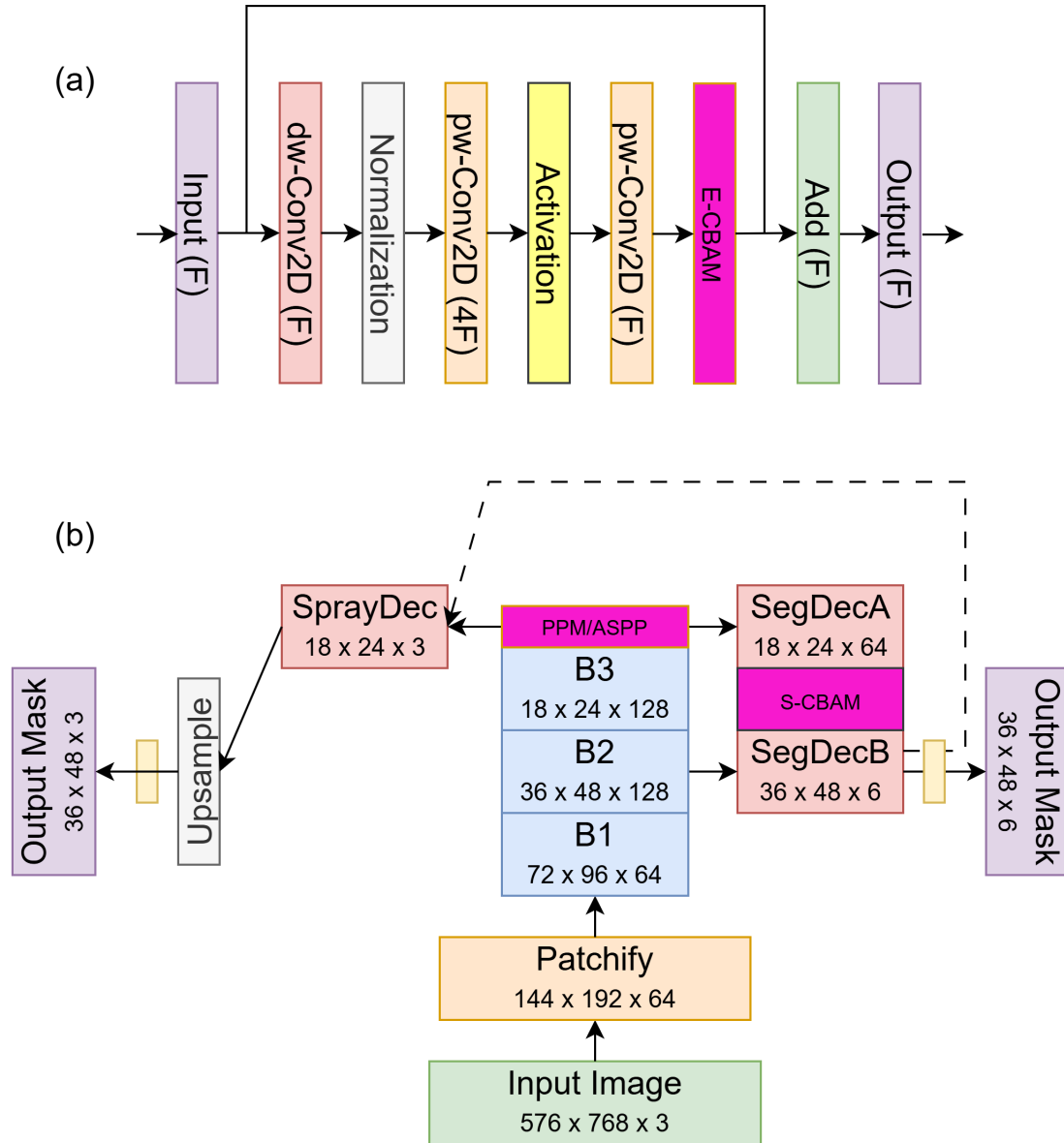


FIGURE A.1: The location within the truncated ConvNeXt (tCN) model at which different modules were inserted. Section (a) shows the insertion of a CBAM module within a residual block, while section (b) shows the insertion of the PPM, ASPP or CBAM modules within the macrostructure of the network. Although all modules are shown in the one diagram, only a single module was included in each of the model variants.

A one-way independent measures ANOVA revealed a significant effect of model type for both the plant segmentation task ($p < 0.0001$) and the spraypoint task ($p = 0.012$). Tukey's Honest Significant Difference (HSD) post hoc test showed that all model variants performed significantly worse than the base model for plant segmentation. However, none of the variants differed significantly from the base model for the spraypoint task.

A.4 Discussion

The use of spatial context in weed localization has been explored using both PSPNet [8, 110, 182, 259] and the DeepLab family of models [84, 114, 216]. However, results have been mixed, with some studies reporting that models that incorporate spatial context outperform others, while others finding them to perform worse.

The Convolutional Block Attention Module (CBAM) has seen limited use in weed localization studies. However, one study [140] reported that incorporating CBAM into an EfficientNet-B2 model [239]—particularly after each residual block, as in our E-CBAM configuration—led to significantly improved feature extraction. This benefit was lost in an ablation study in which the attention modules were removed.

This study investigated the incorporation of spatial context and attention mechanisms as variants of the tCN model used in Chapter 5, to assess whether they could improve performance over the base model. Under the experimental setup used, these modifications did not significantly affect performance on the spraypoint detection task and led to reduced performance on the plant segmentation task. As a result, these variants were not pursued in further experiments.

Bibliography

- [1] Alwaseela Abdalla et al. "Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure". In: *Computers and electronics in agriculture* 167 (2019), p. 105091.
- [2] Radhakrishna Achanta et al. "SLIC superpixels compared to state-of-the-art superpixel methods". In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2274–2282.
- [3] Shyam Prasad Adhikari, Heechan Yang, and Hyongsuk Kim. "Learning semantic graphics using convolutional encoder–decoder network for autonomous weeding in paddy". In: *Frontiers in plant science* 10 (2019), p. 1404.
- [4] "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields". In: *The International Journal of Robotics Research* 36.10 (Sept. 2017), pp. 1045–1052. ISSN: 0278-3649. DOI: [10.1177/0278364917720510](https://doi.org/10.1177/0278364917720510). URL: <http://journals.sagepub.com/doi/10.1177/0278364917720510>.
- [5] Mansoor Alam et al. "Real-Time Machine-Learning Based Crop/Weed Detection and Classification for Variable-Rate Spraying in Precision Agriculture". In: *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*. IEEE, Apr. 2020, pp. 273–280. ISBN: 978-1-7281-6788-6. DOI: [10.1109/ICEEE49618.2020.9102505](https://doi.org/10.1109/ICEEE49618.2020.9102505). URL: <https://ieeexplore.ieee.org/document/9102505/>.
- [6] Abeer M. Almalky and Khaled R. Ahmed. "Deep Learning for Detecting and Classifying the Growth Stages of *Consolida regalis* Weeds on Fields". In: *Agronomy* 13.3 (2023). ISSN: 2073-4395. DOI: [10.3390/agronomy13030934](https://doi.org/10.3390/agronomy13030934). URL: <https://www.mdpi.com/2073-4395/13/3/934>.
- [7] Córdova-Cruzatty Andrea, Barreno Barreno Mauricio Daniel, and Jácome Barrionuevo José Misael. "Precise weed and maize classification through convolutional neuronal networks". In: *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*. IEEE. 2017, pp. 1–6.
- [8] Muhammad Hamza Asad, Saeed Anwar, and Abdul Bais. "Improved Crop and Weed Detection with Diverse Data Ensemble Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2024, pp. 5336–5345.

- [9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.
- [10] M Bah, Adel Hafiane, and Raphael Canals. "Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images". In: *Remote Sensing* 10.11 (Oct. 2018), p. 1690. ISSN: 2072-4292. DOI: [10.3390/rs10111690](https://doi.org/10.3390/rs10111690). URL: <http://www.mdpi.com/2072-4292/10/11/1690>.
- [11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).
- [12] Adel Bakhshipour and Abdolabbas Jafari. "Evaluation of support vector machine and artificial neural networks in weed detection using shape features". In: *Computers and Electronics in Agriculture* 145 (Feb. 2018), pp. 153–160. ISSN: 01681699. DOI: [10.1016/j.compag.2017.12.032](https://doi.org/10.1016/j.compag.2017.12.032). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169917301333>.
- [13] A Shamim Banu and S Deivalakshmi. "AWUNet: leaf area segmentation based on attention gate and wavelet pooling mechanism". In: *Signal, Image and Video Processing* 17.5 (2023), pp. 1915–1924.
- [14] Amir Beck and Marc Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (Jan. 2009), pp. 183–202. ISSN: 1936-4954. DOI: [10.1137/080716542](https://doi.org/10.1137/080716542). URL: <http://epubs.siam.org/doi/10.1137/080716542>.
- [15] Mohammad Mahdi Bejani and Mehdi Ghatee. "A systematic review on overfitting control in shallow and deep neural networks". In: *Artificial Intelligence Review* 54.8 (2021), pp. 6391–6438.
- [16] Asher Bender, Brett Whelan, and Salah Sukkarieh. "A high-resolution, multi-modal data set for agricultural robotics: A Ladybird's-eye view of Brassica". In: *Journal of Field Robotics* 37.1 (2020), pp. 73–96.
- [17] A. Binch and C.W. Fox. "Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland". In: *Computers and Electronics in Agriculture* 140 (Aug. 2017), pp. 123–138. ISSN: 01681699. DOI: [10.1016/j.compag.2017.05.018](https://doi.org/10.1016/j.compag.2017.05.018). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169917300352>.
- [18] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *arXiv preprint arXiv:2004.10934* (2020).
- [19] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 1573-0565. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324). URL: <https://doi.org/10.1023/A:1010933404324>.

- [20] Igor L. Bretas et al. "Detection and mapping of *Amaranthus spinosus* L. in bermudagrass pastures using drone imagery and deep learning for a site-specific weed management". In: *Agronomy Journal* 116.3 (2024), pp. 990–1002. DOI: <https://doi.org/10.1002/agj2.21545>. eprint: <https://access.onlinelibrary.wiley.com/doi/pdf/10.1002/agj2.21545>. URL: <https://access.onlinelibrary.wiley.com/doi/abs/10.1002/agj2.21545>.
- [21] Anderson Brilhador et al. "Classification of weeds and crops at the pixel-level using convolutional neural networks and data augmentation". In: *2019 IEEE latin american conference on computational intelligence (LA-CCI)*. IEEE. 2019, pp. 1–6.
- [22] Brendan Calvert et al. "Autoweed: detecting *Harrisia cactus* in the Goondiwindi region for selective spot-spraying". In: *Proceedings of the 1st Queensland Pest Animal and Weed Symposium*. Weed Society of Queensland Pty. Ltd. 2019, p. 52.
- [23] Brendan Calvert et al. "Robotic spot spraying of *Harrisia cactus* (*Harrisia martinii*) in grazing pastures of the Australian rangelands". In: *Plants* 10.10 (2021), p. 2054.
- [24] Zhao Chang-Tao et al. "Automatic lettuce weed detection and classification based on optimized convolutional neural networks for robotic weed control". In: *Agronomy* 14.12 (2024), p. 2838.
- [25] Łukasz Chechliński, Barbara Siemiątkowska, and Michał Majewski. "A system for weeds and crops identification—reaching over 10 fps on raspberry pi with the usage of mobilenets, densenet and custom modifications". In: *Sensors* 19.17 (2019), p. 3787.
- [26] Liang-Chieh Chen et al. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2017. arXiv: 1606.00915 [cs.CV]. URL: <https://arxiv.org/abs/1606.00915>.
- [27] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: [10.1109/TPAMI.2017.2699184](https://doi.org/10.1109/TPAMI.2017.2699184).
- [28] Liang-Chieh Chen et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.
- [29] Liang-Chieh Chen et al. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017. arXiv: 1706.05587 [cs.CV]. URL: <https://arxiv.org/abs/1706.05587>.

- [30] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, 785–794. ISBN: 9781450342322. DOI: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785). URL: <https://doi.org/10.1145/2939672.2939785>.
- [31] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: (Oct. 2016). arXiv: [1610.02357](https://arxiv.org/abs/1610.02357). URL: <http://arxiv.org/abs/1610.02357>.
- [32] Adam Coates, Andrew Ng, and Honglak Lee. "An Analysis of Single-Layer Networks in Unsupervised Feature Learning". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 215–223. URL: <https://proceedings.mlr.press/v15/coates11a.html>.
- [33] Adam Coates and Andrew Y. Ng. "Learning Feature Representations with K-Means". In: 2012, pp. 561–580. DOI: [10.1007/978-3-642-35289-8_30](https://doi.org/10.1007/978-3-642-35289-8_30). URL: http://link.springer.com/10.1007/978-3-642-35289-8_30.
- [34] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297. ISSN: 0885-6125. DOI: [10.1007/BF00994018](https://doi.org/10.1007/BF00994018). URL: <http://link.springer.com/10.1007/BF00994018>.
- [35] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: [10.1007/BF00994018](https://doi.org/10.1007/BF00994018). URL: <http://link.springer.com/10.1007/BF00994018>.
- [36] Benjamin Costello et al. "Detection of parthenium weed (*Parthenium hysterophorus* L.) and its growth stages using artificial intelligence". In: *Agriculture* 12.11 (2022), p. 1838.
- [37] D. M. Woebbecke et al. "Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions". In: *Transactions of the ASAE* 38.1 (1995), pp. 259–269. ISSN: 2151-0059. DOI: [10.13031/2013.27838](https://doi.org/10.13031/2013.27838). URL: <http://elibrary.asabe.org/abstract.asp??JID=3&AID=27838&CID=t1995&v=38&i=1&T=1>.
- [38] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE, pp. 886–893. ISBN: 0-7695-2372-2. DOI: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177). URL: <http://ieeexplore.ieee.org/document/1467360/>.

- [39] Schori Damian, Anken Thomas, and Seatovic Dejan. "Using Fully Convolutional Networks for Rumex Obtusifolius Segmentation, a Preliminary Report". In: *2019 International Symposium ELMAR*. IEEE, Sept. 2019, pp. 119–122. ISBN: 978-1-7281-2181-9. DOI: [10.1109/ELMAR.2019.8918914](https://doi.org/10.1109/ELMAR.2019.8918914). URL: <https://ieeexplore.ieee.org/document/8918914/>.
- [40] Fengying Dang et al. "YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems". In: *Computers and Electronics in Agriculture* 205 (2023), p. 107655. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2023.107655>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169923000431>.
- [41] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Tamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). URL: <https://aclanthology.org/N19-1423/>.
- [42] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).
- [43] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. "Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm". In: *Procedia Computer Science* 54 (2015). ISSN: 18770509. DOI: [10.1016/j.procs.2015.06.090](https://doi.org/10.1016/j.procs.2015.06.090).
- [44] Foivos I Diakogiannis et al. "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (2020), pp. 94–114.
- [45] Linhao Dong, Shuang Xu, and Bo Xu. "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 5884–5888. DOI: [10.1109/ICASSP.2018.8462506](https://doi.org/10.1109/ICASSP.2018.8462506).
- [46] Alessandro dos Santos Ferreira et al. "Weed detection in soybean crops using ConvNets". In: *Computers and Electronics in Agriculture* 143 (Dec. 2017), pp. 314–324. ISSN: 01681699. DOI: [10.1016/j.compag.2017.10.027](https://doi.org/10.1016/j.compag.2017.10.027). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169917301977>.
- [47] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: [2010.11929](https://arxiv.org/abs/2010.11929) [cs.CV]. URL: <https://arxiv.org/abs/2010.11929>.

- [48] Mads Dyrmann, Henrik Karstoft, and Henrik Skov Midtiby. "Plant species classification using deep convolutional neural network". In: *Biosystems engineering* 151 (2016), pp. 72–80.
- [49] David J Eldridge et al. "Australian dryland soils are acidic and nutrient-depleted, and have unique microbial communities compared with other drylands". In: *Journal of Biogeography* 45.12 (2018), pp. 2803–2814.
- [50] Borja Espejo-Garcia et al. "Towards weeds identification assistance through transfer learning". In: *Computers and Electronics in Agriculture* 171 (2020), p. 105306.
- [51] Andre Esteva et al. "Deep learning-enabled medical computer vision". In: *NPJ digital medicine* 4.1 (2021), p. 5.
- [52] Mark Everingham et al. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88 (2010), pp. 303–338.
- [53] Adnan Farooq, Jiankun Hu, and Xiuping Jia. "Weed Classification in Hyperspectral Remote Sensing Images Via Deep Convolutional Neural Network". In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. 2018, pp. 3816–3819. DOI: [10.1109/IGARSS.2018.8518541](https://doi.org/10.1109/IGARSS.2018.8518541).
- [54] Adnan Farooq, Xiuping Jia, and Jun Zhou. "Texture and shape features for grass weed classification using hyperspectral remote sensing images". In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 7208–7211.
- [55] Adnan Farooq et al. "Multi-resolution weed classification via convolutional neural network and superpixel based local binary pattern using remote sensing images". In: *Remote Sensing* 11.14 (2019), p. 1692.
- [56] H. Fathipoor, R. Shah-Hosseini, and H. Arefi. "CROP AND WEED SEGMENTATION ON GROUND-BASED IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORK". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-4/W1-2022 (2023), pp. 195–200. DOI: [10.5194/isprs-annals-X-4-W1-2022-195-2023](https://doi.org/10.5194/isprs-annals-X-4-W1-2022-195-2023). URL: <https://isprs-annals.copernicus.org/articles/X-4-W1-2022/195/2023/>.
- [57] Hafiza Sundus Fatima et al. "Formation of a Lightweight, Deep Learning-Based Weed Detection System for a Commercial Autonomous Laser Weeding Robot". In: *Applied Sciences* 13.6 (2023). ISSN: 2076-3417. DOI: [10.3390/app13063997](https://doi.org/10.3390/app13063997). URL: <https://www.mdpi.com/2076-3417/13/6/3997>.
- [58] Mulham Fawakherji et al. "Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation". In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2019, pp. 146–152.

- [59] Jonathan Ford, Edmund Sadgrove, and David Paul. "Developing an extreme learning machine based approach to weed segmentation in pastures". In: *Smart Agricultural Technology* 5 (2023), p. 100288. ISSN: 2772-3755. DOI: <https://doi.org/10.1016/j.atech.2023.100288>. URL: <https://www.sciencedirect.com/science/article/pii/S277237552300117X>.
- [60] Jonathan Ford, Edmund Sadgrove, and David Paul. "Dual-Task Multi-Species Network Design and Training Using Advanced Augmentation Techniques". In: *Available at SSRN 5382154* (2025).
- [61] Jonathan Ford, Edmund Sadgrove, and David Paul. "Joint plant-spraypoint detector with ConvNeXt modules and HistMatch normalization". In: *Precision Agriculture* 26.1 (2025), p. 24.
- [62] Yoav Freund and Robert E. Schapire. "Experiments with a New Boosting Algorithm". In: *ICML*. 1996, pp. 148–156.
- [63] Gaitang Wang and Ping Li. "Dynamic Adaboost ensemble extreme learning machine". In: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. IEEE, Aug. 2010, pp. V3–54–V3–58. ISBN: 978-1-4244-6539-2. DOI: [10.1109/ICACTE.2010.5579726](https://doi.org/10.1109/ICACTE.2010.5579726). URL: <http://ieeexplore.ieee.org/document/5579726/>.
- [64] Ignazio Gallo et al. "Deep object detection of crop weeds: Performance of YOLOv7 on a real case dataset from UAV images". In: *Remote Sensing* 15.2 (2023), p. 539.
- [65] Kavita Gandhi et al. "Exposure risk and environmental impacts of glyphosate: Highlights on the toxicity of herbicide co-formulants". In: *Environmental Challenges* 4 (2021), p. 100149. ISSN: 2667-0100. DOI: <https://doi.org/10.1016/j.envc.2021.100149>. URL: <https://www.sciencedirect.com/science/article/pii/S2667010021001281>.
- [66] Junfeng Gao et al. "Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields". In: *Plant Methods* 16.1 (2020), p. 29. ISSN: 1746-4811. DOI: [10.1186/s13007-020-00570-z](https://doi.org/10.1186/s13007-020-00570-z). URL: <https://doi.org/10.1186/s13007-020-00570-z>.
- [67] Junfeng Gao et al. "Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery". In: *Biosystems engineering* 170 (2018), pp. 39–50.
- [68] DL Garden et al. "Effects of grazing management on botanical composition of native grass-based pastures in temperate south-east Australia". In: *Australian Journal of Experimental Agriculture* 40.2 (2000), pp. 225–245.
- [69] Nikita Genze et al. "Manually annotated and curated Dataset of diverse Weed Species in Maize and Sorghum for Computer Vision". In: *Scientific Data* 11.1 (2024), p. 109.

- [70] Ross Girshick. "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.
- [71] Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.
- [72] Thomas Mosgaard Giselsson et al. "A public image database for benchmark of plant seedling classification algorithms". In: *arXiv preprint arXiv:1711.05458* (2017).
- [73] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.
- [74] Russell J Gordon et al. "A review on Pimelea poisoning of livestock". In: *Toxicicon* 186 (2020), pp. 46–57.
- [75] LE Goward et al. "Prospects for diversifying temperate pasture systems in south-eastern Australia with serradella (*Ornithopus* spp.)" In: *Crop & Pasture Science* 76.10 (2025), CP25065.
- [76] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks". In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. IEEE, 2004. ISBN: 0-7803-8359-1. DOI: [10.1109/IJCNN.2004.1380068](https://doi.org/10.1109/IJCNN.2004.1380068).
- [77] Sanjay Kumar Gupta et al. "Multiclass weed identification using semantic segmentation: An automated approach for precision agriculture". In: *Ecological Informatics* 78 (2023), p. 102366.
- [78] Ronja Guldenring et al. "Few-leaf Learning: Weed Segmentation in Grasslands". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 3248–3254. DOI: [10.1109/IROS51168.2021.9636770](https://doi.org/10.1109/IROS51168.2021.9636770).
- [79] Esmael Hamuda et al. "Automatic crop detection under field conditions using the HSV colour space and morphological operations". In: *Computers and electronics in agriculture* 133 (2017), pp. 97–107.
- [80] Sarah M Hamylton et al. "Evaluating techniques for mapping island vegetation from unmanned aerial vehicle (UAV) images: Pixel classification, visual interpretation and machine learning approaches". In: *International Journal of Applied Earth Observation and Geoinformation* 89 (2020), p. 102085.
- [81] Robert M. Haralick, K. Shanmugam, and Its' Hak Dinstein. "Textural Features for Image Classification". In: *IEEE Transactions on Systems, Man, and Cybernetics SMC-3.6* (Nov. 1973), pp. 610–621. ISSN: 0018-9472. DOI: [10.1109/TSMC.1973.4309314](https://doi.org/10.1109/TSMC.1973.4309314). URL: <http://ieeexplore.ieee.org/document/4309314/>.

- [82] ASM Mahmudul Hasan et al. "A survey of deep learning techniques for weed detection from images". In: *Computers and Electronics in Agriculture* 184 (2021), p. 106067.
- [83] ASM Mahmudul Hasan et al. "Weed recognition using deep learning techniques on class-imbalanced imagery". In: *Crop and Pasture Science* (2022).
- [84] Leila Hashemi-Beni et al. "Deep convolutional neural networks for weeds and crops discrimination from UAS imagery". In: *Frontiers in Remote Sensing* 3 (2022), p. 755939.
- [85] Sebastian Haug and Jörn Ostermann. "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks". In: *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part IV* 13. Springer. 2015, pp. 105–116.
- [86] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: (Dec. 2015). arXiv: [1512.03385](https://arxiv.org/abs/1512.03385). URL: <http://arxiv.org/abs/1512.03385>.
- [87] Kaiming He et al. "Identity mappings in deep residual networks". In: *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV* 14. Springer. 2016, pp. 630–645.
- [88] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.
- [89] Dan Hendrycks and Kevin Gimpel. "Gaussian Error Linear Units (GELUs)". In: (2016), pp. 1–10. arXiv: [1606.08415](https://arxiv.org/abs/1606.08415). URL: <http://arxiv.org/abs/1606.08415>.
- [90] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.
- [91] Qibin Hou, Daquan Zhou, and Jiashi Feng. "Coordinate Attention for Efficient Mobile Network Design". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 13713–13722.
- [92] Paul VC Hough. *Method and means for recognizing complex patterns*. US Patent 3,069,654. Dec. 1962.
- [93] Andrew Howard et al. "Searching for MobileNetV3". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [94] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. arXiv: [1704.04861](https://arxiv.org/abs/1704.04861) [cs.CV]. URL: <https://arxiv.org/abs/1704.04861>.
- [95] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

- [96] K Hu et al. "Deep Learning Techniques for In-Crop Weed Identification: A Review. arXiv 2021". In: (Mar. 2021). URL: <https://arxiv.org/abs/2103.14872>.
- [97] G.-B. Huang, L. Chen, and C.-K. Siew. "Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes". In: *IEEE Transactions on Neural Networks* 17.4 (July 2006), pp. 879–892. ISSN: 1045-9227. DOI: [10.1109/TNN.2006.875977](https://doi.org/10.1109/TNN.2006.875977). URL: <http://ieeexplore.ieee.org/document/1650244/>.
- [98] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.
- [99] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: Theory and applications". In: *Neurocomputing* 70.1-3 (Dec. 2006). ISSN: 09252312. DOI: [10.1016/j.neucom.2005.12.126](https://doi.org/10.1016/j.neucom.2005.12.126).
- [100] Guang-Bin Huang et al. "Local Receptive Fields Based Extreme Learning Machine". In: *IEEE Computational Intelligence Magazine* 10.2 (May 2015), pp. 18–29. ISSN: 1556-603X. DOI: [10.1109/MCI.2015.2405316](https://doi.org/10.1109/MCI.2015.2405316). URL: <http://ieeexplore.ieee.org/document/7083684/>.
- [101] Huasheng Huang et al. "A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery". In: *PloS one* 13.4 (2018), e0196302.
- [102] Huasheng Huang et al. "Accurate weed mapping and prescription map generation based on fully convolutional networks using UAV imagery". In: *Sensors* 18.10 (2018), p. 3299.
- [103] Huasheng Huang et al. "Deep learning versus Object-based Image Analysis (OBIA) in weed mapping of UAV imagery". In: *International Journal of Remote Sensing* 41.9 (2020), pp. 3446–3479.
- [104] Calvin Hung, Zhe Xu, and Salah Sukkarieh. "Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a UAV". In: *Remote Sensing* 6.12 (2014), pp. 12037–12054.
- [105] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. "A review of deep transfer learning and recent advancements". In: *Technologies* 11.2 (2023), p. 40.
- [106] Christian Janiesch, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning". In: *Electronic Markets* 31.3 (2021), pp. 685–695.
- [107] Honghua Jiang et al. "CNN feature based graph convolutional network for weed and crop recognition in smart farming". In: *Computers and Electronics in Agriculture* 174 (July 2020), p. 105450. ISSN: 01681699. DOI: [10.1016/j.compag.2020.105450](https://doi.org/10.1016/j.compag.2020.105450). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169919321349>.

- [108] Glenn Jocher. *Ultralytics YOLOv5*. Version 7.0. 2020. DOI: [10.5281/zenodo.3908559](https://doi.org/10.5281/zenodo.3908559). URL: <https://github.com/ultralytics/yolov5>.
- [109] Glenn Jocher et al. *YOLOv8*. <https://github.com/ultralytics/ultralytics>. Accessed: 2025-08-04. 2023.
- [110] Radhika Kamath et al. "Classification of paddy crop and weeds using semantic segmentation". In: *Cogent engineering* 9.1 (2022), p. 2018791.
- [111] Liyanaarachchi Lekamalage Chamara Kasun et al. "Representational Learning with ELMs for Big Data". In: *IEEE Intelligent Systems* 28.6 (Nov. 2013), pp. 31–34. ISSN: 1541-1672. DOI: [10.1109/MIS.2013.140](https://doi.org/10.1109/MIS.2013.140).
- [112] Kensuke Kawamura et al. "Discriminating crops/weeds in an upland rice field from UAV images with the SLIC-RF algorithm". In: *Plant Production Science* 24.2 (Apr. 2021), pp. 198–215. ISSN: 1343-943X. DOI: [10.1080/1343943X.2020.1829490](https://doi.org/10.1080/1343943X.2020.1829490). URL: <https://www.tandfonline.com/doi/full/10.1080/1343943X.2020.1829490>.
- [113] Wajahat Kazmi et al. "Exploiting affine invariant regions and leaf edge shapes for weed detection". In: *Computers and Electronics in Agriculture* 118 (Oct. 2015), pp. 290–299. ISSN: 01681699. DOI: [10.1016/j.compag.2015.08.023](https://doi.org/10.1016/j.compag.2015.08.023). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169915002495>.
- [114] Abbas Khan et al. "CED-Net: Crops and Weeds Segmentation for Smart Farming Using a Small Cascaded Encoder-Decoder Architecture". In: *Electronics* 9.10 (Oct. 2020), p. 1602. ISSN: 2079-9292. DOI: [10.3390/electronics9101602](https://doi.org/10.3390/electronics9101602). URL: <https://www.mdpi.com/2079-9292/9/10/1602>.
- [115] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).
- [116] Florian J Knoll et al. "Improving efficiency of organic farming by using a deep learning classification approach". In: *Computers and electronics in agriculture* 153 (2018), pp. 347–356.
- [117] Florian J Knoll et al. "Real-time classification of weeds in organic carrot production using deep learning algorithms". In: *Computers and Electronics in Agriculture* 167 (2019), p. 105097.
- [118] Shuolin Kong et al. "Real-time detection of crops with dense planting using deep learning at seedling stage". In: *Agronomy* 13.6 (2023), p. 1503.
- [119] Tsampikos Kounalakis, Georgios A. Triantafyllidis, and Lazaros Nalpantidis. "Deep learning-based visual recognition of rumex for robotic precision farming". In: *Computers and Electronics in Agriculture* 165 (Oct. 2019), p. 104973. ISSN: 01681699. DOI: [10.1016/j.compag.2019.104973](https://doi.org/10.1016/j.compag.2019.104973). URL: <https://linkinghub.elsevier.com/retrieve/pii/S016816991930331X>.
- [120] A. Krizhevsky and G. Hinton. *Learning multiple layers of features from tiny images*. 2009.

- [121] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009.
- [122] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- [123] Olee Hoi Ying Lam et al. “An open source workflow for weed mapping in native grassland using unmanned aerial vehicle: Using Rumex obtusifolius as a case study”. In: *European Journal of Remote Sensing* 54.sup1 (2021), pp. 71–88.
- [124] Petre Lameski et al. “Weed Detection Dataset with RGB Images Taken Under Variable Light Conditions”. In: 2017, pp. 112–119. DOI: [10.1007/978-3-319-67597-8_11](https://doi.org/10.1007/978-3-319-67597-8_11). URL: http://link.springer.com/10.1007/978-3-319-67597-8_11.
- [125] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories”. In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06)*. Vol. 2. IEEE, 2006, pp. 2169–2178.
- [126] Vi Nguyen Thanh Le, Selam Ahderom, and Kamal Alameh. “Performances of the lbp based algorithm over cnn models for detecting crops and weeds with similar morphologies”. In: *Sensors* 20.8 (2020), p. 2193.
- [127] Vi Nguyen Thanh Le, Beniamin Apopei, and Kamal Alameh. “Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods”. In: *Information processing in agriculture* 6.1 (2019), pp. 116–131.
- [128] Vi Nguyen Thanh Le, Giang Truong, and Kamal Alameh. “Detecting weeds from crops under complex field environments based on Faster RCNN”. In: *2020 IEEE eighth international conference on communications and electronics (ICCE)*. IEEE, 2021, pp. 350–355.
- [129] Vi Nguyen Thanh Le et al. “A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered Local Binary Pattern operators”. In: *GigaScience* 9.3 (2020), giaa017.
- [130] Y. LeCun, Fu Jie Huang, and L. Bottou. “Learning methods for generic object recognition with invariance to pose and lighting”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. Vol. 2. IEEE, pp. 97–104. ISBN: 0-7695-2158-4. DOI: [10.1109/CVPR.2004.1315150](https://doi.org/10.1109/CVPR.2004.1315150). URL: <http://ieeexplore.ieee.org/document/1315150/>.

- [131] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. ISSN: 00189219. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791). URL: <http://ieeexplore.ieee.org/document/726791/>.
- [132] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. ISSN: 00189219. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791). URL: <http://ieeexplore.ieee.org/document/726791/>.
- [133] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 0028-0836. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539). URL: <http://www.nature.com/articles/nature14539>.
- [134] Simon Leminen Madsen et al. "Open plant phenotype database of common weeds in Denmark". In: *Remote Sensing* 12.8 (2020), p. 1246.
- [135] Yanjie Li et al. "Identification of weeds based on hyperspectral imaging and machine learning". In: *Frontiers in Plant Science* 11 (2021), p. 611622.
- [136] Ming Liang and Xiaolin Hu. "Recurrent convolutional neural network for object recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3367–3375.
- [137] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.
- [138] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V* 13. Springer. 2014, pp. 740–755.
- [139] Bo Liu and Ryan Bruch. "Weed detection for selective spraying: a review". In: *Current Robotics Reports* 1 (2020), pp. 19–26.
- [140] Teng Liu et al. "Semi-supervised learning and attention mechanism for weed detection in wheat". In: *Crop Protection* 174 (2023), p. 106389.
- [141] Yichao Liu, Zongru Shao, and Nico Hoffmann. *Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions*. 2021. arXiv: [2112.05561](https://arxiv.org/abs/2112.05561) [cs.CV]. URL: <https://arxiv.org/abs/2112.05561>.
- [142] Zhuang Liu et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.
- [143] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.
- [144] Francisca López-Granados et al. "Early season weed mapping in sunflower using UAV technology: variability of herbicide treatment maps against weed thresholds". In: *Precision agriculture* 17 (2016), pp. 183–199.
- [145] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

- [146] P. Lottes et al. "An effective classification system for separating sugar beets and weeds for precision farming applications". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2016, pp. 5157–5163. ISBN: 978-1-4673-8026-3. DOI: [10.1109/ICRA.2016.7487720](https://doi.org/10.1109/ICRA.2016.7487720). URL: <http://ieeexplore.ieee.org/document/7487720/>.
- [147] Philipp Lottes and Cyrill Stachniss. "Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sept. 2017, pp. 5155–5161. ISBN: 978-1-5386-2682-5. DOI: [10.1109/IROS.2017.8206403](https://doi.org/10.1109/IROS.2017.8206403). URL: <http://ieeexplore.ieee.org/document/8206403/>.
- [148] Philipp Lottes et al. "Effective Vision-based Classification for Separating Sugar Beets and Weeds for Precision Farming". In: *Journal of Field Robotics* 34.6 (Sept. 2017), pp. 1160–1178. ISSN: 15564959. DOI: [10.1002/rob.21675](https://doi.org/10.1002/rob.21675). URL: <https://onlinelibrary.wiley.com/doi/10.1002/rob.21675>.
- [149] Philipp Lottes et al. "Fully Convolutional Networks With Sequential Information for Robust Crop and Weed Detection in Precision Farming". In: *IEEE Robotics and Automation Letters* 3.4 (Oct. 2018), pp. 2870–2877. ISSN: 2377-3766. DOI: [10.1109/LRA.2018.2846289](https://doi.org/10.1109/LRA.2018.2846289). URL: <https://ieeexplore.ieee.org/document/8379414/>.
- [150] Philipp Lottes et al. "Joint stem detection and crop-weed classification for plant-specific treatment in precision farming". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 8233–8238.
- [151] Philipp Lottes et al. "UAV-based crop and weed classification for smart farming". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3024–3031.
- [152] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. "Effective approaches to attention-based neural machine translation". In: *arXiv preprint arXiv:1508.04025* (2015).
- [153] Xu Ma et al. "Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields". In: *PloS one* 14.4 (2019), e0215676.
- [154] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Atlanta, GA. 2013, p. 3.
- [155] Wouter H Maes and Kathy Steppe. "Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture". In: *Trends in plant science* 24.2 (2019), pp. 152–164.
- [156] Donald W. Marquardt and Ronald D. Snee. "Ridge Regression in Practice". In: *The American Statistician* 29.1 (Feb. 1975). ISSN: 00031305. DOI: [10.2307/2683673](https://doi.org/10.2307/2683673).

- [157] Chris McCool, Tristan Perez, and Ben Upcroft. "Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics". In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1344–1351.
- [158] Mark D. McDonnell and Tony Vladusich. "Enhanced image classification with a fast-learning shallow convolutional neural network". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2015, pp. 1–7. ISBN: 978-1-4799-1960-4. DOI: [10.1109/IJCNN.2015.7280796](https://doi.org/10.1109/IJCNN.2015.7280796). URL: <http://ieeexplore.ieee.org/document/7280796/>.
- [159] Mark D. McDonnell et al. "Fast, Simple and Accurate Handwritten Digit Classification by Training Shallow Neural Network Classifiers with the 'Extreme Learning Machine' Algorithm". In: *PLOS ONE* 10.8 (Aug. 2015). ISSN: 1932-6203. DOI: [10.1371/journal.pone.0134254](https://doi.org/10.1371/journal.pone.0134254).
- [160] Ross McLeod. "Annual Costs of Weeds in Australia. eSYS Development Pty Limited". In: *Centre for Invasive Species Solutions, Canberra* (2018).
- [161] Yoan Miche, Antti Sorjamaa, and Amaury Lendasse. "OP-ELM: Theory, Experiments and a Toolbox". In: *Artificial Neural Networks - ICANN 2008*. Berlin, Heidelberg: Springer Berlin Heidelberg. DOI: [10.1007/978-3-540-87536-9_16](https://doi.org/10.1007/978-3-540-87536-9_16).
- [162] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. "Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), pp. 41–48.
- [163] S Imran Moazzam et al. "Towards automated weed detection through two-stage semantic segmentation of tobacco and weed pixels in aerial Imagery". In: *Smart Agricultural Technology* 4 (2023), p. 100142.
- [164] Syed Imran Moazzam et al. "A w-shaped convolutional network for robust crop and weed classification in agriculture". In: *Precision Agriculture* 24.5 (2023), pp. 2002–2018.
- [165] A.A.O. Monteiro and A. von Wangenheim. *Orthomosaic Dataset of RGB aerial Images for Weed Mapping*. <http://www.lapix.ufsc.br/weed-mapping-sugar-cane>. 2019.
- [166] Anders Krogh Mortensen et al. "The oil radish growth dataset for semantic segmentation and yield estimation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2019, pp. 2703–2710.
- [167] Canicius Mwitta, Glen C Rains, and Eric Prostko. "Evaluation of inference performance of deep learning models for real-time weed detection in an embedded computer". In: *Sensors* 24.2 (2024), p. 514.

- [168] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.
- [169] Nan Liu and Han Wang. "Ensemble Based Extreme Learning Machine". In: *IEEE Signal Processing Letters* 17.8 (Aug. 2010). ISSN: 1070-9908. DOI: [10.1109/LSP.2010.2053356](https://doi.org/10.1109/LSP.2010.2053356).
- [170] João Camargo Neto, George E. Meyer, and David D. Jones. "Individual leaf extractions from young canopy images using Gustafson–Kessel clustering and a genetic algorithm". In: *Computers and Electronics in Agriculture* 51.1-2 (Apr. 2006). ISSN: 01681699. DOI: [10.1016/j.compag.2005.11.002](https://doi.org/10.1016/j.compag.2005.11.002).
- [171] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62.
- [172] Stephen O'Hara and Bruce A Draper. "Introduction to the bag of features paradigm for image classification and retrieval". In: *arXiv preprint arXiv:1101.3354* (2011).
- [173] T. Ojala, M. Pietikainen, and T. Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (July 2002), pp. 971–987. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2002.1017623](https://doi.org/10.1109/TPAMI.2002.1017623). URL: <http://ieeexplore.ieee.org/document/1017623/>.
- [174] Alex Olsen et al. "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning". In: *Scientific Reports* 9.1 (Dec. 2019). ISSN: 2045-2322. DOI: [10.1038/s41598-018-38343-3](https://doi.org/10.1038/s41598-018-38343-3).
- [175] Alex Olsen et al. "In situ leaf classification using histograms of oriented gradients". In: *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE. 2015, pp. 1–8.
- [176] Pauline Ong, Kiat Soon Teo, and Chee Kiong Sia. "UAV-based weed detection in Chinese cabbage using deep learning". In: *Smart Agricultural Technology* 4 (2023), p. 100181.
- [177] Kavir Osorio et al. "A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multispectral Images". In: *AgriEngineering* 2.3 (Aug. 2020), pp. 471–488. ISSN: 2624-7402. DOI: [10.3390/agriengineering2030032](https://doi.org/10.3390/agriengineering2030032). URL: <https://www.mdpi.com/2624-7402/2/3/32>.
- [178] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (Jan. 1979). ISSN: 0018-9472. DOI: [10.1109/TSMC.1979.4310076](https://doi.org/10.1109/TSMC.1979.4310076).

- [179] Emilio Parisotto et al. "Stabilizing Transformers for Reinforcement Learning". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 7487–7498. URL: <https://proceedings.mlr.press/v119/parisotto20a.html>.
- [180] Harsh Pathak et al. "Machine learning and handcrafted image processing methods for classifying common weeds in corn field". In: *Smart Agricultural Technology* 5 (2023), p. 100249.
- [181] Keith G Pembleton et al. "Simple versus diverse pastures: opportunities and challenges in dairy systems". In: *Animal Production Science* 55.7 (2015), pp. 893–901.
- [182] Artzai Picon et al. "Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets". In: *Computers and Electronics in Agriculture* 194 (2022), p. 106719.
- [183] John Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Tech. rep. MSR-TR-98-14. Microsoft, Apr. 1998. URL: <https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/>.
- [184] Ciro Potena, Daniele Nardi, and Alberto Pretto. "Fast and Accurate Crop and Weed Identification with Summarized Train Sets for Precision Agriculture". In: 2017, pp. 105–121. DOI: [10.1007/978-3-319-48036-7_9](https://doi.org/10.1007/978-3-319-48036-7_9). URL: http://link.springer.com/10.1007/978-3-319-48036-7_9.
- [185] Yuanyuan Qing et al. "Deep and wide feature based extreme learning machine for image classification". In: *Neurocomputing* 412 (Oct. 2020). ISSN: 09252312. DOI: [10.1016/j.neucom.2020.06.110](https://doi.org/10.1016/j.neucom.2020.06.110).
- [186] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).
- [187] Abdur Rahman, Yuzhen Lu, and Haifeng Wang. "Performance evaluation of deep learning object detectors for weed detection for cotton". In: *Smart Agricultural Technology* 3 (2023), p. 100126.
- [188] N Rai et al. "SpotWeeds: A multiclass UASs acquired weed image dataset to facilitate site-specific aerial spraying application using Deep Learning". In: *2022 15th International Conference on Precision Agriculture (ICPA)*. 2022.
- [189] Nitin Rai and Xin Sun. "WeedVision: A single-stage deep learning architecture to perform weed detection and segmentation using drone-acquired images". In: *Computers and Electronics in Agriculture* 219 (2024), p. 108792.
- [190] Nitin Rai et al. "Multi-format open-source weed image dataset for real-time weed identification in precision agriculture". In: *Data in Brief* 51 (2023), p. 109691.

- [191] Rekha Raja et al. "Real-time robotic weed knife control system for tomato and lettuce based on geometric appearance of plant labels". In: *Biosystems Engineering* 194 (2020), pp. 152–164.
- [192] W Ramirez et al. "Deep convolutional neural networks for weed detection in agricultural crops using optical aerial images". In: *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*. IEEE. 2020, pp. 133–137.
- [193] Punithavathi Rasappan et al. "Computer Vision and Deep Learning-enabled Weed Detection Model for Precision Agriculture". In: *Computer Systems Science and Engineering* 44 (Aug. 2022), pp. 2759–2774. DOI: [10.32604/csse.2023.027647](https://doi.org/10.32604/csse.2023.027647).
- [194] Najmeh Razfar et al. "Weed detection in soybean crops using custom lightweight deep learning models". In: *Journal of Agriculture and Food Research* 8 (2022), p. 100308.
- [195] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).
- [196] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [197] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.
- [198] Yannik Rist et al. "Weed mapping using very high resolution satellite imagery and fully convolutional neural network". In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 9784–9787.
- [199] Mary-Jane Rogers et al. "The changing role of perennial ryegrass in dairy pastures in northern Victoria, Australia". In: *Grass and Forage Science* 77.2 (2022), pp. 131–140.
- [200] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III* 18. Springer. 2015, pp. 234–241.
- [201] John Wilson Rouse et al. "Monitoring vegetation systems in the Great Plains with ERTS". In: *NASA Spec. Publ 351.1* (1974), p. 309.
- [202] John Wilson Rouse et al. "Monitoring vegetation systems in the Great Plains with ERTS". In: *NASA Spec. Publ 351.1* (1974), p. 309.

- [203] Arno Ruckelshausen et al. "BoniRob—an autonomous field robot platform for individual plant phenotyping". In: *Precision agriculture* 9.841 (2009), p. 1.
- [204] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1986.
- [205] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Dec. 2015), pp. 211–252. ISSN: 0920-5691. DOI: [10.1007/s11263-015-0816-y](https://doi.org/10.1007/s11263-015-0816-y). URL: <http://link.springer.com/10.1007/s11263-015-0816-y>.
- [206] Inkyu Sa et al. "WeedMap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming". In: *Remote Sensing* 10.9 (2018), p. 1423.
- [207] Sajad Sabzi, Yousef Abbaspour-Gilandeh, and Juan Ignacio Arribas. "An automatic visible-range video weed detection, segmentation and classification prototype in potato field". In: *Heliyon* 6.5 (2020).
- [208] E Sadgrove et al. "Fast object detection in pastoral landscapes using a multiple expert colour feature extreme learning machine". In: *Asian-Australasian Conference on Precision Pastures and Livestock Farming* 1 (2017).
- [209] Edmund J. Sadgrove et al. "Fast object detection in pastoral landscapes using a Colour Feature Extreme Learning Machine". In: *Computers and Electronics in Agriculture* 139 (June 2017). ISSN: 01681699. DOI: [10.1016/j.compag.2017.05.017](https://doi.org/10.1016/j.compag.2017.05.017).
- [210] Edmund J. Sadgrove et al. "Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (MEC-ELM)". In: *Computers in Industry* 98 (June 2018). ISSN: 01663615. DOI: [10.1016/j.compind.2018.03.014](https://doi.org/10.1016/j.compind.2018.03.014).
- [211] Edmund J. Sadgrove et al. "The Segmented Colour Feature Extreme Learning Machine: Applications in Agricultural Robotics". In: *Agronomy* 11.11 (Nov. 2021), p. 2290. ISSN: 2073-4395. DOI: [10.3390/agronomy11112290](https://doi.org/10.3390/agronomy11112290). URL: <https://www.mdpi.com/2073-4395/11/11/2290>.
- [212] Juan Sandino et al. "UAVs and machine learning revolutionising invasive grass and vegetation surveys in remote arid lands". In: *Sensors* 18.2 (2018), p. 605.
- [213] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.
- [214] T Sarvini et al. "Performance comparison of weed detection algorithms". In: *2019 international conference on communication and signal processing (ICCSP)*. IEEE. 2019, pp. 0843–0847.

- [215] I.M. Scotford and P.C.H. Miller. "Applications of Spectral Reflectance Techniques in Northern European Cereal Production: A Review". In: *Biosystems Engineering* 90.3 (2005), pp. 235–250. ISSN: 1537-5110. DOI: <https://doi.org/10.1016/j.biosystemseng.2004.11.010>. URL: <https://www.sciencedirect.com/science/article/pii/S1537511004002156>.
- [216] Tej Bahadur Shahi et al. "Deep Learning-Based Weed Detection Using UAV Images: A Comparative Study". In: *Drones* 7.10 (2023). ISSN: 2504-446X. DOI: [10.3390/drones7100624](https://doi.org/10.3390/drones7100624). URL: <https://www.mdpi.com/2504-446X/7/10/624>.
- [217] Ali Sharif Razavian et al. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 806–813.
- [218] Shaun M Sharpe, Arnold W Schumann, and Nathan S Boyd. "Detection of Carolina geranium (*Geranium carolinianum*) growing in competition with strawberry using convolutional neural networks". In: *Weed Science* 67.2 (2019), pp. 239–245.
- [219] Shaun M Sharpe, Arnold W Schumann, and Nathan S Boyd. "Goosegrass detection in strawberry and tomato using a convolutional neural network". In: *Scientific Reports* 10.1 (2020), p. 9548.
- [220] Yuri Shendryk et al. "Leveraging high-resolution satellite imagery and gradient boosting for invasive weed mapping". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 4443–4450.
- [221] Nikolaos G. Silleos et al. "Vegetation Indices: Advances Made in Biomass Estimation and Vegetation Monitoring in the Last 30 Years". In: *Geocarto International* 21.4 (Dec. 2006), pp. 21–28. ISSN: 1010-6049. DOI: [10.1080/10106040608542399](https://doi.org/10.1080/10106040608542399). URL: <http://www.tandfonline.com/doi/abs/10.1080/10106040608542399>.
- [222] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).
- [223] Brian M Sindel. "Management of weeds in pasture systems". In: *Handbook of sustainable weed management*. CRC Press, 2006, pp. 799–818.
- [224] Søren Skovsen et al. "The GrassClover image dataset for semantic and hierarchical species understanding in agriculture". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019, pp. 0–0.
- [225] Søren Kelstrup Skovsen et al. "Robust species distribution mapping of crop mixtures using color images and convolutional neural networks". In: *Sensors* 21.1 (2020), p. 175.

- [226] Lyndon N. Smith et al. "Weed classification in grasslands using convolutional neural networks". In: *Applications of Machine Learning*. Ed. by Michael E. Zelinski et al. SPIE, Sept. 2019, p. 42. ISBN: 9781510629714. DOI: [10.1117/12.2530092](https://doi.org/10.1117/12.2530092). URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11139/2530092/Weed-classification-in-grasslands-using-convolutional-neural-networks/10.1117/12.2530092.full>.
- [227] Mino Sportelli et al. "Evaluation of YOLO object detectors for weed detection in different turfgrass scenarios". In: *Applied Sciences* 13.14 (2023), p. 8502.
- [228] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.
- [229] Augmented Startups. *Weeds Dataset*. <https://universe.roboflow.com/augmented-startups/weeds-nxe1w>. Open Source Dataset. visited on 2024-08-05. Nov. 2021. URL: <https://universe.roboflow.com/augmented-startups/weeds-nxe1w>.
- [230] Bryan L Stegelmeier, T Zane Davis, and Michael J Clayton. "Plant-induced photosensitivity and dermatitis in livestock". In: *Veterinary Clinics of North America: Food Animal Practice* 36.3 (2020), pp. 725–733.
- [231] Daniel Steininger et al. "The CropAndWeed Dataset: a Multi-Modal Learning Approach for Efficient Crop and Weed Manipulation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 3729–3738.
- [232] Wolfram Strothmann et al. "Plant classification with in-field-labeling for crop/weed discrimination using spectral features and 3d surface features from a multi-wavelength laser line profile system". In: *Computers and Electronics in Agriculture* 134 (2017), pp. 79–93.
- [233] Kaspars Sudars et al. "Dataset of annotated food crops and weed images for robotic computer vision control". In: *Data in brief* 31 (2020), p. 105833.
- [234] Hyun K Suh et al. "Transfer learning for the classification of sugar beet and volunteer potato under field conditions". In: *Biosystems engineering* 174 (2018), pp. 50–65.
- [235] Jun Sun et al. "Beet seedling and weed recognition based on convolutional neural network and multi-modality images". In: *Multimedia Tools and Applications* (2022), pp. 1–20.
- [236] Christian Szegedy et al. "Going Deeper With Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

- [237] Christian Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (Feb. 2017). DOI: [10.1609/aaai.v31i1.11231](https://doi.org/10.1609/aaai.v31i1.11231). URL: <https://ojs.aaai.org/index.php/AAAI/article/view/11231>.
- [238] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [239] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.
- [240] Jiexiong Tang, Chenwei Deng, and Guang-Bin Huang. "Extreme Learning Machine for Multilayer Perceptron". In: *IEEE Transactions on Neural Networks and Learning Systems* 27.4 (Apr. 2016). ISSN: 2162-237X. DOI: [10.1109/TNNLS.2015.2424995](https://doi.org/10.1109/TNNLS.2015.2424995).
- [241] JingLei Tang et al. "Research on weeds identification based on K-means feature learning". In: *Soft Computing* 22.22 (Nov. 2018), pp. 7649–7658. ISSN: 1432-7643. DOI: [10.1007/s00500-018-3125-x](https://doi.org/10.1007/s00500-018-3125-x). URL: <http://link.springer.com/10.1007/s00500-018-3125-x>.
- [242] JingLei Tang et al. "Weed identification based on K-means feature learning combined with convolutional neural network". In: *Computers and Electronics in Agriculture* 135 (Apr. 2017), pp. 63–70. ISSN: 01681699. DOI: [10.1016/j.compag.2017.01.001](https://doi.org/10.1016/j.compag.2017.01.001). URL: <https://linkinghub.elsevier.com/retrieve/pii/S016816991630641X>.
- [243] Jonathan Tapson, Philip de Chazal, and André van Schaik. "Explicit Computation of Input Weights in Extreme Learning Machines". In: 2015. DOI: [10.1007/978-3-319-14063-6_4](https://doi.org/10.1007/978-3-319-14063-6_4).
- [244] Nima Teimouri, Rasmus Nyholm Jørgensen, and Ole Green. "Novel assessment of region-based CNNs for detecting monocot/dicot weeds in dense field environments". In: *Agronomy* 12.5 (2022), p. 1167.
- [245] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS". In: *Machine Learning and Knowledge Extraction* 5.4 (2023), pp. 1680–1716. ISSN: 2504-4990. DOI: [10.3390/make5040083](https://doi.org/10.3390/make5040083). URL: <https://www.mdpi.com/2504-4990/5/4/83>.
- [246] Migel D. Tissera and Mark D. McDonnell. "Deep extreme learning machines: supervised autoencoding architecture for classification". In: *Neurocomputing* 174 (Jan. 2016), pp. 42–49. ISSN: 09252312. DOI: [10.1016/j.neucom.2015.03.110](https://doi.org/10.1016/j.neucom.2015.03.110). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0925231215011327>.

- [247] Vo Hoang Trong et al. "Late fusion of multimodal deep neural networks for weeds classification". In: *Computers and Electronics in Agriculture* 175 (2020), p. 105506.
- [248] A. Vadivel, Shamik Sural, and A.K. Majumdar. "An Integrated Color and Intensity Co-occurrence Matrix". In: *Pattern Recognition Letters* 28.8 (June 2007), pp. 974–983. ISSN: 01678655. DOI: [10.1016/j.patrec.2007.01.004](https://doi.org/10.1016/j.patrec.2007.01.004). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0167865507000219>.
- [249] J Valente et al. "Detecting Rumex obtusifolius weed plants in grasslands from UAV RGB imagery using deep learning". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2019), pp. 179–185.
- [250] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [251] Aichen Wang, Wen Zhang, and Xinhua Wei. "A review on weed detection using ground-based machine vision and image processing techniques". In: *Computers and Electronics in Agriculture* 158 (Mar. 2019), pp. 226–240. ISSN: 01681699. DOI: [10.1016/j.compag.2019.02.005](https://doi.org/10.1016/j.compag.2019.02.005). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169918317150>.
- [252] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 7464–7475.
- [253] Pei Wang et al. "Weed25: A deep learning dataset for weed identification". In: *Frontiers in Plant Science* 13 (2022), p. 1053329.
- [254] Alexander Wendel and James Underwood. "Self-supervised weed detection in vegetable crops using ground based hyperspectral imaging". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 5128–5135.
- [255] Sanghyun Woo et al. "Cbam: Convolutional block attention module". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.
- [256] Zhangnan Wu et al. "Review of weed detection methods based on computer vision". In: *Sensors* 21.11 (2021), p. 3647.
- [257] Jinru Xue and Baofeng Su. "Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications". In: *Journal of Sensors* 2017 (2017), pp. 1–17. ISSN: 1687-725X. DOI: [10.1155/2017/1353691](https://doi.org/10.1155/2017/1353691). URL: <https://www.hindawi.com/journals/js/2017/1353691/>.

- [258] Xue Yan, Xiangwu Deng, and Jing Jin. "Classification of weed species in the paddy field with DCNN-Learned features". In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE. 2020, pp. 336–340.
- [259] Biao Yang et al. "FRPNet: An improved Faster-ResNet with PASPP for real-time semantic segmentation in the unstructured field scene". In: *Computers and Electronics in Agriculture* 217 (2024), p. 108623.
- [260] Dong Yu and Li Deng. "Efficient and effective algorithms for training single-hidden-layer neural networks". In: *Pattern Recognition Letters* 33.5 (Apr. 2012), pp. 554–558. ISSN: 01678655. DOI: [10.1016/j.patrec.2011.12.002](https://doi.org/10.1016/j.patrec.2011.12.002). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0167865511004211>.
- [261] Fisher Yu and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions". In: *arXiv preprint arXiv:1511.07122* (2015).
- [262] Fisher Yu et al. "Deep layer aggregation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2403–2412.
- [263] Jialin Yu et al. "Deep learning for image-based weed detection in turfgrass". In: *European Journal of Agronomy* 104 (Mar. 2019), pp. 78–84. ISSN: 11610301. DOI: [10.1016/j.eja.2019.01.004](https://doi.org/10.1016/j.eja.2019.01.004). URL: <https://linkinghub.elsevier.com/retrieve/pii/S1161030118306129>.
- [264] Norio Yuba et al. "Discriminating Pennisetum alopecuoides plants in a grazed pasture from unmanned aerial vehicles using object-based image analysis and random forest classifier". In: *Grassland Science* 67.1 (2021), pp. 73–82. DOI: <https://doi.org/10.1111/grs.12288>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/grs.12288>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/grs.12288>.
- [265] Sangdoo Yun et al. "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6023–6032.
- [266] Seyed Alireza Zamani and Yasser Baleghi. "Early/late fusion structures with optimized feature selection for weed detection using visible and thermal images of paddy fields". In: *Precision Agriculture* 24.2 (2023), pp. 482–510.
- [267] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. "Adaptive deconvolutional networks for mid and high level feature learning". In: *2011 international conference on computer vision*. IEEE. 2011, pp. 2018–2025.
- [268] Matthew D Zeiler et al. "Deconvolutional networks". In: *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE. 2010, pp. 2528–2535.
- [269] Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations (ICLR)*. 2018.

- [270] Lei Zhang et al. "Segmentation algorithm for overlap recognition of seedling lettuce and weeds based on SVM and image blocking". In: *Computers and Electronics in Agriculture* 201 (2022), p. 107284.
- [271] Rufei Zhang et al. "Weed location and recognition based on UAV imaging and deep learning". In: *International Journal of Precision Agricultural Aviation* 3.1 (2020).
- [272] Wenhao Zhang et al. "Broad-Leaf Weed Detection in Pasture". In: *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, June 2018, pp. 101–105. ISBN: 978-1-5386-4991-6. DOI: [10.1109/ICIVC.2018.8492831](https://doi.org/10.1109/ICIVC.2018.8492831). URL: <https://ieeexplore.ieee.org/document/8492831/>.
- [273] Xiaoguang Zhang et al. "A Unified Model for Real-Time Crop Recognition and Stem Localization Exploiting Cross-Task Feature Fusion". In: *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2020, pp. 327–332.
- [274] Hengshuang Zhao et al. "Pyramid scene parsing network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890.
- [275] Bolei Zhou et al. "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [276] Bolei Zhou et al. "Scene parsing through ade20k dataset". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 633–641.
- [277] Zongwei Zhou et al. "Unet++: A nested u-net architecture for medical image segmentation". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer, 2018, pp. 3–11.
- [278] Wentao Zhu, Jun Miao, and Laiyun Qing. *Constrained Extreme Learning Machines: A Study on Classification Cases*. 2015. arXiv: [1501.06115](https://arxiv.org/abs/1501.06115) [cs.LG].
- [279] Jiayao Zhuang et al. "Drought stress impact on the performance of deep convolutional neural networks for weed detection in Bahiagrass". In: *Grass and Forage Science* 78.1 (2023), pp. 214–223.
- [280] Barret Zoph et al. "Learning transferable architectures for scalable image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8697–8710.
- [281] Kunlin Zou et al. "A segmentation network for smart weed management in wheat fields". In: *Computers and Electronics in Agriculture* 202 (2022), p. 107303.

- [282] Tomáš Zoubek et al. "Advancing precision agriculture with computer vision: A comparative study of YOLO models for weed and crop recognition". In: *Crop Protection* 190 (2025), p. 107076.