

# Weed Recognition using Deep Learning Techniques on Class-imbalanced Imagery

A S M Mahmudul Hasan<sup>1,2</sup>, Ferdous Sohel \*<sup>1,2</sup>, Dean Diepeveen<sup>2,3,4</sup>, Hamid Laga<sup>1,5,6</sup>,  
and Michael G.K. Jones<sup>2</sup>

<sup>1</sup>Information Technology, Murdoch University, Murdoch, WA 6150, Australia

<sup>2</sup>Centre for Crop and Food Innovation, Food Futures Institute, Murdoch University,  
Murdoch, WA 6150, Australia

<sup>3</sup>Department of Primary Industries and Regional Development, Western Australia,  
South Perth, WA, 6151, Australia

<sup>4</sup>Centre for Sustainable Farming Systems, Murdoch University, Murdoch, WA 6150,  
Australia

<sup>5</sup>Centre of Biosecurity and One Health, Harry Butler Institute, Murdoch University,  
Murdoch University, Murdoch, WA 6150, Australia

<sup>6</sup>Phenomics and Bioinformatics Research Centre, University of South Australia, SA,  
5000, Australia

---

\*Corresponding author email: F.Sohel@murdoch.edu.au

# Weed Recognition using Deep Learning Techniques on Class-imbalanced Imagery

## Abstract

Most weed species can adversely impact agricultural productivity by competing for nutrients required by high-value crops. Manual weeding is not practical for large cropping areas. Many studies have been undertaken to develop automatic weed management systems for agricultural crops. In this process, one of the major tasks is to recognise the weeds from images. However, weed recognition is a challenging task. It is because weed and crop plants can be similar in colour, texture and shape which can be exacerbated further by the imaging conditions, geographic or weather conditions when the images are recorded. Advanced machine learning techniques can be used to recognise weeds from imagery. In this paper, we have investigated five state-of-the-art deep neural networks, namely VGG16, ResNet-50, Inception-V3, Inception-ResNet-v2 and MobileNetV2, and evaluated their performance for weed recognition. We have used several experimental settings and multiple dataset combinations. In particular, we constructed a large weed-crop dataset by combining several smaller datasets, mitigating class imbalance by data augmentation, and using this dataset in benchmarking the deep neural networks. We investigated the use of transfer learning techniques by preserving the pre-trained weights for extracting the features and fine-tuning them using the images of crop and weed datasets. We found that VGG16 performed better than others on small-scale datasets, while ResNet-50 performed better than other deep networks on the large combined dataset.

**Keywords:** VGG16, ResNet-50, Inception-V3, Inception-ResNet-V2, MobileNetV2, Crop and Weed classification, Machine Learning, Digital agriculture, precision agriculture.

## 1 Introduction

Weeds in crops compete for water, nutrients, space and light, and may decrease product quality (Iqbal et al., 2019). Their control, using a range of herbicides, constitutes a significant part of current agricultural practices. In Australia weed control costs in grain production is estimated at \$4.8 billion per annum. These costs include weed control and the cost of lost production (McLeod, 2018).

The most widely used methods for controlling weeds are chemical-based, where herbicides are applied at an early growth stage of the crop (Harker & O'Donovan, 2013; LÓPEZ-GRANADOS, 2011). Although the weeds spread in small patches in crops, herbicides are usually applied uniformly throughout the

agricultural field. While such an approach works reasonably well against weeds, it also affects the crops. A report from the European Food Safety Authority (EFSA) shows that most of the unprocessed agricultural produces contain harmful substances originating from herbicides (Medina-Pastor & Triacchini, 2020).

Recommended rates of herbicide application are expensive and may also be detrimental to the environment. Thus, new methods that can be used to identify weeds in crops, and then selectively apply herbicides on the weeds, or other methods to control weeds, will reduce production costs to the farmers and benefit the environment. Technologies that enable the rapid discrimination of weeds in crops are now becoming available (Tian et al., 2020).

Recent advances in Deep Learning (DL) have revolutionised the field of Machine Learning (ML). DL has made a significant impact in the area of computer vision by learning features and tasks directly from audio, images or text data without human intervention or predefined rules (Dargan et al., 2019). For image classification, DL methods outperform humans and traditional ML methods in accuracy and speed (Steinberg, 2017). In addition, the availability of computers with powerful GPUs, coupled with the availability of large amounts of labelled data, enable the efficient training of DL models.

As for other computer vision and image analysis problems, digital agriculture and digital farming also benefits from the recent advances in deep learning. Deep learning techniques have been applied for weed and crop management, weed detection, localisation and classification, field conditions and livestock monitoring (Kamilaris & Prenafeta-Boldú, 2018).

ML techniques have been used in commercial solutions to combat weeds. “Robocrop Spot Sprayer” (“Robocrop Spot Sprayer: Weed Removal”, 2018) is a video analysis-based autonomous selective spraying system that can identify potatoes grown in carrots, parsnips, onions or leeks. “WeedSeeker sprayer” (“WeedSeeker 2 Spot Spray System”, n.d.) is a near-infrared reflectance sensor-based system that detects the green component in the field. The machine sprays herbicides only on the plants while reducing the amount of herbicide. Similar technology is offered by a herbicide spraying system known as “WEED-IT” (“Precision Spraying - Weed Sprayer”, n.d.). It can target all green plants on the soil. A fundamental problem with these systems is that they are non-selective of crops or weeds. Therefore the ability to discriminate between crops and weeds is important.

Further development of autonomous weed control systems can be beneficial both economically and environmentally. Labour costs can be reduced by using a machine to identify and remove weeds. Selective spraying can also minimise the amount of herbicides applied (Lameski et al., 2018). The success of an autonomous weed control system will depend on four core modules: (i) weed detection and recognition, (ii) mapping, (iii) guidance and (iv) weed control (Olsen et al., 2019). This paper focuses on the first module: weed detection and recognition, which is a challenging task (Slaughter et al., 2008). This is

because both weeds and crop plants often exhibit similar colours, textures and shapes. Furthermore, the visual properties of both weeds and crop plants can vary depending on the growth stage, lighting conditions, environments and geographical locations (Hasan et al., 2021; Jensen et al., 2020). Also, weeds and crops, exhibit high inter-class similarity as well as high intra-class dissimilarity. The lack of large-scale crop weed datasets is a fundamental problem for DL-based solutions.

There are many approaches to recognise weed and crop classes from images (Wäldchen & Mäder, 2018). High accuracy can be obtained for weed classification using Deep Learning (DL) techniques (Kamilaris & Prenafeta-Boldú, 2018) whereas Chavan and Nandedkar (2018) used Convolutional Neural Network (CNN) models to classify weeds and crop plants. Teimouri et al. (2018) used DL for the classification of weed species and the estimation of growth stages, with an average classification accuracy of 70% and 78% for growth stage estimation.

As a general rule, the accuracy of the methods used for the classification of weed species decreases in multi-class classification when the number of classes is large (Dyrmann et al., 2016; Peteinatos et al., 2020). Class-imbalanced datasets also reduce the performance of DL-based classification techniques because of overfitting (Ali-Gombe & Elyan, 2019). This problem can be addressed using data-level and algorithm-level methods. Data-level methods include oversampling or undersampling of the data. In contrast, algorithm-level methods work by modifying the existing learning algorithms to concentrate less on the majority group and more on the minority classes. The cost-sensitive learning approach is one such approach (Khan et al., 2017; Krawczyk, 2016).

DL techniques have been used extensively for weed recognition, for example Hasan et al. (2021) have provided a comprehensive review of these techniques. Ferreira et al. (2017) compared the performance of CNN with Support Vector Machines (SVM), Adaboost – C4.5, and Random Forest models for discriminating soybean plants, soil, grass, and broadleaf weeds. This study shows that CNN can be used to classify images more accurately than other machine learning approaches. Nkemelu et al. (2018) report that CNN models perform better than SVM and K-Nearest Neighbour (KNN) algorithms.

Transfer learning is an approach that uses the learned features on one problem or data domain for another related problem. Transfer learning mimics classification used by humans, where a person can identify a new thing using previous experience. In deep learning, pre-trained convolutional layers can be used as a feature extractor for a new dataset (Shao et al., 2014). However, most of the well-known CNN models are trained on ImageNet datasets, which contains 1000 classes of objects. That is why, depending on the number of classes in the desired dataset, only the classification layer (fully connected layer) of the models need to be trained again in the transfer learning approach. Suh et al. (2018) applied six CNN models (AlexNet, VGG-19, GoogLeNet, ResNet-50, ResNet-101 and Inception-v3) pre-trained

on the ImageNet dataset to classify sugar beet and volunteer potatoes. They reported that these models can achieve a classification accuracy of about 95% without retraining the pre-trained weights of the convolutional layers. They also observed that the models' performance improved significantly by fine-tuning the pre-trained weights. In the fine-tuning approach, the convolutional layers of the DL models are initialised with the pre-trained weights, and subsequently during the training phase of the model, those weights are retrained for the desired dataset. Instead of training a model from scratch, initialising it with pre-trained weights and fine-tuning them helps the model to achieve better classification accuracy for a new target dataset, and this also saves training time (Gando et al., 2016; Girshick et al., 2014; Hentschel et al., 2016). Olsen et al. (2019) fine-tuned the pre-trained ResNet-50 and Inception-V3 models to classify nine weed species in their study and achieved an average accuracy of 95.7% and 95.1% respectively. In another study, VGG16, ResNet-50 and Inception-V3 pre-trained models were fine-tuned to classify the weed species found in the corn and soybean production system (Ahmad et al., 2021). The VGG16 model achieved the highest classification accuracy of 98.90% in their research.

In this paper, we have performed several experiments: i) First, we evaluated the performance of DL models under the same experimental conditions using small-scale public datasets. ii) We then constructed a large dataset by combining a few small-scale datasets with a variety of weeds in crops. In the dataset construction process, we mitigated the class imbalance problem. In a class-imbalance dataset, certain classes have very high or lower representation compared to others. iii) We then investigated the performance of DL models following several pipelines, e.g. transfer learning and fine-tuning. Finally, we provide a thorough analysis and offer future perspectives (Section 3).

The main contributions of this research are:

- construction of a large data set by combining four small-scale datasets with a variety of weeds and crops.
- addressing the class imbalance issue of the combined dataset using the data augmentation technique.
- comparing the performance of five well-known DL methods using the combined dataset.
- evaluating the efficiency of the pre-trained models on the combined dataset using the transfer learning and fine-tuning approach.

This paper is organised as follows: Section 2 describes the materials and methods, including datasets, pre-processing approaches of images, data augmentation techniques, DL architectures and performance metrics. Section 3 covers the experimental results and analysis, and Section 4 concludes the paper.

## 2 Materials and Methods

### 2.1 Dataset

In this work, four publicly available datasets were used: the “DeepWeeds dataset” (Olsen et al., 2019), the “Soybean weed dataset” (Ferreira et al., 2017), the “Cotton-tomato and weed dataset” (Espejo-Garcia et al., 2020) and the “Corn weed dataset” (Jiang et al., 2020).

#### 2.1.1 DeepWeeds dataset

The DeepWeeds dataset contains images of eight nationally significant species of weeds collected from eight rangeland environments across northern Australia. It also includes another class of images that contain non-weed plants. These are represented as a negative class. In this research, the negative image class was not used as it does not have any weed species. The images were collected using a FLIR Blackfly 23S6C high-resolution ( $1920 \times 1200$  pixel) camera paired with the Fujinon CF25HA-1 machine vision lens (Olsen et al., 2019). The dataset is publicly available through the GitHub repository: <https://github.com/AlexOlsen/DeepWeeds>.

#### 2.1.2 Soybean Weed Dataset

Ferreira et al. (2017) acquired soybean, broadleaf, grass and soil images from Campo Grande in Brazil. We did not use the images from the soil class as they did not contain crop plants or weeds. Ferreira et al. (2017) used a “Sony EXMOR” RGB camera mounted on an Unmanned Aerial Vehicle (UAV - - DJI Phantom 3 Professional). The flights were undertaken in the morning (8 to 10 am) from December 2015 to March 2016 with 400 images captured manually at an average height of four meters above the ground. The images of size  $4000 \times 3000$  were then segmented using the Simple Linear Iterative Clustering (SLIC) superpixels algorithm (Achanta et al., 2012) with manual annotation of the segments to their respective classes. The dataset contained 15336 segments of four classes. This dataset is publicly available at the website: <https://data.mendeley.com/datasets/3fmjm7ncc6/2>.

#### 2.1.3 Cotton Tomato Weed Dataset

This dataset was acquired from three different farms in Greece, covering the south-central, central and northern areas of Greece. The images were captured in the morning (8 to 10 am) from May 2019

to June 2019 to ensure similar light intensities. The images of size  $2272 \times 1704$  were taken manually from about one-meter height using a Nikon D700 camera (Espejo-Garcia et al., 2020). The dataset is available through the GitHub repository: <https://github.com/AUAGroup/early-crop-weed>.

#### 2.1.4 Corn Weed Dataset

This dataset was taken from a corn field in China. A total of 6000 images were captured using a Canon PowerShot SX600 HS camera placed vertically above the crop. To avoid the influence of illumination variations from different backgrounds, the images were taken under various lighting conditions. The original images were large ( $3264 \times 2448$ ), and these were subsequently resized to a resolution of  $800 \times 600$  (Jiang et al., 2020). The dataset is available at the Github: <https://github.com/zhangchuanyin/weed-datasets/tree/master/corn%20weed%20datasets>.

#### 2.1.5 Our Combined Dataset

In this paper, we combine all these datasets to create a single large dataset with weed and crop images sourced from different weather and geographical zones. This has created extra variability and complexity in the dataset with a large number of classes. This is also an opportunity to test the DL models and show their efficacy in complex settings. We used this combined dataset to train the classification models. Table 1 provides a summary of the dataset used. The combined dataset contains four types of crop plants and sixteen species of weeds. The combined dataset is highly class-imbalanced since 27% of images are from the soybean crop, while only 0.2% of images are from the cotton crop (Table 1).

#### 2.1.6 Unseen Test Dataset

Another set of data was collected from the Eden Library website (<https://edenlibrary.ai/>) for this research. The website contains some plant datasets for different research work that use artificial intelligence. The images were collected under field conditions. We used images of five different crop plants from the website namely: Chinese cabbage (142 images), grapevine (33 images), pepper (355 images), red cabbage (52 images) and zucchini (100 images). In addition, we also included 500 images of lettuce plants (Jiang et al., 2020) and 201 images of radish plants (Lameski et al., 2017) in the combined dataset. This dataset was then used to evaluate the performance of the transfer learning approach. This experiment checks the reusability of the DL models in the case of a new dataset.

In the study, the images of each class were randomly assigned for training (60%), validation (20%) and testing (20%). Each image was labelled with one image-level annotation. This means that each

Table 1: Summary of crop and weed datasets used in this research

Dataset	Location	Crop/weed species		Number of images	% of images in the class in the combined dataset	
DeepWeeds	Australia	Weed	Chinee apple	1126	4.17	
			Lantana	1063	3.94	
			Parkinsonia	1031	3.82	
			Parthenium	1022	3.78	
			Prickly acacia	1062	3.93	
			Rubber vine	1009	3.74	
			Siam weed	1074	3.98	
			snakeweed	1016	3.76	
			Crop	Soybean	7376	
Soybean Weed	Brazil	Weed	Broadleaf	1191	4.41	
			Grass	3526	13.06	
			Crop	Cotton	54	
Cotton Tomato Weed	Greece	Crop	Tomato	201	0.74	
			Weed	Black nightshade	123	
			Weed	Velvet leaf	130	
		Crop	Corn	1200	4.44	
Corn Weed	China		Blue Grass	1200	4.44	
			Chenopodium album	1200	4.44	
			Cirsium setosum	1200	4.44	
			Sedge	1200	4.44	

image has only one label, i.e., the name of the weed or crop classes, e.g., Chinee apple or corn. Figure 1 provides sample images in the dataset.

## 2.2 Image Pre-processing

Some level of image pre-processing is needed before the data can be used as input for training the DL model. This includes resizing the images, removing the background, enhancing and denoising the images, colour transformation, morphological transformation etc. In this study, the Keras pre-processing utilities (Chollet et al., 2015) were used to prepare the data for training. This function applies some predefined operations to the data. One of the operations is to increase the dimension of the input. DL models process images in batches. To create the batches of images, additional dimension resizing is needed. An image contains three properties, e.g., image height, width and the number of channels. The pre-processing function adds a dimension to the image for inclusion in the batch information. Pre-processing involves normalising the data so that the pixel values range is from 0 to 1. Each model has a specific pre-processing technique to transform a standard image into an appropriate input. Research suggests that the classification model performance is improved by increasing the input resolution of the images (Sabottke & Spieler, 2020; Sahlsten et al., 2019). However, the model’s computational complexity also increases with a higher resolution of the input image. The default input resolution for all the models used in this research is  $224 \times 224$ .



Figure 1: Sample crop and weed images of each class from the datasets.

### 2.3 Data Augmentation

The combined dataset is highly class-imbalanced. The minority classes are over-sampled using image augmentation to balance the dataset. The augmented data is only used to train the models. Image augmentation is done using the Python image processing library Scikit-image (Van der Walt et al., 2014). After splitting the dataset into training, validation and testing sets, most training images were

from soybean with 4,425 image. By applying augmentation approaches, we obtained 4425 images for all other weed and crop classes; thus we ensured that all classes were balanced. The following operations were applied randomly to the data to generate the augmented images:

- Random rotation in the range of [-25, +25] degrees,
- Horizontal and vertical scaling in the range of 0.5 and 1,
- Horizontal and vertical flip,
- Added random noise (Gaussian noise),
- Blurring the images,
- Applied gamma, sigmoid and logarithmic correction operation, and
- Stretched or shrunk the intensity levels of images.

The models are then trained on both actual data and augmented data without making any discrimination.

## 2.4 Deep Learning

Five state-of-the-art deep learning models with pre-trained weights were used in this research to classify images. These models were made available via the Keras Application Programming Interface (API) (Chollet et al., 2015). TensorFlow (Abadi et al., 2016) was used as a machine learning framework. The selected CNN architectures were:

- **VGG16** (Simonyan & Zisserman, 2014) uses a stack of convolutional layers with a very small receptive field ( $3 \times 3$ ). It was the winner of ImageNet Challenge 2014 in the localisation track. The architecture consists of a stack of 13 convolutional layers, followed by three fully connected layers. A very small receptive field ( $3 \times 3$ ) is used in the convolutional layers. The network fixes the convolutional stride and padding to 1 pixel. Spatial pooling is carried out by the max-pooling layers. However, only five of the convolutional layers are followed by the max-pooling layer. This actual state-of-the-art VGG16 model has 138,357,544 trainable parameters. Of these, about 124 million parameters are contained in the fully connected layers. Those layers were customised in this research.
- **ResNet-50** (He et al., 2016) is deeper than VGG16 but has a lower computational complexity. Generally, with increasing depths of the network, the performance becomes saturated or degraded.

The model uses residual blocks to maintain accuracy with the deeper network. The residual blocks also contain convolutions layers like VGG16. The model uses batch normalisation after each convolutional layer and before the activation layer. The model explicitly reformulates the layers as residual functions with reference to the input layers and skip connections. Although the model contains more layers than VGG16, it only has 25,636,712 trainable parameters.

- **Inception-V3** (Szegedy et al., 2016) uses a deeper network with fewer training parameters (23,851,784).  
The model consists of symmetric and asymmetric building blocks with convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers.
- **Inception-ResNet-V2** (Szegedy et al., 2017) combines the concept of skip connections from ResNet with Inception modules. Each inception block is followed by a filter expansion layer ( $1 \times 1$  convolution without activation). Before concatenation with the input layer the dimensionality expansion is performed to match the depth. The model uses batch normalisation only on the traditional layer, but not for the summation layers. The network is 164 layers deep and has 55,873,736 trainable parameters.
- MobileNetV2 (Sandler et al., 2018) allows memory-efficient inference with a reduced number of parameters. It contains 3,538,984 trainable parameters. The basic building block of the model is a bottleneck depth-separable convolution with residuals. The model has the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. It always uses  $3 \times 3$  kernels and utilises the dropout layer and batch normalisation during training. Instead of ReLU (Rectified Linear Unit), this model uses ReLU6 as an activation function. ReLU6 is a variant of ReLU, where the number 6 is an arbitrary choice of the upper bound, which worked well and the model can easily learn the sparse features.

All the models were initialised with pre-trained weights trained on the ImageNet dataset. As the models were trained to recognise 1000 different objects, the original architecture was slightly modified to classify twenty crops and weed species. The last fully-connected layer of the original model was replaced by a global average pooling layer followed by two dense layers with 1024 neurons and “ReLU” activation function. The output contained another dense layer where the number of neurons depended on the number of classes. The softmax activation function was used in the output layer since the models were multi-class classifiers. The size of the input was  $256 \times 256 \times 3$ , and the batch size was 64. The maximum number of epochs for training the models was 100. However, often the training was completed before reaching the maximum number. The initial learning rate was set to  $1 \times 10^{-4}$  and is randomly decreased down to  $10^{-6}$  by monitoring the validation loss in every epoch. Table 2 shows the number of parameters of each of the models used in this research without the output layer. It was found that the

Table 2: Number of parameters used in the deep learning models

<b>Deep Learning Model</b>	<b>Number of parameters</b>
VGG16	16,289,600
ResNet-50	26,735,488
Inception-V3	24,950,560
Inception-ResNet-V2	56,960,224
MobileNetV2	4,585,216

Inception-Resnet-V2 model has the most parameters, and the MobileNetV2 model has the least.

## 2.5 Transfer Learning and Fine-Tuning

A conventional DL model contains two basic components: a feature extractor and a classifier. Depending on the DL model, different layers in the feature extractor and classifier may vary. However, all the DL architectures, used in this research, contain a series of trainable filters. Their weights are adjusted or trained for classifying images of a target dataset. Figure 2a shows a basic structure of a pre-trained DL model. A pre-trained DL model means that the weights of the filters in the feature extractor and classifier is trained to classify 1000 different classes of images contained in the ImageNet dataset. The concept of transfer learning is to use those pre-trained weights to classify the images of a new unseen dataset (Guo et al., 2019; Pan & Yang, 2009). We used this approach in two different ways. The approaches were categorised as transfer learning and fine-tuning. To train the model using our dataset of crop and weed images, we took the feature extractor from the pre-trained DL model and removed its classifier part since it was designed for a specific classification task. In the transfer learning approach (Figure 2b), we only trained the weights of the filters in the classifier part and kept the pre-trained weights of the layer in the feature extractor. This process eliminates the potential issue of training the complete network on a large number of labelled images. However, in the fine-tuning approach (Figure 2c), the weights in the feature extractor were initialised from the pre-trained model, but not fixed. During the training phase of the model, the weights were retrained together with the classifier part. This process increased the efficiency of the classifier because it was not necessary to train the whole model from scratch. The model can extract discriminating features for the target dataset more accurately. Our experiments used both approaches and evaluated their performance on the crop and weed image dataset. Finally, we trained one state-of-the-art DL architecture from scratch, using our combined dataset (Section 2.1.5) and used its feature extractor to classify the images in an unseen test dataset (Section 2.1.6) using the transfer learning approach. The performance of the pre-trained state-of-the-art model was then compared with the model trained on the crop and weed dataset.

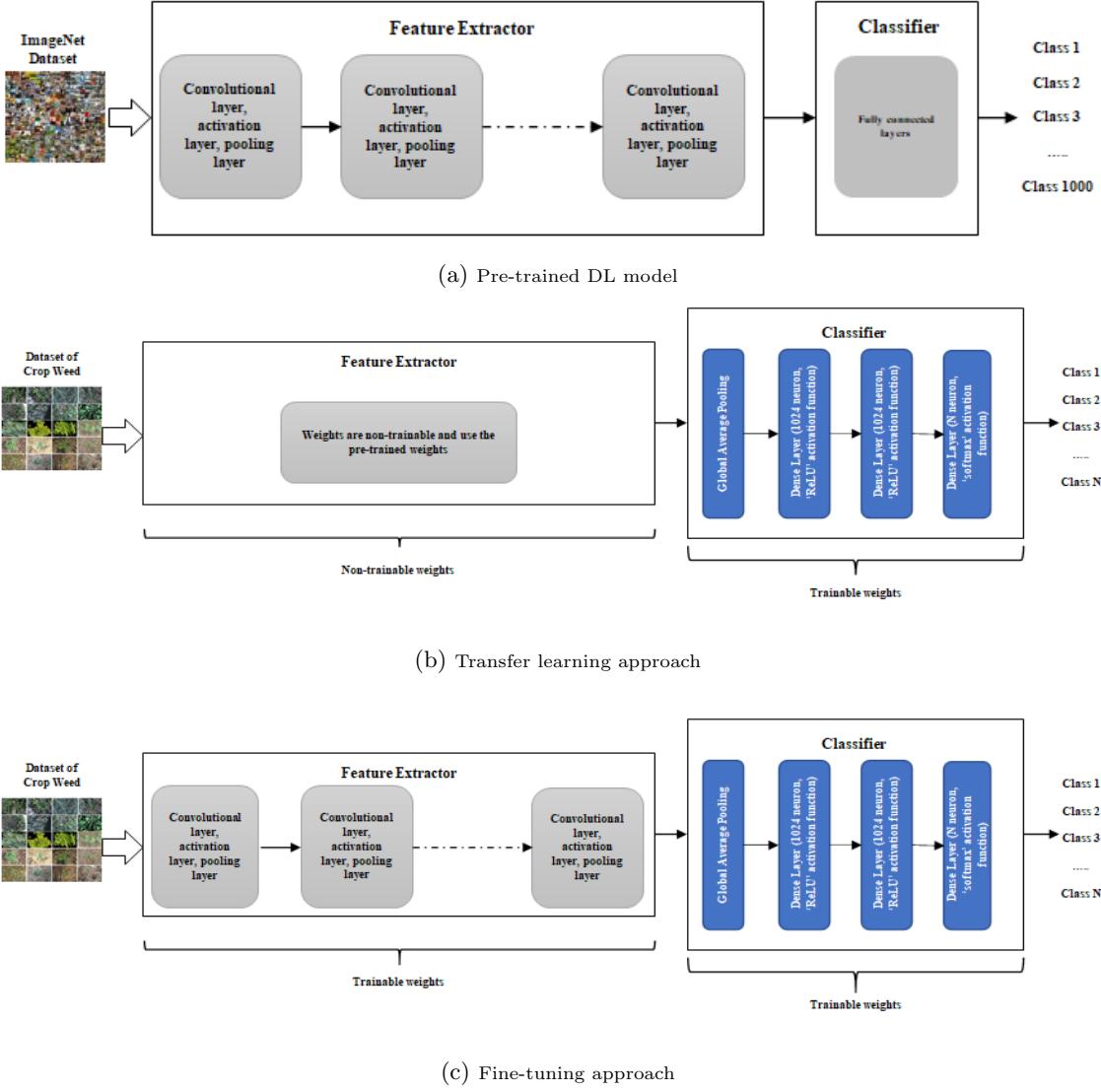


Figure 2: The basic block diagram of DL models used for the experiments.

## 2.6 Performance Metrics

The models were tested and thoroughly evaluated using several metrics: accuracy, precision, recall, and F1 score metrics, which are defined as follows:

- **Accuracy (Acc):** It is the percentage of images whose classes are predicted correctly among all the test images. A higher value represents a better result.
- **Precision (P):** The fraction of correct prediction (True Positive) from the total number of relevant result (Sum of True Positives and False Positives).
- **Recall (R):** The fraction of True Positive from the sum of True Positive and False Negative (number of incorrect predictions).

Table 3: The numbers of images used to train (after augmentation), validate and test the models.

Dataset	Crop and weed species	Training set (number of images)		Validation set (number of images)	Test set (number of images)
		Real images	Real and augmented images		
DeepWeeds	Chinee apple	675	4,425	225	226
	Lantana	637	4,425	212	214
	Parkinsonia	618	4,425	206	207
	Parthenium	613	4,425	204	205
	Prickly acacia	637	4,425	212	213
	Rubber vine	605	4,425	201	203
	Siam weed	644	4,425	214	216
	snakeweed	609	4,425	203	204
Soybean Weed	Soybean	4,425	4,425	1,475	1,476
	Broadleaf	714	4,425	238	239
	Grass	2,112	4,425	704	704
Cotton Tomato Weed	Cotton	32	4,425	10	12
	Tomato	120	4,425	40	41
	Black nightsade	73	4,425	24	26
	Velvet leaf	78	4,425	26	26
Corn Weed	Corn	720	4,425	240	240
	Bluegrass	720	4,425	240	240
	Chenopodium album	720	4,425	240	240
	Cirsium setosum	720	4,425	240	240
	Sedge	718	4,425	239	241

- **F1 Score (F1):** The harmonic mean of precision and recall. This metric is useful to measure the performance of a model on a class-imbalanced dataset.
- **Confusion Matrix:** It is used to measure the performance of machine learning models for classification problems. The confusion matrix tabulates the comparison of the actual target values with the values predicted by the trained model. It helps to visualise how well the classification model is performing and what prediction errors it is making.

In all these metrics, a higher value represents better performance.

### 3 Results and Discussions

We conducted five sets of experiments on the data. Table 3 shows the number of images used for training, validation and testing of the models. Augmentation was applied to generate 4,425 images for each of the classes. However, only actual images were used to validate and test the models. All the experiments were done on a desktop computer, with an Intel(R) Core(TM) i9-9900X processor, 128 gigabyte of RAM and a NVIDIA GeForce RTX 2080 Ti Graphics Processing Unit (GPU). We used the Professional Edition of the Windows 10 operating system. The deep learning models were developed using Python 3.8 and Tensorflow 2.4 framework.

Table 4: Training, validation and testing accuracy for classifying crop and weed species of all four datasets using different DL models.

Dataset	Deep Learning model	Training Accuracy (%)		Validation Accuracy (%)		Testing Accuracy (%)	
		TL	FT	TL	FT	TL	FT
DeepWeeds	VGG16	<b>98.43</b>	99.46	<b>83.84</b>	<b>93.44</b>	<b>84.05</b>	93.36
	ResNet-50	97.56	<b>100.00</b>	46.51	92.96	44.31	<b>93.78</b>
	Inception-V3	81.20	<b>100.00</b>	34.28	86.17	34.77	86.08
	Inception-ResNet-V2	81.02	<b>100.00</b>	35.84	89.09	36.55	89.39
	MobileNetV2	96.47	<b>100.00</b>	35.01	33.09	32.23	31.87
Corn Weed	VGG16	<b>100.00</b>	99.97	<b>96.83</b>	99.33	<b>96.92</b>	<b>99.67</b>
	ResNet-50	<b>100.00</b>	<b>100.00</b>	71.72	99.50	63.11	99.50
	Inception-V3	98.92	<b>100.00</b>	68.39	98.41	59.28	98.42
	Inception-ResNet-V2	97.55	<b>100.00</b>	47.21	<b>99.75</b>	44.96	99.33
	MobileNetV2	99.03	<b>100.00</b>	70.89	89.91	69.03	87.51
Cotton Tomato Weed	VGG16	<b>100.00</b>	96.04	<b>94.00</b>	92.00	<b>99.05</b>	88.57
	ResNet-50	<b>100.00</b>	<b>100.00</b>	54.00	<b>99.00</b>	55.24	<b>99.05</b>
	Inception-V3	<b>100.00</b>	<b>100.00</b>	53.00	96.00	59.05	98.10
	Inception-ResNet-V2	95.71	<b>100.00</b>	64.00	77.00	57.33	77.14
	MobileNetV2	<b>100.00</b>	<b>100.00</b>	64.00	72.00	60.00	78.10
Soybean Weed	VGG16	<b>100.00</b>	99.96	<b>98.97</b>	99.79	<b>98.76</b>	<b>99.88</b>
	ResNet-50	99.98	<b>100.00</b>	82.58	<b>99.91</b>	83.16	99.83
	Inception-V3	99.49	<b>100.00</b>	88.25	99.67	86.77	99.71
	Inception-ResNet-V2	98.80	<b>100.00</b>	90.36	99.79	89.78	99.59
	MobileNetV2	<b>100.00</b>	<b>100.00</b>	94.54	99.54	94.75	99.67

### 3.1 Experiment 1: Comparing the performance of DL models for classifying images in each of the datasets

In this experiment, we trained the five models separately on each dataset using only actual images (see Table 3). Both transfer learning (TL) and fine-tuning (FT) approaches were used to train the models. Table 4 shows the training, validation and testing accuracy for the five models.

On the DeepWeeds dataset, the VGG16 model achieved the highest training, validation and testing accuracy (98.43%, 83.84% and 84.05% respectively) using the transfer learning approach. The training accuracy of the other four models was above 81%. However, the validation and testing accuracy for those models were less than 50%. This suggests that the models are overfitting. After fine-tuning the models, the overfitting problem was mitigated except for the MobileNetV2 architecture. Although four of the models achieved 100% training accuracy after fine-tuning, the validation and testing accuracy was between 86% and 94%. MobileNetV2 model still overfitted even after fine-tuning with about 32% validation and testing accuracy. Overall, the VGG16 model gave the best results for the DeepWeeds dataset as they had the least convolutional layers, which was adequate for small datasets. It should be noted that Olsen et al. (2019), who initially worked on this dataset, achieved an average classification accuracy of 95.1% and 95.7% using Inception-V3 and ResNet-50, respectively. However, they applied

data augmentation techniques to overcome the variable nature of the dataset.

On the Corn Weed and Cotton Tomato Weed datasets, the VGG16 and ResNet-50 models generally gave accurate result, but the accuracy of validation and testing were low for the DL models using the transfer learning approach for both datasets, and the classification performance of the models was substantially improved after fine-tuning. Among the five models, the retrained Inception-ResNet-V2 model gave better results for the Corn Weed dataset with training, validation and testing accuracy of 100%, 99.75% and 99.33% respectively. The ResNet-50 model accurately classified the images of the Cotton Tomato Weed dataset.

VGG16 architecture reached about 99% classification accuracy on both validation and testing data of the “Soybean Weed” dataset using the transfer learning approach. Also, the performance of four other models are better for this dataset using pre-trained weights. Compared to other datasets, the “Soybean Weed” dataset had more training samples, which helped to improve its classification performance. However, after fine-tuning the models on the datasets, all five deep learning architectures achieve more than 99% classification accuracy on the validation and testing data.

According to the results of this experiment, as shown in Table 4, it can be concluded that, for classifying the images of crop and weed species dataset, the transfer learning approach does not work well. Since the pre-trained models were trained on the “ImageNet” dataset (Deng et al., 2009), which does not contain images of crop or weed species, the models cannot accurately classify weed images.

### 3.2 Experiment 2: Combining two datasets

In the previous experiment, we showed that it was unlikely to achieve better classification results using pre-trained weights for the convolutional layers of the DL models. The image classification accuracy improved by fine-tuning the weights of the models for the crop and weed dataset. For that reason, in this experiment, all the models were initialised with pre-trained weights and then retrained for the dataset. In this experiment, the datasets were paired up and used to generate six combinations to train the models. The training, validation and testing accuracies are shown in Table 5. The combinations were-

- “DeepWeeds” with “Corn Weed” dataset (DW-CW),
- “DeepWeeds” with “Cotton Tomato Weed” dataset (DW-CTW),
- “DeepWeeds” with “Soybean Weed” dataset (DW-SW),
- “Corn Weed” with “Cotton Tomato Weed” dataset (CW-CTW),

Table 5: Training, validation and testing accuracy of the DL models after training by combining two of the datasets

<b>DL Models</b>	<b>Accuracy</b>	<b>DW-CW</b>	<b>DW-CTW</b>	<b>DW-SW</b>	<b>CW-CTW</b>	<b>CW-SW</b>	<b>CTW-SW</b>
VGG16	Training	<b>100.00</b>	99.63	99.95	99.97	<b>100.00</b>	<b>100.00</b>
	Validation	<b>96.21</b>	<b>93.64</b>	97.31	98.99	<b>99.67</b>	<b>99.76</b>
	Testing	<b>96.22</b>	<b>94.37</b>	97.25	<b>99.61</b>	<b>99.75</b>	99.76
ResNet-50	Training	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
	Validation	96.10	93.58	<b>97.68</b>	<b>99.53</b>	99.64	99.72
	Testing	95.67	93.25	<b>97.42</b>	99.31	99.61	99.80
Inception-V3	Training	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
	Validation	92.45	87.06	96.07	98.15	99.59	99.44
	Testing	92.06	87.45	96.23	99.16	99.67	<b>99.88</b>
Inception-ResNet-V2	Training	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
	Validation	94.26	89.70	96.43	98.76	99.64	99.56
	Testing	94.25	90.35	96.93	99.46	99.67	99.60
MobileNetV2	Training	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
	Validation	93.01	43.16	96.02	98.31	99.42	99.52
	Testing	92.94	42.49	95.98	98.55	99.61	99.68

- “Corn Weed” with “Soybean Weed” dataset (CW-SW) and
- “Cotton Tomato Weed” with “Soybean Weed” dataset.

After fine-tuning the weights, all the DL models reached 100% training accuracy. The accuracy of the DL architectures also gave better validation and testing results when trained with CW-CTW, CW-SW, CTW-SW combined datasets. However, the models overfitted when trained on the “DeepWeeds” dataset and combined with any of the other three datasets.

The results of the confusion matrix are provided in Figure 3. We found that chinee apple, lantana, prickly acacia and snakeweed had a high confusion rate. This result agrees with that of Olsen et al. (2019). Visually, the images were quite similar and so were difficult to distinguish. That is why the DL model also failed to detect those. Since the dataset was small and did not have enough variations among the images, the models were not able to distinguish among the classes. The datasets also lacked enough images taken under different lighting conditions. The models were unable to detect the actual class of the images because of the illumination effects.

For the DW-CW dataset, the VGG16 model was more accurate. In this case, the model did not distinguish between chinee apple and snakeweed. As shown in the confusion matrix in Figure 3a, out of 224 test images of chinee apple, 16 were classified as snakeweed, and 23 of the 204 test images of snakeweed identified as chinee apple. A significant number of chinee apple and snakeweed images were not correctly predicted by the VGG16 model (see Figure 3b). For the DW-SW dataset, the ResNet-50 model achieved 100% training, 97.68% validation and 97.42% testing accuracy. The confusion matrix is

	bluegrass	chenopodium album	Chinee apple	cirsium setosum	corn	Lantana	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	sedge	Siam weed	Snake weed	
true label	237 (0.99)	0 (0.00)	0 (0.00)	1 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)
predicted label	bluegrass	239 (1.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
bluegrass	237 (0.99)	0 (0.00)	0 (0.00)	1 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)
chenopodium album	0 (0.00)	239 (1.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Chinee apple	0 (0.00)	0 (0.00)	200 (0.88)	0 (0.00)	0 (0.00)	3 (0.01)	2 (0.01)	2 (0.01)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	16 (0.07)
cirsium setosum	0 (0.00)	0 (0.00)	0 (0.02)	240 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
corn	0 (0.00)	0 (0.00)	0 (0.02)	1 (0.00)	238 (0.99)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Lantana	0 (0.00)	0 (0.00)	5 (0.02)	0 (0.00)	0 (0.00)	205 (0.96)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	3 (0.01)
Parkinsonia	0 (0.00)	0 (0.00)	0 (0.02)	0 (0.00)	0 (0.00)	0 (0.00)	203 (0.98)	1 (0.01)	3 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Parthenium	0 (0.00)	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.01)	2 (0.94)	8 (0.04)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)
Prickly acacia	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	4 (0.02)	4 (0.54)	201 (0.01)	2 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	1 (0.00)
Rubber vine	0 (0.00)	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	3 (0.01)	0 (0.00)	197 (0.97)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)
sedge	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	241 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Siam weed	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	214 (0.99)	0 (0.00)	0 (0.00)	0 (0.00)
Snake weed	0 (0.00)	0 (0.00)	0 (0.11)	0 (0.00)	0 (0.01)	2 (0.00)	3 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	3 (0.01)	173 (0.85)	0 (0.00)	0 (0.00)

	black_nightsade	Chinee apple	cotton	Lantana	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	sedge	Siam weed	Snake weed	tomato	velvet_leaf	
true label	26 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
predicted label	black_nightsade	197 (0.87)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
black_nightsade	26 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Chinee apple	0 (0.00)	197 (0.87)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
cotton	0 (0.00)	0 (0.00)	12 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Lantana	0 (0.00)	1 (0.00)	0 (0.97)	207 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Parkinsonia	0 (0.00)	0 (0.00)	0 (0.00)	204 (0.99)	1 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Parthenium	0 (0.00)	3 (0.01)	0 (0.00)	1 (0.94)	192 (0.94)	7 (0.03)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Prickly acacia	0 (0.00)	0 (0.00)	3 (0.01)	5 (0.02)	196 (0.92)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)
Rubber vine	0 (0.00)	3 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	199 (0.98)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Siam weed	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	213 (0.99)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)
Snake weed	0 (0.00)	15 (0.07)	0 (0.00)	2 (0.01)	0 (0.00)	3 (0.01)	1 (0.00)	0 (0.00)	0 (0.00)	4 (0.02)	179 (0.88)	0 (0.00)	0 (0.00)	0 (0.00)
tomato	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	41 (1.00)	0 (0.00)	0 (0.00)
velvet_leaf	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	26 (1.00)	0 (0.00)	0 (0.00)

(a) Confusion matrix for DW-CW dataset (using VGG16 model)
(b) Confusion matrix for DW-CTW dataset (using VGG16 model)

	broadleaf	Chinee apple	grass	Lantana	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	sedge	Siam weed	soybean	
true label	236 (0.99)	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)
predicted label	broadleaf	206 (0.91)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	2 (0.01)	2 (0.01)	0 (0.06)	0 (0.00)	0 (0.00)
broadleaf	236 (0.99)	0 (0.00)	206 (0.91)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Chinee apple	0 (0.00)	703 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
grass	0 (0.00)	0 (0.00)	703 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Lantana	0 (0.00)	5 (0.02)	0 (0.00)	203 (0.95)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.01)	3 (0.01)	2 (0.00)	0 (0.00)
Parkinsonia	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	201 (0.97)	0 (0.00)	0 (0.00)	3 (0.01)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)
Parthenium	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	2 (0.02)	193 (0.94)	7 (0.03)	1 (0.00)	1 (0.00)	1 (0.00)	0 (0.00)
Prickly acacia	0 (0.00)	0 (0.00)	0 (0.00)	5 (0.02)	0 (0.00)	3 (0.01)	203 (0.95)	0 (0.00)	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)
Rubber vine	0 (0.00)	3 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	200 (0.99)	0 (0.00)	0 (0.00)	0 (0.00)
Siam weed	0 (0.00)	0 (0.00)	3 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	212 (0.98)	1 (0.00)	0 (0.00)
Snake weed	0 (0.00)	13 (0.06)	0 (0.00)	9 (0.04)	0 (0.00)	4 (0.02)	1 (0.00)	3 (0.01)	5 (0.02)	169 (0.83)	0 (0.00)	1474 (1.00)
soybean	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)

(c) Confusion matrix for DW-SW dataset (ResNet-50 model)

Figure 3: Confusion matrix of “DeepWeeds” combined with other three dataset.

shown in Figure 3c. The ResNet-50 model identified 13 chinee apple images as snakeweed, and the same number of snakeweed images were classified as chinee apple. The model also identified 9 test images of snakeweed as lantana. Figure 4 shows some sample images which the models classified incorrectly.

By applying data augmentation techniques, one can create more variations among the classes which may also help the model to learn more discriminating features.

17

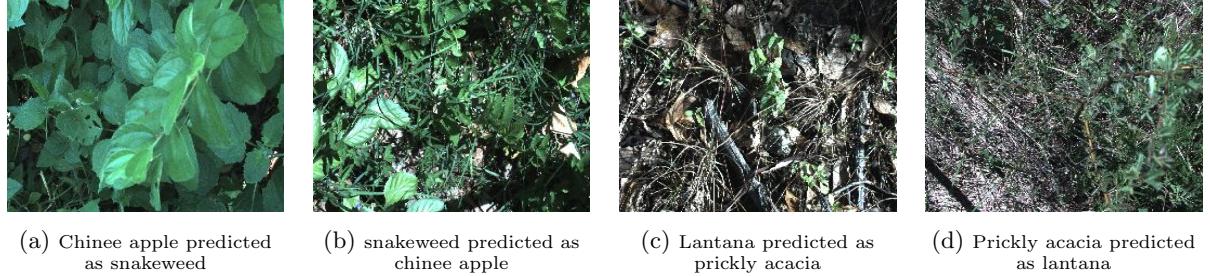


Figure 4: Example of incorrectly classified images.

Table 6: The performance of five deep learning models after training with the combined dataset

DL model	Training accuracy	Validation accuracy	Testing accuracy	Precision (Average)	Recall (Average)	F1 score (Average)
VGG16	99.96	97.53	97.76	96.89	96.83	96.84
ResNet-50	<b>100.00</b>	<b>97.83</b>	<b>98.06</b>	<b>98.06</b>	<b>98.06</b>	<b>98.05</b>
Inception-V3	<b>100.00</b>	96.66	96.09	96.11	97.09	97.09
Inception-Resnet-V2	<b>100.00</b>	96.88	97.17	97.17	97.17	97.16
MobileNetV2	<b>100.00</b>	96.94	97.17	97.18	97.17	97.17

### 3.3 Experiment 3: Training the model with all four datasets together

In this experiment, all the datasets were combined to train the deep learning models. Classifying the images of the combined dataset is much more complex, as the data is highly class-imbalanced. The models were initialised with pre-trained weights and then fine-tuned. Table 6 shows the training, validation and testing accuracy and average precision, recall, and F1 scores achieved by the models on the test data.

After training the models with the combined dataset, the ResNet-50 model performed better. Though all the models except VGG16 achieved 100% training accuracy, the validation (97.83%) and testing (98.06%) accuracies of ResNet-50 architecture were higher. The average precision, recall and F1 score also verified these results. However, the models still did not correctly classify the chinee apple and snakeweed species mentioned in the previous experiment (Section 3.2). A confusion matrix for predicting the classes of images using ResNet-50 is shown in Figure 5. The confusion of ResNet-50 is chosen, since the highest accuracy is achieved in this experiment using this model. Seventeen chinee apple images were classified as snakeweed, and fifteen snakeweeds images were classified incorrectly as chinee apple. In addition, the model also incorrectly classified some lantana and prickly acacia weed images. To overcome this classification problem, both actual and augmented data were used in the following experiment.

	black_nightsade	bluegrass	broadleaf	chenopodium_album	Chinee apple	cirsium_setosum	corn	cotton	grass	Lantana	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	sedge	Siam weed	Snake weed	soybean	tomato	velvet_leaf
true label	26 (1.00) (0.00)	0 (0.00) (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
predicted label	black_nightsade	bluegrass	broadleaf	chenopodium_album	Chinee apple	cirsium_setosum	corn	cotton	grass	Lantana	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	sedge	Siam weed	Snake weed	soybean	tomato	velvet_leaf
black_nightsade	26 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
bluegrass	0 (0.00)	239 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
broadleaf	0 (0.00)	0 (0.00)	236 (0.99) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	2 (0.01) (0.00)	0 (0.00)	0 (0.00)
chenopodium_album	0 (0.00)	0 (0.00)	0 (0.00)	238 (0.99) (0.00)	0 (0.00)	0 (0.01)	2 (0.01) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
Chinee apple	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	201 (0.89) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	2 (0.01) (0.00)	2 (0.01) (0.00)	1 (0.00)	2 (0.01) (0.00)	0 (0.00)	17 (0.08) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
cirsium_setosum	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	240 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
corn	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (1.00) (0.00)	240 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
cotton	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	12 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
grass	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	704 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Lantana	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	2 (0.01) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	200 (0.93) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	7 (0.03) (0.02)	4 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Parkinsonia	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	205 (0.99) (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Parthenium	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.01)	196 (0.96) (0.01)	3 (0.00)	1 (0.00)	0 (0.00)	2 (0.01) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Prickly acacia	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	6 (0.03) (0.01)	3 (0.01) (0.00)	201 (0.94) (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	2 (0.01) (0.00)	0 (0.00)	0 (0.00)	
Rubber vine	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.01)	3 (0.01) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	200 (0.99) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
sedge	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	241 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Siam weed	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	215 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Snake weed	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	15 (0.07) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	4 (0.02) (0.00)	1 (0.00)	1 (0.00)	0 (0.00)	3 (0.01) (0.00)	0 (0.00)	4 (0.02) (0.00)	176 (0.86) (0.00)	0 (0.00)	0 (0.00)	
soybean	0 (0.00)	0 (0.00)	2 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1472 (1.00) (0.00)	0 (0.00)	0 (0.00)	
tomato	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.02) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	40 (0.98) (0.00)	0 (0.00)	0 (0.00)	
velvet_leaf	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.04) (0.00)	25 (0.96) (0.00)	0 (0.00)	

Figure 5: Confusion matrix after combining four dataset using ResNet-50 model

### 3.4 Experiment 4: Training the models using both real and augmented images of the four datasets

Augmented data were used together with the real data in the training phase to address the misclassification problem in the previous experiment (Section 3.3). All the weed species and crop plant images had the same training data for this experiment. The models were initialised with pre-trained weights, and all the parameters were fine-tuned. Table 7 shows the result of this experiment.

From Table 7, we can see that the training accuracy for all the DL models is 100%. Also the validation and testing accuracies were reasonably high. In this experiment, the ResNet-50 models achieved the

Table 7: Performance of five deep learning models after training with the real and augmented data

<b>DL model</b>	<b>Training accuracy</b>	<b>Validation accuracy</b>	<b>Testing accuracy</b>	<b>Precision (Average)</b>	<b>Recall (Average)</b>	<b>F1 score (Average)</b>
VGG16	100.00	97.96	97.83	97.83	97.84	97.83
ResNet-50	<b>100.00</b>	<b>98.31</b>	<b>98.30</b>	<b>98.29</b>	<b>98.30</b>	<b>98.30</b>
Inception-V3	100.00	97.31	98.02	98.02	98.02	98.01
Inception-Resnet-V2	100.00	97.85	97.76	97.76	97.76	97.76
MobileNetV2	100.00	97.68	98.02	98.02	98.02	98.02

highest precision, recall and F1 score for the test data. Figure 6 shows the confusion matrix for the ResNet-50 model. We compared the performance of the model using the confusion matrix with the previous experiment. The performance of the model was improved using both actual and augmented data. The classification accuracy increased for chinee apple, lantana, prickly acacia and snakeweed species by 2%.

In this research, the ResNet-50 model attained the highest accuracy using actual and augmented images. The Inception-ResNet-V2 model gave similar results. The explanation is that both of the models used residual layers. Residual connections help train a deeper neural network with better performance and reduced computational complexity. A deeper convolutional network works better when trained using a large dataset (Szegedy et al., 2017). Since we have used the augmented data and actual images, the dataset size has increased by several times.

### **3.5 Experiment 5: Comparing the performance of two ResNet-50 models individually trained on ImageNet dataset, and the combined dataset, and testing on the Unseen Test dataset**

In this experiment, we used two ResNet-50 models. The first was trained on our combined dataset with actual and augmented data (Sec. 2.1.5). Here, the top layers were removed from the model and a global average pooling layer and three dense layers were added as before. Other than the top layers, all the layers used pre-trained weights, which were not fine-tuned. This model termed as “CW ResNet-50”. The same arrangement was used for the pre-trained ResNet-50 model, which was instead trained on the ImageNet dataset. It was named as “SOTA ResNet-50” model for further use. We trained the top layers of both models using the training split of the Unseen Test Dataset (2.1.6). Both models were tested using the test split of the Unseen Test Dataset. The confusion matrix for CW ResNet-50 and SOTA ResNet-50 model is shown in Figure 7.

We can see in Figure 7 that the performance of the two models is very similar. The “SOTA ResNet-50” model detected all the classes of crop and weeds accurately. However, the pre-trained “CW

	black_nightsade	bluegrass	broadleaf	chenopodium_album	Chinee apple	cirsium_setosum	corn	cotton	grass	Lantana	Parkinsonia	Parthenium	Prickly acacia	Rubber vine	sedge	Siam weed	Snake weed	soybean	tomato	velvet_leaf
true label	26 (1.00) (0.00)	0 (0.00) (0.99)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
black_nightsade	26 (1.00) (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
bluegrass	0 (0.00)	238 (0.99)	0 (0.00)	0 (0.00)	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
broadleaf	0 (0.00)	0 (0.00)	237 (0.99)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	2 (0.01)	0 (0.00)	0 (0.00)	
chenopodium_album	0 (0.00)	0 (0.00)	0 (1.00)	239 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Chinee apple	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.91)	206 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.01)	2 (0.00)	1 (0.00)	3 (0.01)	1 (0.00)	13 (0.06)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
cirsium_setosum	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (1.00)	240 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
corn	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.99)	237 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
cotton	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (1.00)	12 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
grass	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (1.00)	704 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Lantana	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.01)	3 (0.00)	0 (0.00)	0 (0.00)	204 (0.95)	0 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	4 (0.02)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Parkinsonia	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	202 (0.98)	2 (0.01)	2 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Parthenium	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1 (0.01)	2 (0.97)	199 (0.97)	2 (0.01)	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Prickly acacia	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.02)	4 (0.01)	204 (0.96)	3 (0.01)	0 (0.00)	1 (0.00)	1 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Rubber vine	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.02)	4 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.98)	199 (0.98)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
sedge	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (1.00)	241 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Siam weed	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.01)	3 (0.01)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.99)	213 (0.99)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
Snake weed	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.05)	11 (0.00)	0 (0.00)	0 (0.00)	6 (0.03)	0 (0.00)	1 (0.00)	1 (0.00)	3 (0.01)	0 (0.00)	3 (0.88)	179 (0.88)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
soybean	0 (0.00)	0 (0.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	1474 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
tomato	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	41 (1.00)	0 (0.00)	0 (0.00)	
velvet_leaf	1 (0.04)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	25 (0.96)	0 (0.00)	

Figure 6: Confusion matrix for ResNet-50 model using combined dataset with augmentation

Resnet-50” model only identified two images incorrectly. As the “SOTA ResNet-50” model was trained on a large dataset containing millions of images, it detected the discriminating features more accurately. On the other hand, the “CW Resnet-50” model was only trained on 88,500 images. If this model were trained with more data, it is probable that it would be more accurate using the transfer learning approach. This type of pre-trained model could be used for classifying the images of new crop and weed datasets, which would eventually make the training process faster.

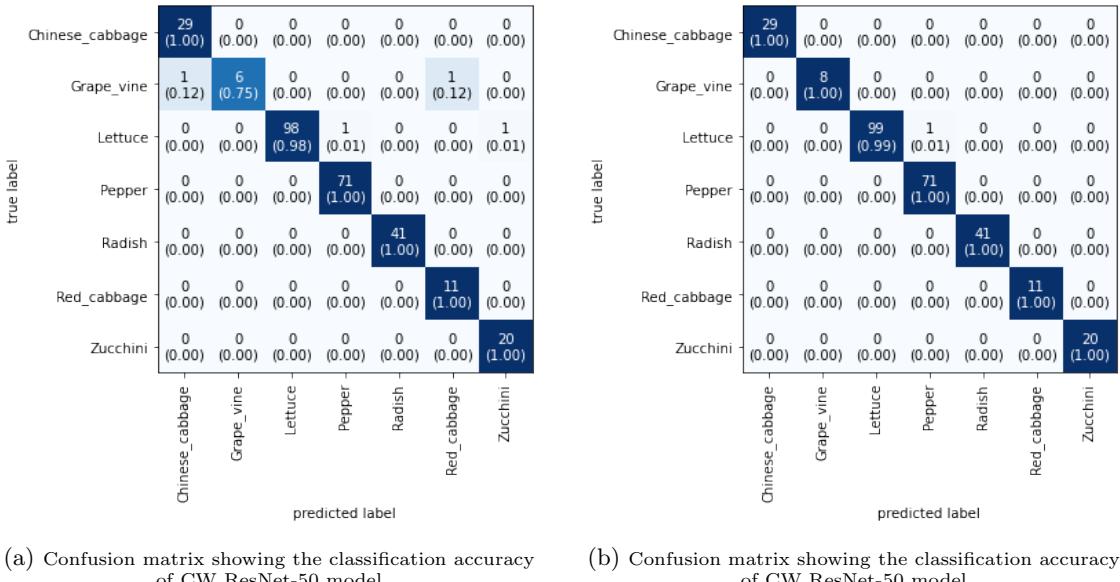


Figure 7: Confusion matrix for CW ResNet-50 and SOTA ResNet-50 model.

## 4 Conclusion

This study was undertaken on four image datasets of crop and weed species collected from four different geographical locations. The datasets contained a total of 20 different species of crops and weeds. We used five state-of-the-art CNN models, namely VGG16, ResNet-50, Inception-V3, Inception-ResNet-V2, MobileNetV2, to classify the images of these crops and weeds.

First, we evaluated the performance of transfer learning and fine-tuning approaches of the models by training them on each dataset. The results showed that fine-tuning of the models could improve classification of the images more accurately than the transfer learning approach.

To add more complexity to the classification problem, we combined the datasets together. After combining two of the datasets, the performance decreased due to some of the species of weeds in the “DeepWeeds” dataset. The weed species that were confused were chinee apple, snakeweed, lantana and prickly acacia. We then combined all four datasets to train the models. Since the dataset was class-imbalanced, it was difficult to achieve high classification accuracy by only training the model with actual images. Consequently, we used augmentation to balance the classes of the dataset. However, it was evident that the models had problems in distinguishing between chinee apple and snakeweed. The performance of the models improved using both actual and augmented data. The models could distinguish chinee apple and snake weed more accurately. The results showed that the ResNet-50 was most accurate.

Another finding was that using the transfer learning method was that in most cases the models did

not achieve the desired accuracy. As ResNet-50 was the most accurate system, we ran a test using this pre-trained model. The model was used to classify the images of a new dataset using the transfer learning approach. Although the model was not more accurate than the state-of-the-art pre-trained ResNet-50 model, it was very close to that. We could expect a higher accuracy using the transfer learning approach if the model can be trained using a large crop and weed dataset.

This research shows that the data augmentation technique can help address the class imbalance problem and add more variations to the dataset. The variations in the images of the training dataset improve the training accuracy of the deep learning models. Moreover, the transfer learning approach can mitigate the requirement of large data sets to train the deep learning models from scratch. The pre-trained models are trained on a large dataset to capture the detailed generalised features from the imagery, e.g., ImageNet in our case. However, because, ImageNet data set was not categorically labelled for weeds or crops, fine-tuning the pre-trained weights with crop and weed datasets help capture the dataset or task-specific features. Consequently, fine-tuning improves classification accuracy.

For training a deep learning model for classifying images, it is essential to have a large dataset like ImageNet (Deng et al., 2009) and MS-COCO (Lin et al., 2014). Classification of crop and weed species cannot be generalised unless a benchmark dataset is available. Most studies in this area are site-specific. A large dataset is needed to generalise the classification of crop and weed plants, and as an initial approach, large datasets can be generated by combining multiple small datasets, as demonstrated here. In this work, the images only had image-level labels. A benchmark dataset can be created by combining many datasets annotated with a variety of image labelling techniques. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) based image sample generation can also be used to mitigate class-imbalance issues. Moreover, it is needed to develop a crop & weed dataset annotated at the object level. For implementing a real-time selective herbicide sprayer, the classification of weed species is not enough. It is also necessary to locate the weeds in crops. Deep learning-based object detection models can be used for detecting weeds.

## 5 Declaration of Funding

This research did not receive any specific funding.

## 6 Conflicts of interest

The authors declare no conflict of interest.

## **7 Data availability**

The data that support this study will be shared upon reasonable request to the corresponding author.

## References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283.
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), 2274–2282.
- Ahmad, A., Saraswat, D., Aggarwal, V., Etienne, A., & Hancock, B. (2021). Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems. *Computers and Electronics in Agriculture*, 184, 106081.
- Ali-Gombe, A., & Elyan, E. (2019). Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network. *Neurocomputing*, 361, 212–221.
- Chavan, T. R., & Nandedkar, A. V. (2018). Agroavnet for crops and weeds classification: A step forward in automatic farming. *Computers and Electronics in Agriculture*, 154, 361–372.
- Chollet, F. et al. (2015). *Keras*. <https://github.com/fchollet/keras>
- Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2019). A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 1–22.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.
- Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*, 151, 72–80.
- Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., & Vasilakoglou, I. (2020). Towards weeds identification assistance through transfer learning. *Computers and Electronics in Agriculture*, 171, 105306.
- Ferreira, A. d. S., Freitas, D. M., da Silva, G. G., Pistori, H., & Folhes, M. T. (2017). Weed detection in soybean crops using convnets. *Computers and Electronics in Agriculture*, 143, 314–324.
- Gando, G., Yamada, T., Sato, H., Oyama, S., & Kurihara, M. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. *Expert Systems with Applications*, 66, 295–301.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.
- Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., & Feris, R. (2019). Spottune: Transfer learning through adaptive fine-tuning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4805–4814.
- Harker, K. N., & O'Donovan, J. T. (2013). Recent weed control, weed management, and integrated weed management. *Weed Technology*, 27(1), 1–11.
- Hasan, A. M., Sohel, F., Diepeveen, D., Laga, H., & Jones, M. G. (2021). A survey of deep learning techniques for weed detection from images. *Computers and Electronics in Agriculture*, 184, 106–067.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hentschel, C., Wiradarma, T. P., & Sack, H. (2016). Fine tuning cnns with scarce training data—adapting imangenet to art epoch classification. *2016 IEEE International Conference on Image Processing (ICIP)*, 3693–3697.
- Iqbal, N., Manalil, S., Chauhan, B. S., & Adkins, S. W. (2019). Investigation of alternate herbicides for effective weed management in glyphosate-tolerant cotton. *Archives of Agronomy and Soil Science*, 65(13), 1885–1899.
- Jensen, T. A., Smith, B., & Defeo, L. F. (2020). An automated site-specific fallow weed management system using unmanned aerial vehicles.
- Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., & Song, C. (2020). Cnn feature based graph convolutional network for weed and crop recognition in smart farming. *Computers and Electronics in Agriculture*, 174, 105450.
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147, 70–90.
- Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8), 3573–3587.
- Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4), 221–232.
- Lameski, P., Zdravevski, E., & Kulakov, A. (2018). Review of automated weed control approaches: An environmental impact perspective. *International Conference on Telecommunications*, 132–147.
- Lameski, P., Zdravevski, E., Trajkovik, V., & Kulakov, A. (2017). Weed detection dataset with rgb images taken under variable light conditions. *International Conference on ICT Innovations*, 112–119.

- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, 740–755.
- LÓPEZ-GRANADOS, F. (2011). Weed detection for site-specific weed management: Mapping and real-time approaches. *Weed Research*, 51(1), 1–11.
- McLeod, R. (2018). Annual costs of weeds in australia. <https://invasives.com.au/wp-content/uploads/2019/01/Cost-of-weeds-report.pdf>
- Medina-Pastor, P., & Triacchini, G. (2020). The 2018 european union report on pesticide residues in food. *EFSA Journal*, 18(4), e06057.
- Nkemelu, D. K., Omeiza, D., & Lubalo, N. (2018). Deep convolutional neural network for plant seedlings classification. *arXiv preprint arXiv:1811.08404*.
- Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., Banks, W., Grgenti, B., Kenny, O., Whinney, J., et al. (2019). Deepweeds: A multiclass weed species image dataset for deep learning. *Scientific reports*, 9(1), 1–12.
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345–1359.
- Peteinatos, G., Reichel, P., Karouta, J., Andújar, D., & Gerhards, R. (2020). Weed identification in maize, sunflower, and potatoes with the aid of convolutional neural networks. *Remote Sensing*, 12(24), 4185.
- Precision spraying - weed sprayer. (n.d.). Retrieved January 25, 2021, from <https://www.weed-it.com/>
- Robocrop spot sprayer: Weed removal. (2018). Retrieved January 25, 2021, from <https://garford.com/products/robocrop-spot-sprayer/>
- Sabottke, C. F., & Spieler, B. M. (2020). The effect of image resolution on deep learning in radiography. *Radiology: Artificial Intelligence*, 2(1), e190015.
- Sahlsten, J., Jaskari, J., Kivinen, J., Turunen, L., Jaanio, E., Hietala, K., & Kaski, K. (2019). Deep learning fundus image analysis for diabetic retinopathy and macular edema grading. *Scientific reports*, 9(1), 1–11.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.
- Shao, L., Zhu, F., & Li, X. (2014). Transfer learning for visual categorization: A survey. *IEEE transactions on neural networks and learning systems*, 26(5), 1019–1034.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Slaughter, D., Giles, D., & Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1), 63–78.

- Steinberg, R. (2017). 6 areas where artificial neural networks outperform humans. Retrieved December 25, 2020, from <https://venturebeat.com/2017/12/08/6-areas-where-artificial-neural-networks-outperform-humans/>
- Suh, H. K., Ijsselmuiden, J., Hofstee, J. W., & van Henten, E. J. (2018). Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosystems engineering*, 174, 50–65.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Teimouri, N., Dyrmann, M., Nielsen, P. R., Mathiassen, S. K., Somerville, G. J., & Jørgensen, R. N. (2018). Weed growth stage estimator using deep convolutional neural networks. *Sensors*, 18(5), 1580.
- Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation—a review. *Information Processing in Agriculture*, 7(1), 1–19.
- Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). Scikit-image: Image processing in python. *PeerJ*, 2, e453.
- Wäldchen, J., & Mäder, P. (2018). Plant species identification using computer vision techniques: A systematic literature review. *Archives of Computational Methods in Engineering*, 25(2), 507–543.
- Weedseeker 2 spot spray system. (n.d.). Retrieved January 25, 2021, from <https://agriculture.trimble.com/product/weedseeker-2-spot-spray-system/>