Progress Report on the Segmentation and Classification of Weeds

Student Engineer: Isaac Tan, N9960210
Supervisors: Jasmine Banks, Khamael Al-Dulaimi

***Abstract*— Weed management plays a crucial role in the outcome of harvests across the industry of agriculture. Through advancements in software systems, more emphasis is being placed on the optimisation of weed management and how emerging technologies can be used to reduce labour, reduce chemical contamination, and increase profitability of agriculture. This report investigates the use of a convolutional neural network in the task of identifying and segmenting weeds in images of grass for the future task of automatically precision spraying weeds. The neural network was built upon the U-Net architecture and the results of this report aimed to compare the accuracy and speed of different constraints on the architecture and asses the overall viability for a segmentation system using this architecture. Findings showed that system resulted in accuracies up to 86.5% and taking at maximum 1.69 seconds to segment a dataset of 135 images 256x256 pixels in size.**

I. Introduction

One industry that will be ever vital to the sustainability of the population is agriculture. Since the dawn of the industry, farmers have investigated ways of optimising the industry so that they can produce the maximum yield while minimising the loss of yield due to factors like weeds, pests, and natural disasters. With technology continuously evolving, people are looking towards artificial neural networks (ANNs) and machine learning to help overcome these issues and make significant improvements in the agriculture industry.

This project will investigate the problem of weeds in agriculture and aim to construct a software system that aids in the alleviation of this problem. To do that, this report will delve into the current literature of machine learning and ANNs in the agriculture industry in relation to weeds. Finally, it will discuss an implementation of a CNN to segment weeds and grass, the CNN's architecture, and the results from its implementation. It will also discuss the risks of such a project and how its implementation is influenced by ethics and sustainability.

This report is in partial fulfilment of EGH400-1 unit assessment requirements and **submitted on 12/06/2022.**
`isaac.tan@connect.edu.au`

II. Literature Review

Before creating the software system, it was important that the task, the technology, and the current approaches were fully understood. The purpose of this literature review is to justify the need for the segmentation software system proposed in this project. Additionally, it will provide an understanding of what an artificial neural network is and how it works. Finally, it will investigate current methods and case studies in similar problem spaces. The literature explored in this review will be contained to sources written in relevant agricultural, computer science or engineering journals or articles from January 2012 onwards. These sources will be obtained through online databases such as IEEE Xplore, Scopus and Inspec.

The need for the software system proposed in this project stems from the huge annual loss of crop to weeds. Studies in both Australia and India showed billion-dollar losses per year to weeds with Australia losing an estimated $5 billion in 2018 (McLeod, 2018), and India losing $11 billion in 2020 (Talaviya, Shah, Patel, Yagnik, & Shah, 2020). Additionally, current methods of controlling weeds are cost ineffective, labour intensive and harmful to the environment. Current methods include hand picking or spraying with targeted herbicide, and broader spraying with a general herbicide. The disadvantage of hand picking and spraying is that it is especially labour intensive on larger farms. On the other hand, using a general herbicide and mass spraying farms is costly, less effective, and harmful to the environment (Hlaing & Khaing, 2014). Different weed species require different herbicides and as such broader spraying cannot eliminate all weeds and instead aim to eliminate the most prevalent species (Tao & Wei, 2022). Additionally, failure to eliminate a weed with a herbicide creates herbicide resistant species as they build tolerance (Kamath, Balachandra, Vardhan, & Maheshwari, 2022). Research in Australia has also shown that herbicides used to treat weeds on sugarcane farms flow off into rivers and then into the Great Barrier Reef, harming one of the natural wonders (Sugar Research Australia, 2022).

Automated targeted spraying aims to solve these issues as weed species are precisely sprayed with the correct herbicide and manual labour is replaced with an automated system. These automated systems can be split into two major tasks: identifying the weed species and applying herbicide. This report focuses solely on the task of identifying the weeds.

One technology that has shown promise in identifying weeds is the use of machine learning and artificial neural networks. Artificial neural networks (ANNs) are algorithms mimicking the biological structure of the human brain (Wen, Yihui, Ying, & Jiaxu, 2022). They make decisions based upon the output of a stream of connections between neurons from an input. These neurons are separated into layers and connected through weight, bias, and an activation function. Information is passed between layers with the weights applied and assessed under the activation function. If the output of the activation function meets the bias or threshold, then the data passed forward to the next layer. This process repeats until the output layer, at which the ANN arrives at a decision (Dave & Dutta, 2014). A simplified diagram of this can be seen in Fig 2.1.

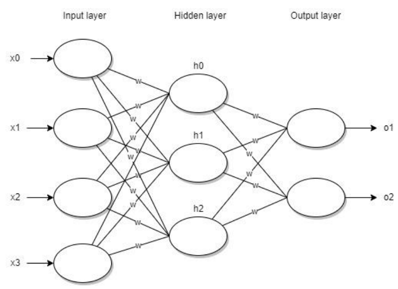

Fig 2.1 Neural Network Diagram. Adapted from (Shiruru, 2016)

Neural networks are made more advanced by adding multiple layers, forming a deep neural network (DNN). A subclass of deep neural networks, convolutional neural networks (CNNs) are particularly useful when it comes to tasks of identifying patterns and similarities in images. CNNs utilise a convolutional filter which when repeated creates a feature map of the input (Grace, Anitha, Sivaramakrishnan, & Sivakumari, 2021). This automatically created feature map removes the manual task of feature extraction which further differentiates ANNs from DNNs. Manually performing feature extraction and applying the weights between neurons is called supervised learning and creating convolutions that automatically adjust these weights is termed unsupervised learning (Abiodun, et al., 2018). The goal of both learning types is to arrive at predictions closer to the expected outcome with each iteration of training. This distance between predicted outcome and expected outcome or ground truth is measured using a loss function and with each step of training, the algorithm adjusts its feature weights to reduce the loss. As this process is repeated the algorithm eventually optimises the loss function and increases accuracy (Yu, Wang, Zou, & Wang, 2020). Convolutions create linear functions of the output of the previous layer and feed it forward, however for binary classification like the classes of weed and not weed in this project, a decision needs to be mapped to 0 or 1. In this case, a sigmoid activation function is used as the last layer of the network to fit the decision to these values (Li, Yang, Peng, & Zhou, 2021). During the training phase, the CNN is constantly updating feature weights to minimise the value loss. However, if the model is trained for too long, it over optimises feature weights specifically for the training dataset causing something called overfitting. This means that when the model is evaluated on the testing dataset, the overall accuracy is lowered as the model has been trained to perform well specifically for the training dataset (Narasinga Rao, Venkatesh Prasad, Sai Teja, Zindavali, & Phanindra Reddy, 2018).

Before a software system was created for this project, case studies on similar projects were investigated. One study in the International Journal of Electrical and Computer Engineering investigated six CNN architectures for classifying weeds amongst bean and beet crops. It investigated VGG (16 & 19), GoogLeNet (Inception v3 & v4), and MobileNet (v1 & v2) and found that Inception v4 was the most accurate with 99.41% and 99.51% accuracy on from scratch and pre-trained models respectively. It found that all models gave a 98%+ accuracy (Adil, Ahmed, Mohammed, & Soufiane, 2022).

When classifying tobacco crop and weeds researchers compared region and grid-based CNNs using R-CNN and YOLOv5 respectively. The study published in Applied Sciences found that the R-CNN resulted in a 98% accuracy and the system using YOLOv5 yielded a 94% accuracy. Additionally, it stated that integrating a system that automatically detected and precision sprayed weeds decreased the use of herbicide by 52% (Alam, et al., 2022).

Another study published in the International Conference on Artificial Intelligence and Smart Systems (ICAIS) compared a series of CNNs to other decision-making algorithms with a varied segmentation and feature extraction techniques to classify crops and weeds. It found that the CNNs were more effective than the other machine learning algorithms and that of the CNNs focusing on segmentation, SegNet and U-Net were the most accurate with results of 95%+ (Veeragandham & Santhi, 2021). It also detailed the common challenges of classifying weed species using automated systems such as the varying light conditions. The study states that these problems can be overcome with different image pre-processing techniques.

Delving deeper into Segnet and U-Net, a study published in IEEE International Conference on Signal, Information and Data Processing (ICSIDP) investigated an application of image segmentation for the blind and compares the results of Segnet and U-Net. Their method found that for segmentation, U-Net achieved the better result of 84.32% accuracy against 70.54% with SegNet. Additionally, their system was fast, taking less than 0.5 seconds to segment any random image (Liu, Wang, & Zhao, 2019).

Additionally, Ronneberger et al. who first introduced U-Net, states that for segmentation, U-Net achieves very good performance on varying images when they investigated its use in neuronal structures and electron-microscopic recordings. They found that it achieved 92.03% accuracy taking less than one second to segment a 512x512 pixel image. (Ronneberger, Fischer, & Brox, 2015). The appeal of using U-Net was that it concatenates the features from encoding-decoding with higher resolution features from before max pooling. This allowed for contextual awareness when up-scaling.

III. METHOD

The general method of image segmentation and classification follows five steps: image acquisition, image pre-processing, image segmentation, feature extraction and classification (Vasavi, Punitha, & Rao, 2022). However, for this project, a deep neural network was used thus removing the need for manual feature extraction. As such, the steps covered in this section of the report include acquisition, pre-processing, segmentation, and classification.

The images used for this project were acquired as part of a dataset provided by project supervisor, Khamael Al-Dulaimi. Typically, images in a dataset are of different sizes and are randomly cropped and rescaled in pre-processing to a determined smaller size so that all images are of equal shapes. This was not the case with this dataset as all images were the same size. The images in this dataset did however require random separation into training and testing datasets with a split of 75% and 25% respectively. The training dataset was then split further to allow 10% of it to be used as validation. To decrease the volume of data and thus decrease training and testing time, images were downscaled. The effect of this is discussed later in this report. In the literature review it was discovered that one of the biggest obstacles in real-world application of automated classification is varying light conditions. This is often alleviated with image pre-processing with varied light conditions. However, since the scope of this project is contained to the images in the dataset, which are all taken under similar lighting conditions, this step was purposefully ignored here.

The dataset provided for this project also included markup files for each image. These markup files contained bounding box information for where in the image a weed was located. This was helpful when constructing image labels for the training of the neural network. A Python program was created that parsed all the annotation files, located bounding box information, and created a mask image with the information. An example output of this Python program can be seen below in Fig 3.1 next to its corresponding dataset image where red indicates weeds and green indicates background.

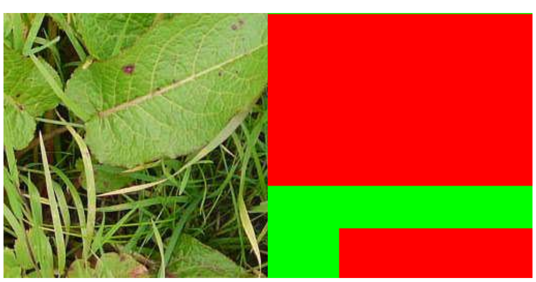

Fig 3.1 Image from dataset (left), output mask from Python program (right)

The mask was used to create a one hot encoded map for each image which was used as a ground truth label. This was because one hot maps are easily manipulated and rescaled. While this was not the most accurate method of creating ground truth labels, it was the most time-effective method. Creating accurate ground truth labels would have involved manually and precisely outlining each weed object in all 540 images in the dataset.

The segmentation model was built in Python using the TensorFlow library and Keras API. Following findings in the literature review, it was decided that the neural network be built based upon the U-Net architecture. This was so that it took features at a higher resolution and applied them with deeper convolutions. The architecture took an input frame and applied two 3x3 convolutions followed by a 2x2 max pooling operation. This process was repeated for a total of 4 times with the number of feature channels of the convolutions doubling after each max pooling. An up-sampling filter of 2x2 was then applied and concatenated with the feature map of the same size from before the max pooling operation. This was then followed by two more 3x3 convolutions and the process was repeated for four up-samples. Each filter application used a ReLU activation function. The final filter was a 1x1 convolution with a sigmoid activation function to determine the binary output of weed or not weed. A diagram of this architecture can be seen below in Fig 3.2 where the grey boxes are the feature maps, and the white boxes are the copied and concatenated feature maps from before the max pools.

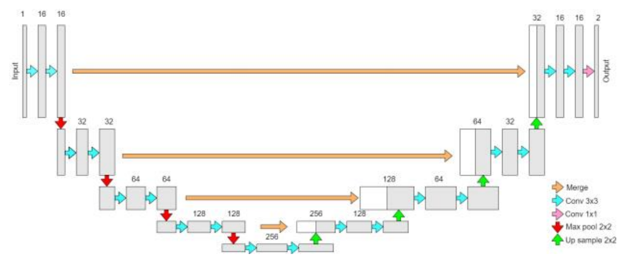

Fig 3.2 U-Net Architecture Adapted from (Ronneberger, Fischer, & Brox, 2015)

The 405 images from the training dataset were used to train and validate the neural network on this architecture. The model was then compiled with the Adam optimiser. Since the task was one of binary segmentation, a loss function of binary cross entropy was used when compiling. The model was initialised for training with an arbitrary number of 15 epochs. When fitting the model however, a call back function was created to combat the overfitting problem discussed in the literature review. This call back function monitored the validation loss at the end of each epoch and if the validation loss had not improved for more than 5 epochs then training would be stopped, and the previous feature weights would be restored.

Following training, the model was then evaluated with the testing dataset of 135 images. The results of this are discussed later in section VI. To visualise these results, the output feature maps for each image were converted back to masks.

IV. RISKS

Given this project focused solely on the software of a weed segmentation system, there were no physical risks involved. There were risks in terms of deliverable dependencies as well as software dependencies. The

software system was developed in the Python programming language with the TensorFlow library and the Keras API to interface. This meant that the software was dependent on the libraries importing correctly and the functions in the library working as expected. Python was used as it was a language that had been used by the student engineer previously and they were comfortable with. TensorFlow and Keras were chosen as they were well documented and had a large online support community of others that share their troubleshooting tips for issues that arose. This largely mitigated this risk.
The overall completion of the software system had a few dependencies on the completion of smaller software tasks. These tasks were given estimated completion times to maintain an appropriate completion trajectory and assess the overall state of the project. Additionally, weekly, or fortnightly meetings were conducted between the student engineer and the project supervisors to check in and allow for questions to be asked or feedback to be provided.

V. Ethics and Sustainability

Throughout this project, the four cores of ethics in engineering were consistently considered. The in-depth literature review meant that decisions made for this project were well-informed and made based on adequate knowledge. The literature review and development of the software system enabled continued learning as there was no single source that answered all questions.

Consistent meetings were held between the student engineer and the project supervisors. This allowed for effective and honest communication with the stakeholders of the project and gave opportunity for support for the student engineer.

The effect on sustainability of creating the software system described in this report was also considered prior to its development. Using a software system to identify weed species and precision spraying them holds up all three pillars of engineering sustainability. The theoretical system would reduce the cost of weed management and increase the yield of crops thus increasing profit. By applying precision spraying, the volume of herbicide that flows into the ground and then greater environment is also reduced, benefitting the environment. Finally, the physical labour of precision spraying large farms is reduced, benefiting the social aspect of sustainability. It could be said that implementation of an automated weeding system could potentially make farmers redundant and lead to a loss of jobs. However, the operation of these theorised systems would still require human upkeep, just less physically straining compared to hand spraying large farms.

The project was conducted with a clear conscience knowing that the overall impact would be positive and solve real-world issues.

VI. Results & Deliverables

The model described in section III was developed and trained with the training dataset of 405 images. This model resulted in a total of 1,962,642 parameters, all of which were trainable. Table 6.1 shows the training summary of the model as it trains.

Table 6.1 Training summary of CNN with input of 64x64

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.1374	0.7543	0.1267	0.8038
2	0.0962	0.8529	0.1354	0.7790
3	0.0881	0.8636	0.1320	0.8297
4	0.0771	0.8803	0.1249	0.8341
5	0.0646	0.9013	0.1354	0.8365
6	0.0580	0.9120	0.1455	0.8403
7	0.0449	0.9319	0.1942	0.8337
8	0.0373	0.9443	0.2701	0.8404
9	0.0274	0.9596	0.2746	0.8430

The model was then assessed under the criteria of accuracy and latency to assess the potential viability of implementing this model with live cameras as the input in real time. It was previously stated that the input image size could be scaled down to reduce computation and improve speed. As such, the model was tested with three different input sizes to compare accuracy and speed. Accuracy was measured by counting the number of times the predicted class matched the ground truth label and then dividing that by the total number of comparisons (TensorFlow, 2021). The latency of the system was calculated by deducting the time at the start of the predictions from the time at the end of the predictions. The testing and results in this section were generated solely on the testing dataset of 135 images. Furthermore, the training and testing of the CNN was done on a laptop with no GPU and an 11th Gen Intel Core i7-1165G7 CPU running at 2.80GHz. Table 6.2 below shows the results for each input image scaling.

Table 6.2 Accuracy and Latency of CNN model assessed with varying input image scaling

Input image size (pixel length & width)	Accuracy (%)	Time (ms)
64	80.56	304.255
128	83.09	796.657
256	86.54	1690.865

The results show that as images were downscaled less, i.e. the input image size was larger, the software took longer to make predictions with it taking up to 1.69 seconds to make predictions on the whole testing dataset when images were 256x256 pixels. This is still an acceptable latency on the criteria of implementing into a live camera feed as 1.69 seconds to make predictions on 135 images is 12.54ms per image. The results show that the larger input image led to higher accuracy with the 256x256 input resulting in the highest accuracy of 86.54%. The notion that increasing the input image size leads to higher accuracy was confirmed when comparing the results of the 64x64 images and 128x128 images which resulted in accuracies of 80.56% and 83.09%. Overall, the system achieved a high level of accuracy on all image sizes tested.

To visualise the predictions made by the CNN, the output can be overlayed over the original input image. This can be seen in Fig 6.1 below where the left shows the original image, and the right shows the prediction overlayed with red assigned to weed class and green assigned to not weed class.

Fig 6.1 CNN segmentation predictions of two images. Original images (left) & predictions overlayed with original image (right)

When observing the predictions, it was seen that the system often over predicts the boundary of the weeds and misclassifies some of the surrounding grass as a weed. Additionally, there were some images of grass with no weeds that had small predictions of weeds in the image. This is likely due to the ground truth labels used to train the system. Because the ground truth labels used bounding box rectangles to identify regions in the image where there was a weed and weeds are not rectangles, there were often bounding boxes that included regions of grass. As such, the system was trained to overestimate. These ground truth labels often being larger than the weed also effects the calculation of accuracy obtained above. This is because the calculation of accuracy compares the prediction to the ground truth label, which if the ground truth is not accurate, means the accuracy assessment isn't a precise assessment of accuracy.

As mentioned in the literature review and method, overfitting of a training model is a common issue when training CNNs. The method of stopping the training early if validation loss doesn't improve after a set number of epochs was applied to this model. This resulted in the model being trained for 9 epochs as opposed to 15 which was stated in the initialisation of the model fitting. This can be seen in table 6.1 above.

The deliverables of this project altered slightly between the project proposal and this progress report. The initial scope of this project included the creation of a software system that would use area thresholding and canny edge detection on the dataset to create label masks for each image. These masks would be used as ground truth labels for training and testing the segmentation system. When this was proposed, the existence of the annotation bounding box information was not known. Once this was discovered it didn't make sense to spend a large amount of time developing software that created masks when a much simpler program could be created that parsed the annotations and created these masks instead. An updated version of the deliverables is included in the appendix.

All other deliverables outlined in the initial scope have been delivered up to current date. This includes the literature review, the conceptual design, ground truth labels, the software for the segmentation system, the testing, and results.

VII. Limitation and Future Works

Analysing the results of the segmentation system highlighted opportunities for improvement in the future. As discussed above, the pixelwise accuracy of the ground truth labels was not great. This led to false positives of weed detection. In future, these ground truth labels could be created such that they are much more accurate and follow the actual shape of the weed as opposed to a rectangle. This would also help alleviate the issue of some weeds being partially obscured by grass in the image and the segmentation software predicting that the overlapping grass was weed.

It was stated in some articles in the literature review that data augmentation of the input images can increase the accuracy of a neural network. In future, this can be applied to the dataset used in this project to increase the volume of training data. This can be done through altering brightness, contrast, saturation, dimensions, and rotations of images.

Because the task of segmentation was binary in that there were only two classes, weed and not weed, the model was compiled with the binary cross entropy loss function. However, there are other loss functions that show promise in improving accuracy in cases where there is class imbalance. One such loss function that could be investigated is the focal loss function which down-weights easy examples and targets hard negatives.

VIII. Conclusion

To summarise, this project aimed create a software system capable of segmenting weeds from grass for potential use in a system of automatically detecting and precision spraying weeds. This was done due to the huge cost in terms of labour, monetary and impact on the environment that weeds pose in the agriculture industry. The entire project was conducted with engineering ethics and sustainability in mind.

The software system was created using convolutional neural networks to extract features from images and make decisions based upon unsupervised learning. The neural network was built upon the U-Net architecture as it was able to maintain contextual awareness after encoding by merging feature maps with those from before max pooling.

Results from testing the neural network show good accuracy and latency with all constraints of the system showing 80% accuracy. The highest accuracy came when the network was trained with the highest resolution input images of 256x256, resulting in 86.54% accuracy. Predictions did take longer when inputs were larger, however even at the highest resolution, predictions only took 1.69 seconds for 135 test images. Applying image rescaling to reduce image size did also decrease the time take to make predictions on the data, it did however come at the cost of accuracy. The software system was able to perform predictions at a high speed. As such, using a live camera feed as an input data and using the software system in real-time is viable given the results obtained in this report.

Overall, the U-Net architecture was successful, though some improvements could be made to how ground truth labels are created, or the loss function used to account for the false positives observed in the results. Furthermore, image augmentation can be investigated for future works to increase the size and variability of the training dataset.

The project successfully delivered the segmentation software system as well as other deliverables outlined in the project proposal. While there were some changes to the deliverables scope, the changes were made following meetings with the project supervisors and still resulted in the same expected outcome.

APPENDIX

Table 1 Updated project deliverables

#	Focus	Deliverable	Dependant	Milestone
1	Literature Review	Literature Review Report		1/5/2022
2	Conceptual design	Outline of methodology for segmentation and classification	1	1/5/2022
3	Ground truth label software	Python program	1,2	22/5/2022
4	Semantic segmentation software	Python program	1,2,3	5/6/2022
5	Test & validate	Accuracy, latency, visualisation of predictions	4	
6	Interim Report	Progress on software system, results, changes to scope	4,5	14/6/2022
7	Interim Presentation	Oral presentation of project so far and interim results	6	29/6/2022
8	Stakeholder consultation	Project trajectory analysis		21/6/2022
9	Classification software	Image segmentation and classification software	8	16/10/2022

ACKNOWLEDGMENTS

This project was made possible by the supervision and mentorship of Khamael Al-Dulaimi and Jasmine Banks. Additionally, the dataset of 540 images used for the training and testing of the neural network was provided with corresponding annotation markups by Khamael Al-Dulaimi.

IX. REFERENCES

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*(11).

Adil, T., Ahmed, G., Mohammed, B., & Soufiane, B. (2022). Weeds detection efficiency through different convolutional neural networks technology. *International Journal of Electrical and Computer Engineering, 12*(1), 1048-1055.

Alam, M. S., Alam, M., Tufail, M., Khan, M. U., Güneş, A., Salah, B., . . . Khan, M. T. (2022). TobSet: A New Tobacco Crop and Weeds Image Dataset and Its Utilization for Vision-Based Spraying by Agricultural Robots. *Applied Sciences, 12*(3).

Dave, V. S., & Dutta, K. (2014). Neural network based models for software effort estimation: a review. *Artificial Intelligence Review*, 295-307.

Grace, R. K., Anitha, J., Sivaramakrishnan, R., & Sivakumari, M. S. (2021). Crop and Weed Classification Using Deep Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 935-938.

Hlaing, S. H., & Khaing, A. S. (2014). Weed and Crop Segmentation and CLassifcation Using Area Thresholding. *International Journal of Research in Engineering and Technology, 3*, 375-382.

Kamath, R., Balachandra, M., Vardhan, A., & Maheshwari, U. (2022). Classification of paddy crop and weeds using semantic segmentaion. *Cogent Engineering, 9*, 3-15.

Li, Z., Yang, W., Peng, S., & Zhou, J. (2021). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 1-21.

Liu, L., Wang, Y., & Zhao, H. (2019). An Image Segmentation Method for the blind sidewalks recognition by using the convolutional neural network U-net. *IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*.

McLeod, R. (2018). *Annual Costs of Weeds in Australia.* Canberra: Centre for Invasive Species Solutions.

Narasinga Rao, M., Venkatesh Prasad, V., Sai Teja, P., Zindavali, M., & Phanindra Reddy, O. (2018). A survey on prevention of overfitting in convolution neural networks using machine learning techniques. *International Journal of Engineering and Technology (UAE)*, 177-180.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 234-241.

Shiruru, K. (2016). An Introduction to Artificial Neural Network. *International Journal Of Advance Research And Innovative Ideas In Education*, 27-30.

Sugar Research Australia. (2022). *Reducing herbicide usage on sugarcane farms in reef catchment areas with precise robotic weed control*. Retrieved 4 15, 2022, from https://sugarresearch.com.au/current-research-projects/

Talaviya, T., Shah, D., Patel, N., Yagnik, H., & Shah, M. (2020). Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artificial Intelligence in Agriculture, 4*, 58-73.

Tao, T., & Wei, X. (2022). A hybrid CNN–SVM classifer for weed recognition in winter rape field. *Plant Methods, 18*(29).

TensorFlow. (2021, 4 22). *TensorFlow Core v2.9.1*. Retrieved from TensorFlow: https://www.tensorflow.org/api_docs/python/tf/keras/metrics/Accuracy

Vasavi, P., Punitha, A., & Rao, T. V. (2022). Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: a review. *International Journal of Electrical and Computer Engineering (IJECE)*, 2079-2086.

Veeragandham, S., & Santhi, H. (2021). A Detailed Review on Challenges and Imperatives of Various CNN Algorithms in Weed Detection. *International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 1068-1073.

Wen, J., Yihui, R., Ying, L., & Jiaxu, L. (2022). Artificial Neural Networks and Deep Learning Techniques Applied to Radar Target Detection: A Review. *Electronics (Switzerland), 11*(1).

Yu, R., Wang, Y., Zou, Z., & Wang, L. (2020). Convolutional neural networks with refined loss functions for the real-time crash risk analysis. *Transportation Research Part C*.