Summer 7-24-2019

Mid to Late Season Weed Detection in Soybean Production Fields Using Unmanned Aerial Vehicle and Machine Learning

Arun Narenthiran Veeranampalayam Sivakumar
University of Nebraska - Lincoln, arun-narenthiran@huskers.unl.edu

Veeranampalayam Sivakumar, Arun Narenthiran, "Mid to Late Season Weed Detection in Soybean Production Fields Using Unmanned Aerial Vehicle and Machine Learning" (2019). *Biological Systems Engineering--Dissertations, Theses, and Student Research*. 91.
https://digitalcommons.unl.edu/biosysengdiss/91

MID TO LATE SEASON WEED DETECTION IN SOYBEAN PRODUCTION FIELDS USING UNMANNED AERIAL VEHICLE AND MACHINE LEARNING

by

Arun Narenthiran Veeranampalayam Sivakumar

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Agricultural and Biological Systems Engineering

Under the Supervision of Professor Yeyin Shi

Lincoln, Nebraska

August, 2019

MID TO LATE SEASON WEED DETECTION IN SOYBEAN PRODUCTION FIELDS USING UNMANNED AERIAL VEHICLE AND MACHINE LEARNING

Arun Narenthiran Veeranampalayam Sivakumar, M.S.

University of Nebraska, 2019

Advisor: Yeyin Shi

Mid to late season weeds are those that escape the early season herbicide applications and those that emerge late in the season. They might not affect the crop yield, but if uncontrolled, will produce a large number of seeds causing problems in the subsequent years. In this study, high-resolution aerial imagery of mid-season weeds in soybean fields was captured using an unmanned aerial vehicle (UAV) and the performance of two different automated weed detection approaches – patch-based classification and object detection was studied for site-specific weed management. For the patch-based classification approach, several conventional machine learning models on Haralick texture features were compared with the Mobilenet v2 based convolutional neural network (CNN) model for their classification performance. The results showed that the CNN model had the best classification performance for individual patches. Two different image slicing approaches – patches with and without overlap were tested, and it was found that slicing with overlap leads to improved weed detection but with higher inference time. For the object detection approach, two models with different network architectures, namely Faster RCNN and SSD were evaluated and compared. It was found that Faster RCNN had better overall weed detection performance than the SSD with similar inference time. Also, it was found that Faster RCNN had better detection performance and shorter inference time compared to the patch-based CNN with overlapping image slicing. The influence of spatial resolution on weed detection accuracy was investigated by simulating the UAV imagery captured at different altitudes. It was found that Faster RCNN achieves similar performance at a lower spatial resolution. The inference time of Faster RCNN was evaluated using a regular laptop. The results showed the potential of on-farm near real-time weed detection in soybean production fields by capturing UAV imagery with lesser overlap and processing them with a pre-trained deep learning model, such as Faster RCNN, in regular laptops and mobile devices.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Yeyin Shi for her encouragement and support throughout my Master’s program. She motivated and challenged me and gave me the freedom to work on my interest in the application of machine learning in agriculture. I also would like to thank my committee members Dr. Stephen Scott, Dr. Joe Luck and Dr. Amit Jhala for their time to serve on my committee and review this long document and providing invaluable feedback for the successful completion of this project. Also, I would like to acknowledge my teammates Jiating Li and Lin Wang for their great support, Jon Forbes for his help in data collection and Suresh Thapa for his great contribution to Chapter 2. Finally, I would like to thank my family - my parents for their numerous sacrifices to give me the best education possible since childhood and my sister for her support all the time. Also, I would like to thank my significant other, Gowthami Venkateswaran, for being with me during my highs and lows in these two years and always motivating me.

TABLE OF CONTENTS

LIST OF MULTIMEDIA OBJECTS

CHAPTER 1 INTRODUCTION

The world population, currently 7.6 billion, is expected to reach more than 9 billion by 2050. With an increase in gross domestic product (GDP) per capita of the developing countries, the food consumption per capita is also expected to grow (Tilman, Balzer, Hill, & Befort, 2011). In order to secure food for growing population, global food production should almost double by 2050. Hence, in addition to breeding higher yielding varieties of crops, it is necessary to address the various factors contributing to yield loss such as nutrients, water, weeds, insects, and diseases. Weeds are unwanted plants that grow in the field and compete with crops for resources, thereby suppressing the growth of the crop and thereby the yield. Typically, weeds are controlled by the application of pre-emergence and post-emergence herbicides at uniform rates throughout the field, which often leads to overuse resulting in environmental impacts, economic loss, and evolution of herbicide-resistant weeds. However, the spatial distribution of weeds is not uniform across the field but rather is found to occur in patches. Hence, the approach of site specific weed management was proposed in which the weeds are sensed and spot sprayed using variable rate applicators (CHRISTENSEN et al., 2009).

Earlier studies on site specific weed management focused on real time sensing and spraying systems. They were limited in computational power and so were slow in operation thereby not being able to cover large areas. Remote sensing imagery from satellite covering large areas was proposed as a solution but was limited in the resolution of the imagery (Thorp & Tian, 2004). Unmanned aerial vehicles (UAVs) with their ability to get very high- resolution aerial imagery and cover large areas proved to be a better alternative. Because of these advantages they have been used for several applications in agriculture such as water stress detection, disease detection, nitrogen management, weed detection, high throughput phenotyping (Sankaran et al., 2015; Shi et al., 2016). Advances in the field of machine learning in the past decade has led to various applications of machine learning algorithms and convolutional neural networks (CNN) for applications in agriculture including weed detection. However, most of these studies are focused on early season weed detection since it is the critical period for weed removal. Mid to late season weeds are those escaped the early season herbicide application or those that emerged late in the season. Even though they might not affect the crop yield in that season, if uncontrolled, will produce large number of seeds thereby creating seedbank and causing problems in the future. With limited herbicide options available for application late in the season, automated detection of these mid to late season weeds will enable farmers to act quickly to control them. Therefore, in this study, the focus is on studying the use of patch-classification and object detection approaches to detect weeds from UAV imagery and evaluate the feasibility of on-farm near-real time detection using regular laptop in commercial scale soybean fields. The specific objectives were:

1. To evaluate and compare conventional machine learning models with a deep convolutional neural network model on patch-classification in terms of the detection performance and the inference time
2. To evaluate and compare the detection performance and inference time of object detection and patch-based classification deep learning model

3. To evaluate the operational feasibility of object detection and patch-based classification deep learning models for near real time weed detection using UAVs in commercial scale soybean fields

This thesis has been organized as five chapters with Chapter 1 and Chapter 5 being the introduction and conclusion. The studies corresponding to the three specific objectives mentioned above have been prepared as three manuscripts to be submitted to journals and named as Chapter 2, Chapter 3 and Chapter 4 respectively.

1.1 REFERENCES

CHRISTENSEN, S., SØGAARD, H. T., KUDSK, P., NØRREMARK, M., LUND, I., NADIMI, E. S., & JØRGENSEN, R. (2009). Site-specific weed control technologies. Weed Research, 49(3), 233–241. https://doi.org/10.1111/j.1365-3180.2009.00696.x

Sankaran, S., Khot, L. R., Espinoza, C. Z., Jarolmasjed, S., Sathuvalli, V. R., Vandemark, G. J., … Pavek, M. J. (2015). Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review. European Journal of Agronomy, 70, 112–123. https://doi.org/10.1016/J.EJA.2015.07.004

Shi, Y., Thomasson, J. A., Murray, S. C., Pugh, N. A., Rooney, W. L., Shafian, S., … Yang, C. (2016). Unmanned Aerial Vehicles for High-Throughput Phenotyping and Agronomic Research. PLOS ONE, 11(7), e0159781.

https://doi.org/10.1371/journal.pone.0159781

Thorp, K. R., & Tian, L. F. (2004). A Review on Remote Sensing of Weeds in Agriculture. Precision Agriculture, 5(5), 477–508. https://doi.org/10.1007/s11119-004-5321-1

Tilman, D., Balzer, C., Hill, J., & Befort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. Proceedings of the National Academy of Sciences of the United States of America, 108(50), 20260–20264. https://doi.org/10.1073/pnas.1116437108

CHAPTER 2 CONVENTIONAL MACHINE LEARNING OR DEEP LEARNING: WHICH IS BETTER FOR PATCH-BASED MID TO LATE SEASON WEED DETECTION IN UAV IMAGERY?

This manuscript has been prepared for journal submission

2.1 INTRODUCTION

Weeds are one of the most important factors contributing to yield loss in crops (CENTENARY REVIEW, 2019). There are two ways of control to combat weeds, namely mechanical and chemical, of which chemical application is the most common practice. Use of chemicals such as herbicides after the emergence of weeds is the most common approach in weed management. Typically the herbicides are applied at a constant rate across the field, which has negative consequences such as economic loss, impacts on environment and increase in the evolution of herbicide resistant weeds due to overuse. However, the spatial distribution of weeds is not uniform across the field as they have been found to occur in patches. With the availability of remote sensing and global positioning systems for civilian applications, site-specific application was proposed as a core of precision farming that accounts for the spatial variability in the field to increase productivity and minimize environmental footprints (Zhang, Wang, & Wang, 2002). In weed management, this led to the site-specific herbicide application in which the herbicides are spot sprayed only where the weeds are using a sensing, decision making and variable-rate application system (CHRISTENSEN et al., 2009; Weis et al., 2008).

Earlier studies on weed sensing focused on extracting different features from digital images to distinguish between the crops and different weed species. Color based indices were found to be effective in distinguishing the plant pixels from the soil background but were difficult to differentiate between the crops and the weeds. Texture features that capture the spatial variation of pixel intensities as well as shape features such as roundness, perimeter among others, were able to distinguish between broadleaf and grassy plants. However, they were not able to differentiate individual species of weeds (D. M. Woebbecke, G. E. Meyer, K. Von Bargen, & D. A. Mortensen, 1995a, 1995b; G. E. Meyer, T. Mehta, M. F. Kocher, D. A. Mortensen, & A. Samal, 1998). Following this, with advancements in variable rate implements, several studies focused on site-specific spraying. These systems used image processing techniques like discrete wavelet transform as well as nonlinear classifiers to recognize the weeds. Besides, these systems used the information about crop row to detect the inter-row weeds. However, the speed of these systems was limited by the computational capacity of the hardware (L. Tian, J. F. Reid, & J. W. Hummel, 1999; W. S. Lee, Slaughter, & Giles, 1999). Thorp & Tian (2004) studied the potential of satellite and manned aircraft-based remote sensing technologies to locate the weed patches. Using aerial imagery enables sensing large areas to develop a prescription map for herbicide application which can then be used by variable rate applicators without having to sense weeds real time on the ground. The low spatial resolution, the occurrence of mixed-image pixels and the similar spectral nature of a lot of weeds and the crops were the major difficulties with using satellite imagery.

Unmanned aerial vehicles (UAVs) with their ability to obtain ultra-high-resolution aerial imagery at centimeter scales helped overcome the limitation of resolution in satellite imagery and enabled various applications in precision farming such as weed, disease and pest detections (Sankaran et al., 2015). Most of the early studies on weed detection using UAV based aerial imagery was in sunflower fields. Following that, several researchers have investigated different weed detection algorithms from aerial imagery in different crops such as maize, sugarcane, and sugar beets. The potential of using only color-based indices from multispectral imagery to segment the weed pixels from crop and soil pixels was studied, but it was not found to be as effective because of spectral similarity of crops and weeds (Torres-Sánchez, López-Granados, De Castro, & Peña-Barragán, 2013). Following this, object-based image analysis (OBIA) was the most widely studied technique to detect early season weeds. In this approach, the image was converted into objects with spatially and spectrally homogenous pixels after which the vegetation objects were segmented from the soil using color indices such as Excess Green index and Normalized Difference Vegetation Index. The crop rows were then found using the orientation of the largest object and the inter-row vegetation objects were classified as by masking the crop rows (López-Granados et al., 2016; J. M. Peña, Torres-Sánchez, de Castro, Kelly, & López-Granados, 2013; J. Peña et al., 2015). But the limitation of OBIA approach is the inability to detect the weeds in the crop row and the tuning of parameters needed to optimally segment the objects.

Advances in the field of machine learning have led to improvements in precision farming applications. As an alternative to OBIA, Hough transform was used to find crop rows and machine learning based classifiers were then used to classify small patches using spectral features and their relative position to crop rows (Perez-Ortiz et al., 2016; Pérez-Ortiz et al., 2015). With the availability compact hyperspectral cameras for UAV systems, the potential of pixel level classification using machine learning approaches for weed detection from hyperspectral images was studied (Gao, Nuyttens, Lootens, He, & Pieters, 2018; Koot, 2014; Yano et al., 2017). Various machine learning classifiers such as support vector machines, artificial neural networks, random forest were used in these studies. However, the very high cost of hyperspectral cameras and complexity in their data processing currently limit their adoption in commercial applications. In the past several years, convolutional neural networks (CNNs) have revolutionized computer vision. Various researchers in the recent years have studied the application of CNNs for pixel wise classification of UAV images into weed, crop and soil for precision weed management (Huang et al., 2018; Lottes, Khanna, Pfeifer, Siegwart, & Stachniss, 2017; Sa et al., 2018).

It is to be noted that all these studies focus on early season weed detection for site specific application of post emergence herbicide. In addition to early season weeds, mid and late season weeds pose several problems to the farmers. Mid and late season weeds refer to those that escaped the post emergence herbicide application and those emerged late in the season. Though they may not affect the yield of the crop, if left uncontrolled, will produce large amounts of seeds thereby creating a seedbank to cause problems for years in the future. Unlike tall crops like maize and sorghum in which late season weeds are hidden under the canopy and cannot be seen from aerial imagery, soybean being a short stature crop, aerial imagery can be used for this problem. The major challenge in the detection of late season weeds in the imagery is that the crop and weed objects are overlapping with each other in a cluttered manner thus limiting the performance of the segmentation algorithms used in OBIA. For practical applications in production agriculture, a RGB camera based solution is preferred over others due to the low cost and convenience in data processing.

2.1.1 Objective

Develop a patch-based method for automatic mid to late season weed detection from UAV imagery in soybean after canopy closure and compare between conventional machine learning and convolutional neural network (CNN) based classifiers for detection performance and speed.

2.1.2 Specific objectives

1. Tuning of Gray Level Co-Occurrence Matrix based feature extraction parameters
2. Comparison between conventional machine learning models such as support vector machine, logistic regression, artificial neural networks, and k-nearest neighbors, and CNN deep learning on their patch classification performance using accuracy, precision, recall and f1 score as evaluation metrics and their inference time
3. Comparison between overlap and non-overlap image division methods for testing using Intersection over Union (IoU) as the evaluation metric

2.2 MATERIALS AND METHODS

2.2.1 Study site

The study site is located in South Central Agricultural Laboratory of the University of Nebraska, Lincoln at Clay Center, NE, USA (40.575188, -98.130909). The study area consisted of soybean weed management research plots. Figure 2.1 shows the study area.

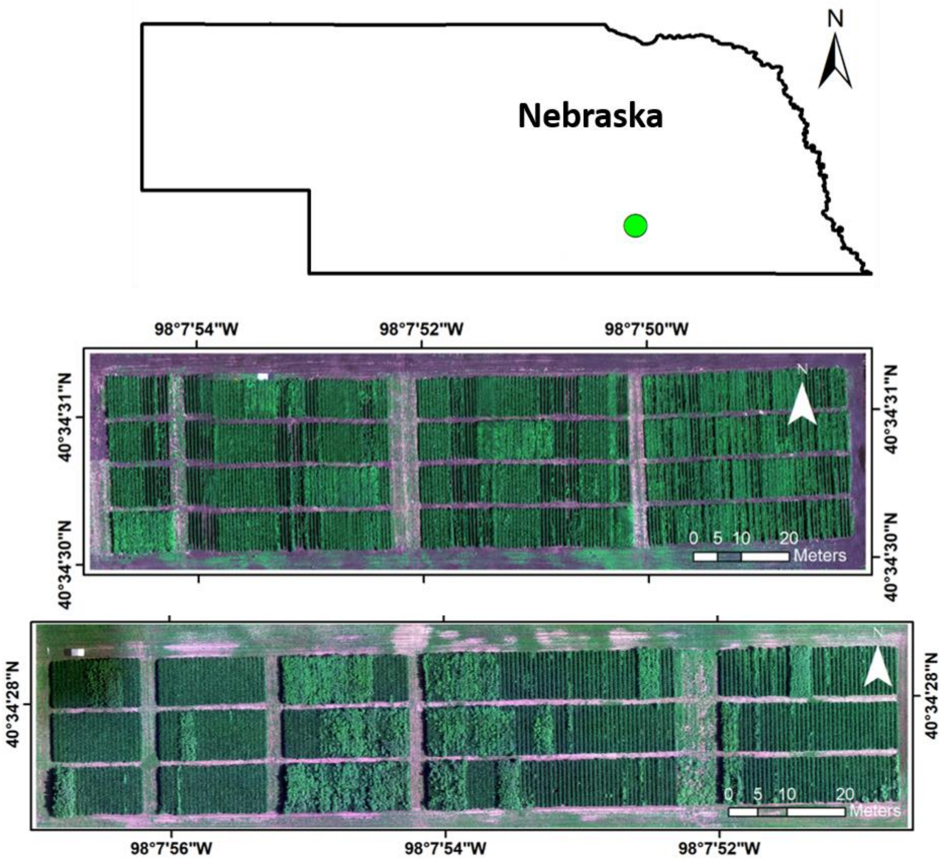


Figure 2.1. Study area at South Central Agricultural Laboratory in Clay Center, NE

2.2.2 UAV data collection

A DJI Matrice 600 pro UAV (DJI, Shenzhen, China) with a 16 megapixels (4608 × 3456) RGB camera (Zenmuse X5R, DJI, Shenzhen, China) (Figure 2.2) was used to capture aerial imagery of soybean fields with weeds. The data collections were conducted late morning on two dates: July 2nd, 2018, in the north field, and July 12th, 2018, in the south field. This resulted in variability in illumination during data collection as well as the growth stage of weeds in the data, thereby adding to the robustness of the classification models. DJI Ground Station pro application was used to plan the flight mission, and the images were obtained at an altitude of 20 m above ground level with 90 % forward overlap and 85% side overlap. This resulted in a spatial resolution of 0.5 cm/pixel.

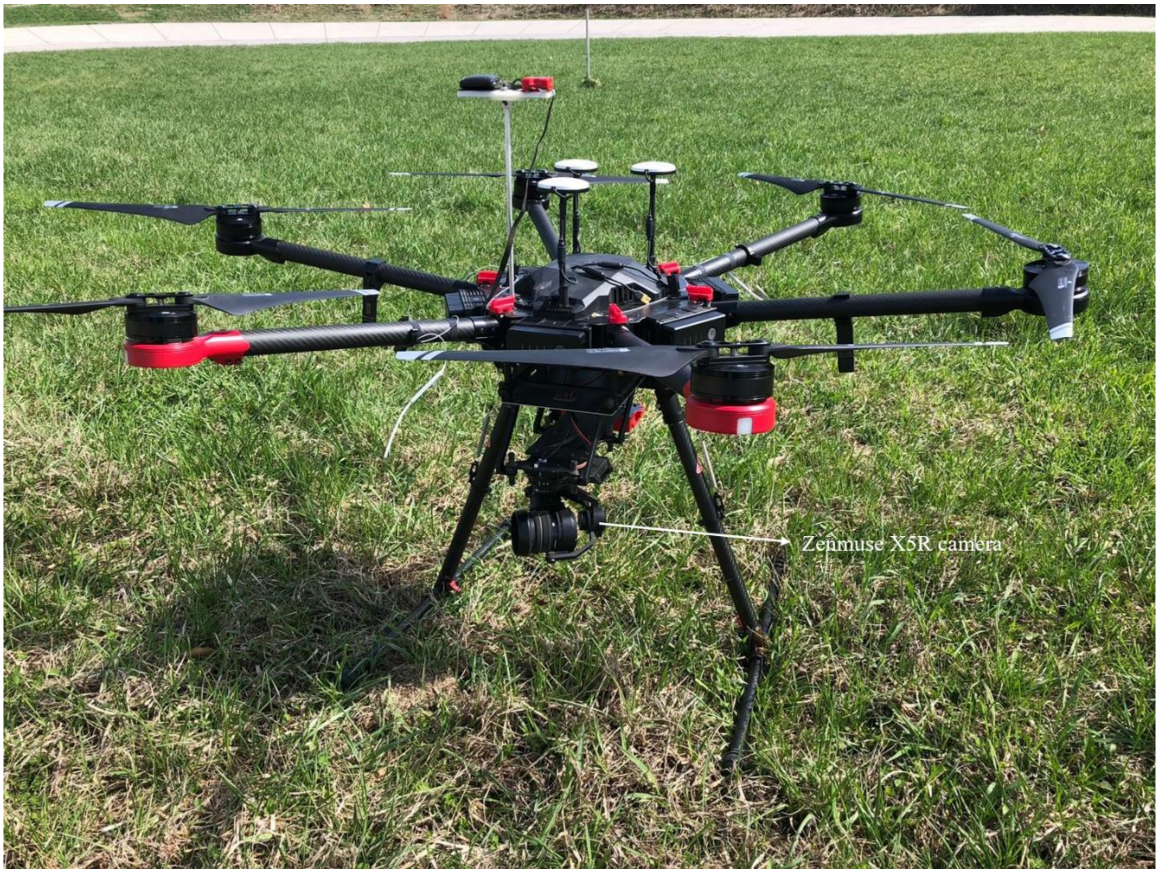


Figure 2.2. The DJI Matrice 600 pro UAV platform with Zenmuse X5R sensor used in this study.

2.2.3 Data annotation and preprocessing

The images obtained from the UAV was preprocessed, the weed areas annotated and the patch dataset was created as follows. Figure 2.3 shows a flowchart explaining the methodology used in this study from the preprocessing of raw images to the evaluation of results. From the obtained dataset, the overlapping raw images were removed to exclude duplicate data. Since the original size of the raw image of 4608 × 3456 pixels is too large to fit in memory, each 16 MP raw image was sliced into 12 sub-images of size 1152 × 1152 pixels. After this process, the dataset contained 450 images of size 1152 × 1152. The image annotation tool LabelImg (Tzutalin, 2015) was used to draw bounding boxes on the areas containing weeds in the images. From these images with annotations, bounding box areas containing weeds were cropped out. The cropped out weed areas of varying sizes were then further cropped into a maximum possible number of patches of size 128 × 128 to form the ‘weed’ class. Then, the areas remained in the original image, i.e., soybean and soil areas were cropped into the maximum possible number of small patches of size 128 × 128 to form the ‘background’ class. Both classes were randomly split into 90% as training data and 10% as test data.

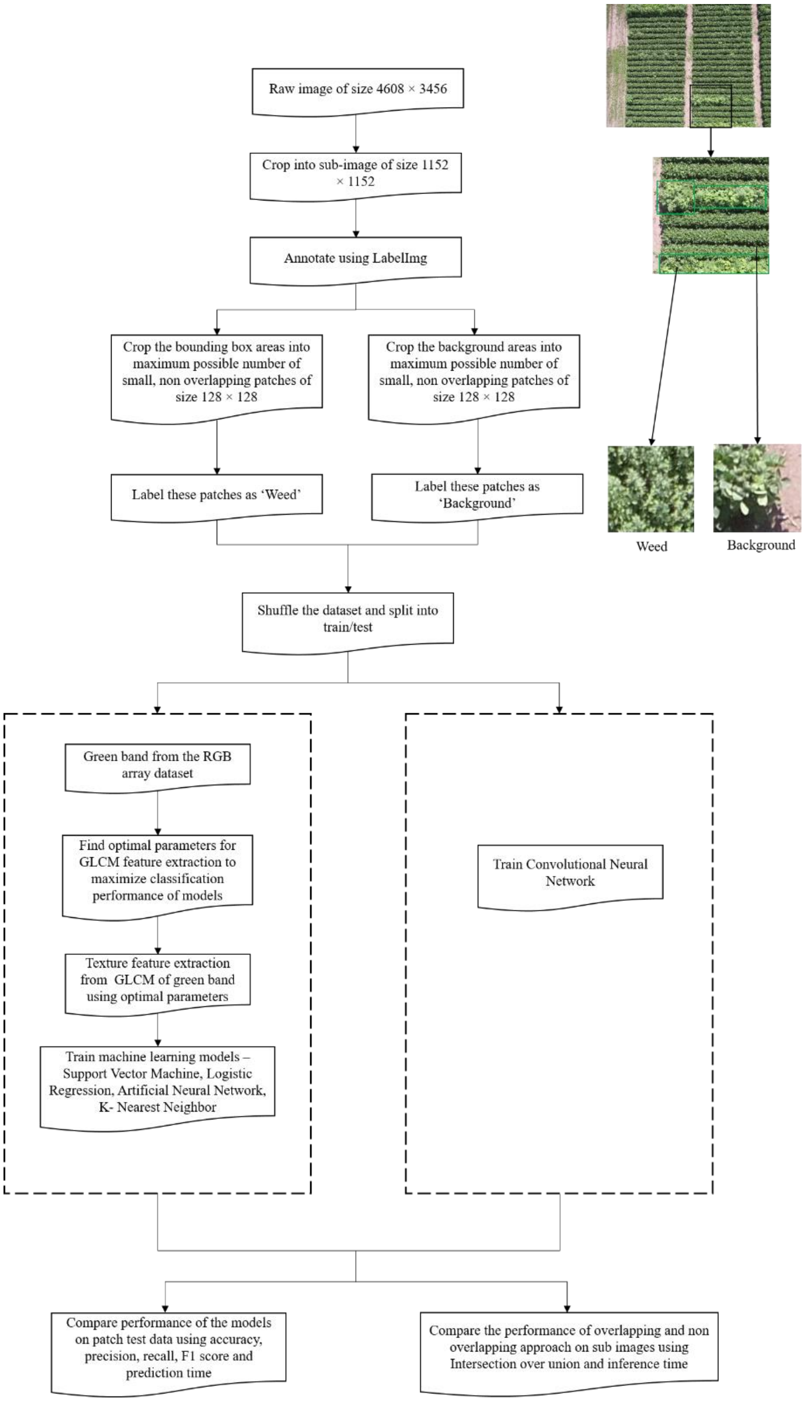

Raw image of size 4608 × 3456
Crop into sub-image of size 1152 × 1152
Annotate using LabelImg
Crop the bounding box areas into maximum possible number of small, non overlapping patches of size 128 × 128
Crop the background areas into maximum possible number of small, non overlapping patches of size 128 × 128
Label these patches as 'Weed'
Label these patches as 'Background'
Weed
Background
Shuffle the dataset and split into train/test
Green band from the RGB array dataset
Find optimal parameters for GLCM feature extraction to maximize classification performance of models
Texture feature extraction from GLCM of green band using optimal parameters
Train machine learning models – Support Vector Machine, Logistic Regression, Artificial Neural Network, K- Nearest Neighbor
Train Convolutional Neural Network
Compare performance of the models on patch test data using accuracy, precision, recall, F1 score and prediction time
Compare the performance of overlapping and non overlapping approach on sub images using Intersection over union and inference time

Figure 2.3. Flowchart showing data annotation, feature extraction, training of different machine learning models and convolutional neural network, comparison of the performance of these models and evaluation approaches

2.2.4 Feature extraction using Gray Level Co-occurrence Matrix for conventional machine learning models

In case of image classification using machine learning methods, feature extraction is one of the most important steps. Color, texture, and shape features are most widely used in agricultural applications. Since the data was collected after soybean canopy closure, there was a minimum amount of soil pixels presented between crop rows. The major classes in this study were the weed and the soybean canopy. It can be seen from the histogram of the two classes (Figure 2.4) that color features merely were not helpful enough to discriminate between the two classes. However, texture features extracted from Gray Level Co-Occurrence Matrix (GLCM) has proven successful in early season weed detection and plant species identification in several crops (Ahmed, Al-Mamun, Bari, Hossain, & Kwan, 2012; G. E. Meyer et al., 1998; HAWARI Ghazali, Mustafa, Hussain, Hawari Ghazali, & Marzuki Mustafa, 2007; Pulido Rojas, Solaque Guzmán, & Velasco Toledo, 2017; Wu & Wen, 2009). With only green band showing slight changes in intensity between the crop and weed pixels only the green band of the patches were used for texture feature extraction.

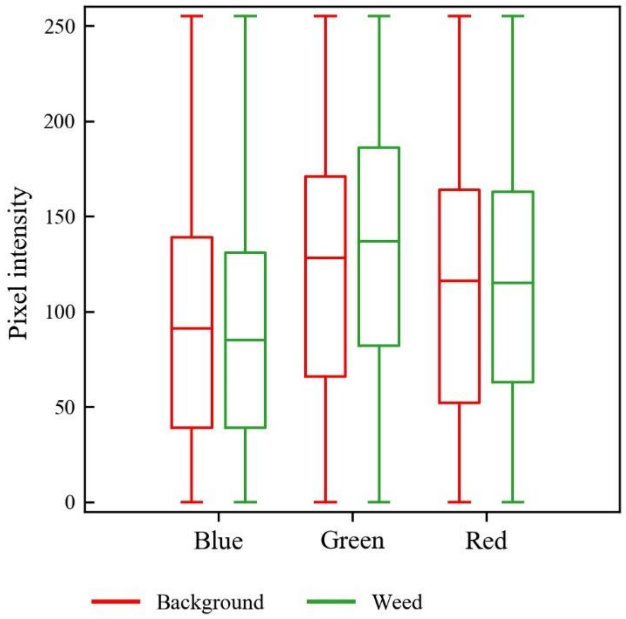
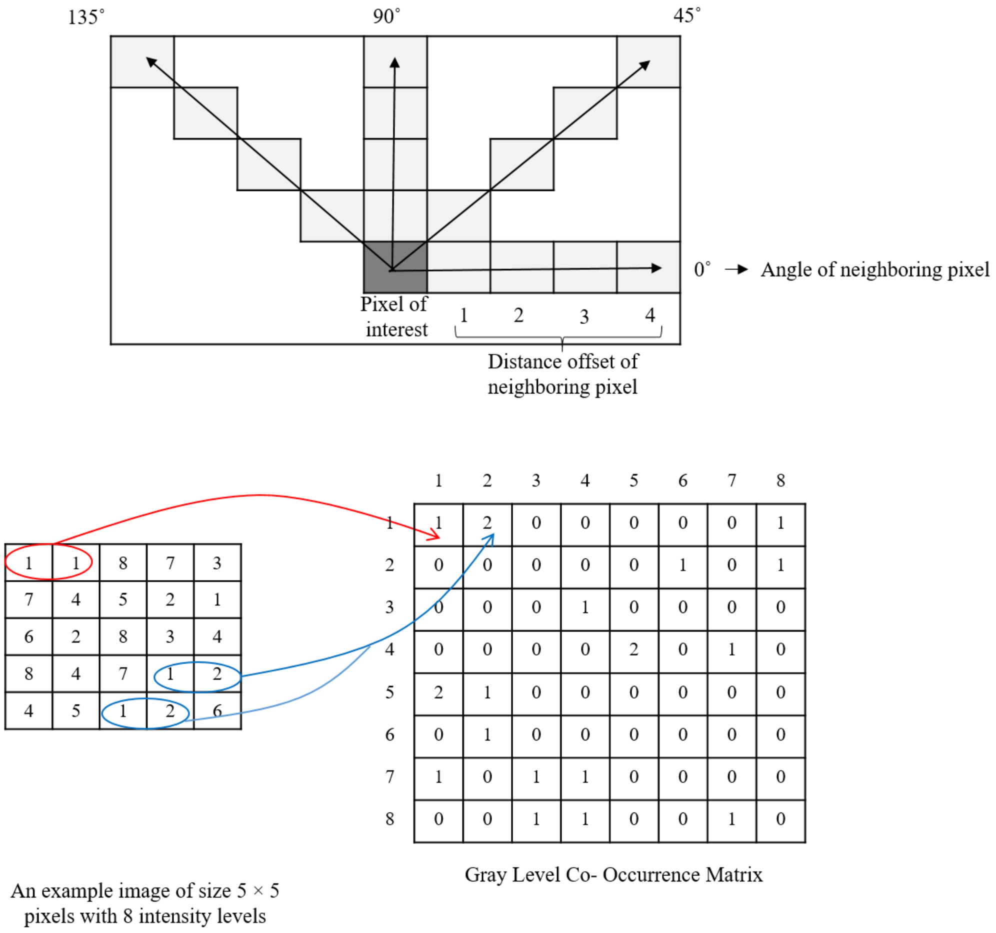


Figure 2.4. Boxplot of the distribution of the “background” class including soybean and soil pixels and the “weed” class in blue, green and red bands.

Gray level co-occurrence matrix (GLCM) is a statistical measure of the pixel intensity distribution in an image. In order to measure the variation in texture in the image, rather than individual pixel intensities, intensities of a pair of pixels defined by two parameters – distance offset *d* and angle *Θ* are measured (Chapter 7. Texture analysis, 2017). The size of the Co-occurrence matrix depends on the number of gray levels at which the image intensities are measured. Hence, GLCM of n gray levels maps an image of size $i \times j$ into a matrix of size $n \times n$ which represents the frequency of occurrence of each pair of gray levels (Figure 2.5). To represent the textural properties of an image with few numbers, (Haralick, Shanmugam, & Dinstein, 1973) proposed 14 features of which the following 6 – Contrast, Dissimilarity, Homogeneity, ASM, Energy, Correlation were calculated using scikit-image, an image processing package in python (van der Walt et al., 2014).


135˚
90˚
45˚
0˚ → Angle of neighboring pixel
Pixel of interest
1
2
3
4
Distance offset of neighboring pixel
An example image of size 5 × 5 pixels with 8 intensity levels
Gray Level Co- Occurrence Matrix

1	1	8	7	3
7	4	5	2	1
6	2	8	3	4
8	4	7	1	2
4	5	1	2	6

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	0	0	1
2	0	0	0	0	0	1	0	1
3	0	0	0	1	0	0	0	0
4	0	0	0	0	2	0	1	0
5	2	1	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0
7	1	0	1	1	0	0	0	0
8	0	0	1	1	0	0	1	0

Figure 2.5. Example showing the calculation of Gray Level Co-Occurrence Matrix at an angle of 0°, distance offset 1 and 8 intensity levels for an image of size 5×5

Feature engineering greatly influences the performance of a machine learning classifier (Domingos, 2012). In addition to choosing the right feature extraction method, it is necessary to find the optimal parameters for that method for each problem to maximize the performance of the classifier. In case of feature extraction using GLCM, distance offset and angle are the two parameters that determine the number of co-occurrence matrices created. Previous studies have either looked at using only the neighboring pixel (a distance offset of 1) at multiple directions or average of various distance offsets and directions (S. N. Ondimu & H. Murase, 2008; T. F. Burks, S. A. Shearer, & F. A. Payne, 2000; Y. K. Chang et al., 2012). Based on visual observation of the patches, it was decided to use distance offsets in the range 1-5 and four different angles - 0°, 45°, 90°, and 135° thereby resulting in 20 matrices in total. Since 6 different features are extracted from each matrix, the dataset had 120 features in total.

In order to find the influence of distance offset and direction on the classification performance, experiments were conducted with 4 different combinations of features:1) Averaging across all distance offsets and directions resulting in 6 features, 2) Retaining all 120 features, 3) Averaging across all directions but retaining all distance offsets resulting in 30 features and 4) Averaging across all distance offsets but retaining all directions resulting in 24 features. Following this, with the best combination of feature, experiments were conducted with a different number of distance offsets to find the optimal value. The same procedure was repeated to find the optimal number of gray levels in the GLCM. Classification accuracy and computation time for feature extraction were used as the evaluation metrics in all these experiments.

2.2.5 Conventional Machine Learning models

The hypothesis of this study was that the dataset is linearly separable in using Haralick features from GLCM and hence logistic regression was the first machine learning model that was investigated. Though logistic regression model had a very good classification accuracy of 93.32, potential for further improvement in classification accuracy and other evaluation metrics (as can be seen in the results) indicated that there might be slight non linearity existing in this dataset and hence 3 different models capable of learning non-linear decision boundaries – support vector machine (SVM) with Gaussian kernel or radial basis function (RBF), k-nearest neighbor and artificial neural networks were studied. During the training of each model, hyper parameter tuning was done using k-fold cross validation with different sets of hyper parameters to obtain the optimal hyper parameters. The best performing models with optimal hyper parameters in each case were then compared within them as well as with convolutional neural network using different evaluation metrics.

Logistic regression is a linear binary classifier and a probabilistic discriminative model where the model learns a mapping function to directly output the posterior class probability distribution without modeling the likelihood function of the features. It is widely used for classification tasks in case of linearly separable data. In case of non-linear data, it can be used by augmenting the features to a high dimensional space but the time complexity suffers from the curse of dimensionality and hence is not suitable for non-linear data with a large number of dimensions (Bishop, 2006).

Support vector machine (SVM) is a linear binary classifier, which constructs a hyperplane to linearly separate the data in the feature dimension space. The objective of SVM is to maximize the width of the margin between the two classes thereby leading to low generalization error. The kernel trick in SVM allows us to augment the features to high dimensional polynomial features or Gaussian similarity features as in Radial Basis Function kernel. This property enables SVM to classify nonlinear data by augmenting the features to higher dimensions. The major difference between SVM and logistic regression is that in logistic regression the objective is to find a decision boundary, which would correctly classifies all the training data whereas in case of SVM the objective is to maximize the width of the decision margin with the constraint of correctly classifying all the training data. In addition, the kernel trick enables SVM to be used in case of nonlinear data with a large number of features. (Bishop, 2006).

K-nearest neighbor (KNN) is an analogy based machine learning algorithm that is non-parametric. Its advantage is its ability to learn highly complex decision boundary. It works by finding the similarity of a test point with all the training data points and assigns the class of the test point based on the class of the k-nearest neighbors. It suffers from overfitting. Also, since all the training data has to be loaded into the memory to make a prediction for every test data point, it is computationally expensive but there are some ways to increase the speed. The other disadvantages are the optimal of parameter *k* and the appropriate distance metric has to be found for each problem (Bishop, 2006).

Artificial neural networks (ANN) mimic the human biological neuron in the structure and are highly non-linear in nature. They have units called neurons similar neurons in the human brain and have weights associated with the connections between the neuron. At the neuron, the weighted sum of signals from all neurons in previous layers is done followed by an activation function which maps the weighted sum to a non-linear output. These layers, other than the input and output layer, are called hidden layers. At the end, there is an output layer the outputs the class label in case of classification or prediction in case of regression problem in which a real valued number is needed as output (Bishop, 2006).

2.2.6 Convolutional neural network model

Even though artificial neural networks with several hidden layers are very good in learning highly non-linear decision boundaries, in case of data with multiple arrays such as images and audio signals, their performance is limited by the information contained in the extracted features. In case of no feature extraction, artificial neural networks use the pixel intensities of images in different bands as features. However, since the spatial correlation of pixels in an image is not considered in such approach, the performance is limited. To overcome this limitation, convolutional neural networks (CNNs) were proposed. Most of the fully connected hidden layers in ANN were replaced by sliding windows or convolutional layers that learn spatial features in the image. These sliding windows are called feature maps. Similar to ANN, the output from the feature maps from each hidden layer is mapped to a non-linear space by using activation functions. The abstraction is that the feature maps in the earlier hidden layers learn generalized features such as textural features whereas feature maps in the hidden layers at the end learn task-specific features. They are trained using the backpropagation algorithm similar to ANN.

However, because of the large number of parameters, they need a large amount of data to train a CNN from scratch (Krizhevsky, Sutskever, & Hinton, 2012; LeCun, Bengio, & Hinton, 2015).

To overcome this limitation, the transfer learning approach was proposed. In this case, the weights and graphs of CNN trained on large image datasets for another task were used to initialize the weights. In this case, CNN converged faster for smaller datasets. In this study, MobileNet v2 model was used for transfer learning (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). It has been trained on the ImageNet dataset with 1.4 million images of 1000 classes. In the first 10 epochs, the convolutional layers in the model were frozen and only the fully connected layer was trained to distinguish weed patches from the background patches. After this, the convolutional layers were trained as well to fine tune the performance of the model (Chollet, 2017).

2.2.7 Evaluation metrics

Two different sets of evaluation metrics were used in this study- one to evaluate the performance of the classifiers on individual patches and another to evaluate the performance of the classifiers on the sub-image. All the models were trained and evaluated on a computer with Intel i9 processor with 18 cores and 64 GB of RAM and NVIDIA GeForce RTX 2080 Ti graphics card.

2.2.7.1 Evaluate on patches

Accuracy, precision, recall and f1 score were used as the evaluation metrics to evaluate the classification performance on patches. To calculate all these metrics, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) were calculated. TP refers to the weed patches that have been classified as weed. TN are the background patches that have been classified as background. FP refers to the background patches that have been misclassified as weed while FN are the weed patches that have been misclassified as background (Géron, 2017). In addition, the time needed for feature extraction as well as prediction of test patches was calculated and used as a metric.

$$\text{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}}$$

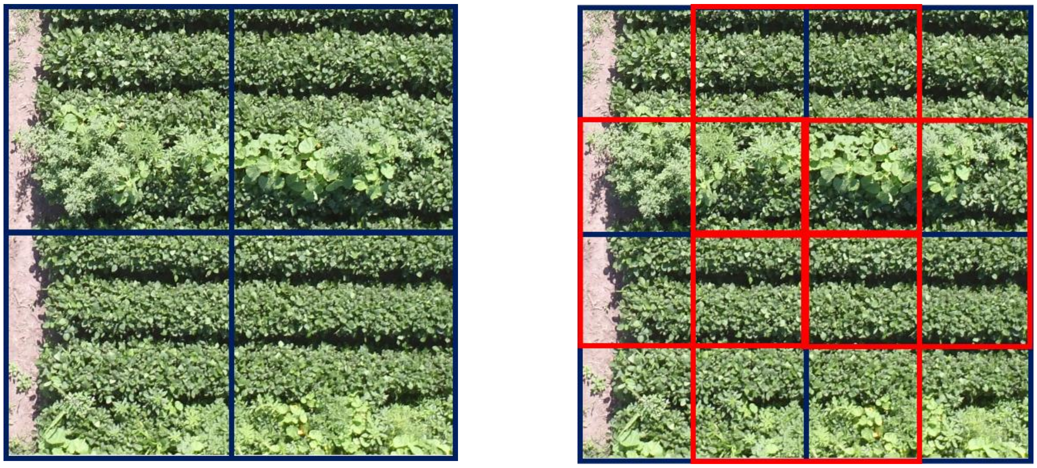
$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$

$$\text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$$

$$\text{F1 score} = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}\ +\mathrm{Recall}}$$

2.2.7.2 Evaluate on raw images

To use the patch-based classification methods, patches have to be cropped from the big raw image from the UAV system. The image slicing was done in two ways: overlapping and non-overlapping and their effect on detection of weeds was studied. In case of non-overlapping approach, a 1152×1152 sub-image was sliced into 81 non-overlapping images of size 128×128 and the class for each of the non-overlapping patch is predicted using the classifier. In case of the overlapping approach, rather than non-overlapping 128×128 patches, the image is sliced with 75% horizontal overlap and 75% vertical overlap (Figure 2.6). This helps to reduce the edge effects in the big image and helps to improve the performance.



(a) Non-overlapping approach

(b) Overlapping approach

Figure 2.6. Example of slicing an image using non-overlapping and overlapping approach. (a) a 1152 × 1152 image sliced into 4 non-overlapping small images (b) same image sliced using 50% horizontal and vertical overlap resulting in 8 small images

To compare the two approaches, mean IoU of the output images from the two approaches with respect to the ground truth binary image was used.

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}}$$

2.3 RESULTS AND DISCUSSION

2.3.1 Optimal parameters of feature extraction for conventional machine learning models

The effect of each combination of features on the classification accuracy of the 4 different classifiers was evaluated using hyper parameter tuning and cross validation and shown in Figure 2.7. It has been found that in case of all the classifiers, retaining all the 120 features leads to the best classification accuracy. Also, it has been found that, other than KNN, in the other 3 classifiers, retaining all the distance offsets (30 features) leads to a significant increase in classification accuracy compared to retaining all directions (24 features). This shows that the 5 different distance offsets contain more information than the 4 different directions for classifying between the weed and the background. This could be because of the almost radial symmetry of the plant in the top view as in aerial imagery.

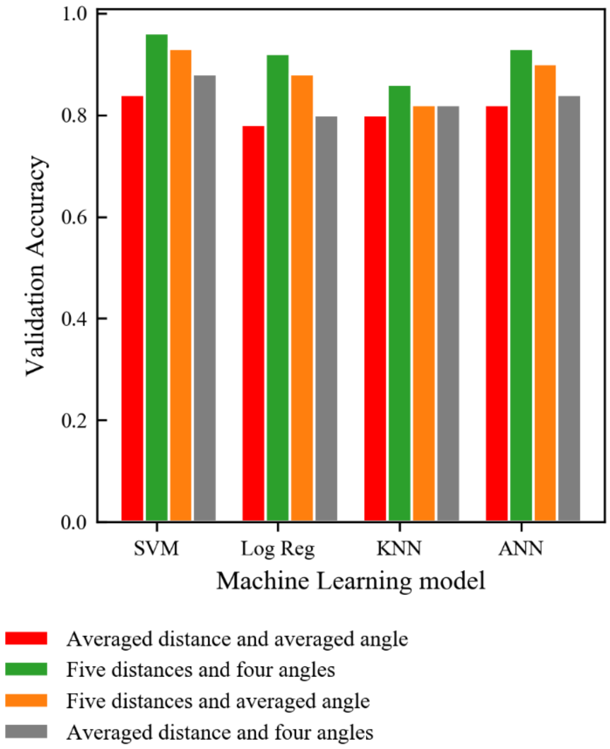


Figure 2.7. Validation accuracy of each model for different number of features. It shows that not averaging across distance and angle and retaining the features from all the distance offsets and angles result in the best performance in case of all the models

To find the optimal number of distance offsets to be calculated, experiments were conducted at various ranges from 1 to 15. In each case, features from all the distance offsets and directions were retained in the dataset. The effect of the various number of distance offsets on classification accuracy as well as the time needed for computing the features was investigated and shown in Figure 2.8. It should be noted that these experiments were performed on processors with high computing power compared to the ones used on the farm for real-time applications. Hence, rather than absolute values of computation time, the trend of change in computation time with varying distance offset should be observed. It can been seen that in case of all the models, the validation accuracy increases with increase in the number of distance offsets up to a value of 5 after which it plateaus. From the plot showing the computation time of features, it can be seen that the feature extraction using GLCM has a linear time complexity with the number of distance offsets being calculated. Hence, for this problem at this spatial resolution, Figure 2.8 shows that extracting features using 5 distance offsets (from 1 to 5) results in the best performance in terms of classification accuracy and the time needed to compute the features.

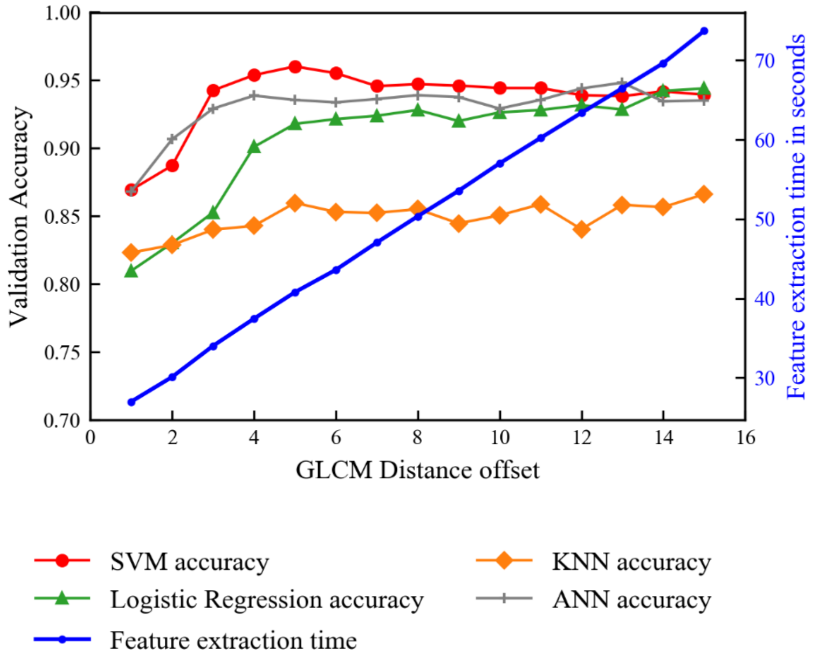


Figure 2.8. Effect of the number of distance offsets in GLCM on model performance and features computation time. 5 distance offsets (1-5) is the optimal value leading to the best accuracy and optimal feature extraction time in case of all the models.

In addition to distance offset and direction, the number of gray levels is another parameter in the calculation of GLCM that influences the calculation of Haralick features. In this case, the input images are 8 bit and so have values in the range 0-255. But it may not be needed to calculate GLCM at 256 gray levels and hence experiments were conducted by varying the gray levels from 8 to 128. The effect of varying the number of gray levels in GLCM on classification accuracy of different models as well as the time needed for computation of features is shown in Figure 2.9. It can be seen that the computational time needed for calculating features from GLCM has a piecewise linear time complexity with an increase in the number of gray levels in GLCM. In case of classification accuracy, other than KNN, for the other 3 models, gray level of 16 was found to result in the best performance whereas, in case of KNN, gray level of 32 resulted in best performance after which there was no significant performance gain with an increase in gray levels. Hence, GLCM was computed with 16 gray levels, 5 distance offsets and 4 directions resulting in 120 features in case of SVM, logistic regression and ANN whereas in case of KNN the only difference being 32 gray levels.

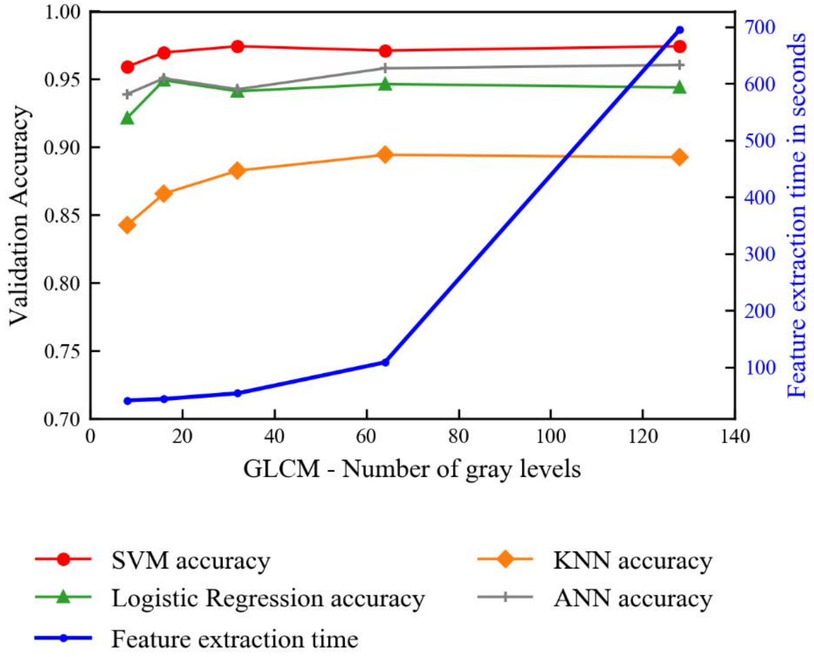


Figure 2.9. Effect of gray level in GLCM on validation accuracy and feature extraction time. 16 gray levels are found to be the optimal number of gray levels leading to best accuracy and feature extraction time trade-off.

2.3.2 Training and validation of machine learning models

Learning curves were then plotted (as shown in Figure 2.10) in case of each model by varying the training data size in steps of 500 from 500 to 10000. It can be seen that the validation accuracy increases with an increase in training data size up to 4000 after which, there is no significant increase in validation accuracy of the model. This shows that using the GLCM based feature extraction technique, all the 4 models have achieved their maximum performance for this problem and that adding more training data would not lead to any performance gain. In addition, it can be seen that in case of SVM the gap between the validation accuracy and training accuracy is very high which might be because SVM with Gaussian kernel suffers from overfitting. This can be attributed to the ability of the Gaussian kernel to create highly nonlinear decision boundaries. ANN is a non-linear model and is prone to overfitting and hence the similar gap between training accuracy curve and validation accuracy curve is obtained in ANN. It is to be noted that the learning curves are monotonic for all models but ANN. This is because of the stochastic gradient descent based solver used in training the ANN. Of all the 4 models, KNN suffers from very severe overfitting. This could be attributed to KNN being an analogy based lazy learner. Also, the performance of KNN is very sensitive to the distance function used to calculate similarity. In this case, during hyper parameter tuning, only various degrees of Minkowski distance was tested. But, it is to be noted that there is a very high correlation between the various Haralick features. Hence, using a distance function that accounts for the covariance such as Mahalanobis distance might have reduced the severity of overfitting observed here. Logistic regression is found to have no overfitting with similar performance in training data and validation data. This is due to its nature of being a simple linear classifier. Several previous research on using machine learning methods for weeds detection methods have found ANN to be a better classifier (Koot, 2014; Yano et al., 2017). Even though the results of ANN performance of this study are similar to those studies, the major difference in this study is the feature used. Previous work mentioned either used other feature extraction techniques from RGB imagery or hyperspectral imagery with pixel intensities in various bands as features. Hence, using Haralick features, for patch-based mid and late season weed detection in soybean fields from UAV imagery, we find SVM and Logistic regression to be the best performing classifiers. Also, the high classification performance from RGB is similar to the findings of (Koot, 2014) where RGB images performed better than multispectral images in classifying the weed pixels.

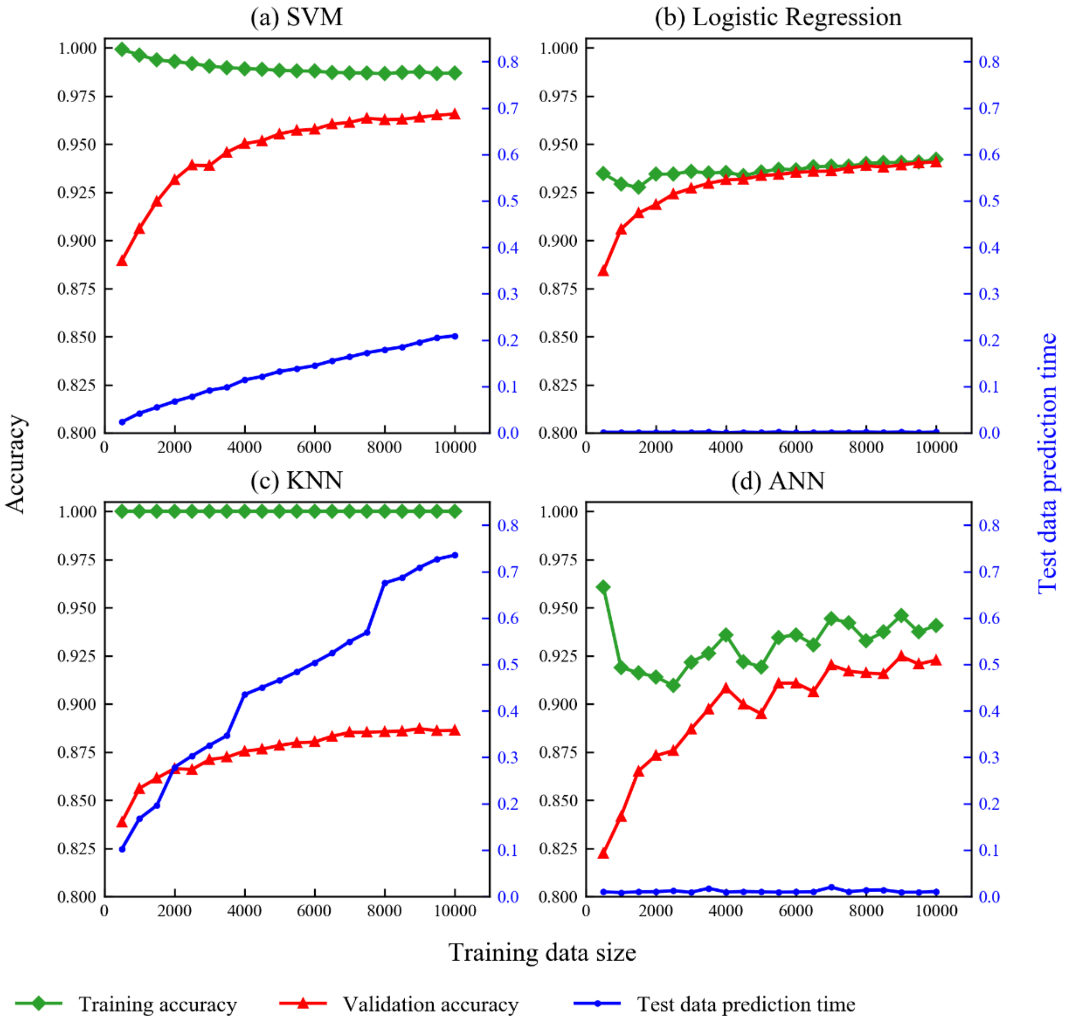


Figure 2.10. Curves showing the change in validation and training accuracy as well as prediction time of test data with an increase in training data size. It can be seen that a training data size of 4000 to 6000 results in the best performance of the models with no significant improvement after that.

In addition to studying the increase in validation accuracy with increase in training data size, its effect on the prediction time on test data is plotted in Figure 2.10. Test data prediction time indicates only the time needed for the model to predict the class label for all the test dataset with features extracted. As seen from Figure 2.10, the time needed for prediction of test data remained the same with increasing training data size for Logistic regression and ANN. This is because in case of Logistic regression and ANN, the number of parameters remains the same irrespective of the size of the training data size. Whereas in case of SVM with Gaussian kernel and KNN, it can be seen that the prediction time of test data has an almost linear time complexity with the training data size. This could be because, in case of SVM with Gaussian kernel, prediction for test data is made by calculating the similarity of the test data point with all the support vectors and the number of support vectors might be increasing with an increase in training data size thereby increasing the prediction time. Hence, in case of this application which requires on-farm data processing, if SVM with Gaussian kernel is to be used, using a training data size of 6000 results in better trade-off in terms of classification accuracy and prediction time.

2.3.3 Training and validation of convolutional neural network model

Figure 2.11 shows the training graph of the convolutional neural network. The change in training and validation accuracy and loss with an increase in training epochs is shown. During the first 10 epochs when the convolutional layers are frozen, the convolutional layers act as feature extractor and only the fully connected classification layers are trained. Hence, using the features that were extracted on the large dataset on which MobileNet v2 was trained, the model achieves a training and validation accuracy of about 94% in 10 epochs. Also, the training accuracy shows no improvement in accuracy from epochs 6 to 10. Hence, training was stopped after 10 epochs after which the convolutional layers are allowed to train as well. The model then tweaks the features learned to the current classification task after epoch as evident from the sudden increase in training and validation accuracy from epoch 10 to 11. With the convolutional layers unfrozen, the model was trained for 10 epochs. Since no performance gain was achieved after a total of 16 epochs, the training process was stopped after 20 epochs. The classification performance of the model achieved is similar to previous research on using transfer learning on CNN for related tasks such as plant species identification and plant disease detection (S. H. Lee, Chan, Wilkin, & Remagnino, 2015; Mohanty, Hughes, & Salathé, 2016).

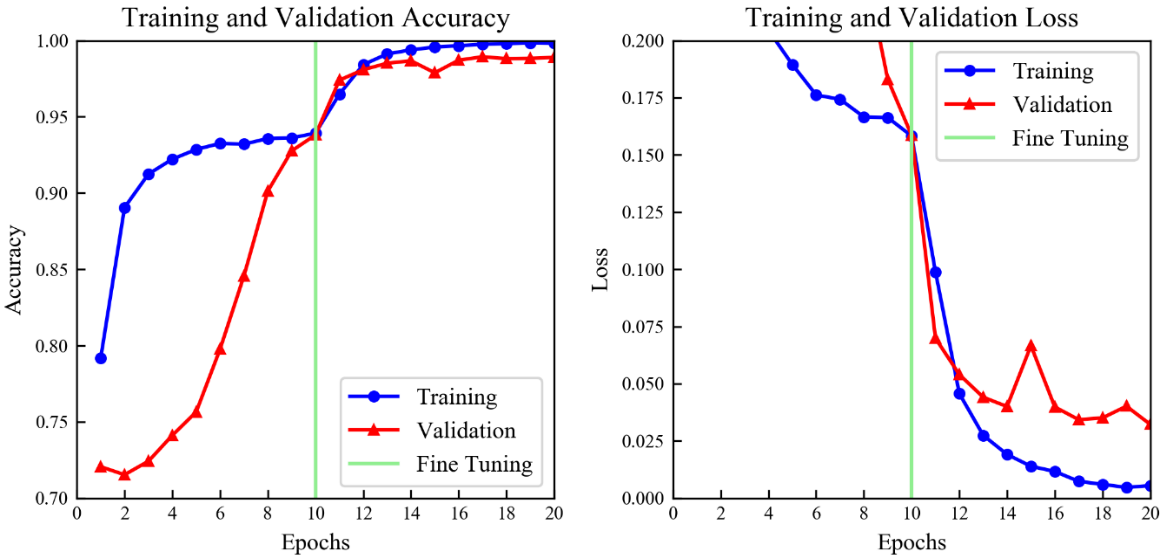


Figure 2.11. Training graph of CNN showing the change in training and validation accuracy as well as loss during the training process. The sudden jump in accuracy after fine-tuning can be seen.

2.3.4 Comparison of performance on test data

The following table shows the performance of each model on test data using 4 evaluation metrics – accuracy, precision, recall, and f1 score. Also, the prediction time of each model on the test data set as well as the total inference time (time for feature extraction and prediction) on test data is shown in Table 2.1.

Table 2.1. Model performance on test patches. SVM results in best classification performance among machine learning models but is lesser than CNN. But, CNN has longer inference time. Feature extraction contributes the most to the inference time in case of machine learning models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**Prediction time of 1 test patch in seconds**	**Processing time of 1 test patch (Feature extraction + Prediction) in seconds**
SVM	96.25	96.25	96.25	96.25	0.0001	0.0018
Logistic regression	93.32	92.62	94.13	93.37	0.000001	0.0017
ANN	91.2	90.44	92.13	91.28	0.000004	0.0019
KNN	85.14	85.86	84.14	84.99	0.0003	0.0024
CNN	98.64	97.75	97.88	97.82	0.0023	0.0023

Among the 4 machine learning methods based on manually engineered features, SVM with Gaussian kernel has the best performance in terms of all the evaluation metrics. But it takes longer for prediction compared to the other three methods. The very good performance of logistic regression indicates that the data is almost linear in the Haralick feature space but not perfectly linear. It is important to note that even though logistic regression is significantly faster in prediction than the other machine learning methods, the difference in time between models become negligible when the feature extraction time is taken into account. As seen from the table, the total inference time in case of manually engineered features based machine learning methods is limited by the time taken for feature extraction rather than prediction. In case of CNN, there is no feature extraction step involved since the convolutional layers learn the features by themselves. However, because of the large number of parameters in CNN, the inference time in case of CNN is longer than all the machine learning based methods.

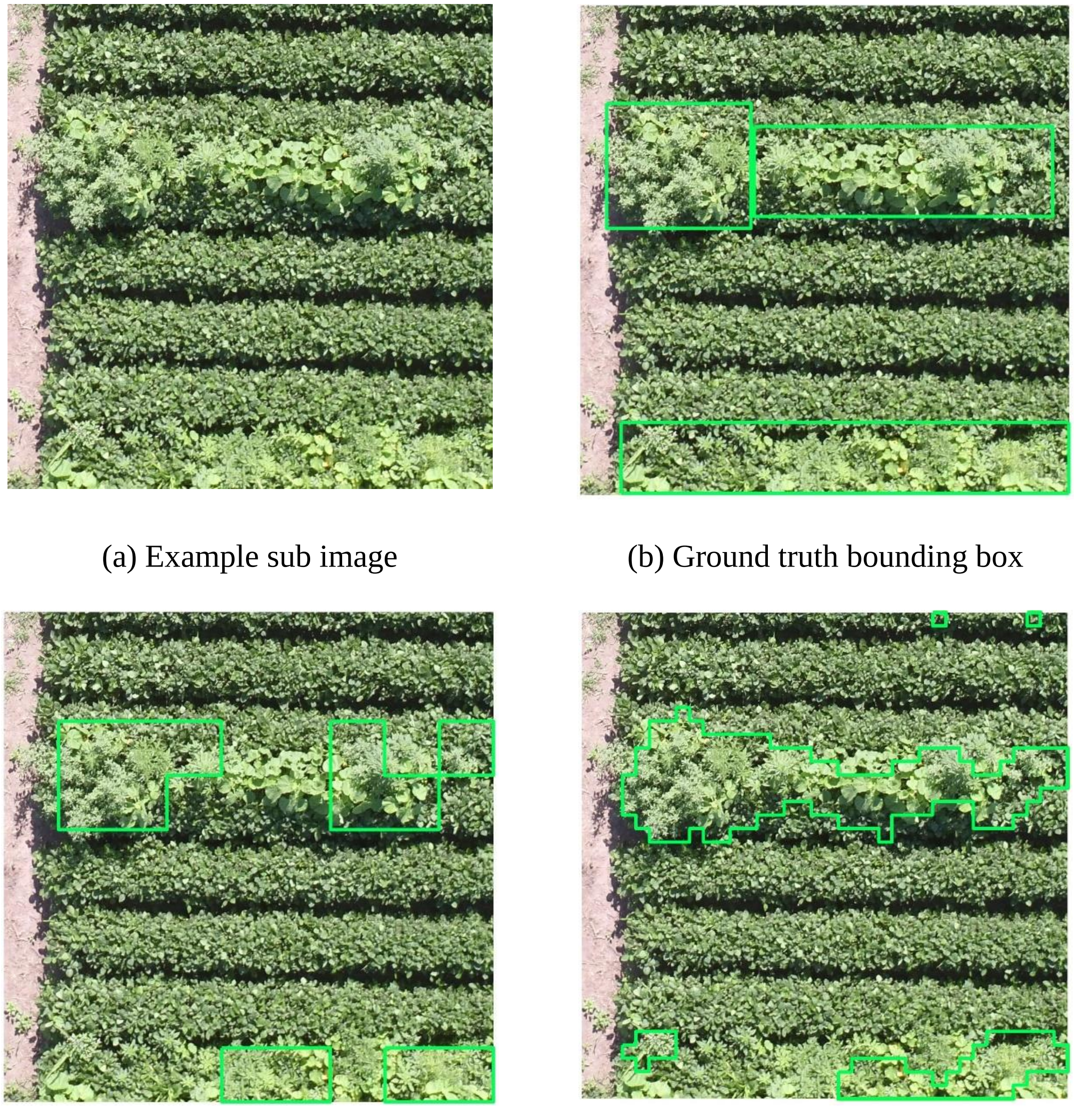
2.3.5 Comparison of overlapping and non-overlapping image slicing methods

The mean intersection over union (IoU) and the inference time on the test dataset using the overlap approach and the non-overlapping approach is shown in Table 2.2. It is found that the inference time for the overlapping approach is significantly longer compared to the non-overlapping approach. This is due to the large number of image patches that have to be evaluated in case of overlapping approach. But it can be seen that the IoU of the overlapping approach is better than the nonoverlapping approach in case of every model that was studied. Of all the models, CNN was found to have the highest IoU. This is because it was seen in Table 2 that CNN has the highest classification performance on the patch of all the models and so it leads to comparatively higher IoU than the other models. It is to be noted that 0.61 was the highest IoU that could be obtained using the patch-based approach. But as we can see in Figure 2.10, even though the IoU is lesser, the overlapping approach outputs a localized boundary of the weed patches. Since the ground truth boxes are rectangular, the ground truth does not refer to all the weed areas but the greatest area of a rectangle within which all the weed areas are present. Hence, in case of overlapping approach, higher IoU could be obtained if the ground truth refers only to the area of weed but not bounding box area.

Table 2.2. Mean IoU of different models with two approaches on test data. Image slicing with overlap results in better IoU than without overlap in case of all the models but with a significant cost in the form of inference time. Hence, in case of computational resource constraints, without overlap approach is suggested.

Model	**IoU with overlap (%)**	**Processing time of a sub-image (1152×1152) with overlap in seconds**	**IoU without overlap**	**Processing time of a sub-image (1152×1152) without overlap in seconds**
SVM	59	2.42	56	0.19
Logistic regression	58	2.12	56	0.17
ANN	52	2.17	49	0.17
KNN	52	3.47	50	0.24
CNN	61	1.03	60	0.22

The relatively less difference in IoU between overlapping and non overlapping approach in case of CNN might indicate that CNN performs relatively better in case of images with mixed pixels of crop and weed in them. The difference in inference time between overlapping and non overlapping is relatively lesser in case of CNN than other models. This could be because feature extraction within the network and prediction is done in batches in CNN whereas in case of machine learning models, feature extraction is done individually for each data point. In case of the crop rows being at an angle of 0° in the image, it should be noted that the horizontal overlap is less important than the vertical overlap. This is because the probability of mixed pixels is more in case of sliced rows than the sliced columns. Hence, in case of regular shaped fields, since the crop rows are perfectly planted in one direction, the overlapping approach may not include horizontal overlap thereby leading to increase in speed of processing without significant decrease in IoU.

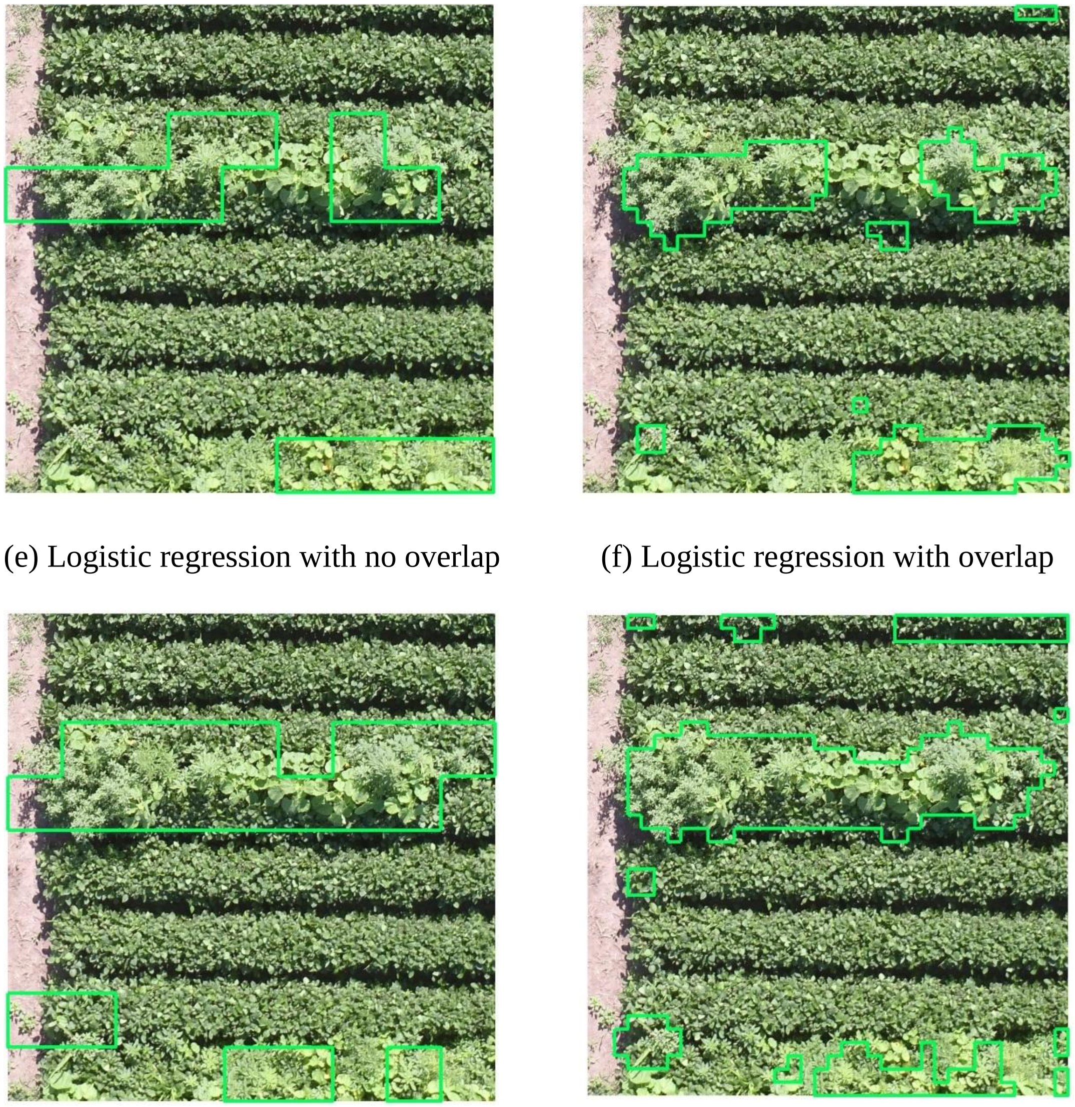


(a) Example sub image

(b) Ground truth bounding box

(c) SVM with no overlap

(d) SVM with overlap

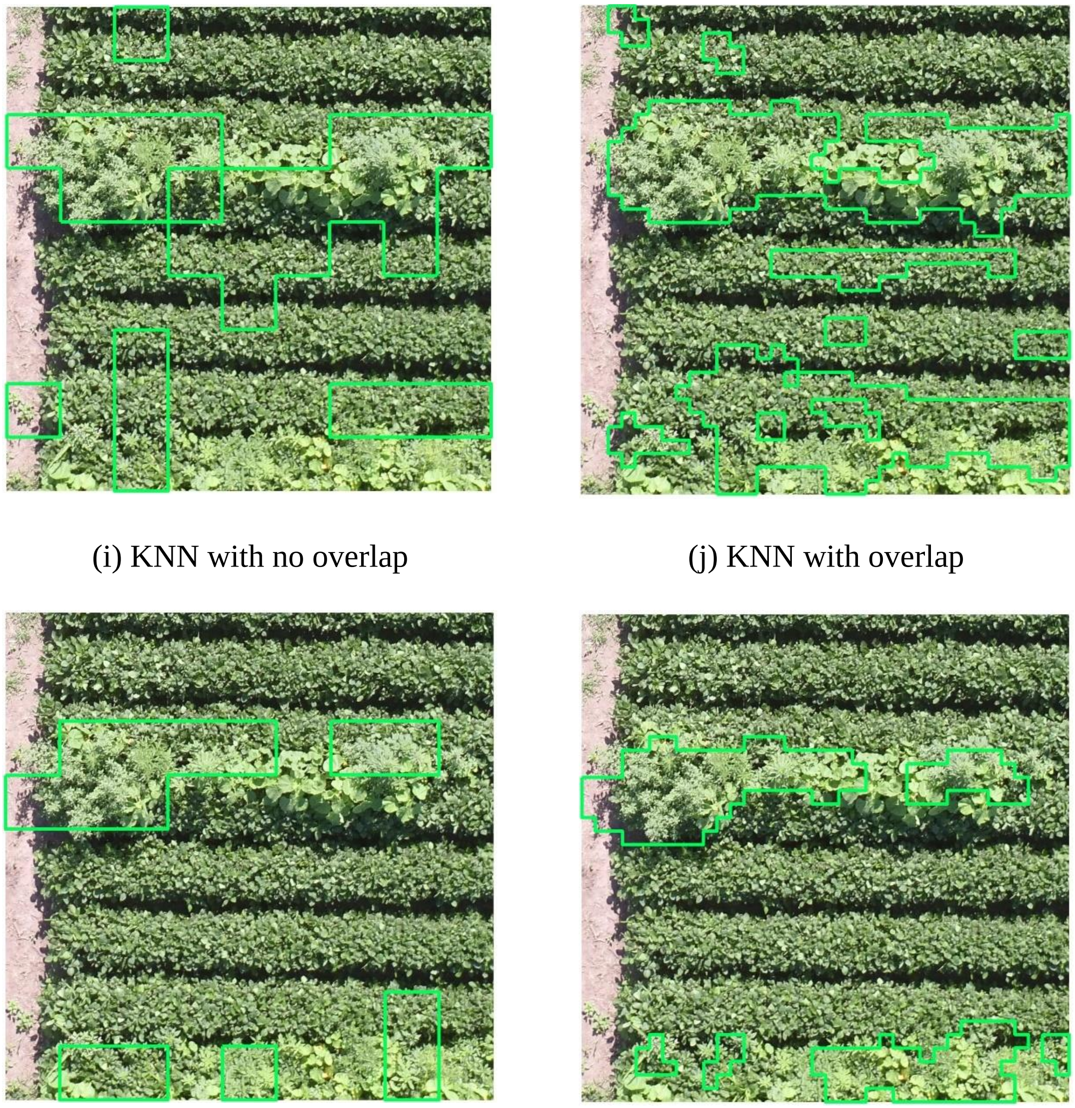


(e) Logistic regression with no overlap

(f) Logistic regression with overlap

(g) ANN with no overlap

(h) ANN with overlap



(i) KNN with no overlap

(j) KNN with overlap

(k) CNN with no overlap

(l) CNN with overlap

Figure 2.12. a) Raw image b) Ground truth bounding box c) SVM with no overlap d) SVM with overlap e) Logistic regression with no overlap f) Logistic regression with overlap g) ANN with no overlap h) ANN with overlap i) KNN with no overlap j) KNN with overlap k) CNN with no overlap l) CNN with overlap

2.4 CONCLUSION

This study investigated the potential of using patch-based machine learning and deep learning methods to detect mid and late season weeds from UAV imagery. In case of machine learning methods, experiments were done to find the optimal parameters for GLCM to extract Haralick’s features. Four directions - 0°, 45°, 90°, and 135°, 5 distance offsets from 1 to 5 and 16 gray levels were found to be the optimal parameters for feature extraction. Also, results from this experiment show that the optimal value of these parameters would vary with the problem as well as the spatial resolution of the imagery since the size of the object in the image varies. Hence, further studies are needed at different spatial resolutions to find the optimal parameters in order to make suggestions for UAV imagery collected from different altitudes. Among conventional machine learning models, SVM resulted in the best classification performance but CNN was found to have better patch classification performance than all the conventional machine learning models. In case of processing time, SVM, logistic regression and ANN was found to be faster than CNN. The results also showed the bottleneck of feature extraction time associated with machine learning methods although their prediction time is significantly faster. Among the overlapping and non-overlapping image division methods, overlapping method had better IoU with the ground truth image than non-overlapping method but there was a significant increase in processing time associated with overlapping method. The processing time of all the models in addition to the classification performance provides useful information to choose the appropriate model and evaluation approach based on computational resource availability and model performance requirement.

2.5 ACKNOWLEDGEMENT

This project is based on research that was supported by the University of Nebraska Collaboration Initiative/System Science Seed Grant, and the funding supported by the Nebraska Agricultural Experiment Station through the Hatch Act capacity funding program (Accession Number 1011130) from the USDA National Institute of Food and Agriculture.

2.6 REFERENCES

Ahmed, F., Al-Mamun, H. A., Bari, A. S. M. H., Hossain, E., & Kwan, P. (2012). Classification of crops and weeds from digital images: A support vector machine approach. Crop Protection, 40, 98–104. https://doi.org/10.1016/J.CROPRO.2012.04.024

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

CENTENARY REVIEW. (2019). https://doi.org/10.1017/S0021859605005708

Chapter 7. Texture analysis. (2017). https://doi.org/10.1016/B978-0-12-809284-2.00007-1

Chollet, F. (2017). Transfer Learning Using Pretrained ConvNets. Retrieved from TensorFlow Tutorials website: https://www.tensorflow.org/alpha/tutorials/images/transfer_learning

CHRISTENSEN, S., SØGAARD, H. T., KUDSK, P., NØRREMARK, M., LUND, I., NADIMI, E. S., & JØRGENSEN, R. (2009). Site-specific weed control technologies. Weed Research, 49(3), 233–241. https://doi.org/10.1111/j.1365-3180.2009.00696.x

D. M. Woebbecke, D. M., G. E. Meyer, G. E., K. Von Bargen, K. Von, & D. A. Mortensen, D. A. (1995a). Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions. Transactions of the ASAE, 38(1), 259–269. https://doi.org/10.13031/2013.27838

D. M. Woebbecke, D. M., G. E. Meyer, G. E., K. Von Bargen, K. Von, & D. A. Mortensen, D. A. (1995b). Shape Features for Identifying Young Weeds Using Image Analysis. Transactions of the ASAE, 38(1), 271–281. https://doi.org/10.13031/2013.27839

Domingos, P. (2012). A Few Useful Things to Know about Machine Learning. In Communications of the ACM (Vol. 55). Retrieved from https://pdfs.semanticscholar.org/c3b6/0802b56eeec611e9def0fdfbcaf42b851b99.pdf

G. E. Meyer, G. E., T. Mehta, T., M. F. Kocher, M. F., D. A. Mortensen, D. A., & A. Samal, A. (1998). TEXTURAL IMAGING AND DISCRIMINANT ANALYSIS FOR DISTINGUISHINGWEEDS FOR SPOT SPRAYING. Transactions of the ASAE, 41(4), 1189–1197. https://doi.org/10.13031/2013.17244

Gao, J., Nuyttens, D., Lootens, P., He, Y., & Pieters, J. G. (2018). Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery. Biosystems Engineering, 170, 39–50. https://doi.org/10.1016/J.BIOSYSTEMSENG.2018.03.006

Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. Retrieved from https://books.google.com/books?hl=en&lr=&id=khpYDgAAQBAJ&oi=fnd&pg=PP1&dq=geron+machine+learning&ots=kLFuJPxoq1&sig=5WQlCySn6gycgpESjQeytsTrQtQ

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314

HAWARI Ghazali, K., Mustafa, M., Hussain, A., Hawari Ghazali, K., & Marzuki Mustafa, M. (2007). Color Image Processing of Weed Classification: A comparison of two Feature Extraction Technique Intelligent Traffic Light Using Vision Sensor View project Intelligent Controller View project Color Image Processing of Weed Classification: A comparison of two Feature Extraction Technique. Retrieved from https://www.researchgate.net/publication/266450840

Huang, H., Deng, J., Lan, Y., Yang, A., Deng, X., & Zhang, L. (2018). A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery. PLOS ONE, 13(4), e0196302. https://doi.org/10.1371/journal.pone.0196302

Koot, T. (2014). Weed detection with unmanned aerial vehicles in agricultural systems. Retrieved from http://edepot.wur.nl/333537

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep

Convolutional Neural Networks (pp. 1097–1105). pp. 1097–1105. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networ

L. Tian, L., J. F. Reid, J. F., & J. W. Hummel, J. W. (1999). DEVELOPMENT OF A PRECISION SPRAYER FOR SITE-SPECIFIC WEED MANAGEMENT. Transactions of the ASAE, 42(4), 893–900. https://doi.org/10.13031/2013.13269

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, S. H., Chan, C. S., Wilkin, P., & Remagnino, P. (2015). Deep-plant: Plant identification with convolutional neural networks. 2015 IEEE International Conference on Image Processing (ICIP), 452–456. https://doi.org/10.1109/ICIP.2015.7350839

Lee, W. S., Slaughter, D. C., & Giles, D. K. (1999). Precision Agriculture, 1. Retrieved from Kluwer Academic Publishers website: https://link.springer.com/content/pdf/10.1023%2FA%3A1009977903204.pdf

López-Granados, F., Torres-Sánchez, J., Serrano-Pérez, A., de Castro, A. I., Mesas-Carrascosa, F.-J., & Peña, J.-M. (2016). Early season weed mapping in sunflower using UAV technology: variability of herbicide treatment maps against weed thresholds. Precision Agriculture, 17(2), 183–199. https://doi.org/10.1007/s11119-015-9415-8

Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., & Stachniss, C. (2017). UAV-based crop and weed classification for smart farming. 2017 IEEE International Conference on Robotics and Automation (ICRA), 3024–3031. https://doi.org/10.1109/ICRA.2017.7989347

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7, 1419. https://doi.org/10.3389/fpls.2016.01419

Peña, J. M., Torres-Sánchez, J., de Castro, A. I., Kelly, M., & López-Granados, F. (2013). Weed Mapping in Early-Season Maize Fields Using Object-Based Analysis of Unmanned Aerial Vehicle (UAV) Images. PLoS ONE, 8(10), e77151. https://doi.org/10.1371/journal.pone.0077151

Peña, J., Torres-Sánchez, J., Serrano-Pérez, A., de Castro, A., López-Granados, F., Peña, J. M., … López-Granados, F. (2015). Quantifying Efficacy and Limits of Unmanned Aerial Vehicle (UAV) Technology for Weed Seedling Detection as Affected by Sensor Resolution. Sensors, 15(3), 5609–5626. https://doi.org/10.3390/s150305609

Perez-Ortiz, M., Gutierrez, P. A., Pena, J. M., Torres-Sanchez, J., Lopez-Granados, F., & Hervas-Martinez, C. (2016). Machine learning paradigms for weed mapping via unmanned aerial vehicles. 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 1–8. https://doi.org/10.1109/SSCI.2016.7849987

Pérez-Ortiz, M., Peña, J. M., Gutiérrez, P. A., Torres-Sánchez, J., Hervás-Martínez, C., & López-Granados, F. (2015). A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method.

Applied Soft Computing, 37, 533–544. https://doi.org/10.1016/J.ASOC.2015.08.027

Pulido Rojas, C., Solaque Guzmán, L., & Velasco Toledo, N. (2017). Weed recognition by SVM texture feature classification in outdoor vegetable crops images. Ingeniería e Investigación, 37(1), 68. https://doi.org/10.15446/ing.investig.v37n1.54703

S. N. Ondimu, S. N., & H. Murase, H. (2008). Comparison of Plant Water Stress Detection Ability of Color and Gray-Level Texture in Sunagoke Moss. Transactions of the ASABE, 51(3), 1111–1120. https://doi.org/10.13031/2013.24513

Sa, I., Chen, Z., Popovic, M., Khanna, R., Liebisch, F., Nieto, J., & Siegwart, R. (2018). weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. IEEE Robotics and Automation Letters, 3(1), 588–595. https://doi.org/10.1109/LRA.2017.2774979

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks (pp. 4510–4520). pp. 4510–4520. Retrieved from http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html

Sankaran, S., Khot, L. R., Espinoza, C. Z., Jarolmasjed, S., Sathuvalli, V. R., Vandemark, G. J., … Pavek, M. J. (2015). Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review. European Journal of Agronomy, 70, 112–123. https://doi.org/10.1016/J.EJA.2015.07.004

T. F. Burks, T. F., S. A. Shearer, S. A., & F. A. Payne, F. A. (2000). CLASSIFICATION

OF WEED SPECIES USING COLOR TEXTURE FEATURES AND DISCRIMINANT ANALYSIS. Transactions of the ASAE, 43(2), 441–448. https://doi.org/10.13031/2013.2723

Thorp, K. R., & Tian, L. F. (2004). A Review on Remote Sensing of Weeds in Agriculture. Precision Agriculture, 5(5), 477–508. https://doi.org/10.1007/s11119-004-5321-1

Torres-Sánchez, J., López-Granados, F., De Castro, A. I., & Peña-Barragán, J. M. (2013). Configuration and Specifications of an Unmanned Aerial Vehicle (UAV) for Early Site Specific Weed Management. PLoS ONE, 8(3), e58210. https://doi.org/10.1371/journal.pone.0058210

Tzutalin. (2015). LabelImg. Git code.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., … Yu, T. (2014). scikit-image: image processing in Python. PeerJ, 2, e453. https://doi.org/10.7717/peerj.453

Weis, M., Gutjahr, C., Rueda Ayala, V., Gerhards, R., Ritter, C., & Schölderle, F. (2008). Precision farming for weed management: techniques. Gesunde Pflanzen, 60(4), 171–181. https://doi.org/10.1007/s10343-008-0195-1

Wu, L., & Wen, Y. (2009). Weed/corn seedling recognition by support vector machine using texture features. In African Journal of Agricultural Research (Vol. 4). Retrieved from http://www.academicjournals.org/ajar

Y. K. Chang, Y. K., Q. U. Zaman, Q. U., A. W. Schumann, A. W., D. C. Percival, D. C.,

T. J. Esau, T. J., & G. Ayalew, G. (2012). Development of Color Co-occurrence Matrix Based Machine Vision Algorithms for Wild Blueberry Fields. Applied Engineering in Agriculture, 28(3), 315–323. https://doi.org/10.13031/2013.42321

Yano, I. H., Santiago, W. E., Jose, ;, Alves, R., Toledo, L., Mota, M., & Teruel, ; Barbara. (2017). Agricultural Academy. In Bulgarian Journal of Agricultural Science (Vol. 23). Retrieved from https://www.agrojournal.org/23/03-21.pdf

Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—a worldwide overview. Computers and Electronics in Agriculture, 36(2–3), 113–132. https://doi.org/10.1016/S0168-1699(02)00096-0

CHAPTER 3 COMPARISON OF OBJECT DETECTION AND PATCH-BASED CLASSIFICATION DEEP LEARNING MODELS ON MID TO LATE SEASON WEED DETECTION IN UAV IMAGERY

This manuscript has been prepared for journal submission

3.1 INTRODUCTION

To feed the increasing population, it is necessary to increase global agricultural productivity (Godfray et al., 2010). Hence, it becomes critical to address the various crop yield-limiting factors such as weeds and other biotic and abiotic stresses. (“CENTENARY REVIEW,” 2019). Weeds are unwanted plants that grow in the field and compete with the crops for water, light, nutrients, and space. If uncontrolled, weeds can have several negative consequences such as crop yield loss, production of a large number of seeds thereby creating a weed seed bank in the field and contamination of grain during harvesting to name a few (dos Santos Ferreira, Matte Freitas, Gonçalves da Silva, Pistori, & Theophilo Folhes, 2017). Traditionally, weed management programs involve control of weeds through chemical or mechanical means such as uniform application of herbicides throughout the field. However, the spatial density of weeds is not uniform across the field, thereby leading to overuse of chemicals which results in environmental concerns and evolution of herbicide-resistant weeds. To overcome this issue, a concept of site-specific weed management, which refers to detecting weed patches and spot spraying or removal by mechanical means, was proposed in the early ’90s. (CHRISTENSEN et al., 2009; Weis et al., 2008; Zhang, Wang, & Wang, 2002).

Earlier studies on weed detection often used Color Co-occurrence Matrix based texture analysis for digital images (G. E. Meyer, T. Mehta, M. F. Kocher, D. A. Mortensen, & A. Samal, 1998; T. F. Burks, S. A. Shearer, & F. A. Payne, 2000). Following this, there were several studies on combining optical sensing, image processing algorithms, and variable rate application implements for real-time site-specific herbicide application on weeds. However, the speed of these systems was limited by computational power constraints for real-time detection, which in turn limited their ability to cover large areas of fields. Unmanned aerial vehicles (UAVs) with their ability to cover large areas in a short amount of time and payload capacity to carry optical sensors provided an alternative. UAVs have been studied for various applications in precision farming such as weed, disease, and pest detection using high-resolution aerial imagery (Sankaran et al., 2015). Multiple studies have investigated several algorithms to detect weeds from the crop in aerial imagery. A common approach is to use vegetation indices to segment the vegetation pixels from the soil pixels, followed by crop row detection for weed classification using techniques such as object-based image analysis (OBIA) and Hough Transform (López-Granados et al., 2016; Peña, Torres-Sánchez, de Castro, Kelly, & López-Granados, 2013; Pérez-Ortiz et al., 2015). However, crop row detection-based approaches cannot detect intra-row weeds. Hence machine learning based classifiers using features computed from OBIA were used to detect intra-row weeds as well (de Castro et al., 2018). However, the performance of OBIA is sensitive to the segmentation accuracy and so optimal parameters for segmentation step in OBIA has to be found for different crops and field conditions (D. Liu & Xia, 2010). Also, this approach is limited to early season weed detection when crops and weed objects do not overlap. In case of overlapping crop and weed objects, Lottes, Khanna, Pfeifer, Siegwart, & Stachniss (2017) proposed a key point based feature extraction approach that was found to detect weed objects that overlap with the crop. However, with color based vegetation segmentation being the first preprocessing step in the above studies to segment the vegetation objects, they are limited in the application after crop canopy closure when there is no soil background.

With advancements in parallel processing computing and availability of large datasets, convolutional neural networks (CNN) was found to perform very well in computer vision tasks such as classification, prediction and object detection (Krizhevsky, Sutskever, & Hinton, 2012a). In addition to performance, another principal advantage of CNN is that the network learns the features by itself during the training process, and hence manual feature engineering is not necessary. CNNs have been studied for various image-based applications in agriculture such as weed detection, disease detection, fruit counting, crop yield estimation, obstacle detection for autonomous farm machines and soil moisture content estimation (Kuwata & Shibasaki, 2015; Mohanty, Hughes, & Salathé, 2016; Rahnemoonfar, Sheppard, Rahnemoonfar, & Sheppard, 2017; Song et al., 2016; Steen et al., 2016). CNNs have been used for weed detection using data obtained from three different ways – using UAVs, using the autonomous ground robot and high-resolution images obtained manually in the field. A simple CNN binary classifier was trained to classify manually collected small high-resolution images of maize and weed. The performance of the classifier with transfer learning on various pre-trained networks such as LeNet and AlexNet was compared, but this study was limited in variability in the obtained dataset and on the evaluation of the classification approach with large images (Andrea, Mauricio Daniel, & Jose Misael, 2017). Dyrmann, Mortensen, Midtiby, & Jørgensen (2016) used a pre-trained VGG-16 network and replaced the fully connected layer with a deconvolution layer to output a pixel-wise classification map of maize, weed, and soil. The training images were simulated by overlapping a small number of available images of soil, maize, and weed in various orientations and proportions. The use of encoder- decoder architecture for real-time output of pixel-wise classification map for site-specific spraying was studied. It was found that by adding hand-crafted features such as vegetation indices, different color spaces and edges as input channels to CNN, the generalization performance of the model in different locations at the different growth stage of the crop improved (Milioto, Lottes, & Stachniss, 2018). Also, to improve the generalization performance of the CNN-based weed detection system, Lottes, Behley, Milioto, & Stachniss (2018) studied the use of fully convolutional DenseNet with spatiotemporal fusion and spatiotemporal decoder with sequential images to learn the local geometry of crops in fixed straight lines along the path of a ground robot. In addition to weed detection, for effective removal of weeds in case of mechanical or laser means, it is necessary to detect the stem of weeds for actuation. A fully convolutional DenseNet was trained to output the stem location of crop and weed as well as a pixel-wise segmentation map of crop and weed (Lottes, Behley, Chebrolu, Milioto, & Stachniss, 2018).

In case of weed detection using UAV imagery, similar to OBIA approaches mentioned above, dos Santos Ferreira et al. (2017) used a Superpixel segmentation algorithm to segment objects are clusters from an image and trained CNN to classify these clusters and compared the performance with other machine learning classifiers which use handcrafted features. Sa, Chen, et al. (2018) studied the use of an encoder - decoder architecture, Segnet for pixel-wise classification of multispectral imagery and followed up with a study on performance evaluation of this detection system using different UAV platforms and multispectral cameras (Sa, Popović, et al., 2018). Bah, Dericquebourg, Hafiane, & Canals (2019) used Hough transform along with patch-based CNN to detect weeds from UAV imagery and found that overlapping weed and crop objects led to some errors in this approach. It is to be noted that in this approach the patches are sliced from the large image in a non-overlapping manner. H. Huang et al. (2018) studied the performance of various deep learning architectures for pixel-wise classification of rice and weeds and found that the Fully Convolutional Network architecture outperformed other architectures.

From the literature reviewed, it can be seen that automated weed detection has been primarily focused on early season weeds since that is found to be the critical period for weed management to prevent crop yield loss. However, it is to be noted that mid to late season weeds escaped from the routine early-season management also threatens the production in a longer term by creating a large number of seeds for several future growing seasons. With the herbicide resistance issue currently, escaped herbicide-resistance weeds become prominent. Studies on early season weeds use vegetation segmentation as a preprocessing step to reduce the memory requirements, but with no soil pixels due to canopy closure, this does not apply to mid to late season weed imaging. Also, because of the significant overlap between crop and weed, the performance of the object-based feature extraction algorithm will be limited because of the challenges in segmenting weed and crop objects in such a cluttered environment. With deep learning-based object detection methods having proven successful for tasks such as fruit counting which has a cluttered background as in this case, it is hypothesized that such methods would be able to detect mid to late season weeds from UAV imagery. Also, as seen in Chapter 2, with patch-based CNN method using overlapping evaluation performing well, the objective of this study is to study deep learning based object detection methods to detect mid to late season weeds and compare their performance with patch-based CNN method. The specific objectives are

1. Evaluate the performance of two object detection algorithms with different detection performance and inference speed - Faster RCNN and Single Shot Detector algorithm in detecting mid to late season weeds from UAV imagery using precision, recall, f1 score and mean IoU as the evaluation metrics for their detection performance and inference time as the metric for their speed
2. Compare the performance of object detection model with better detection performance and speed with patch-based CNN in terms of weed detection performance using mean IoU and inference time

3.2 MATERIALS AND METHODS

3.2.1 Study Site

The study sites were located in the South Central Agricultural Laboratory of the University of Nebraska, Lincoln at Clay Center, NE, USA (40.575188, -98.130909). The two study sites were located adjacent to each other. They were different soybean weed management research plots. Figure 3.1 shows the stitched maps of the study sites.

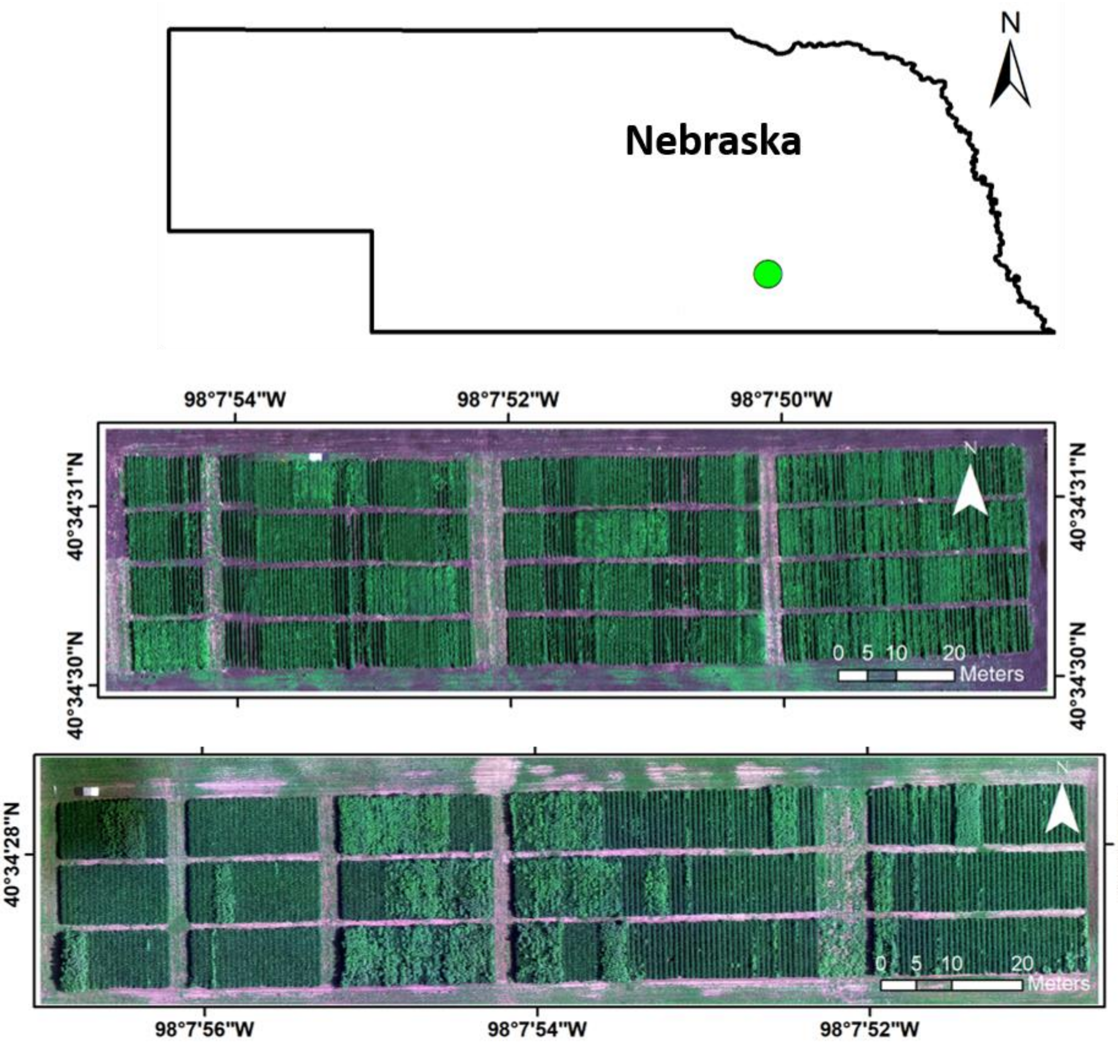


Figure 3.1 Study area at South Central Ag Laboratory in Clay Center, NE

3.2.2 UAV data collection

A DJI Matrice 600 pro unmanned aerial vehicle (UAV) platform (Figure 3.2) was used with a Zenmuse X5R camera to capture aerial imagery. In order to collect data with varying growth stage of crop as well as variations in illumination conditions, the images from study site 1 (shown at the top in Figure 3.1) were collected on July 2nd, 2018 whereas the images from study site 2 (shown at the bottom in Figure 3.1) were collected on July 12th, 2018. The flight altitude in both the cases was 20m above ground level. The Zenmuse X5R camera used is a 16 megapixel camera with 4/3” sensor and 72 degree diagonal field of view. The dimension of the captured images is 4608×3456 pixels in three bands – Red, Green, and Blue. To develop an economical solution, this study focuses on only using RGB imagery. At 20m altitude, for the given sensor specifications, the spatial resolution of the output image is 0.5 cm/pixel. DJI Ground Station pro software was used for flight control and the data collection mission was flown with 90% forward overlap and 85% side overlap.

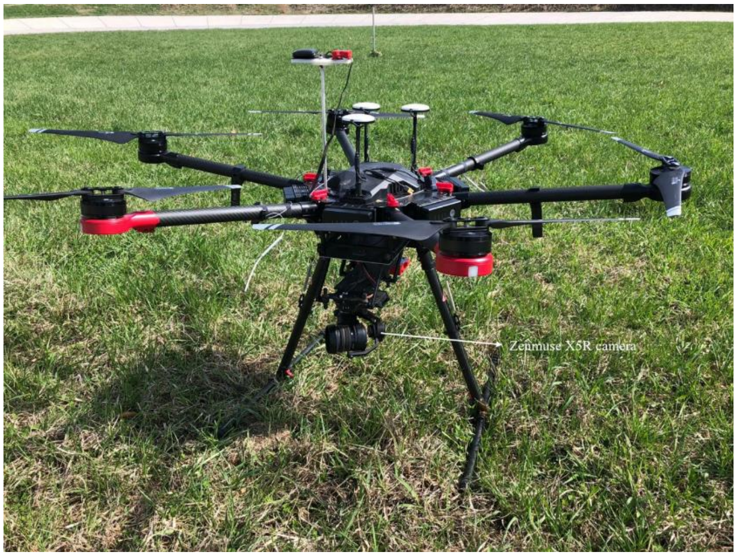


Figure 3.2 DJI Matrice 600 pro UAV platform with Zenmuse X5R camera

3.2.3 Data annotation and processing

The objective of the study is to develop a weed detection system with on-farm data processing capability. Since the mosaicking of overlapping aerial images is the time-consuming process in the workflow and is not required in this case, overlapping images were removed, and only the non-overlapping raw images were retained. The original dimension of the raw image is too large to fit in the memory for processing: each raw image of size 4608×3456 pixels was sliced into 12 sub-images of size 1152×1152 pixels. The weed areas in each sub-image were annotated as rectangular bounding boxes using the python labeling tool LabelImg (Tzutalin, 2015). A total of 450 sub-images were annotated and was then randomly split into 90% training images and 10% test images. Using the annotation information, small patches of size 128×128 were cropped from weed areas and the background areas in sub-images. These images were saved as binary class dataset belonging to two classes – ‘Weed’ and ‘Background’. 7098 patches in each class were extracted and used for training, whereas 801 patches belonging to each class were used as test data.

3.2.4 Patch based CNN

Convolutional neural networks (CNNs) are feedforward artificial neural networks with the fully connected layers in the input hidden layers replaced with convolutional filters. This reduces the number of filters in each layer and enables CNNs to learn spatial patterns in images and other two-dimensional data. The advantage of a CNN is its ability to learn the features by itself, thereby preventing the need for time-consuming hand engineering of features needed in case of other Computer Vision algorithms. CNN architectures have been proposed and its use in applications such as document recognition by using backpropagation for training has been studied much earlier (LeCun, Bottou, Bengio, & Haffner, 1998). However, their applications were limited because of the need for very large datasets to train a large number of parameters in deep networks and also the computational needs for training. In the last decade, with advancements in parallel processing capabilities using graphical processing units and increases in availability in large datasets, (Krizhevsky, Sutskever, & Hinton, 2012b) showed the potential of CNN in complex multiclass image classification tasks. But in most cases, it was found that there was not enough data available that is needed to train a deep CNN from scratch. Transfer learning helped overcome this limitation. Transfer learning is the technique of using the weights of pre-trained networks trained on very large datasets such as Alexnet, Googlenet and retraining them with small datasets for other applications (Torrey & Shavlik, 2010). This has been found to lead to exceptional classification performance and hypothesis for its performance is that the features learned in the initial convolutional layers are global features which are common across image classification tasks.

In this study, a pre-trained network called Mobilenet v2 has been used for transfer learning. The Mobilenet v2 was developed primarily for use in mobile or devices with lesser memory capabilities. Hence, in order to reduce the number of parameters, in each convolutional block, Mobilenet v2 consists of an expansion layer with a convolutional kernel of window size 1. This layer increases the number of channels in the input. This is followed by a normal convolutional layer which is then followed by a projection layer that consists of a convolutional kernel of window size 1. This depthwise layer reduces the number of channels in the output thereby reducing the number of parameters in the next convolutional block. Hence in each block, feature maps are projected to a high dimensional space followed by learning higher dimensional features which are then encoded using a depthwise convolutional projection layer. The Mobilenet v2 network was trained ImageNet dataset containing 1.4 million images belonging to 1000 classes (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). This network was retrained using the training patches belonging to both the classes. Initially, for the first 10 epochs, only the classifier layer of the network was trained by freezing the weights of all other layers. This was done to use the global features learned on the ImageNet dataset and fine tune the classifier for this specific application. After this, fine-tuning was performed in which all the top layers were unfrozen and were allowed to fine tune the convolutional features to this specific application. The fine tuning was performed for 10 epochs and hence the model was only trained for 20 epochs in total (Chollet, 2017).

3.2.5 Object detection models

Object detection refers to the task of localization of an object in an image in addition to classifying the object. Hence, for every object in the image, the model is expected to regress the coordinates of the bounding box of the object in addition to the class probabilities for classification. Two different models have been investigated – Faster RCNN with Inception v2 as feature extractor and SSD with Inception v2 as a feature extractor. Faster RCNN and SSD was chosen since Faster RCNN was found to have better performance whereas SSD was found to have better speed and performance tradeoff. Also, in both cases, Inception v2 architecture as extractor was found to be faster. Since our objective is to develop a weed detection system with on-farm real-time data processing capabilities, Inception v2 was chosen (Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, 2017).

3.2.5.1 Faster RCNN

Faster RCNN model consists of three sections namely the feature extractor, region proposal network followed by the Region of Interest (RoI) and classification layer as shown in Figure 3.3.

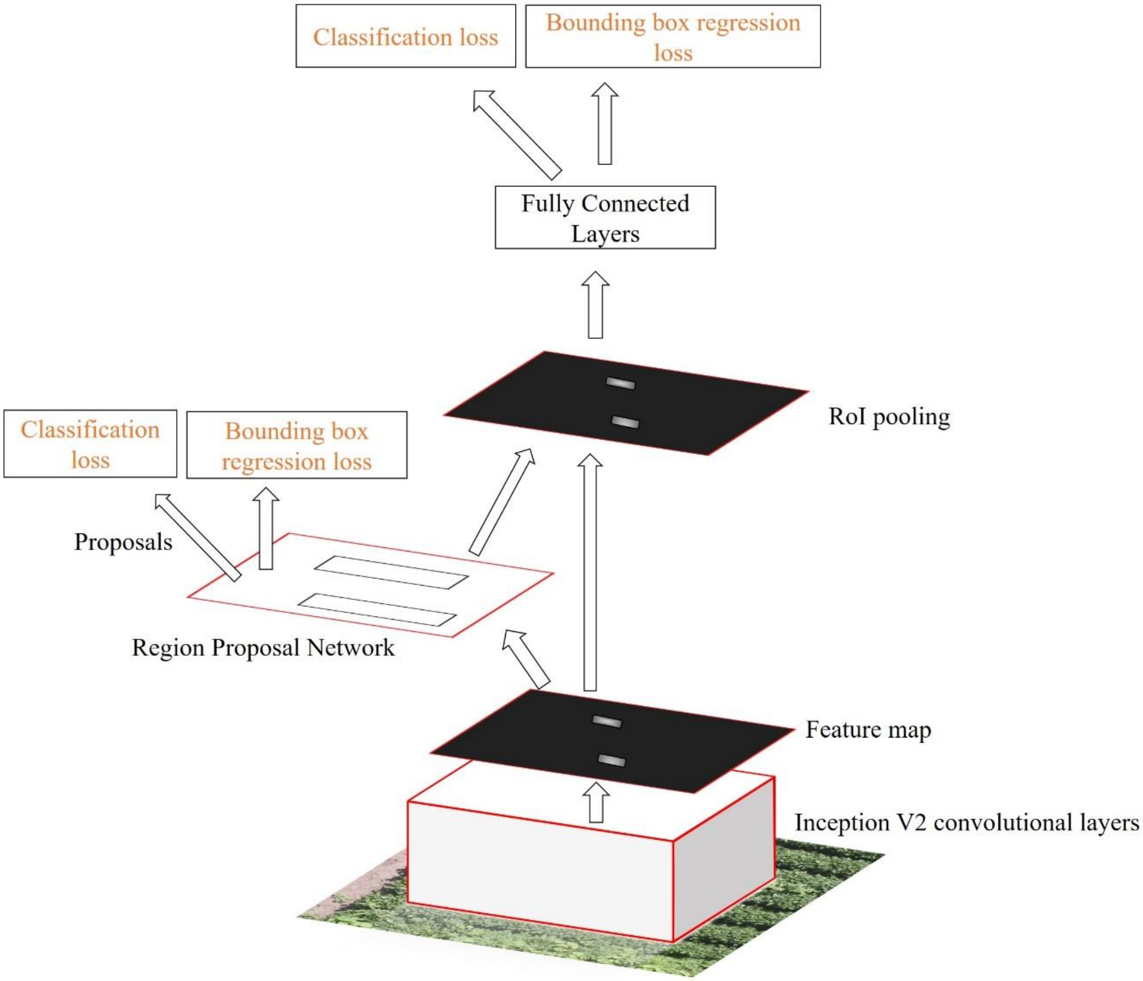


Figure 3.3. Faster RCNN architecture

For feature extraction, the convolutional layers from Inception v2 architecture is used. The advantage of Inception v2 network is its use of wider networks with filters of different kernel sizes in each layer which makes it translation and scale invariant. Hence, the Inception v2 architecture outputs a reduced dimensional feature map to the region proposal layer. The region proposal network is defined by anchors or fixed boundary boxes at each location. At each location, anchors of different scale and aspect ratio are defined, thereby enabling the region proposal network to make scale invariant proposals. The region proposal layer uses a convolutional filter on the feature map to output a confidence score for two classes, namely object, and background. This is called the objectness score. Also, the convolutional filter outputs regression offsets for anchor boxes. Hence, assuming there are k anchors at a location, the convolutional filter in the region proposal network outputs 6k value, namely 4k coordinates and 2k scores. Two losses are calculated from this output – classification loss and bounding box regression loss. The bounding box coordinates of anchors classified as objects are then combined with the feature map from feature extractor. In the RoI pooling layer, bounding box regions of different sizes and aspect ratios are resized to fixed size outputs using max pooling. The max pooled feature map of a fixed size corresponding to each output is then classified, and its bounding box offsets with respect to ground truth boxes are regressed. Hence, as in region proposal layer, two losses are computed at this output, namely the classification loss and bounding box regression loss.

3.2.5.2 Hyperparameters of the architecture

In the framework that was used, the input images to the Faster RCNN network were resized to images of fixed size 1024 × 1024 pixels. At each location in the region proposal layer, 4 different scales namely 0.25, 0.5, 1.0, 2.0 and 3 different aspect ratios namely 0.5, 1.0 and 2.0 were used. Hence, in total there were 12 anchors at each location. The model was trained for 25000 epochs with a batch size of 1 using momentum optimizer. The training dataset was split into training and validation dataset and the performance of the model on validation data was continuously monitored during training to check if the model starts to overfit. Random horizontal flip and random crop operations were performed to augment the training data.

3.2.5.3 SSD

Single Shot Detector (SSD) model was proposed to improve the inference time of objection detection models with region proposal network such as Faster RCNN. The main difference in SSD compared to Faster RCNN is the generation of detection outputs without a separate region proposal layer. Similar to Faster RCNN, SSD uses a feature extractor which is Inception v2 architecture in this case. At each location of feature map output, the model outputs a set of bounding boxes of different scales and aspect ratios. This is very similar to Faster RCNN but the difference being the convolutional filter on the feature map outputs directly the confidence scores corresponding to the output classes along with regression box offsets. Hence, the class and bounding box offsets are output in a single shot as the name suggests. In order for the model to be scale and translation invariant, rather than outputting bounding boxes from only the feature map output, extra feature layers are added to the feature map output and detection boxes are output at different scales from each output. Hence, in total, the SSD model has 6 layers that output detection boxes at different scales (W. Liu et al., 2015) (Figure 3.4).

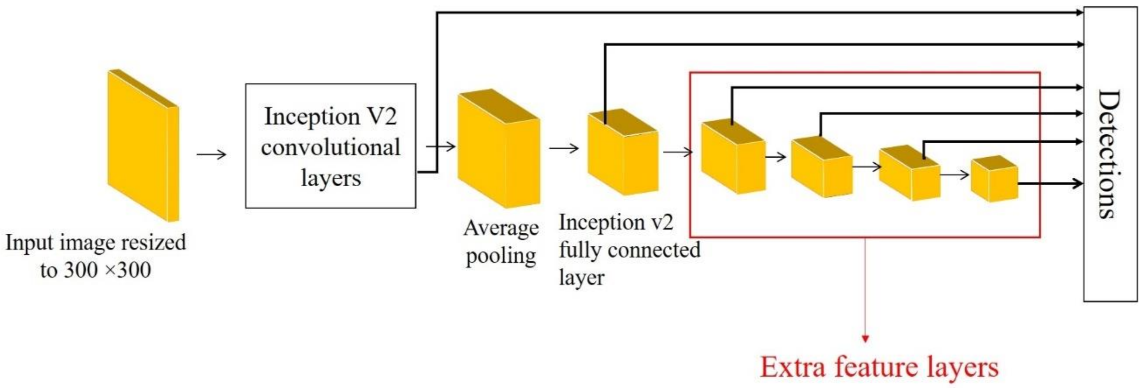


Figure 3.4. SSD architecture

3.2.5.4 Hyperparameters of the architecture

In case of SSD, in the framework that has been used, the input images are always reshaped to a fixed dimension of 300 × 300 pixels. After the feature extraction, in 6 different layers that output detection boxes, 6 different scales in the range 0.2-0.95 was used. Five different aspect ratios namely 1.0, 2.0, 0.5, 3.0 and 0.333 were generated at each location. The model was trained for 25000 epochs as in the case of Faster RCNN. A batch size of 24 was used in training and RMS prop optimizer was used. Data augmentation was applied with random horizontal flipping and random cropping of images. Validation images were evaluated periodically during the training to check if the model is overfitting.

3.2.6 Hardware and software used

The models were trained, and evaluation of the models was performed on a computer with Intel i9 processor with 18 cores and 64 GB of RAM and NVIDIA GeForce RTX 2080 Ti graphics card. Tensorflow object detection API (Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, 2017) in Python was used to train and evaluate Faster RCNN and SSD. Tensorflow tutorial on transfer learning (Chollet, 2017) was used to train the MobileNet v2 architecture for patch-based CNN.

3.2.7 Evaluation metrics

Precision, recall, f1 score, and Intersection over Union (IoU) are the evaluation metrics used in this study.

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$

$$\text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$$

$$\text{F1 score} = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}\ +\mathrm{Recall}}$$

Here TP refers to True Positive, FP refers to False Positive, and FN refers to False negative. Also these, mean Average Precision (mAP) is another metric that is commonly used in object detection problems. It is the mean of the average precision at all recall values at different IoU of prediction and ground truth thresholds from 0.5 to 0.95. It is to be noted that these metrics were primarily formulated for object detection. Even though in this study we use object detection models, the objective is not to find weed objects but rather all the area covered by weeds for management purpose. However, in case of above metrics, for each ground truth box, only one prediction box with the highest class score is assigned as a True Positive. In this case, some prediction boxes might end up being considered as False Positive even though they cover an area of a weed patch not covered by the True Positive for a corresponding ground truth box. As can be seen in the following Figure 3.5, the output of this image has two prediction boxes covering the weed area in the left but in the ground truth it was marked as one bounding box. Hence, if precision is used as the evaluation metric, the box on the bottom will be regarded as False Positive even though that box adds to more weed area being detected. Therefore, Intersection over Union (IoU) of binary output image representing weed and background pixels with the ground truth binary image is used as the primary evaluation metric.

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}}$$

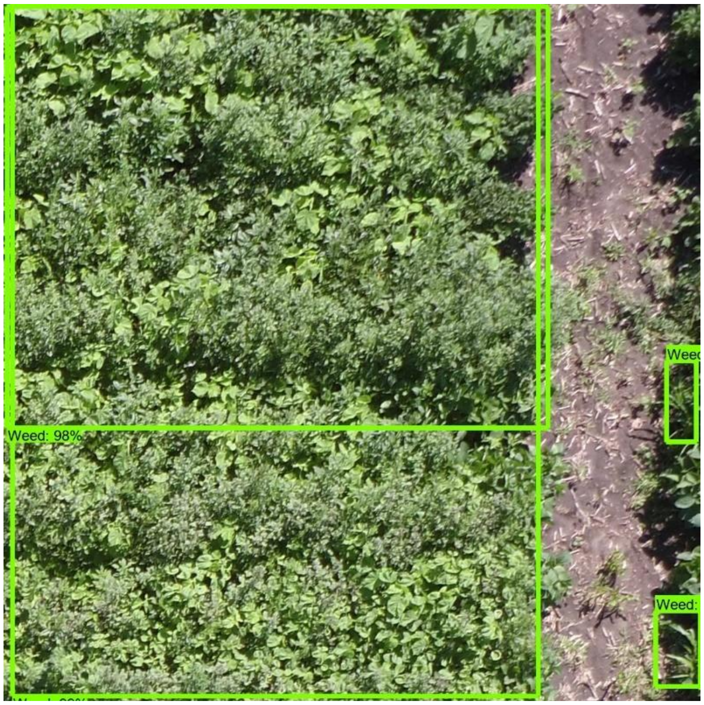


Figure 3.5. Example output image showing a weed patch annotated with single box in ground truth image detected as two boxes in output. This will lead to lesser precision as only the bigger box is considered true positive and therefore IoU is a better evaluation metric for this problem

To evaluate the patch-based CNN on the sub-image, an overlapping slicing approach is used. The sub-image of size 1152 × 1152 pixels is sliced into patches of size 128 × 128 pixels with a stride of 32 on the horizontal and vertical. Therefore, the sliced patches have 75% horizontal and vertical overlap. Hence, each small area of size 32 × 32 is part of 8 patches and the class with maximum votes from the 4 patches is assigned as the class of the small area. To evaluate this result with ground truth and to compare with the results of Faster RCNN and SSD, IoU is used as the evaluation metric.

3.3 RESULTS AND DISCUSSION

3.3.1 Training of Faster RCNN and SSD

Figure 3.6 shows the training graph for Faster RCNN and SSD. The decrease in training loss and the increase in mAP of the validation data with training epochs can be seen. By the end of the training, very little difference in the mAP of Faster RCNN and SSD validation dat was obtained. It can be seen that Faster RCNN converges faster than SSD. The training process of Faster RCNN might appear to be more oscillating to be than SSD which could be due to the different batch sizes and optimizers being used by the two models. However, it should be noted that the scale of the two loss plots is different. The different batch size and optimizer could also be the reason for the Faster RCNN model converging to high validation mAP earlier than SSD since a batch size of 1 Faster RCNN leads to 24 times more gradient updates being performed than SSD with a batch size of 24.

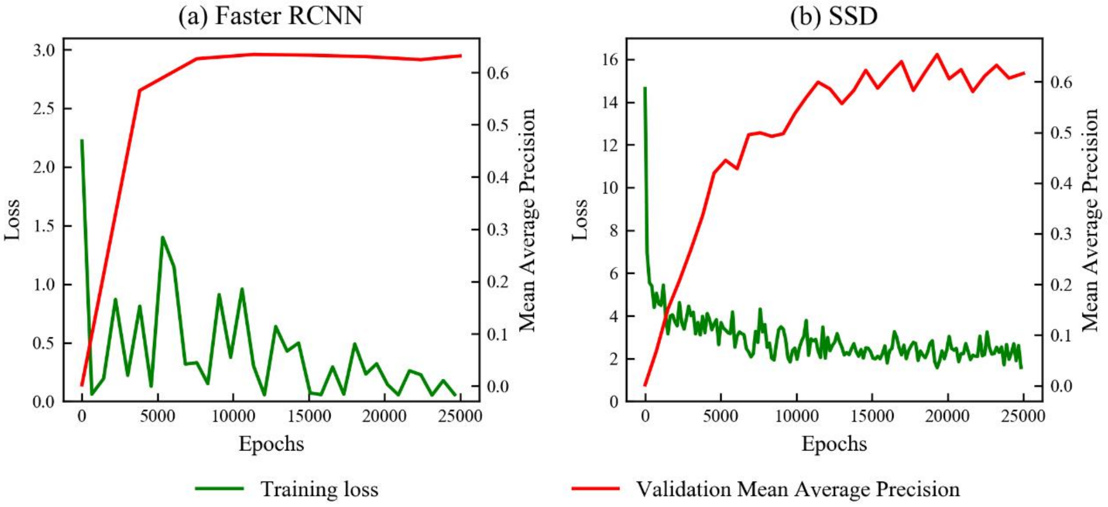


Figure 3.6. Change in training loss and Validation Mean Average Precision with number of epochs of (a) Faster RCNN and (b) SSD

3.3.2 Optimal IoU and confidence thresholds for Faster RCNN and SSD

In order to find the optimal threshold for IoU of the prediction boxes and ground truth boxes that would result in best performance of the model, precision recall curve was drawn using various confidence thresholds from 0 to 1 at various IoU thresholds ranging from 0.5 to 0.95 (Figure 3.7).

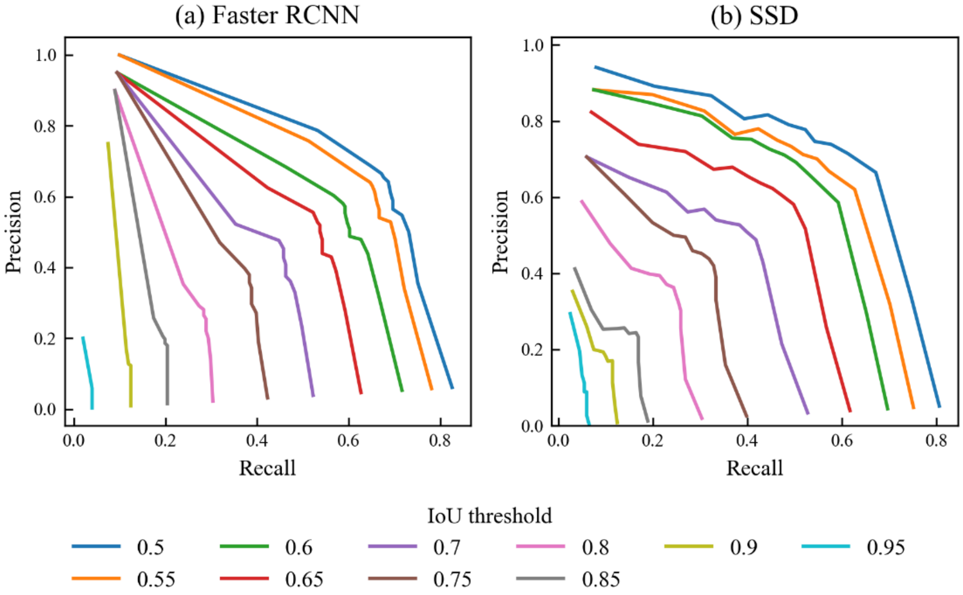


Figure 3.7. Precision-recall curve at different thresholds for IoU of the predicted box and ground truth box (a) Faster RCNN and (b) SSD

It can be seen that the area under the precision-recall curve is almost the same in case of Faster RCNN and SSD which explains the fact that the validation mAP during the final epochs as seen from the training graph was very similar (0.63 in Faster RCNN and 0.62 in SSD). Also, it can be seen that, both Faster RCNN and SSD achieve the maximum area under the precision-recall curve at an IoU threshold of 0.5 for the prediction box and ground truth box. Hence, for each ground truth box, among all prediction boxes with a confidence score greater than the threshold, the prediction box which has an IoU with that ground box greater than the threshold as well as the highest value of IoU among all prediction boxes is considered a true positive. All prediction boxes that were not a true positive with any ground truth box are regarded as false positives. The number of false negatives is equal to the number of ground truth box that does not have a corresponding true positive. With the optimal IoU threshold found for Faster RCNN and SSD, the following graph (Figure 3.8) was plotted to find the optimal confidence threshold for Faster RCNN and SSD that results in the best performance.

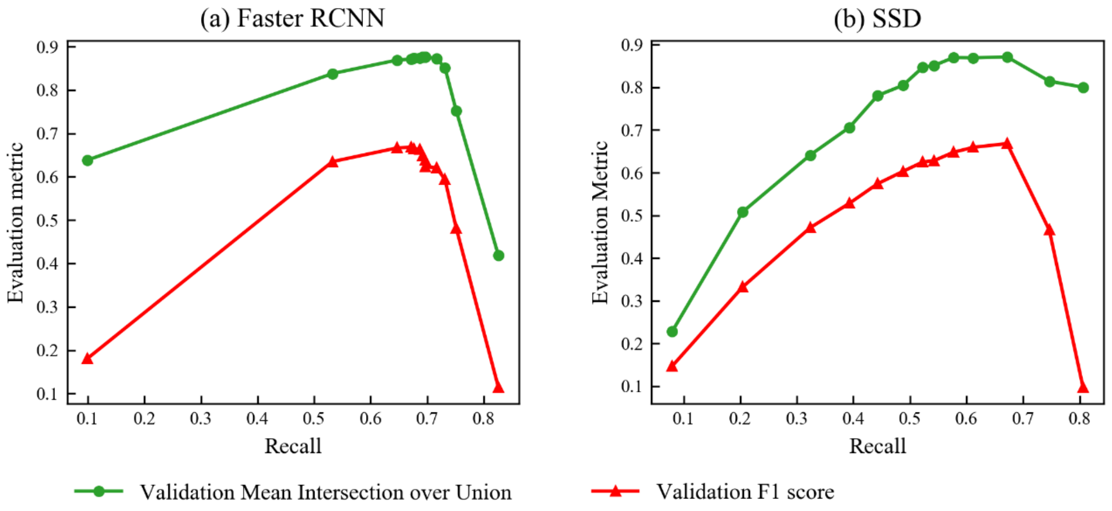


Figure 3.8. Change in IoU of output binary image and ground truth binary image as well as f1 score with change in recall

The above graph shows the change in f1 score and the mean IoU of the output binary image of the model with the ground truth binary image with change in recall. It was found that the recall at which the best performance of mean IoU and f1 score was observed was at a corresponding confidence threshold of 0.6 in case of Faster RCNN and 0.1 in case of SSD. It is to be noted that mean IoU here refers to the Intersection over Union of the whole binary model output image with the ground truth binary image whereas the IoU mentioned earlier was the Intersection over Union of individual prediction bounding boxes with individual ground truth bounding boxes.

3.3.3 Comparison of performance of Faster RCNN and SSD

Table 3.1 shows the precision, recall, f1 score, mean IoU of the model output binary image and the ground truth binary and the inference time of a 1152 × 1152 image. It can be seen that the precision, recall, f1 score and mean IoU of both the models were similar but the SSD model was slightly faster in execution than Faster RCNN. It is to be noted that the above performance was in case of Faster RCNN network that outputs 300 proposals from the region proposal network. However, Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy (2017) found that by reducing the number of proposals output by Faster RCNN, the inference time of Faster RCNN can be improved but with a slight cost in precision, recall and f1 score. Therefore, experiments were conducted to study the change in inference time, precision, recall, f1 score and mean IoU by varying the number of proposal boxes from the Faster RCNN network from 50 to 300 and the results are plotted in Figure 3.9.

Table 3.1. Performance of test data in Faster RCNN and SSD

Model	**Precision**	**Recall**	**F1 score**	**Mean IoU**	**Inference time of 1152 × 1152 image in seconds**
Faster RCNN	0.65	0.68	0.66	0.85	0.23
SSD	0.66	0.68	0.67	0.84	0.21

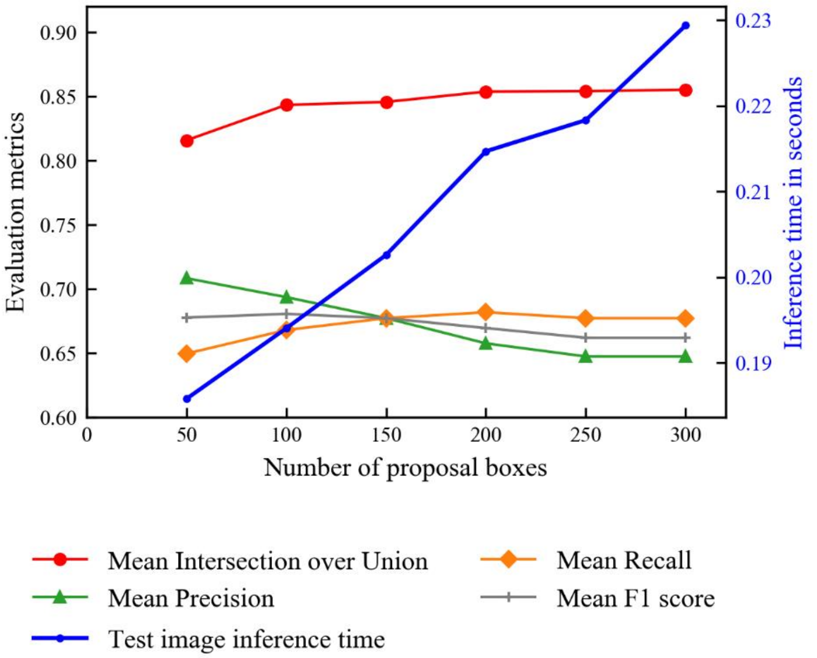
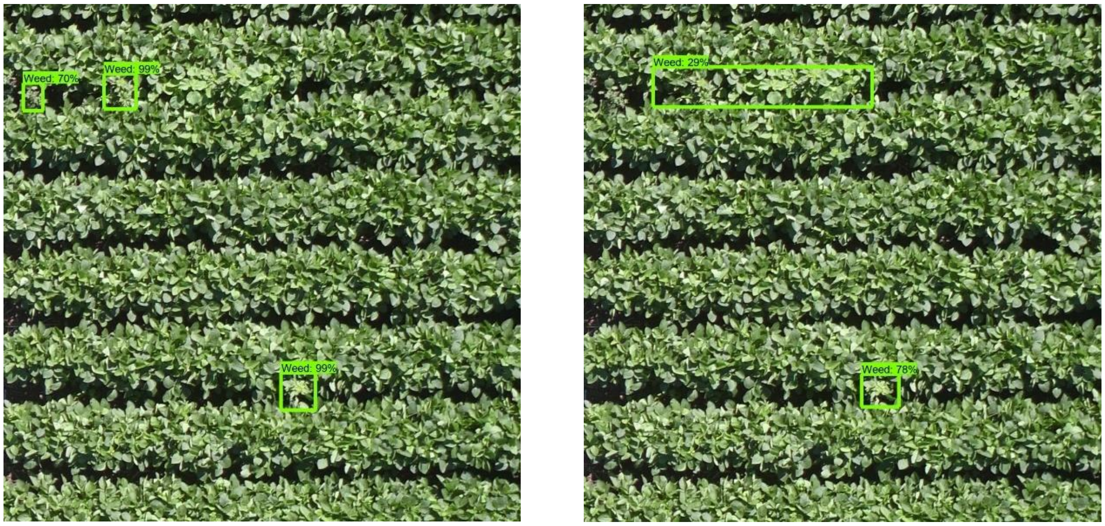


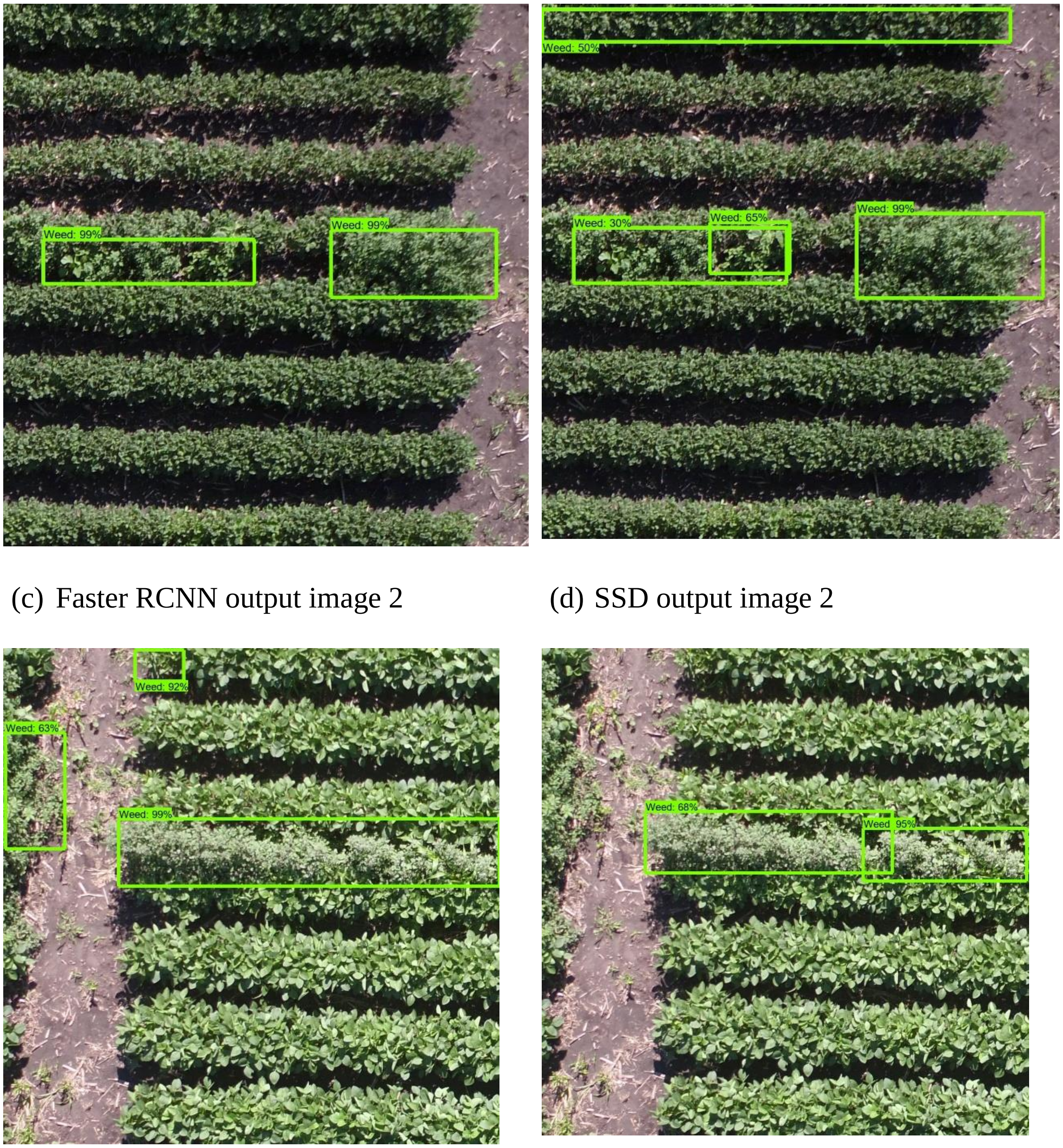
Figure 3.9. Change in evaluation metrics and inference time of Faster RCNN model with increase in number of proposals

It can be seen that the inference time of Faster RCNN has a linear time complexity with the number of proposal boxes output from the region proposal network. It can be seen that from 200 to 300 proposals, there was no change in performance of the model but the inference time decreased and hence it can be concluded that 200 proposals is the optimal number of proposals for this dataset. At 200 proposals, the inference time of Faster RCNN was 0.21 seconds which was the same as SSD. Hence, no difference in performance was found between Faster RCNN with 200 proposals and SSD in terms of evaluation metrics used in this study. However, it is to be noted that, even with same performance metric, Faster RCNN outputs weed objects with high confidence compared to SSD since the confidence threshold being used for Faster RCNN was 0.6 whereas it was a very low 0.1 for SSD. Though this threshold might result in the best performance with the current validation test, it might affect the generalization performance of the model in case of test dataset that is from a different location or from a field with different management practices. In such cases, the low threshold might lead to reduced precision. On visual observation of the outputs of all the 44 test images, it was found that in 41 images, both the images detected all the weed areas. Hence, in these images, the difference in IoU between the model output and the ground truth is only because of the slight displacements of the boundaries of the bounding boxes from each other. As mentioned in section 3.2.7, the low values of precision, recall and f1 score obtained is primarily because of the way these metrics are calculated since only one bounding box is considered as a true positive for one ground truth box whereas the model in case of some weed areas with slight discontinuities outputs multiple prediction boxes to detect those areas. Therefore, as mentioned earlier, mean IoU of the binary output image with the binary image of the ground truth is the appropriate metric. In 3 of the test images (shown in Figure 3.10), there was a difference in the output of Faster RCNN and SSD. In the output image 1, Faster RCNN couldn’t detect a small strip of weed between the crop rows but has been detected by SSD. But by looking at the confidence score of the weed object from SSD, it can be understood that SSD was able to detect this weed object only because of the very low confidence threshold set for it. Whereas in output image 2, it can be seen that SSD misclassified a row of soybean crop with herbicide drift injury as weed. Also, in case of output image 3, SSD could not detect the weeds on the left vertical border of the image. With both the failure areas being present in the border of the images, this might show the susceptibility of the SSD model in the image border. This could be due to the architecture of SSD that does detection of objects and classification into its class in a single shot, unlike Faster RCNN. Another possible reason could be that, by default, the API used to train both the models was resizing the input images to Faster RCNN to 600 × 600 whereas in case of SSD it was resized to 300 × 300. Therefore this further loss of detail in the input image compared to the Faster RCNN input image might have led to the misclassifications in the border. Hence, further study with the same input image resolution is needed for a fair comparison.



(a) Faster RCNN output image 1

(b) SSD output image 1



(c) Faster RCNN output image 2

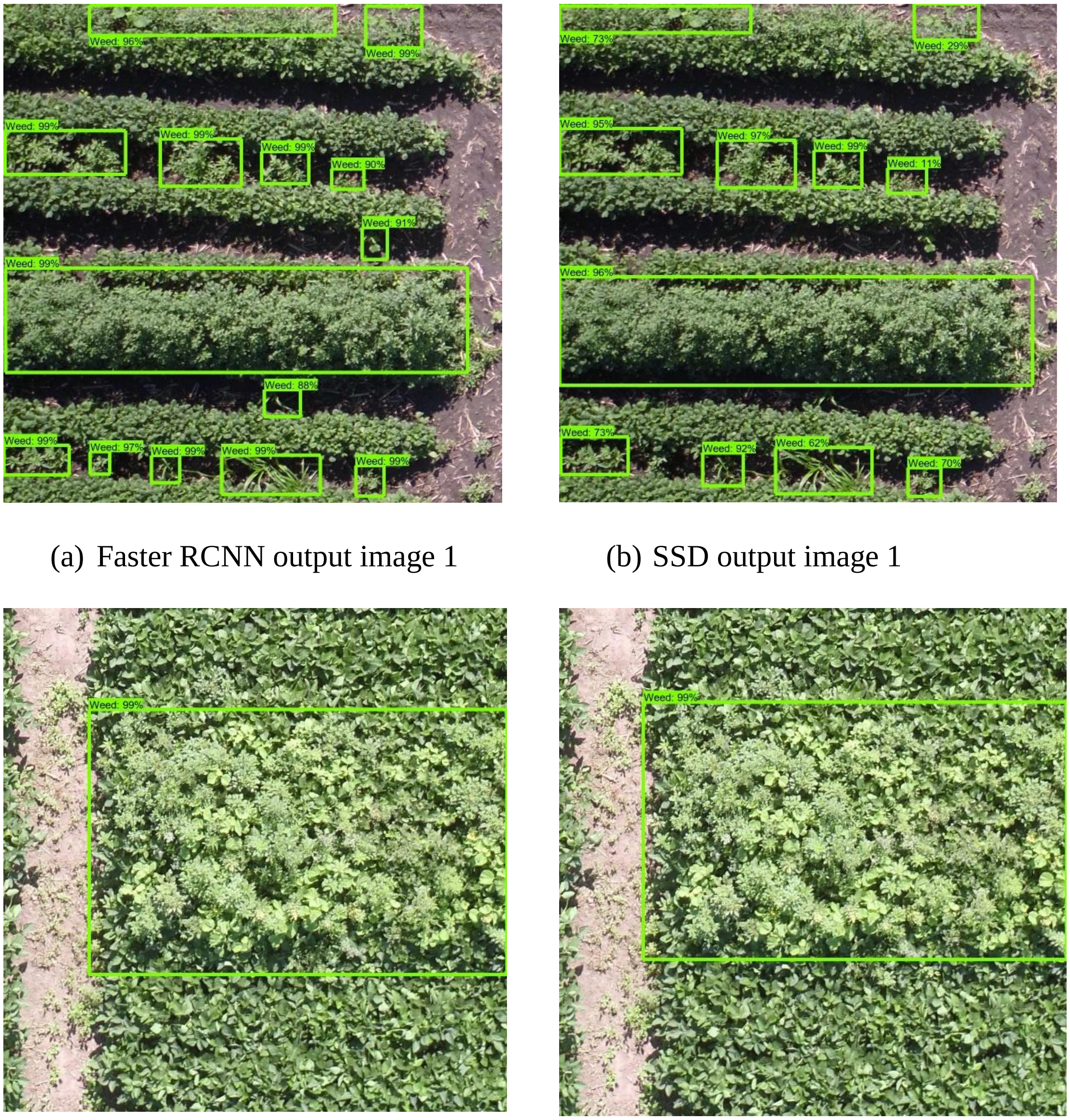
(d) SSD output image 2

(e) Faster RCNN output image 3

(f) SSD output image 3

Figure 3.10. Output images with discrepancies between Faster RCNN and SSD

Other than the above mentioned 3 images, Faster RCNN, as well as SSD, performed exceptionally well in detecting weed objects of various scales as seen in Figure 3.11. As mentioned earlier, it can be seen that though SSD detects all the weed objects that were detected Faster RCNN, the confidence of a lot of those predictions are very low and ended up as true positive because of the low confidence threshold.



(a) Faster RCNN output image 1

(b) SSD output image 1

(c) Faster RCNN output image 2

(d) Faster RCNN output image 2

Figure 3.11. Example output images with good model performance

Since by reducing the number of proposals to 200, Faster RCNN can be as fast SSD in terms of inference time, it can be concluded that for this application with respect to this dataset, Faster RCNN has better speed performance tradeoff.

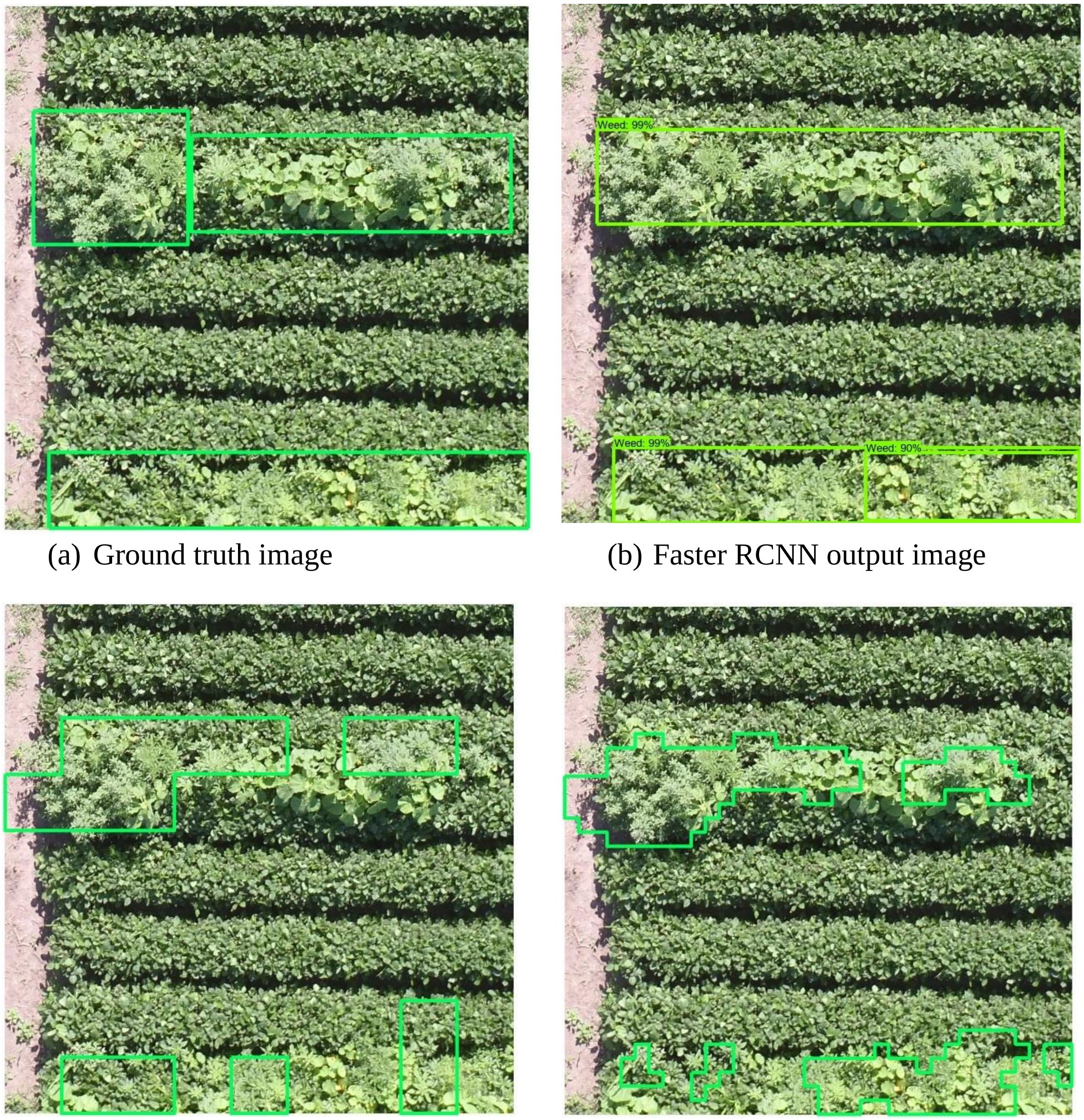
3.4 Comparison of performance of Faster RCNN and patch-based CNN

The Mobilenet v2 network trained on the training patches showed very high performance in classifying test patches with an f1 score of 0.98. But in order to evaluate its performance in detecting the weed objects in the sub-image and compare its performance with Faster RCNN object detection model, the overlapping approach explained earlier was used. The following table shows the mean IoU of the output binary image from Faster RCNN and patch-based CNN with the ground truth binary image. Also, the table shows the time taken to evaluate one sub-image by both the models.

Table 3.2. Performance of Faster RCNN and patch-based CNN in test sub-images

Model	**Mean IoU**	**Inference time in seconds for each sub-image (1152×1152)**
Faster RCNN with 200 proposals	0.85	0.21
Patch based CNN sliced with overlap	0.61	1.03
Patch based CNN sliced without overlap	0.6	0.22

It can be seen that Faster RCNN has better performance than patch-based CNN with overlap both in terms of mean IoU and inference time. But patch-based CNN without overlap has an inference time which is almost the same as Faster RCNN. The low values of IoU of patch based CNN without overlap were because of the coarse nature of this algorithm. Since each sub-image is split into 81 patches in this approach, weeds smaller in size would not be detected in this approach. Also, because of the way the patches are sliced, there could be a lot of patches with weeds and background in equal proportion. Whereas the Mobilenet model has only been trained with patches with only weed or only background and hence the model is prone to error in this approach. To reduce this error, the slicing with overlap approach is tested. Since, for each small block within a patch, the class is determined by majority vote in 8 patches, the problem of mixed patches can be solved to some extent. But the almost similar IoU of slicing with overlap and without overlap is because the ground truth binary image represents weed objects are rectangular boxes whereas output binary image from patch-based overlap approach consists of weed objects which are polygonal in nature because of the majority vote as can be seen in Figure 3.12. Therefore, patch-based CNN with overlap has better performance than the IoU value with ground truth image suggests. But the drawback of this approach is the very high inference compared to Faster RCNN and patch based RCNN without overlap. Further studies can be done with different levels of horizontal and vertical overlap and its influence on the inference time of this approach. But with the inference time of Faster RCNN is the same as the patch based CNN without overlap, any amount of overlap would lead to more patches to be evaluated than the non-overlap approach and hence greater inference time. Therefore, among the approaches investigated in this study, Faster RCNN has the best overall performance. But in order to implement this system for on-farm detection, further studies are needed to evaluate the performance of these approaches at higher altitudes. At an altitude of 20m in which this data was collected, it is practically impossible to cover the large soybean fields with the current limited battery capacity of the UAV systems. Therefore, evaluation of the performance of these models at low-resolution images from high altitude are needed for practical adoption of these systems. Similar to SSD, it can be seen that there is a higher misclassification rate of patches in the border of the images. In case of using this approach, it is suggested to collect images with some overlap such as 15% so that weed objects present in the border of one image end up in the interior of the next image.



(a) Ground truth image

(b) Faster RCNN output image

(c) Patch based CNN without overlap output image

(d) Patch based with overlap output image

Figure 3.12. Output images of patch based CNN and Faster RCNN

3.4 CONCLUSION

Faster RCNN and SSD object detection models were trained and evaluated for mid to late season weed detections in soybean fields. It was found that the Faster RCNN model with 200 box proposals and SSD had similar weed detection performance in terms of precision, recall, f1 score and IoU as well as similar inference time. But, the optimal confidence threshold of SSD was found to be 0.1 which resulted in lower confidence in case of weed objects detected whereas the optimal confidence threshold was found to be 0.6 in case of Faster RCNN which led to weed objects detected with higher confidence. Also, it was found that SSD was susceptible to misclassification in the border of some test images. These findings indicated that SSD might have relatively lower generalization performance than Faster RCNN for mid to late season weed detection in soybean using UAV imagery and hence Faster RCNN was concluded as the better performing model among the two. Between Faster RCNN and patch-based CNN, it was found that Faster RCNN had better weed detection performance than patch-based CNN with overlap as well as without overlap. The inference time of Faster RCNN was found to be similar as patch-based CNN without overlap but significantly lesser than patch-based CNN with overlap. Hence, Faster RCNN was found to be the best model in terms of weed detection performance and inference time among the different models compared in this study. By resampling high-resolution images to low-resolution images, the performance of Faster RCNN at different altitudes can be evaluated. Also, the inference time experiments at different altitudes should be performed on low computational power devices such as regular laptops and mini-PCs used for flight control of UAV systems. This would help understand the potential of using such devices for on-farm near real-time data processing.

3.5 ACKNOWLEDGEMENT

This project is based on research that was supported by the University of Nebraska Collaboration Initiative/System Science Seed Grant, and the funding supported by the Nebraska Agricultural Experiment Station through the Hatch Act capacity funding program (Accession Number 1011130) from the USDA National Institute of Food and Agriculture.

3.6 REFERENCES

Andrea, C.-C., Mauricio Daniel, B. B., & Jose Misael, J. B. (2017). Precise weed and maize classification through convolutional neuronal networks. In 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM) (pp. 1–6). IEEE. https://doi.org/10.1109/ETCM.2017.8247469

Bah, M. D., Dericquebourg, E., Hafiane, A., & Canals, R. (2019). Deep Learning Based Classification System for Identifying Weeds Using High-Resolution UAV Imagery (pp. 176–187). Springer, Cham. https://doi.org/10.1007/978-3-030-01177-2_13

CENTENARY REVIEW. (2019). https://doi.org/10.1017/S0021859605005708

Chollet, F. (2017). Transfer Learning Using Pretrained ConvNets. Retrieved from https://www.tensorflow.org/alpha/tutorials/images/transfer_learning

CHRISTENSEN, S., SØGAARD, H. T., KUDSK, P., NØRREMARK, M., LUND, I., NADIMI, E. S., & JØRGENSEN, R. (2009). Site-specific weed control technologies. Weed Research, 49(3), 233–241. https://doi.org/10.1111/j.1365-3180.2009.00696.x

de Castro, A., Torres-Sánchez, J., Peña, J., Jiménez-Brenes, F., Csillik, O., López-Granados, F., … López-Granados, F. (2018). An Automatic Random Forest-OBIA Algorithm for Early Weed Mapping between and within Crop Rows Using UAV Imagery. Remote Sensing, 10(3), 285. https://doi.org/10.3390/rs10020285

dos Santos Ferreira, A., Matte Freitas, D., Gonçalves da Silva, G., Pistori, H., & Theophilo Folhes, M. (2017). Weed detection in soybean crops using ConvNets. Computers and Electronics in Agriculture, 143, 314–324. https://doi.org/10.1016/J.COMPAG.2017.10.027

Dyrmann, M., Mortensen, A., Midtiby, H., & Jørgensen, R. (2016). Pixel-wise classification of weeds and crops in images by using a fully convolutional neural network. In Proceedings of the International Conference on Agricultural Engineering, Aarhus, Denmark (pp. 26-29). Retrieved from https://core.ac.uk/download/pdf/50630438.pdf

G. E. Meyer, G. E., T. Mehta, T., M. F. Kocher, M. F., D. A. Mortensen, D. A., & A. Samal, A. (1998). TEXTURAL IMAGING AND DISCRIMINANT ANALYSIS FOR DISTINGUISHINGWEEDS FOR SPOT SPRAYING. Transactions of the ASAE, 41(4), 1189–1197. https://doi.org/10.13031/2013.17244

Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F.,

… Toulmin, C. (2010). Food security: the challenge of feeding 9 billion people. Science (New York, N.Y.), 327(5967), 812–818. https://doi.org/10.1126/science.1185383

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7310-7311). Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Huang_SpeedAccuracy_Trade-Offs_for_CVPR_2017_paper.html

Huang, H., Deng, J., Lan, Y., Yang, A., Deng, X., & Zhang, L. (2018). A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery. PLOS ONE, 13(4), e0196302. https://doi.org/10.1371/journal.pone.0196302

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012a). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networ

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networ

Kuwata, K., & Shibasaki, R. (2015). Estimating crop yields with deep learning and remotely sensed data. In 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 858–861). IEEE. https://doi.org/10.1109/IGARSS.2015.7325900

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In Proceedings of the IEEE, 86(11), 2278-2324. Retrieved from http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

Liu, D., & Xia, F. (2010). Remote Sensing Letters Assessing object-based classification: advantages and limitations Assessing object-based classification: advantages and limitations. https://doi.org/10.1080/01431161003743173

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). SSD: Single Shot MultiBox Detector. https://doi.org/10.1007/978-3-319-46448-0_2

López-Granados, F., Torres-Sánchez, J., Serrano-Pérez, A., de Castro, A. I., Mesas-Carrascosa, F.-J., & Peña, J.-M. (2016). Early season weed mapping in sunflower using UAV technology: variability of herbicide treatment maps against weed thresholds. Precision Agriculture, 17(2), 183–199. https://doi.org/10.1007/s11119-015-9415-8

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2018). Joint Stem Detection and Crop-Weed Classification for Plant-Specific Treatment in Precision Farming. In 2018 IEEE/RSJ International Conference on Intelligent Robots and

Systems (IROS) (pp. 8233–8238). IEEE. https://doi.org/10.1109/IROS.2018.8593678

Lottes, P., Behley, J., Milioto, A., & Stachniss, C. (2018). Fully Convolutional Networks With Sequential Information for Robust Crop and Weed Detection in Precision Farming. IEEE Robotics and Automation Letters, 3(4), 2870–2877. https://doi.org/10.1109/LRA.2018.2846289

Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., & Stachniss, C. (2017). UAV-based crop and weed classification for smart farming. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3024–3031). IEEE. https://doi.org/10.1109/ICRA.2017.7989347

Milioto, A., Lottes, P., & Stachniss, C. (2018). Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2229–2235). IEEE. https://doi.org/10.1109/ICRA.2018.8460962

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7, 1419. https://doi.org/10.3389/fpls.2016.01419

Peña, J. M., Torres-Sánchez, J., de Castro, A. I., Kelly, M., & López-Granados, F. (2013). Weed Mapping in Early-Season Maize Fields Using Object-Based Analysis of Unmanned Aerial Vehicle (UAV) Images. PLoS ONE, 8(10), e77151.

https://doi.org/10.1371/journal.pone.0077151

Pérez-Ortiz, M., Peña, J. M., Gutiérrez, P. A., Torres-Sánchez, J., Hervás-Martínez, C., & López-Granados, F. (2015). A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method. Applied Soft Computing, 37, 533–544. https://doi.org/10.1016/J.ASOC.2015.08.027

Rahnemoonfar, M., Sheppard, C., Rahnemoonfar, M., & Sheppard, C. (2017). Deep Count: Fruit Counting Based on Deep Simulated Learning. Sensors, 17(4), 905. https://doi.org/10.3390/s17040905

Sa, I., Chen, Z., Popovic, M., Khanna, R., Liebisch, F., Nieto, J., & Siegwart, R. (2018). weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. IEEE Robotics and Automation Letters, 3(1), 588–595. https://doi.org/10.1109/LRA.2017.2774979

Sa, I., Popović, M., Khanna, R., Chen, Z., Lottes, P., Liebisch, F., … Siegwart, R. (2018). WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming. Remote Sensing, 10(9), 1423. https://doi.org/10.3390/rs10091423

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Retrieved from http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html

Sankaran, S., Khot, L. R., Espinoza, C. Z., Jarolmasjed, S., Sathuvalli, V. R., Vandemark,

G. J., … Pavek, M. J. (2015). Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review. European Journal of Agronomy, 70, 112–123. https://doi.org/10.1016/J.EJA.2015.07.004

Song, X., Zhang, G., Liu, F., Li, D., Zhao, Y., & Yang, J. (2016). Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model. Journal of Arid Land, 8(5), 734–748. https://doi.org/10.1007/s40333-016-0049-0

Steen, K., Christiansen, P., Karstoft, H., Jørgensen, R., Steen, K. A., Christiansen, P., … Jørgensen, R. N. (2016). Using Deep Learning to Challenge Safety Standard for Highly Autonomous Machines in Agriculture. Journal of Imaging, 2(1), 6. https://doi.org/10.3390/jimaging2010006

T. F. Burks, T. F., S. A. Shearer, S. A., & F. A. Payne, F. A. (2000). CLASSIFICATION OF WEED SPECIES USING COLOR TEXTURE FEATURES AND DISCRIMINANT ANALYSIS. Transactions of the ASAE, 43(2), 441–448. https://doi.org/10.13031/2013.2723

Torrey, L., & Shavlik, J. (2010). Transfer learning. In In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI Global. Retrieved from https://www.igi-global.com/chapter/transfer-learning/36988

Tzutalin. (2015). LabelImg. Git code.

Weis, M., Gutjahr, C., Rueda Ayala, V., Gerhards, R., Ritter, C., & Schölderle, F. (2008).

Precision farming for weed management: techniques. Gesunde Pflanzen, 60(4), 171–181. https://doi.org/10.1007/s10343-008-0195-1

Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—a worldwide overview. Computers and Electronics in Agriculture, 36(2–3), 113–132. https://doi.org/10.1016/S0168-1699(02)00096-0

CHAPTER 4 OPERATIONAL FEASIBILITY OF NEAR REAL TIME MID TO LATE SEASON WEED DETECTION IN COMMERCIAL SCALE SOYBEAN FIELDS USING UAV AND MACHINE LEARNING

This manuscript has been prepared for journal submission

4.1 INTRODUCTION

Advancements in the unmanned aerial vehicle (UAV) technology in the past decade has led to various applications in agriculture such as irrigation management, nitrogen management, pest detection, weed detection, and high throughput phenotyping. The main advantage of UAV in remote sensing is in their ability to cover a large area and collect high-resolution aerial imagery using various sensors such as RGB camera, multispectral camera, hyperspectral camera, thermal camera and LIDAR (Sankaran et al., 2015). In case of most of the applications mentioned above, UAVs are used as a sensing platform to collect data which is then processed later away from the field. A typical workflow involves using UAV to collect images with an overlap which are then stitched using computer vision algorithms into an orthorectified map to visualize the whole field (Pérez, Agüera, & Carvajal, 2013). It includes algorithms such as feature detection and pixel matching on a large number of captured images and hence needs computational power and time. In most of the applications, the orthomosaic from the UAV in different spectral bands are then combined as different vegetation indices and used to create prescription maps for variable rate application of farm inputs (Rasmussen et al., 2016).

Since these variable rate application operations are not affected by a latency period of a day or two in processing, the stitching of images to create the orthomosaic is usually done away from the farm because of the computational constraints mentioned above. However, in case of using the UAV to complement crop scouting, if the images can be processed near real time, it will help the farmers to scout the problem areas thereby saving time and effort.

Several studies have looked at various applications of UAVs in different types of crops. However, the limited battery time of the UAVs is regarded as the main barrier for adoption of this technology by the farmers, especially in case of large farms growing row crops such as corn, soybean. But, if examined further, it can be seen that the major reason for the long time needed to cover large areas is the practice of collecting images with more than 50% overlap along and across the flight path. This is done since overlapping images are needed to use pixel matching algorithms to stitch the images into an orthomosaic. However, in case of applications such as crop scouting, providing near real-time information about the problem areas in the field from the images can be more valuable than the orthomosaic with high geometric accuracy provided by the pixel matching algorithms. With the extraordinary performance of convolutional neural networks (CNNs) in image related tasks such as classification and object detection, several studies have looked at the performance of CNNs for various agricultural applications (Kamilaris & Prenafeta-Boldú, 2018). However, there is limited work on the evaluation of the inference time of CNNs on low computational power devices for on-farm data processing.

Weeds are one of the major crop yield-limiting factors. Several studies have focused on UAV based early season weed detection in several crops. Mid to late season weeds, though might not affect the crop yield in that season, will produce a large number of seeds and cause problems for several years in the future. In Chapter 1 and Chapter 2, two different approaches have been studied in detail, and their performance and inference time has been evaluated using imagery captured at 20m altitude. However, for farmers to adopt these systems, the feasibility of these systems for the commercial scale fields need to be evaluated. Therefore, the objective of this study is to evaluate the operational feasibility of data collection and near real-time data processing of UAV based mid to late season weed detection systems for commercial scale soybean fields.

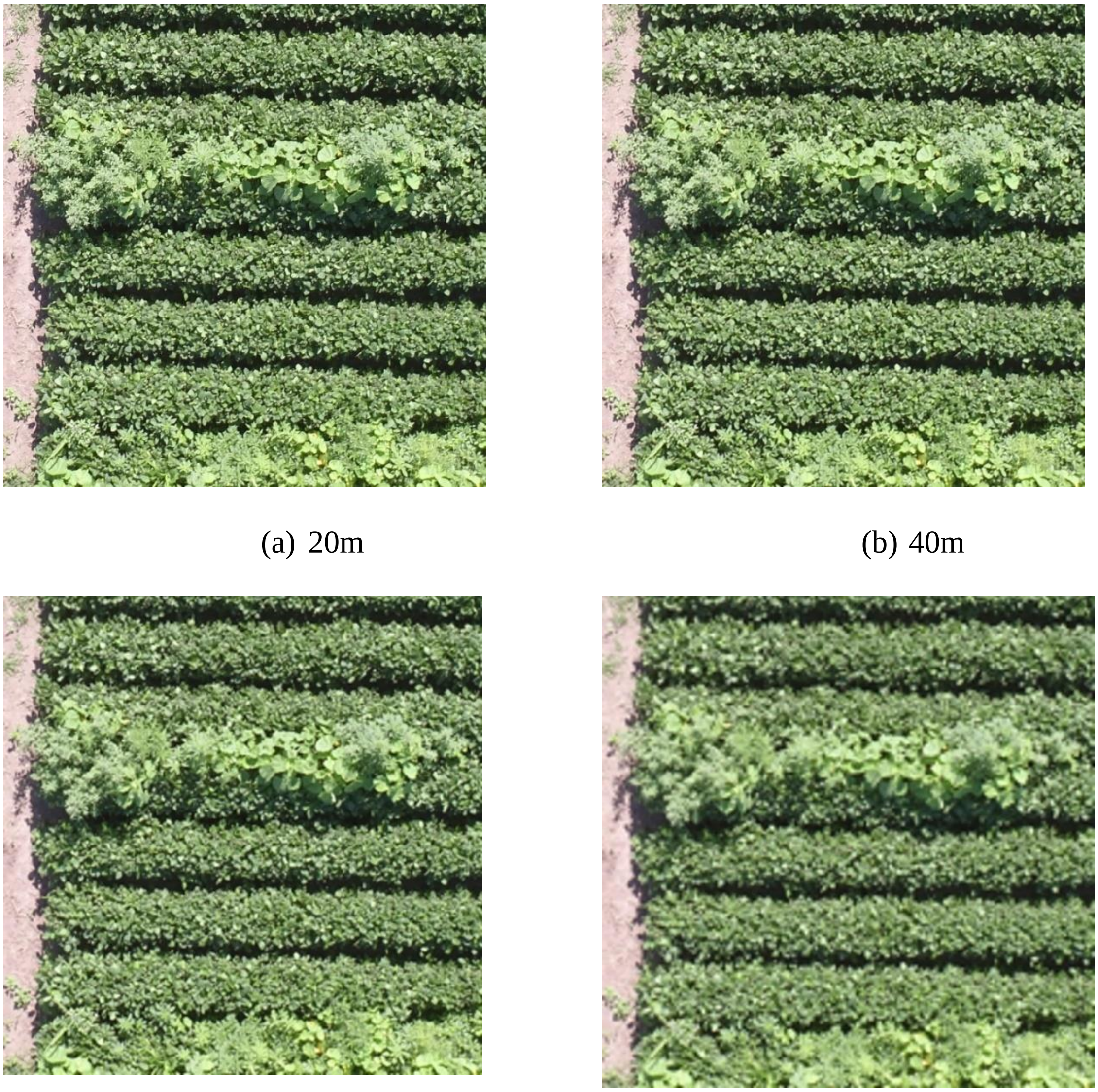
The specific objectives of the study are

1. To evaluate the performance of patch-based CNN and Faster RCNN weed detection systems in terms of weed detection performance using precision, recall, f1 score and mean Intersection over Union (IoU), and inference time at the different spatial resolutions of input images
2. To estimate the time needed for data collection and data processing at different data collection altitudes for a virtual square soybean field of quarter section area using the specified sensor and weed detection models
3. Discuss the potential of near real-time weed detection system using existing technologies

4.2 METHODOLOGY

4.2.1 Resizing of images to simulate images captured at different flight altitudes

As mentioned in earlier chapters, the data used in this study was collected using the DJI Zenmuse X5R camera at 20m altitude above ground level which corresponds to a spatial resolution of 0.5 cm/pixel for this particular sensor. In Chapter 2, the performance of Faster RCNN and patch based CNN approaches on mid to late season weed detection has been studied in detail. In order to evaluate the performance of the models at different spatial resolutions, the images were resized by bicubic interpolation using the ‘resize’ function in Matlab Image processing toolbox to images of 4 different spatial resolutions namely 0.5 cm/pixel, 1 cm/pixel, 2cm/pixel, 3cm/pixel. These 4 resolutions correspond to flight altitudes 20m, 40m, 80m, and 120m respectively for this sensor configuration (Figure 4.1). After resolution reductions, in order to maintain the same dimension of the input images for comparison purpose, the images were resized back to the original image dimension by bicubic interpolation. For example, in case of 1 cm/pixel resolution, the original image of size 1152 × 1152 pixels was resized to 576 × 576 pixels and was again resized back to 1152 × 1152 pixels. In this case, even though the field of view and the number of pixels in the image remains the same, the detail in the image is reduced and hence the image becomes equivalent to have been captured at 1 cm/pixel resolution. In case of both Faster RCNN and patch-based CNN without overlap, the performance of two models was evaluated – the performance at different altitudes of a model trained with only images at 20m and a model trained with images at all altitudes. Mean intersection over union, precision, recall and f1 score were the metrics used to evaluate the performance of Faster RCNN at different simulated flight altitudes. In case of patch-based CNN, only mean intersection over union was used for evaluation.



(a) 20m

(b) 40m

(c) 80m

(d) 120m

Figure 4.1. Example showing an image resized to 4 different altitudes (a) original image taken at 20m AGL with 0.5 cm/pixel resolution (b) image with reduced resolution simulating one taken at 40m AGL with 1cm/pixel resolution (c) image with reduced

resolution simulating one taken at 80m AGL with 2 cm/pixel resolution (d) image with reduced resolution simulating one taken at 120m AGL with 3 cm/pixel resolution

4.2.2 Estimation of flight time and data processing time at different altitudes

In order to investigate the implementation of a UAV based near real-time mid to late season weed detection system, a case study based approach has been followed. Since this study is focused on large row crop farms in the US Midwest, a squared area of 160 acres was created in the flight control software for this case study. Using the specifications of the DJI Zenmuse X5R camera that has been used in this study, the flight time and the number of images that would be captured at different flight altitudes were calculated using the DJI Ground Station Pro flight planning application (DJI, Shenzhen, China) for the virtual field. It was assumed that a forward and side overlap of 15% was maintained during data capture. It is to be noted that an overlap of more than 50% is not used as is the usual case since the generation of orthomosaic using pixel-matching algorithm is not the objective of this study. But, a buffer of 15% overlap is used because in Chapter 2, it has been found that in some case, Single Shot Detector (SSD) algorithm, as well as patch based convolutional neural networks (CNN), has more misclassification in the edges of the images. Hence, by using a 15% overlap, a weed area which was in the edge in one image would still be covered in the interior in the next image and so has very less chance of getting misclassified if SSD and patch based CNN models are used. In Chapter 1 and Chapter 2, the inference time experiments were run on devices with high computational power and Graphical Processing Unit (GPU). But in order to evaluate the feasibility of on-farm data processing using regular laptops and mini-PCs, the inference time experiments have been conducted using a Microsoft Surface Pro mini PC with 8 GB of RAM and Intel i5 processor with no GPU. The inference time was then used along with the number of image information to calculate the data processing time for the virtual field.

4.3 RESULTS AND DISCUSSION

4.3.1 Performance of Faster RCNN and patch based CNN models at different spatial resolutions

Figure 4.2 shows the change in evaluation metrics such as mean Intersection over Union (IoU) of the model output binary image and ground truth image, precision, recall and f1 score of Faster RCNN model with the change in the altitude at which the test images were captured. Two different Faster RCNN models – a model trained only with images at 20m and a model trained with resized images equivalent to all the altitudes (20m, 40m, 80m, 120m) are compared. It can be seen that the performance metrics remain constant at all test image altitudes in case of a model trained with images from all altitudes whereas the model trained with images from only 20m performs well for test images at 40m but its performance falls drastically thereafter. This shows that the model trained only with images at 20m can only generalize up to an altitude an 40m after which the loss of detail in the image is so high that the parameters learned in the network do not generalize well.

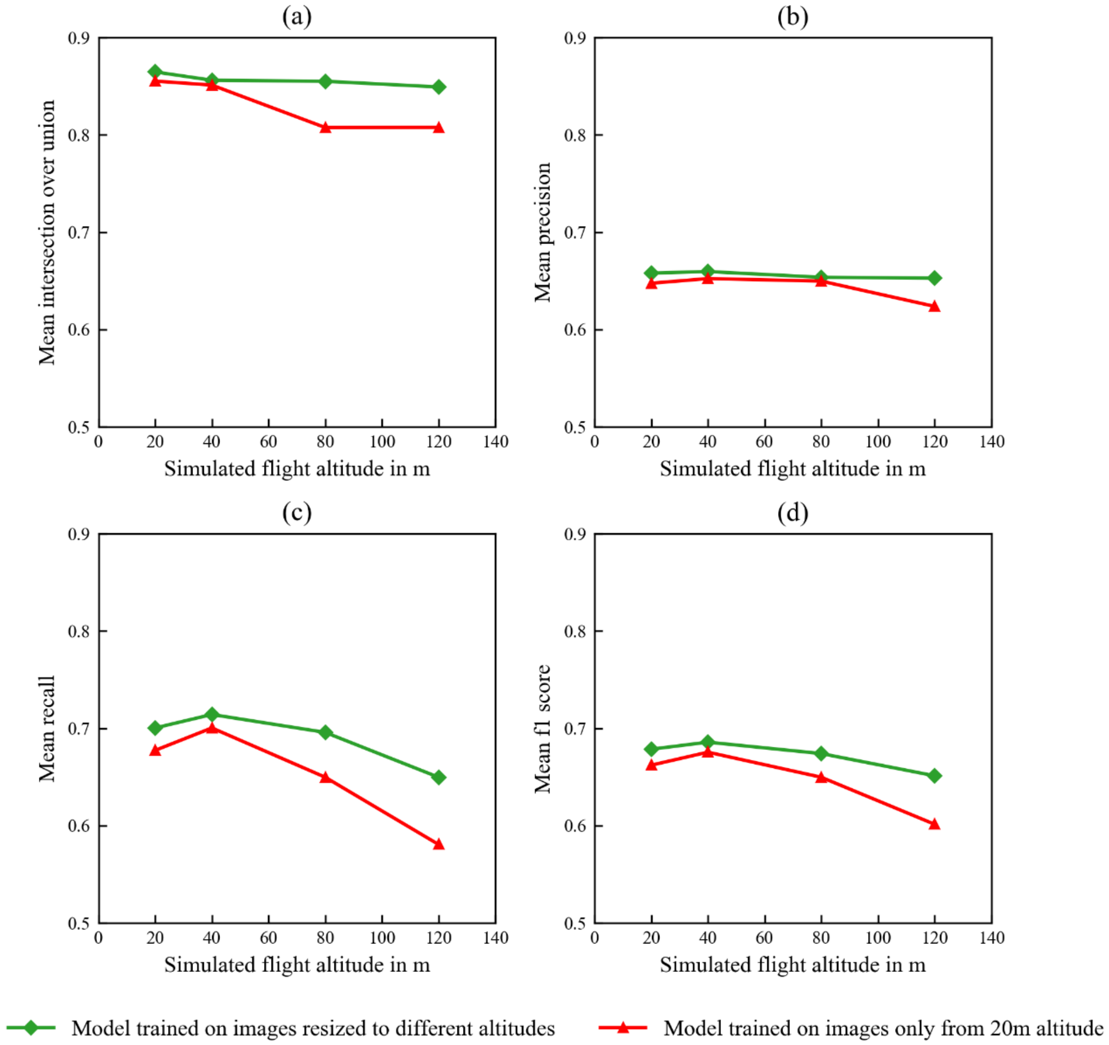


Figure 4.2. Performance of Faster RCNN on images at different altitudes

Figure 4.3 shows the change in mean Intersection over Union of the output binary image from the patch based CNN algorithm without overlap with the ground truth binary image. It is to be noted that in case of using patch-based CNN algorithm for test images from higher altitudes, the size of the patches that are cropped from the big image has to be reduced accordingly. For example, in case of 16-megapixel image with 0.5 cm/pixel spatial resolution in Chapter 1 and 2, a patch size of 128 × 128 was used. This size was chosen based on the approximate number of pixels that covered the crop or weed row. Since the width of a crop row at that resolution was found to be 128, the above dimension was chosen for the patches. But in case of test images at an altitude greater than 40m, in proportion to an increase in spatial resolution, the patch size has to be reduced for such images. In this case, as mentioned earlier, each test image was resized to a lower resolution and then resized back to the original number of pixels but with details lost. Therefore, testing with the same patch size of 128 × 128 in this resized image is equivalent to testing the performance of reduced patch size in higher altitude images. Because of the reduced patch size in higher altitudes, the number of images to be evaluated will increase. In Chapter 1, we found that even though patch based CNN with overlap shows better performance than patch based CNN without overlap, it is significantly slower because of the large number of images to be evaluated. Therefore, in this altitude experiments, only the patch-based CNN without overlap is considered. It can be seen in Figure 4.3 that when the model is trained with patches from all altitude, the mean IoU, though slightly decreases, does not show significant change. However, in case of the model trained only with patches at 20m, the mean IoU decreases significantly showing the poor generalization performance. Also, comparing with Figure 4.2, it can be seen that the generalization performance of Faster RCNN model trained with only 20m images is significantly better than patch based CNN without overlap model trained with only 20m images since the decrease in mean IoU in Faster RCNN has been found to be lesser than that of patch-based CNN without overlap.

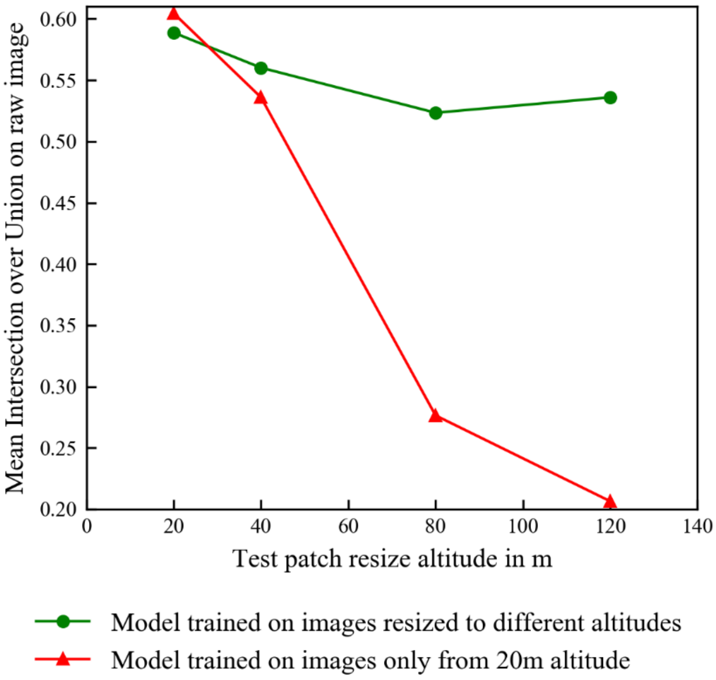


Figure 4.3. Change in mean Intersection over Union of patch-based CNN output with the test image altitude

4.3.2 Estimation of flight time and data processing time at different altitudes

Figure 4.4 and Table 4.1 shows the number of images that will be captured and the time to fly using DJI Zenmuse X5R camera at 15% forward and side overlap to cover a large square shaped field of 160-acre area. It can be seen that, in case of 80m and 120m altitude, the number of images and the flight time increase by small amounts whereas with further decrease in altitude, the number of images as well as flight time increase significantly. It is to be noted that in case of 80m and 120m altitude, the flight time is less than 20 minutes. On average, most of the commercially available UAV platforms have an actual flight time of 20 minutes. Therefore, with 80m and 120m altitude, a 160-acre field can be flown without having to use additional batteries.

Table 4.1. Number of images and flight time at different altitudes calculated to cover a 160 acre field with 15% overlap at different altitude. Results were calculated based on a camera with 4608 × 3456 pixels and a focal length of 15mm

Flight altitude	**Ground sampling distance**	**Flight speed in miles per hour**	**Number of images**	**Flight time in minutes**
20 m	0.5 cm	16.4	2377	93.1
40 m	1 cm	21.9	594	36.4
80 m	2 cm	21.9	154	19.7
120 m	3 cm	22.2	64	11.4

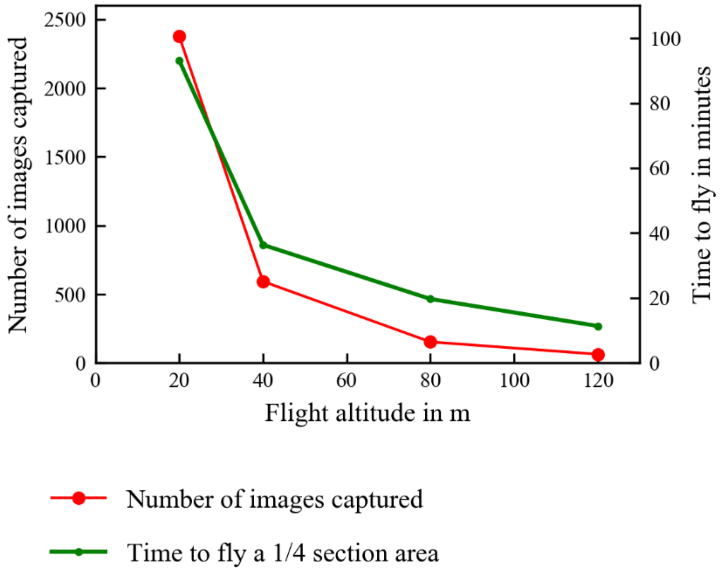


Figure 4.4.Change in number of images captured and the time to fly at different flight altitudes for 160 acres square shaped field

Figure 4.5 shows the change in time needed to process an image of size 1152 × 1152 at different altitudes in case of Faster RCNN and patch based CNN without overlap. As mentioned in 4.3.1, in case of patch-based CNN, at higher altitudes, the patch size has to be reduced proportionally to the change in altitude from 20m. Therefore, the number of patches to be evaluated for a test image collected at 120m will be significantly higher than the number of patches to be evaluated for a test image collected at 40m since the patch size in terms of pixels varies. Hence, the inference time for an 1152 × 1152 test image will vary at different altitudes for patch-based CNN whereas, in case of Faster RCNN, it remains the same.

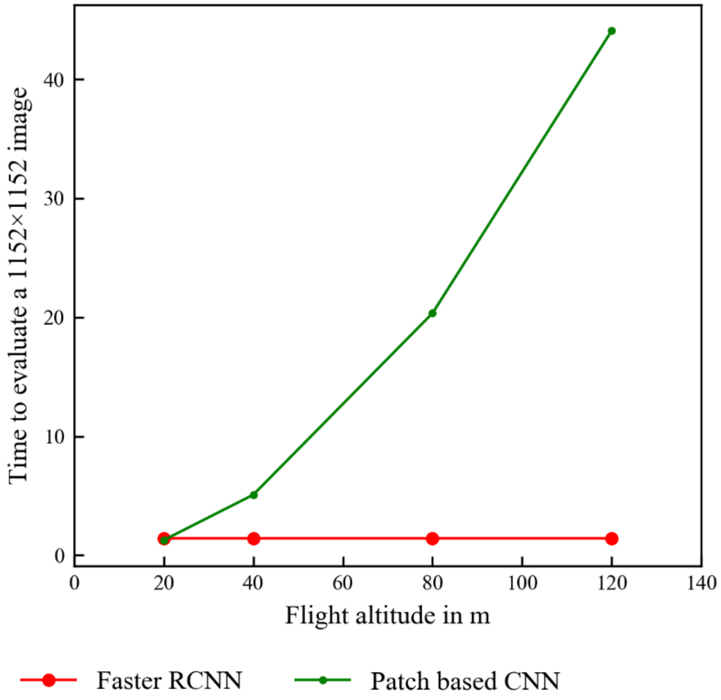


Figure 4.5. Change in inference time of an 1152 × 1152 image at different altitudes

At 20m altitude, with a spatial resolution of 0.5cm/pixel and patch size of 128×128, the patch covers an area of 64cm × 64cm on the ground. For different flight altitudes with different spatial resolution in images, the proportional patch sizes were calculated that would cover the same 64cm × 64 cm on the ground. Then, by combining the information from Figure 4.4 and Figure 4.5, the time taken for data processing at different altitudes for Faster RCNN and patch based CNN was calculated and is shown in Figure 4.6 and. It can be seen that, in case of Faster RCNN, since the inference time of the test image remains the same at all altitudes, the time taken for data processing decreases with increase in altitude in proportion to the decrease in the number of images captured at different altitudes. However, in case of patch-based CNN, since with the increase in altitude even though there is a decrease in the number of images captured, the patch size decreases and hence the number of patches to be evaluated increases proportionally. So the data processing time almost remains the same at all altitudes in case of patch-based CNN. Therefore patch based CNN is not recommended for near real-time on farm data processing.

Table 4.2. Time taken for data processing at different altitudes for Faster RCNN and patch based CNN without overlap

Altitude	**Faster RCNN data processing time for 160 acre field in minutes**	**Patch based CNN data processing time for 160 acre field in minutes**
20m	662. 84	604.5
40m	165.64	604.24
80m	42.94	626.62
120m	17.85	564.43

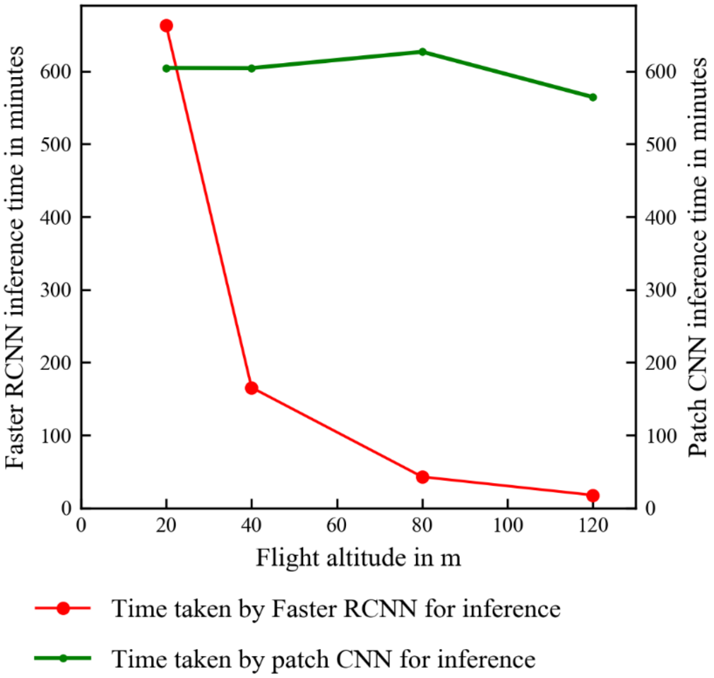


Figure 4.6. Time taken for data processing at different flight altitudes

Hence, it can be seen from the results that, in case of using DJI Zenmuse X5R camera and Faster RCNN based weed detection model, data collected from 160 acres of field 120m altitude can be processed on farm in about 18 minutes of time using a mini PC. Persson & Andersson, 2016) studied different orthomosaic generating algorithms for on-farm orthomosaic creation in near real-time as data is collected from a DJI Matrice 100 UAV system. It was found that, using DJI mobile and on board SDK, images captured by DJI UAV systems can be transmitted real time thereby enabling near real time processing. A 12 megapixel took about 8 seconds to download. In our case, since the image size is 16 megapixel, multiplying by the corresponding factor, it would take about

10.6 seconds to download an image from DJI Zenmuse X5R camera. Therefore, by downloading the images in near real time, before the completion of the flight, most of the images can be downloaded and processed. Also, Persson & Andersson, (2016) found that simple image stitching methods such as cropping and alpha blending, though less accurate, can be done in near real time and can be used as a visualization tool in our weed detection system. Since, the raw 16-megapixel image has GPS coordinates available as metadata, using the pitch, roll and yaw information, each image can be converted to an orthorectified image which is in nadir view. This would enable to use simple interpolation of GPS coordinate of the middle of the image to obtain GPS of the all the pixels in the image thereby enabling georeferencing of the weed output binary image from the model. The georeferencing accuracy of this system is limited by the accuracy of the GPS coordinates of the image which will be improved in the near future with advancements in low-cost RTK GPS technology.

4.4 CONCLUSION

Faster RCNN and patch-based CNN model was trained with images of different spatial resolutions and their weed detection performance at various spatial resolutions was tested using precision, recall, f1 score and mean IoU as evaluation metrics. It was found that both the models had similar weed detection performance at all spatial resolutions when trained with images of different spatial resolutions whereas in case of a model trained with only one spatial resolution, the models did not perform well at other spatial resolutions. Considering a virtual square shaped soybean field of 160 acres area, the time needed to capture UAV imagery using a DJI Zenmuse X5R camera with 15%

forward and side overlap and the time needed to process all these images using Faster RCNN and patch-based CNN model in a regular laptop was estimated at different altitudes. It was found that at an altitude of 120m, the 160 acre virtual soybean field can be captured in aerial imagery using the mentioned sensor within 12 minutes and the captured images can be processed using Faster RCNN model in a Microsoft surface pro laptop within 18 minutes. In case of using patch-based CNN approach, the data processing time remained almost same at more than 550 minutes at all altitudes. Hence, it was concluded on farm near real time mid to late season weed detection in commercial scale fields is feasible with existing UAV sensor technology and regular laptops by capturing imagery at the maximum legally permissible altitude of 120m and Faster RCNN model. Hence, by developing a mobile application for regular laptops and mobile devices, Faster RCNN model can be used for near real-time weed detection using mobile phones or miniPCs used for flight control of the UAV system. This system thus shows a processing workflow to help overcome the data processing bottlenecks for near real-time applications to aid crop scouting.

4.5 ACKNOWLEDGEMENT

This project is based on research that was supported by the University of Nebraska Collaboration Initiative/System Science Seed Grant, and the funding supported by the Nebraska Agricultural Experiment Station through the Hatch Act capacity funding program (Accession Number 1011130) from the USDA National Institute of Food and Agriculture.

4.6 REFERENCES

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70–90. https://doi.org/10.1016/J.COMPAG.2018.02.016

Pérez, M., Agüera, F., & Carvajal, F. (2013). LOW COST SURVEYING USING AN UNMANNED AERIAL VEHICLE. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 40, 311-315. Retrieved from www.microdrones.com

Persson, D., & Andersson, J. (2016). Real-time image processing on handheld devices and UAV. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2:947096

Rasmussen, J., Ntakos, G., Nielsen, J., Svensgaard, J., Poulsen, R. N., & Christensen, S. (2016). Are vegetation indices derived from consumer-grade cameras mounted on UAVs sufficiently reliable for assessing experimental plots? European Journal of Agronomy, 74, 75–92. https://doi.org/10.1016/J.EJA.2015.11.026

Sankaran, S., Khot, L. R., Espinoza, C. Z., Jarolmasjed, S., Sathuvalli, V. R., Vandemark, G. J., … Pavek, M. J. (2015). Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review. European Journal of Agronomy, 70, 112–123. https://doi.org/10.1016/J.EJA.2015.07.004

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

Even though several studies have focused on using UAV imagery and machine learning to detect early season weeds, there is no literature focused on automated detection of mid to late season weeds. In my research, patch-based classification using conventional machine learning as well as CNN and object detection using Faster RCNN and SSD models were studied for mid to late season weed detection in UAV imagery. Patch based classification method using conventional machine learning models such as support vector machine, logistic regression, ANN and KNN, though resulted in higher prediction time, suffered from the bottleneck of longer time needed for feature extraction using Gray Level Co-Occurrence Matrix. CNN model using Mobilenet v2 showed the best classification performance compared to conventional machine learning models in case of patch-based weed detection. Also, two different approaches to test the patch-based classification models on raw images were evaluated – slicing raw images into patches with and without overlap. It was found that slicing the raw images into patches with overlap (75% horizontal and vertical overlap) improved the weed detection performance of the model on the raw images. However, it has a significant increase in evaluation time compared to slicing without overlap. In case of object detection models, Faster RCNN model with 300 proposals was found to have slightly better detection performance compared to the SSD model with a slightly longer inference time. However, it was observed that reducing the proposals to 200 decreased the inference time of Faster RCNN model to the same amount as what was needed by the SSD model without any significant decrease in detection performance. In order to evaluate the feasibility of using these models for near real-time weed detection of commercial scale soybean fields, experiments were conducted by resizing the images from 20m altitude to higher altitudes and the change in performance with loss of detail was studied. It was found in our case that Faster RCNN model was able to detect weeds in images simulated at 120m altitude without any decrease in detection performance compared to images obtained at 20m. Also, with 15% forward and side overlap, it was estimated that a 160-acre soybean field of square shape could be covered within 12 minutes of flight time and the captured images could be evaluated using Faster RCNN model on a regular laptop (Microsoft Surface Pro) within 18 minutes.

It is known that one of the major drawbacks of using individual raw images collected rather than a stitched orthomosaic map is the error in geolocation. However, with the availability of RTK or close to RTK level GPS in the latest commercial UAV and sensor systems, it is possible now to have very high geolocation precision for individual images relative to a base station. This enables obtaining images with very low deviation in the overlap between images. Also, RTK GPS improves the stability of flight and possibly less deviation from the nadir view of the camera. Using the GPS coordinates and pitch, roll and yaw information, orthorectification of individual images can be achieved with very high geolocation accuracy. Further studies can be conducted to estimate the error in the geolocation using orthorectified individual images without RTK GPS to know the tradeoff in geolocation accuracy with the cost of buying an RTK GPS enabled UAV system.

As mentioned earlier, it was estimated to take about 12 minutes to cover 160 acres at 120m, in this case, using DJI Zenmuse X5R camera, and so an area as big as 500 acres can be covered within 40 minutes and the data processed in a regular laptop within an hour. Also, it is to be noted that with the average flight time of UAV battery being around 20 minutes, using just two sets of batteries, the above mentioned 500 acres can be covered using existing technologies. In case of commercial UAV systems such as those from the DJI, software development kits are available to develop flight control applications and also to access real-time transmission of data from the UAV. Typically, mini PCs are used to run the flight control applications. Hence, by using the software development kit, an integrated application can be developed that enables mission planning and flight control and also accesses the captured images real-time and processes the images on the background using deep learning based models on the same mini PC or regular laptop. This would enable near real-time weed detection using existing UAV technology and mobile devices which we believe is part of the future of digital agriculture.