

Detection of insect-induced defoliation in soybeans with deep learning and object
detection

by

Hayden Craig Walker

B.S., Kansas State University, 2019

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF ARTS

Department of Geography and Geospatial Sciences
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Major Professor
Douglas Goodin

Copyright

© Hayden Walker 2021.

Abstract

This thesis utilizes a modified Faster Region-based convolutional neural network (R-CNN) model framework with a Visual Geometry Group 16 (VGG16) feature extraction network to explore two similar but different applications.

The first study aimed to evaluate the practicality and accuracy of detecting and labeling soybean leaflets based on their specific defoliation level captured via smartphone. This study was conducted by training and testing the model with images of individual soybean leaflets with varying defoliation levels. Using a defoliation analysis application (Bioleaf), the leaflets were categorized as either exceeding 30% defoliation or below 30% defoliation. One hundred fifty images from each category (300 images total) were used for training data, and 30 images from each category were used for test data (60 images total). The results produced an average precision (AP) of 88.96% and an average recall (AR) of 90.55%. Overall, the model identified and labeled 49 of the 60 test images correctly.

The second study aimed to evaluate the practicality and accuracy of detecting and labeling soybean defoliation from canopy level RGB images via an unmanned aircraft vehicle (UAV). This study was conducted through training and testing the model with images of the soybean canopy collected with a flying height of approximately 1 meter. Two hundred images were used to train the model, and 40 images were used as a test dataset. Two hundred images were used for training, and 40 images were used for the test data. The results produced a precision of 25.16% and a recall of 65.00%.

Table of Contents

List of Figures	vii
List of Tables	ix
List of Equations	x
Acknowledgments	xi
Dedication	xii
Chapter 1 - Introduction	1
1.1 - Background	1
1.2 – Introduction to crucial insect defoliators and thresholds	2
1.3 – Computer vision applications in agriculture	4
1.4– Introduction to object classification and detection	5
1.4.1 – R-CNN	6
1.4.2 – Fast R-CNN	7
1.4.3 – Faster R-CNN	8
1.5 – Structure of the thesis	9
1.6 – Chapter 1 References	11
Chapter 2 Abstract	15
Chapter 2 - Detecting and Classifying Soybean Leaflet Defoliation with Deep Learning	16
2.1 Introduction	16
2.2 Materials and Methods	18
2.2.1 Image Acquisition and Labeling	18
2.2.2 Faster R-CNN	21
2.2.3 Anchor Size Justification	23
2.2.4 Convolutional Neural Network Architecture	23
2.2.5 Training	25
2.2.6 Model Evaluation	27
2.3 Results	28
2.3.1 Loss Curves	28
2.3.2 Precision and Recall	29
2.3.3 Sample of Results	31

2.4 Discussion and Conclusion	32
2.5 Chapter 2 References	34
Chapter 3 Abstract	38
Chapter 3 - Detection of Insect-Induced Soybean Defoliation from UAV RGB Imagery with Faster R-CNN	39
3.1 Introduction.....	39
3.2 Materials and Methods.....	42
3.2.1 Image Acquisition and Labeling	42
3.2.2 Faster R-CNN Description	44
3.2.3 Anchor Size Justification.....	46
3.2.4 Convolutional Neural Network Architecture	47
3.2.5 Training.....	47
3.2.6 Model Evaluation	50
3.3 Results.....	51
3.3.2 Precision and Recall	52
3.3.3 Sample of Results	54
3.4 Discussion and Conclusion	55
3.5 References.....	57
Chapter 4 – Conclusion and Synthesis.....	61
4.1 – Manuscript 1	61
4.1.1 - Summary of Findings	61
4.1.2 - Implications for Crop Pest Management.....	61
4.1.3 - Suggestions for Future Work	62
4.2 – Manuscript 2	63
4.2.1 - Summary of Findings.....	63
4.2.2 - Implications for Crop Pest Management.....	64
4.2.3 - Suggestions for Future Work	64
4.3 – Synthesis	65
Appendix A - Result Images.....	68
Study One – Leaflet Dataset	68
Study Two – Canopy Dataset.....	74

Appendix B - Codes.....	78
Image Resize and Rotate Code	78

List of Figures

Figure 1-1 Leading-soybean producing countries worldwide from 2012/13 to 2020/21. Brazil and the U.S. have consistently been the top producing countries over the displayed years.....	1
Figure 1-2 Common invasive soybean insects. (a) Bean Leaf Beetle (b) Woollybear Caterpillar (c) Thistle Caterpillar (d) Green Cloverworm (e) Japanese Beetle (f) Blister Beetle (g) Grasshopper.	3
Figure 1-3 R-CNN architecture. The model takes an input image, extracts approximately 2000 region proposals, and warps them into a fixed size. Then each proposal is fed into the convolution network. Lastly, using a class-specific linear support vector machine, classifications are issued. Image adapted from Girshick, 2015.	7
Figure 1-4 Fast R-CNN architecture. An input image and regions of interest (RoI) are input into the convolution network. Each RoI is then pooled into a fixed-size feature map. The network has two separate output vectors for each RoI: softmax probability classifiers and a bounding box regressor to adjust for offset. Image adapted from Girshick, 2015.	8
Figure 1-5 Faster R-CNN architecture. An initial feature map is created, and regions on interest (RoI) are then located. Each RoI is fed into the convolutional network. The network has two separate output vectors for each RoI: softmax probability classifiers and a bounding box regressor to adjust for offset.....	9
Figure 2-1 Soybean leaflet images resized to 600 x 600 pixels, transposed and flipped to create five times the amount of training data synthetically. This particular leaflet contains greater than 30% defoliation.	19
Figure 2-2 Example of a bounding box interactively drawn using the LabelImage software.	19
Figure 2-3 Image datasets were collected at an experiment site southwest of Manhattan, Kansas, United States. Images were collected using a mobile phone, approximately 20cm from the lens to the leaf tissue. Acquisition dates were July 22 nd and 23 rd , 2020.	20
Figure 2-4 Diagram illustrating the Faster R-CNN model. The CNN (VGG16) generates a feature map through five convolutional layers. The region proposal network (RPN) determines if the object in question is foreground or background. Lastly, region of interest (RoI) pooling occurs, and the region proposals are corrected and classified.	22

Figure 2-5 Examples of anchors applied through the region proposal network (RPN). All nine of the anchor are applied to the proposed region.	23
Figure 2-6 Diagram of the VGG16 architecture for feature extraction.	24
Figure 2-7 The total loss curve with VGG16 as the feature map extraction in a Faster R-CNN model.....	28
Figure 2-8 Examples of results from the model. (a-c) represent the three types of results from the over 30% classification: TP, FP, and FN. (d-f) represents the three types from the under 30% classification.	29
Figure 2-9 Results from 12 out of the 60 test images. A green border represents a correct classification and detection, while a red border represents an incorrect classification or detection.	31
Figure 3-1 Example of a raw UAV image downsized to approximately 600x600 pixels to contain defoliation.	42
Figure 3-2 Image datasets were collected at an experiment site southwest of Manhattan, Kansas, United States. Images were collected using a DJI Mavic Pro, with a flying height of approximately 1 meter above the canopy. Acquisition dates were July 22 nd and 23 rd , 2020.	43
Figure 3-3 Example of a bounding box interactively drawn using the LabelImage software.	44
Figure 3-4 Diagram illustrating the Faster R-CNN model. The CNN (VGG16) generates a feature map through five convolutional layers. The region proposal network (RPN) determines if the object in question is foreground or background. Lastly, region of interest (RoI) pooling occurs, and the region proposals are corrected and classified.	45
Figure 3-5 Examples of anchors applied through the region proposal network (RPN). All nine of the anchor are applied to the proposed region.	46
Figure 3-6 Diagram of the VGG16 architecture for feature extraction.	48
Figure 3-7 The total loss curve with VGG16 as the feature map extraction in a Faster R-CNN model.....	52
Figure 3-8 Model result with examples of instances of true positive (green box), false positive (red box), and false negative (yellow box).	52
Figure 3-9 Results from 9 out of the 40 test images. The blue boxes are the applied bounding boxes.	54

List of Tables

Table 2-1 The resulting true positive, false positive, and false negative values for the test dataset. There are a total of 30 images in each class.....	30
Table 2-2 Precision and recall for the over 30% defoliation class, under 30% defoliation class, the average precision (AP), and the average recall (AR).....	30
Table 3-1 The resulting true positive, false positive, and false negative values for the test dataset. There were a total of 40 test images in the dataset.	53
Table 3-2 Precision and recall values for the test dataset.	53

List of Equations

Equation 2-1	25
Equation 2-2	26
Equation 2-3	26
Equation 2-4	26
Equation 2-5	26
Equation 2-6	27
Equation 2-7	27
Equation 2-8	28
Equation 3-1	49
Equation 3-2	49
Equation 3-3	49
Equation 3-4	49
Equation 3-5	50
Equation 3-6	50
Equation 3-7	51
Equation 3-8	51

Acknowledgments

This thesis's completion would not have been achievable without a few significant individuals' guidance and support. I am incredibly thankful for my major professor Dr. Douglas G. Goodin. He was patient and supportive in finding a direction for my research. When I doubted myself, he always seemed to believe I would find a way. To my committee members, Dr. Brian McCornack and Dr. Shawn Hutchinson, your insightful perspectives and approaches encouraged me to complete my thesis, and I would never have gotten to where I am today without you. To Dong Luo, no words can express my thankfulness for sharing your expertise in machine learning and coding. I am confident that I would have never come close to accomplishing my goals without your guidance and knowledge.

Thank you to my fellow graduate students within the Geography and Geospatial Science department. Thank you for being friends I could always talk to and share my stress and struggles. Not only did we have the stress of graduate school, but also the pandemic and everything that came along with it. I firmly believe we are stronger for it and benefit from our shared experiences.

Last but not least, thank you to my mom and dad. You have always believed and supported me through my academic endeavors. From kindergarten to now, my 7th year at Kansas State University, you have believed in me more than I have believed in myself. In moments of anxiety and feelings of doubt, you were always there to offer words of support and get me back on the path of self-belief. Without your love and support, I would not be the man I am today.

Dedication

This thesis is dedicated to my family. Without your love and support, none of this would have been possible.

Chapter 1 - Introduction

1.1 - Background

With a worldwide production estimated at 362.1 million metric tons, soybeans (*Glycine max L.*) carry great importance in the realm of both economics and nutrition. Globally, soybean production supplies nearly half of the demand for vegetable oils (A. de F. Bueno et al., 2011). Global production of soybeans has steadily increased, with a 34.8% increase in metric tons produced since the 2012/13 growing season (Shahbandeh, 2021). During the 2020/21 growing season, the United States produced roughly 113.5 million metric tons of soybeans. This value makes it the second-largest producer globally, behind Brazil, which produced approximately 133 million metric tons during the same period (Shahbandeh, 2021).

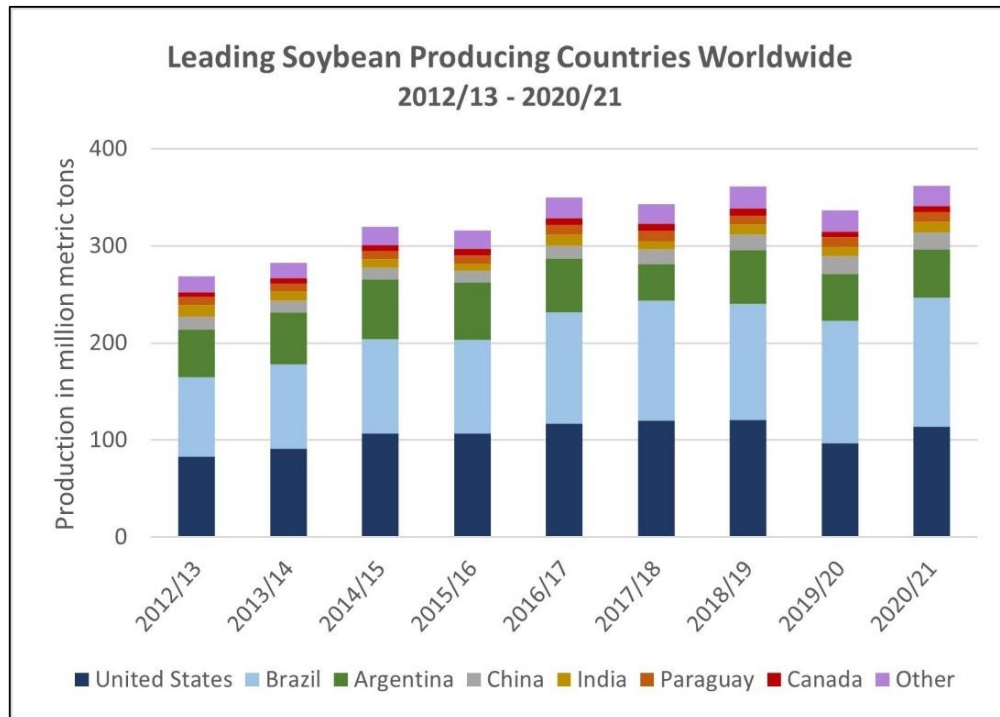


Figure 1-1 Leading-soybean producing countries worldwide from 2012/13 to 2020/21. Brazil and the U.S. have consistently been the top producing countries over the displayed years. Data collected from Shahbandeh, 2021.

These production levels can further increase by 29% if problems with insect pests could be mitigated or avoided altogether (Oerke, 2006). Approximately 10-16% of the potential global crop production is lost due to invasive insects before harvest, with similar losses post-harvest (Bradshaw et al., 2016). Losses caused by these pests are believed to cost the global economy an estimated US\$70 billion annually (Bradshaw et al., 2016; da Silva et al., 2019). While commercial soybean production has a long history both internationally and in the US, these losses due to invasive insect pests have only recently become a concern (Deutsch et al., 2018). Studies have shown that these increasing losses because of invasive insects have come due to a warming climate (Deutsch et al., 2018). This development highlights the need for new and improved management strategies moving forward to combat the increasing levels of infestation.

1.2 – Introduction to crucial insect defoliators and thresholds

Common invasive insects that consume the highest percentage of soybean foliage include the Bean Leaf Beetle (*Cerotoma trifurcata*), Woollybear Caterpillar (*Pyrrharctia isabella*), Thistle Caterpillar (*Vanessa cardui*), Green Clover Worm (*Hypena scabra*), Japanese Beetle (*Popillia japonica*), Blister Beetles (*Meloidae*), and various types of grasshopper. This is by no means a comprehensive list of all soybean defoliators. However, it does encompass the key species typically responsible for defoliation and economic loss in Kansas and across the North Central Region of the U.S. Below, Figure 1-2 shows images of each of the previously mentioned insect species and displays the typical feeding pattern characteristics associated with each. These species typically feed in mixed populations, and therefore defoliation thresholds should be applied to individual fields rather than particular species (A. F. Bueno et al., 2013).

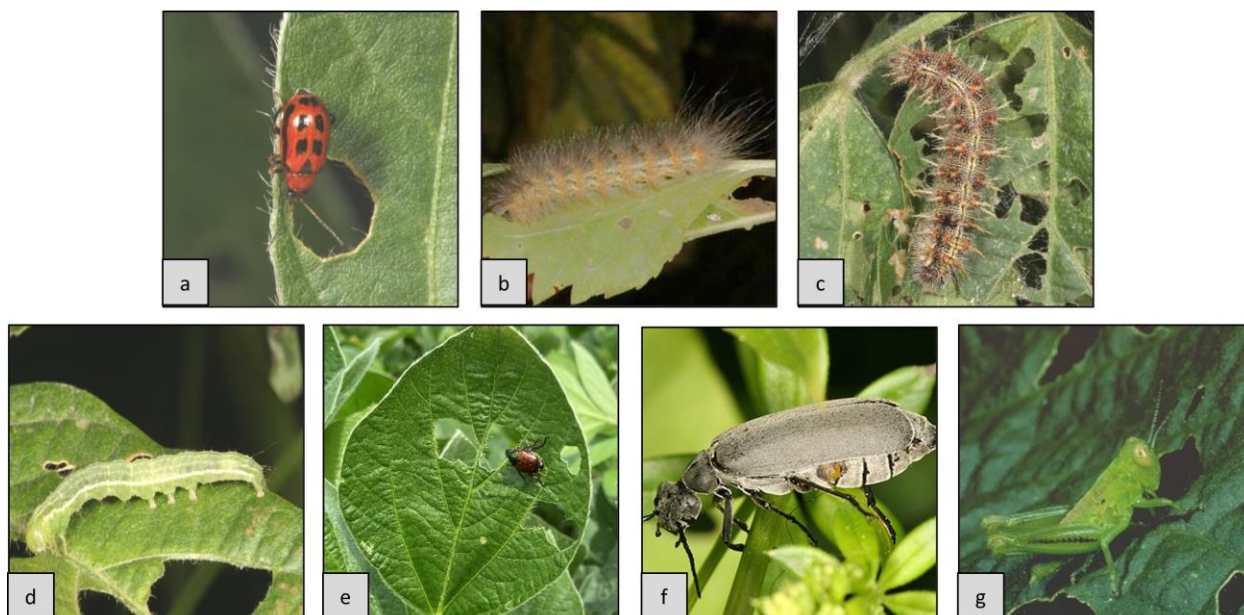


Figure 1-2 Common invasive soybean insects. (a) Bean Leaf Beetle (b) Woollybear Caterpillar (c) Thistle Caterpillar (d) Green Cloverworm (e) Japanese Beetle (f) Blister Beetle (g) Grasshopper.
 Images source: Kansas State University Entomology Department
<https://entomology.k-state.edu/extension/insect-information/crop-pests/soybeans/>

Research studies over the years have established that the loss of light interception by the leaves is the driving force in yield loss from defoliation (Grymes et al., 1999; Haile et al., 1998). Soybean plants have a remarkable ability to compensate for defoliation from these invasive insect species. Studies have shown that defoliation levels in the 15%-25% have increased yields due to sunlight now accessing leaves lower in the canopy (Turnipseed, 1972). Today, the accepted economic thresholds for insect-induced soybean leaf defoliation are 25% during vegetative stages (V1-V6), 15% during bloom to pod-fill (R1-R5), and 25% during pod-fill to maturity (R5 – R8). These economic thresholds could also be called “action” thresholds; when defoliation rises to or above these levels, it is then economically beneficial to prohibit the level of defoliation from increasing. Traditionally, monitoring of defoliation is performed manually and considered to be a tedious and time-expensive assignment.

1.3 – Computer vision applications in agriculture

With the growth in computer vision applications, there has also come a rise in innovative and exciting applications. Computer vision has opened the door for advanced disease detection, grain quality inspection, and crop phenotyping. These innovative approaches aid in the production of the most cultivated crops in the world, including maize (*Zea mays*), winter wheat (*Triticum aestivum*), rice (*Oryza glaberrima*), and soybean (*Glycine max*). Shrivastave et al. (2016) and Pires et al. (2016) utilized computer vision methods to detect and categorize soybean foliar diseases. Both methods used a support vector machine (SVM) as a classifier for the specific diseases with a high level of success. Computer vision and machine learning have also been applied to detect wheat disease (Serranti et al., 2013) and rice (Sun et al., 2016). Looking specifically at maize, studies have shown computer vision's ability to detect maize tassels from UAV RGB imagery (Kumar et al., 2019; Liu et al., 2020). These studies employed the Faster R-CNN architecture to detect and classify tassels.

There have also been several publications focused on estimating soybean defoliation employing computer vision and machine learning procedures. A study devised to assess soybean leaf defoliation using convolutional neural networks, de Silva et al. (2019) applied AlexNet, VGGNet, and ResNet convolutional neural network architectures. As a result of this work, the author has assigned estimated percent defoliation to individual soybean leaves at an accurate level (root mean square error: 4.57). Along with the studies mentioned, there are many other computer vision applications within the agriculture realm. Presently, the ability to detect and assess defoliation is on an individual leaf basis. There are currently no proposed methods to detect defoliation from the canopy level.

1.4– Introduction to object classification and detection

Computer vision has been gaining traction in recent years and continues to be applied in many different disciplines. Two of the leading computer vision mechanisms are object detection and object classification. While they are interrelated, there is a fundamental difference between the two methods. With object classification, the goal is to classify the object in the image. Object detection differs in that it aims to locate the image of interest and draw a bounding box around the object. Additionally, object detection is a much more complex and challenging task than object classification. Recently, object detection has allowed for vehicle detection (Xu et al., 2017), crop disease identification (Ha et al., 2017), tree species identification (Santos et al., 2019), and has also proven helpful for many other applications.

One of the most prominent and influential algorithm architectures for computer vision has been found by developing convolutional neural networks (CNN) (LeCun, 1998). A CNN algorithm's focal objective is to differentiate between all the input images and “learn” the unique features that belong to desired objects. The algorithm works to identify and locate the desired items in new input images that were not used in the model's training. In human brains, this process occurs subconsciously. Using a banana as an example, our minds look for identifiable features such as the unique shape, the color yellow, a stem, and so on until the brain recognizes the object is, in fact, a banana. In a very similar way, the convolutional neural network can perform image detection by first looking for low-level characteristics such as curves, edges, or angles. After identifying these standard features, the algorithm then moves through a series of convolutional layers, going deeper to extract more abstract characteristics (LeCun, 1998).

Initially, convolutional neural networks displayed an impressive ability to perform object detection and classification with a vast list of applications (Albawi et al., 2017; Kalchbrenner et

al., 2014; O'Shea & Nash, 2015). The initial CNN structure was first proposed by LeCun et al. in 1998. The algorithm's architecture contains convolution and max-pooling layers to accomplish feature learning. Each convolution layer creates a feature map using several filters to extract features from the preceding layer. Additionally, each convolution layer decreases in size, and the extracted features become more complex than the prior. The output layer in the CNN then uses all the extracted features for final classification (Albawi et al., 2017).

1.4.1 – R-CNN

The original CNN method possesses limitations, specifically in its need for many proposed regions, which is quite computationally taxing. To address this issue, Girshick et al. proposed a region-based CNN (R-CNN) (Girshick et al., 2014). This method offered a selective search to extract 2000 regions from the input image (Uijlings et al., 2013). Using this technique, each candidate region's proposals are warped and resized into a square, then fed into the convolutional neural network. The CNN extracts the features from the image. The extracted features are then introduced into a support vector machine to determine the object's classification within that specific region proposal. Not only does the CNN predict the presence of an item within the proposal, but it also employs a bounding box regression to increase the precision of the estimated bounding box around the identified object (Girshick et al., 2014). For example, if a region proposal predicted a dog's presence, but the head of that dog wasn't included in the original proposal, the bounding box regressor would adjust the bounding box to create a more accurate localization of the entire dog (Girshick et al., 2014).

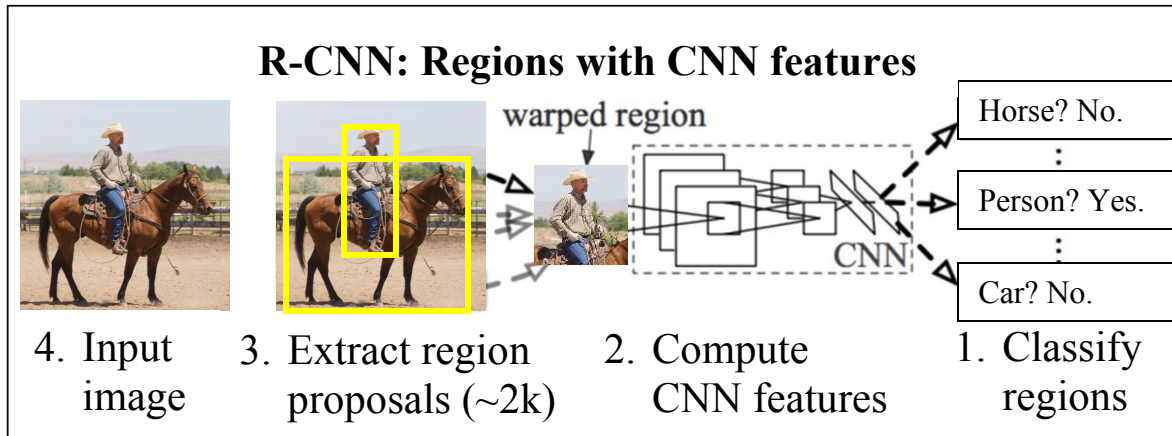


Figure 1-3 R-CNN architecture. The model takes an input image, extracts approximately 2000 region proposals, and warps them into a fixed size. Then each proposal is fed into the convolution network. Lastly, using a class-specific linear support vector machine, classifications are issued. Image adapted from Girshick, 2015.

Though R-CNN significantly improved previous architectures, it still possesses fundamental shortcomings. It requires a significant amount of time to train the network due to classifying all 2000 region proposals per image and does not share any computation across the proposals, meaning each proposal's computation is completed independently of other proposals (Girshick, 2015). Additionally, implementing in real-time is not practical since it takes approximately 47 seconds to test each input image (Girshick, 2015).

1.4.2 – Fast R-CNN

To address R-CNN's drawbacks, Girshick developed a new and faster object detection algorithm called Fast R-CNN (Girshick, 2015). The architecture of this new algorithm is comparable to the previous R-CNN method. However, in the Fast R-CNN algorithm, the input image is fed directly into the CNN. The image is explicitly directed to the convolutional and max pooling layers to create a convolutional feature map (Girshick, 2015). Region proposals are

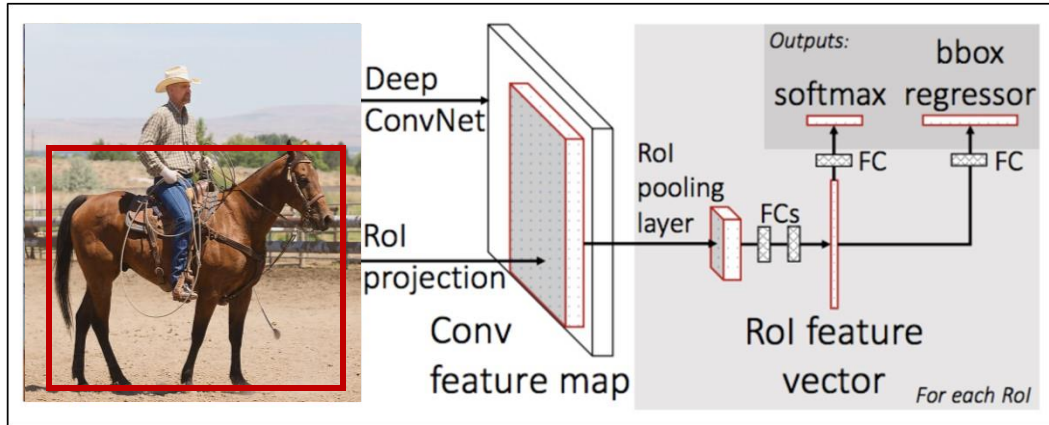


Figure 1-4 Fast R-CNN architecture. An input image and regions of interest (RoI) are input into the convolution network. Each RoI is then pooled into a fixed-size feature map. The network has two separate output vectors for each RoI: softmax probability classifiers and a bounding box regressor to adjust for offset. Image adapted from Girshick, 2015.

applied for the feature map. For each region proposal, region of interest (RoI) pooling layers are used to reshape the images into a fixed size to be fed into a sequence of fully connected layers. These fully connected layers then branch into two output layers. The first layer produces softmax probability estimates for the designated number of object classes plus a background class. The second of these layers apply a bounding box regressor to adjust the applied bounding boxes' positions to the extracted objects (Girshick, 2015).

1.4.3 – Faster R-CNN

Both of the previous architectures utilized the selective search method, and it was here in which computational bottlenecks arose (Ren et al., 2017). Ren et al. introduced a Region Proposal Network (RPN) that shares convolutional features with the detection network to address these bottleneck issues, thus allowing for computationally cost-free region proposals. Faster R-CNN has two separate networks: the RPN for creating the region proposals and the following network utilizing these proposals to detect objects, create labels, and apply bounding boxes. Compared to selective search, the cost of creating region proposals is much less significant with RPN because of its ability to share the computation with the object detection network.

An important concept in Faster R-CNN is the application of anchors. There are nine anchors at a given position of an image in the algorithm's default configuration. Each location has 3 different pixel scales (eg. 128x128, 256x256, and 512x512) and 3 different ratios (eg. 1:1, 1:2, and 2:1). These anchors are then passed on to the RPN, predicting if the anchor possesses mostly foreground (object) or background (non-object). The predicted regions are then resized using an RoI pooling layer used to classify the object and apply the bounding box regressor just as in Fast R-CNN (Ren et al., 2017).

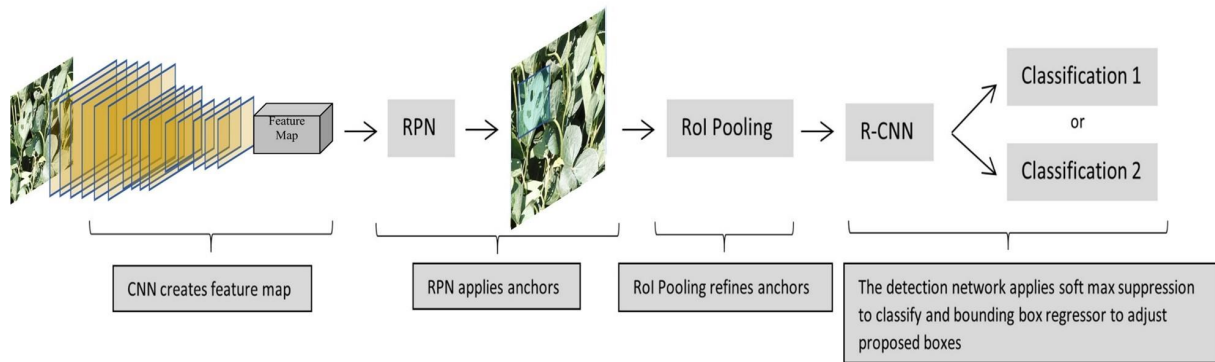


Figure 1-5 Faster R-CNN architecture. An initial feature map is created, and regions on interest (RoI) are then located. Each RoI is fed into the convolutional network. The network has two separate output vectors for each RoI: softmax probability classifiers and a bounding box regressor to adjust for offset.

1.5 – Structure of the thesis

This thesis's remainder consists of two stand-alone research manuscripts to be submitted for publication. The first manuscript, titled “Detecting and Classifying Soybean Leaflet Defoliation with a Threshold Approach with Faster R-CNN” will be submitted to *Computers and Electronics in Agriculture*. This manuscript utilizes the Faster R-CNN architecture mentioned above to detect and classify lone soybean leaves' images as either exceeding or below a specific threshold of insect-induced defoliation on the leaf. Following is a manuscript titled “Detection of

Insect-Induced Soybean Defoliation from UAV RGB Imagery with Faster R-CNN” and will be submitted for publication to *Computers and Electronics in Agriculture*. This second manuscript utilizes a similar method. However, instead of individual soybean leaves, the novel model attempts to detect insect-induced defoliation within the canopy from low altitude, RGB images captured using a small unmanned aerial vehicle (DJI Mavic). The fourth and final chapter includes an in-depth synthesis of the included manuscripts.

1.6 – Chapter 1 References

- Albawi, S., Abed Mohammed, T., & ALZAWI, S. (2017). *Understanding of a Convolutional Neural Network*. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
- Bradshaw, C. J. A., Leroy, B., Bellard, C., Roiz, D., Albert, C., Fournier, A., Barbet-Massin, M., Salles, J.-M., Simard, F., & Courchamp, F. (2016). Massive yet grossly underestimated global costs of invasive insects. *Nature Communications*, 7(1), 12986. <https://doi.org/10.1038/ncomms12986>
- Bueno, A. de F., Batistela, M. J., Bueno, R. C. O. de F., França-Neto, J. de B., Naime Nishikawa, M. A., & Filho, A. L. (2011). Effects of integrated pest management, biological control and prophylactic use of insecticides on the management and sustainability of soybean. *Crop Protection*, 30(7), 937–945. <https://doi.org/10.1016/j.cropro.2011.02.021>
- Bueno, A. F., Paula-Moraes, S. V., Gazzoni, D. L., & Pomari, A. F. (2013). Economic Thresholds in Soybean-Integrated Pest Management: Old Concepts, Current Adoption, and Adequacy. *Neotropical Entomology*, 42(5), 439–447. <https://doi.org/10.1007/s13744-013-0167-8>
- da Silva, L. A., Bressan, P. O., Gonçalves, D. N., Freitas, D. M., Machado, B. B., & Gonçalves, W. N. (2019). Estimating soybean leaf defoliation using convolutional neural networks and synthetic images. *Computers and Electronics in Agriculture*, 156, 360–368. <https://doi.org/10.1016/j.compag.2018.11.040>
- Deutsch, C. A., Tewksbury, J. J., Tigchelaar, M., Battisti, D. S., Merrill, S. C., Huey, R. B., & Naylor, R. L. (2018). Increase in crop losses to insect pests in a warming climate. *Science*, 361(6405), 916–919. <https://doi.org/10.1126/science.aat3466>

- Girshick, R. (2015). Fast_R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. <https://doi.org/10.1109/CVPR.2014.81>
- Grymes, C. F., Griffin, J. L., Boethel, D. J., Leonard, B. R., Jordan, D. L., & Russin, J. S. (1999). Soybean Response to Weed Interference and Defoliation. *Weed Science*, 47(1), 90–94.
- Ha, J. G., Moon, H., Kwak, J. T., Hassan, S. I., Dang, M., Lee, O. N., & Park, H. Y. (2017). Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *Journal of Applied Remote Sensing*, 11(04), 1. <https://doi.org/10.1117/1.JRS.11.042621>
- Haile, F. J., Higley, L. G., & Specht, J. E. (1998). Soybean Cultivars and Insect Defoliation: Yield Loss and Economic Injury Levels. *Agronomy Journal*, 90(3), 344–352. <https://doi.org/10.2134/agronj1998.00021962009000030006x>
- Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. *ArXiv:1404.2188 [Cs]*. <http://arxiv.org/abs/1404.2188>
- Kumar, A., Taparia, M., Rajalakshmi, P., Desai, U. B., Naik, B., & Guo, W. (2019). UAV Based Remote Sensing for Tassel Detection and Growth Stage Estimation of Maize Crop using F-RCNN. *Computer Vision Problems in Plant Phenotyping*, 3.
- LeCun, Y. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278–2323.

- Liu, Y., Cen, C., Che, Y., Ke, R., Ma, Y., & Ma, Y. (2020). Detection of Maize Tassels from UAV RGB Imagery with Faster R-CNN. *Remote Sensing*, 12(2), 338.
<https://doi.org/10.3390/rs12020338>
- Oerke, E.-C. (2006). Crop losses to pests. *The Journal of Agricultural Science*, 144(1), 31–43.
<https://doi.org/10.1017/S0021859605005708>
- O’Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks.
ArXiv:1511.08458 [Cs]. <http://arxiv.org/abs/1511.08458>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Santos, A. A. dos, Marcato Junior, J., Araújo, M. S., Di Martini, D. R., Tetila, E. C., Siqueira, H. L., Aoki, C., Eltner, A., Matsubara, E. T., Pistori, H., Feitosa, R. Q., Liesenberg, V., & Gonçalves, W. N. (2019). Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs. *Sensors*, 19(16), 3595. <https://doi.org/10.3390/s19163595>
- Serranti, S., Cesare, D., & Bonifazi, G. (2013). The development of a hyperspectral imaging method for the detection of Fusarium-damaged, yellow berry and vitreous Italian durum wheat kernels. *Biosystems Engineering*, 115(1), 20–30.
<https://doi.org/10.1016/j.biosystemseng.2013.01.011>
- Sun, K., Wang, Z., Tu, K., Wang, S., & Pan, L. (2016). Recognition of Mould Colony on Unhulled Paddy Based on Computer Vision using Conventional Machine-learning and Deep Learning Techniques. *Scientific Reports*, 6(1), 37994.
<https://doi.org/10.1038/srep37994>

- Turnipseed, S. G. (1972). Response of soybeans to foliage losses in South Carolina. *Journal of Economic Entomology*, 65(1), 224–229.
- Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., & Smeulders, A. W. M. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2), 154–171. <https://doi.org/10.1007/s11263-013-0620-5>
- Xu, Y., Yu, G., Wang, Y., Wu, X., & Ma, Y. (2017). Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN. *Journal of Advanced Transportation*, 2017, 1–10. <https://doi.org/10.1155/2017/2823617>

Chapter 2 Abstract

Invasive insect-induced defoliation is a significant concern for maximizing potential soybean yields. The prevalence of deep learning provides an opportunity to improve upon defoliation detection and assessment. This study uses images of individual soybean leaflets captured via smartphone to test the performance of a modified Faster Region-based Convolutional Network (Faster R-CNN) with a visual geometry group neural network (VGG16). The proposed model generated an output image with an applied bounding box and classification of either above 30% defoliation or below 30% defoliation. The results showed an overall prediction accuracy of 81.67%. The average precision and recall over both classes were 88.96% and 90.55%, respectively. In future work, further improvements to the model's performance could be made by enlarging the training dataset to include more leaflets with greater than 30% defoliation.

Chapter 2 - Detecting and Classifying Soybean Leaflet Defoliation with Deep Learning

2.1 Introduction

Demand for food and other agricultural commodities is estimated to rise by 50 percent by 2050 (FAO, 2017). This surge in demand is due to population growth, shifts towards increasing urbanization, and rises in per capita income (FAO, 2017; Tilman et al., 2011). To combat this looming concern, improved crop management strategies, specifically invasive insect control, are vital to improving productivity.

Annually, pest-induced crop production losses total 20 to 40 percent (FAO, 2017) and cost the global economy an estimated US\$70 billion (Bradshaw et al., 2016; da Silva et al., 2019). The primary concern with invasive insects is herbivory, especially defoliation, of the canopy. Defoliation lowers the plant's photosynthetic and energy production potential and, consequently, its final production capacity (Briske & Richards, 1995). For this reason, it is paramount to monitor defoliation levels within the canopy in case preventative actions must be implemented. For soybeans in the vegetative state, prevention measures become necessary when defoliation levels reach an average of 30% throughout the canopy (Kogan et al., 1977).

Quantifying insect-induced defoliation is necessary for farmers, crop scouts, and entomologists for enhanced integrated pest management (IPM) practices. The monitoring of defoliation is usually performed manually and is considered tedious and time-expensive. Typically, leaf defoliation is visually approximated by an expert using a guide that compares leaf samples to example leaves with known defoliation levels (Kogan et al., 1977). The grid counting method is another commonly used technique. This practice involves an expert positioning the leaf on a grid and counting the number of squares to assess the level of defoliation (Kvet & Marshal,

1971). Defoliation levels are habitually overestimated due to human error, which has led to the over-application of insecticides (Machado et al., 2016). Consequently, this leads to potential threats to public health (Pimentel, 2005).

Leaf area meters can also assess defoliation levels. However, these meters only quantify the leaf area and do not directly quantify defoliation level (Clément et al., 2015; Easlon & Bloom, 2014; Igathinathane et al., 2006). To assess defoliation levels, measurements must be collected before and after defoliation, making these techniques impractical to implement in real-time. Smartphones have recently become a viable option making data acquisition more manageable and less timely. Mechado et al. (2016) constructed a mobile application called BioLeaf to estimate insect-induced defoliation. This novel approach accepts input soybean leaf images and estimates the defoliation level by progressing through several steps, including image thresholding to reduce noise and employing Bezier curves to recreate the original leaf border. It then calculates the defoliation level in relation to leaf area as a percent defoliated. Although these methods improve upon previous work regarding speed and accuracy, they still require expert intervention.

Computer and information technologies have become increasingly robust and reliable in recent years in object detection applications. When computers become the primary medium for object detection tasks, complex algorithms and models must be employed. One of the leading outcomes of these newly developed models is the implementation of machine learning models. These models typically have multiple layers and are referred to as Deep Neural Networks (Larochelle et al., 2019). One of the most popular deep neural networks is the Convolutional Neural Network (CNN) (Albawi et al., 2017). CNN's have paved the way for tasks such as face recognition (Lawrence et al., 1997), self-driving cars (Bojarski et al., 2016), and cancer detection (Hirasawa et al., 2018). Da Silva et al. (2018) constructed a fully automated method to estimate

leaf defoliation level utilizing a convolutional neural network (CNN). Their model reconstructed the damaged leaf and assigned estimated percent defoliation. Although this approach further improved defoliation assessment speed and reliability, it required that the images first be preprocessed and converted into a binary color scale. One opportunity to improve upon this method would be through employing a R-CNN model that can directly take in raw RGB images and rapidly process and label them.

The following study uses the VGG16 (Visual Geometric Group – 16) (Simonyan & Zisserman, 2015) as the feature extraction network in Faster R-CNN (Region-based Convolution Neural Network) (S. Ren et al., 2017) to fully automate the classification of soybean defoliation as above or below the 30% management threshold. The remainder of this paper is organized as follows: In section 2.2, the proposed novel's methods and materials are introduced. This section also contains information about image acquisition, labeling, and model description. Results are included in section 2.3. Lastly, section 2.4 concludes the paper and discusses potential directions for future work.

2.2 Materials and Methods

2.2.1 Image Acquisition and Labeling

Image datasets were collected on July 22nd and 23rd, 2020, from experimental fields near Manhattan, Kansas, United States (39.115837, -96.635032), as shown in Figure 2-3. A total of 371 images of mid-vegetative stage soybean leaflets were collected with a Samsung s10 device. Distance from lens to leaf tissue was approximately 20cm, and the camera resolution was 3024 x 4032 pixels. Images were interactively processed using the BioLeaf application created by Machado et al. (2016) to assess defoliation level and sort into categories of above 30% defoliation

and below 30% defoliation. After processing all 371 images, 332 displayed less than 30% defoliation, and 39 displayed defoliation levels exceeding 30%. At least one hundred eighty leaflets (150 for training, 30 for testing) greater than 30% was the desired number to obtain adequate training and test data. To produce more data, leaves containing defoliation levels greater than 30% would have to be synthetically produced. The leaflet images were resized to 600 x 600 pixels (to reduce the data processing time while running the model), transposed, and rotated to produce five times the number of images shown in Figure 2-1.



Figure 2-1 Soybean leaflet images resized to 600 x 600 pixels, transposed and flipped to create five times the amount of training data synthetically. This particular leaflet contains greater than 30% defoliation.

The graphical image annotation tool Labellmg (Labellmg, n.d.) was used to generate the bounding boxes in the images, as shown in Figure 2-2. The annotated images were then randomly divided into train and test datasets. In total, there were 300 total training images (150 greater than 30% defoliation, 150 less than 30% defoliation) and 60 test images (30 greater than 30% defoliation, 30 less than 30% defoliation). The training set and the test set were separated according to the ratio 5:1.



Figure 2-2 Example of a bounding box interactively drawn using the LabellImage software.

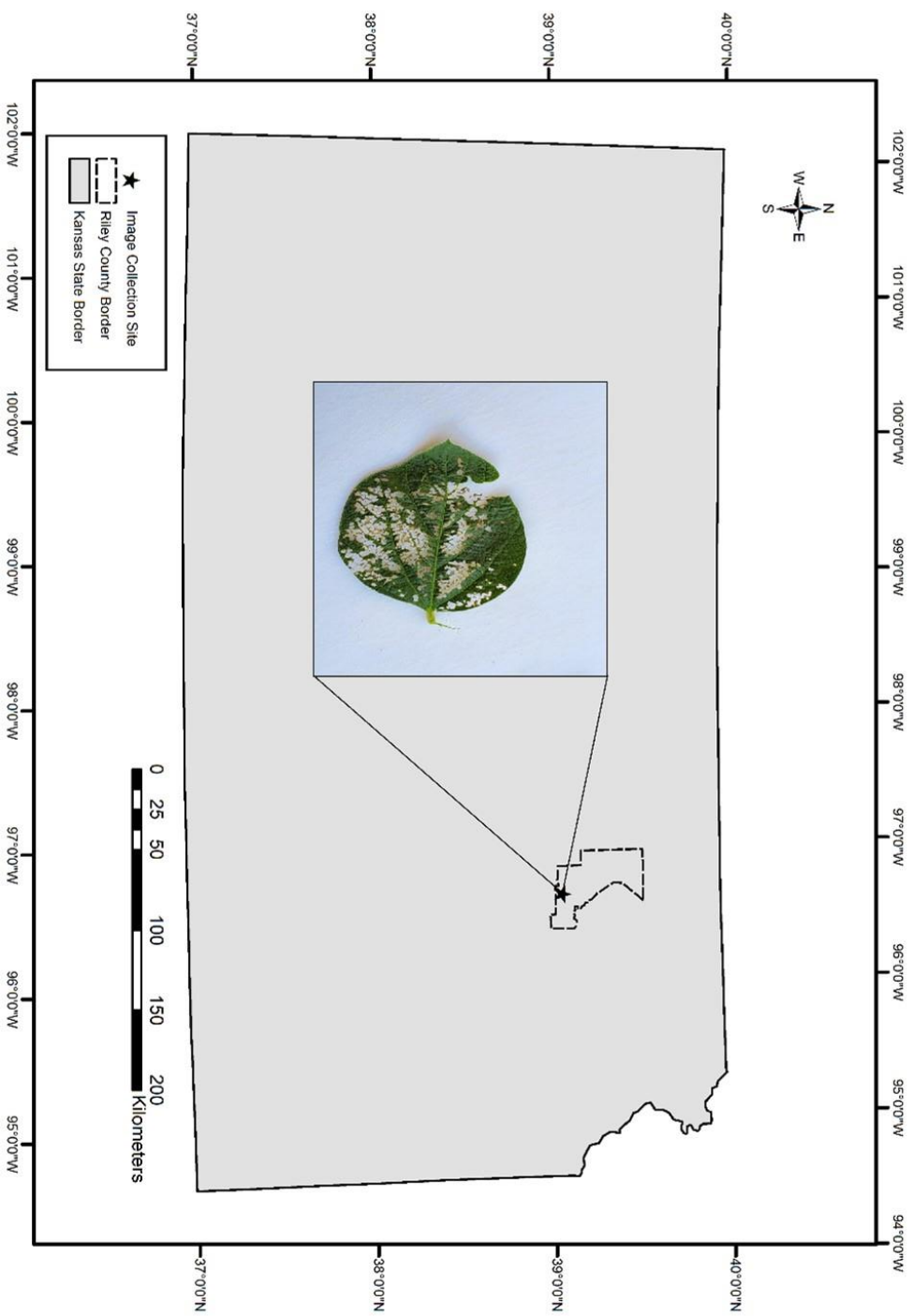


Figure 2-3 Image datasets were collected at an experiment site southwest of Manhattan, Kansas, United States. Images were collected using a mobile phone, approximately 20cm from the lens to the leaf tissue. Acquisition dates were July 22nd and 23rd, 2020.

2.2.2 *Faster R-CNN*

Faster R-CNN allows for an end-to-end training algorithm for object detection. The general structure is shown in Figure 2-4. Feature maps are generated for each annotated image by imputing the image into a convolutional neural network (CNN). The region proposal network (RPN) determines if an object is foreground or background. These region proposals from the RPN are then processed by the region of interest (RoI) pooling layer. The model then splits into two fully connected layers (FC). The first is a bounding box regressor. This FC layer corrects the bounding box's position to fit the detected object more closely. The second FC layer uses a softmax classification to determine which category the detected object belongs.

Previous R-CNN architectures (R-CNN and Fast R-CNN) (Girshick, 2015; Girshick et al., 2014) used a selective search method to predict the presence of an object. Selective search applied approximately 2000 proposals to each image, and each proposal had to be passed through a CNN, which became very computationally taxing. Faster R-CNN is superior to previous architectures by employing a region proposal network (RPN). This RPN allows each input image to pass through the CNN to create a feature map and then predict features. The RPN uses the user-applied ground truth boxes from the image annotation task and applies its bounding boxes, called anchors (Figure 2-5), with different aspect ratios. Using the ground truth boxes and the applied anchors, it calculates the intersection over union (IoU). IoU simply measures how much overlap is present between the model-applied anchors and ground truth boxes. Anchors with an IoU greater than .7 are considered foreground (an object), and anchors with an IoU of less than .3 are considered background (non-object).

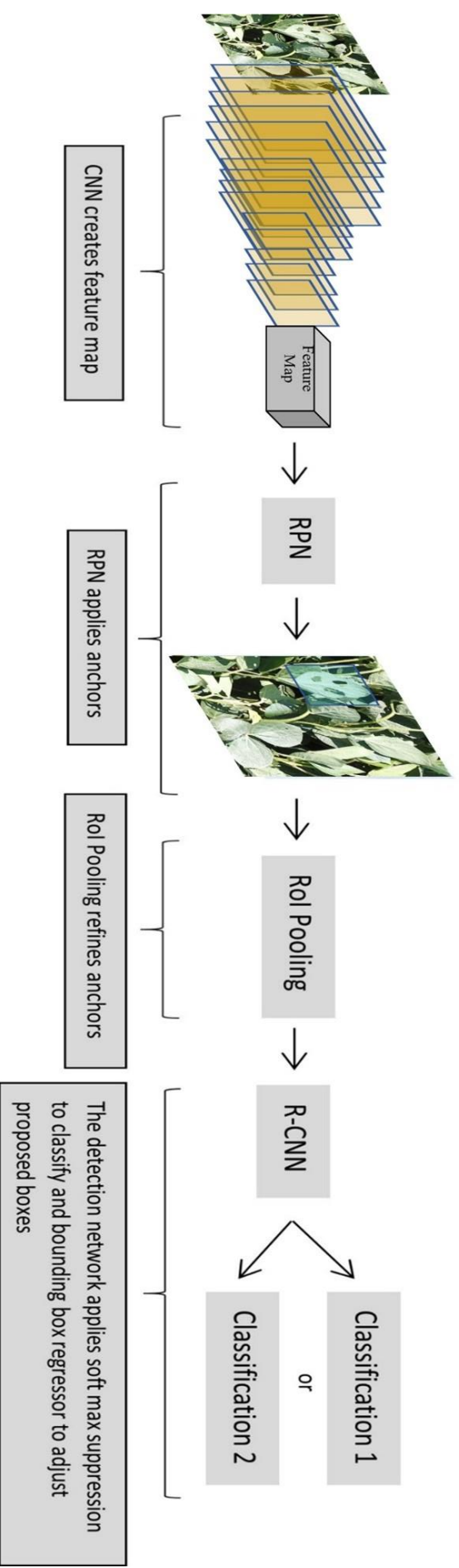


Figure 2-4 Diagram illustrating the Faster R-CNN model. The CNN (VGG16) generates a feature map through five convolutional layers. The region proposal network (RPN) determines if the object in question is foreground or background. Lastly, region of interest (RoI) pooling occurs, and the region proposals are corrected and classified.

2.2.3 Anchor Size Justification

For Faster R-CNN, the anchor size's default parameters are $[128^2, 256^2, 512^2]$. It was found that some of the soybean leaflets required smaller anchors. Therefore, anchor size was adjusted to $[64^2, 128^2, 256^2]$. This practice was inspired by previous work showing that utilization of smaller anchor sizes can improve detection results when the dataset contains objects much tinier than the smallest anchor size (Y. Ren et al., 2018).

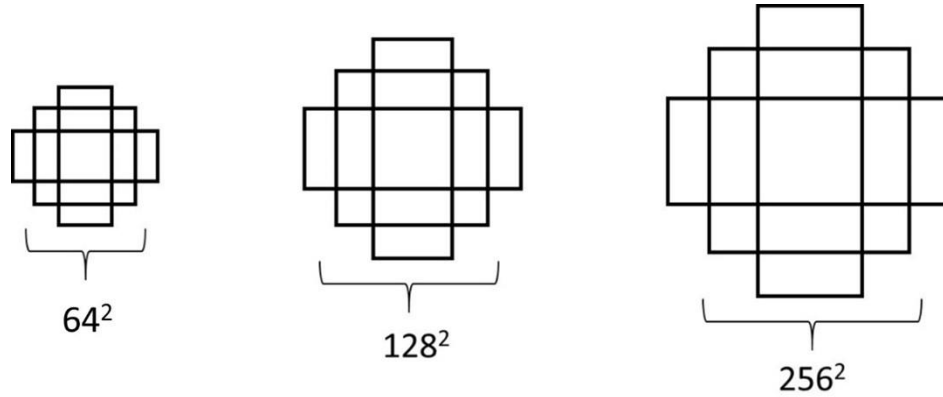


Figure 2-5 Examples of anchors applied through the region proposal network (RPN). All nine of the anchor are applied to the proposed region.

2.2.4 Convolutional Neural Network Architecture

The Visual Geometry Group first conceived the visual geometry group network (VGGNet) at Oxford University (Simonyan & Zisserman, 2015). A state-of-the-art Faster R-CNN with the VGG16 architecture was used for the object detection network. In 2014, the VGG16 network architecture placed 2nd in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) annual competition. However, it performed better than the first place finisher (GoogleNet) to detect multiple objects (Deng et al., 2009). The original VGG16 network is pre-trained with the ImageNet dataset and then retrained with the soybean leaflet dataset. The default dimensions for the input image is 224 x 224 for the

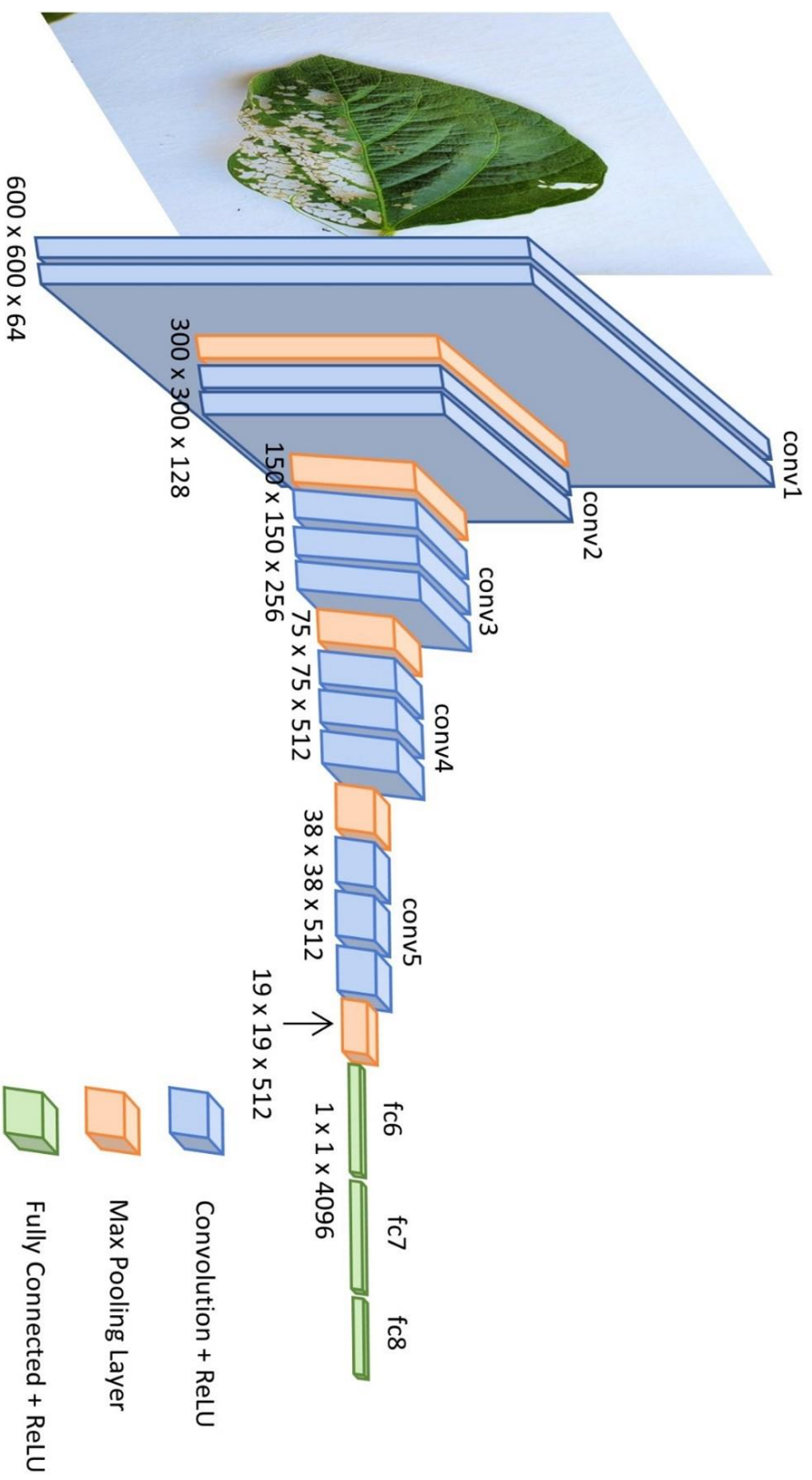


Figure 2-6 Diagram of the VGG16 architecture for feature extraction.

VGG16 model. This parameter was changed to 600 x 600 pixels in this experiment to better fit the dataset. The VGG16 model used for feature map extraction in Faster R-CNN in this study can be seen in Figure 2-6.

2.2.5 Training

The Faster R-CNN model was implemented using the Keras framework on an Ubuntu 20.04 LTS (64-bit) workstation with an Intel Xeon 10 core Silver 4210 CPU and an 8 gigabyte GeForce RTX 2080 Super with 3072 CUDA cores GPU. Weight parameters from a pre-trained model were used to reduce the training time. A concept known as transfer learning (Pan & Yang, 2010). The model was trained through two separate training runs of 75 epochs for a total of 150 epochs. The epoch number is a parameter that determines the number of times the training process works through the entire dataset. Epoch length (also known as batch size) was set to 300. This parameter determines how many of the dataset images are fed into the model at once. The iteration number parameter was set to zero because it was unnecessary to iterate the training process. In total, it took the model two hours and five minutes to train to completion.

Classification and location regression loss functions for individual images are referenced from Liu et al., 2020 and S. Ren et al., 2017. The functions are as follows:

Equation 2-1

$$L(\{p_i, t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \alpha \frac{1}{N_{loc}} \sum_i p_i^* L_{loc}(t_i, t_i^*)$$

Here N_{cls} , N_{loc} , and α represent loss parameters. The index of an anchor in a mini-batch is represented by i .

Equation 2-2

$$L_{cls}(p_i, p_i^*) = -\log [p_i p_i^* + (1 - p_i)(1 - p_i^*)]$$

p_i is the predicted probability of anchor i being an object. p_i^* is zero if the anchor is negative and is one if it is positive.

Equation 2-3

$$L_{loc}(t_i, t_i^*) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L1}(t_i - t_i^*)$$

Equation 2-4

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Equation 2-5

$$\begin{cases} t_x = \frac{(x - x_a)}{w_a}, & t_y = \frac{(y - y_a)}{h_a} \\ t_w = \log\left(\frac{w}{w_a}\right), & t_h = \log\left(\frac{h}{h_a}\right) \\ t_x^* = \frac{(x^* - x_a)}{w_a}, & t_y^* = \frac{(y^* - y_a)}{h_a} \\ t_w^* = \log\left(\frac{w^*}{w_a}\right), & t_h^* = \log\left(\frac{h^*}{h_a}\right) \end{cases}$$

x, y are coordinates of the center point of the ground truth bounding box; w, h is the width and height of the ground truth bounding box; and x, x_a , and x^* correspond to the predicted bounding box, the applied anchor box, and the ground-truth bounding box, respectively.

2.2.6 Model Evaluation

The evaluation methods are based on the model's ability and accuracy in detecting and labeling the objects of interest. For each input image, the object detection model applied a set of regression boxes with an attached confidence score ranging from zero to one. Each predicted bounding box possessed the same format as the ground truth box: {Xmin, Ymin, Xmax, Ymax}.

Intersection over Union (IoU) is a popular evaluation metric used to measure an object detection model's accuracy. The formula for IoU is:

Equation 2-6

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

The detection model applies the predicted bounding box to the image, and the IoU is calculated. If the ground truth bounding box and the predicted bounding box match up perfectly, the IoU is equal to one. For this experiment, the IoU threshold is set to 0.5, meaning if the IoU value exceeds the threshold, the detected object is considered correct. Correctly applied bounding boxes are marked as a true positive (TP) while bounding boxes less than the threshold are supposed false positives (FP). Average Precision (AP) is the average precision over all detected object categories. The precision of a given classification is given as the ratio of true positives and the total number of predicted positives. The formula to find precision is as follows:

Equation 2-7

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

The recall of classification, also known as the true positive rate, is the ratio of true positives and the total number of ground truth positives. The formula to find recall is as follows:

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

A high precision value indicates a low number of false positives, and a high recall value indicates a high number of the targets can be detected via the model.

2.3 Results

2.3.1 Loss Curves

The total loss drop curve from training is shown below in Figure 2-7. The loss dropped drastically in the first 30 epochs and then leveled off after 100 epochs at a loss value of approximately 0.1.

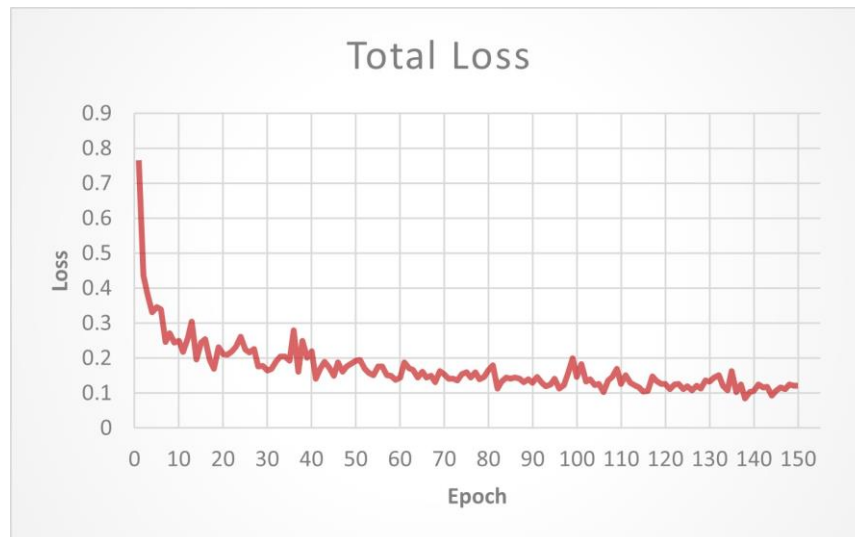


Figure 2-7 The total loss curve with VGG16 as the feature map extraction in a Faster R-CNN model.

2.3.2 Precision and Recall

Detection results are classified in one of three ways: true positive (TP), false positive (FP), or false negative (FN). Examples of each of these can be seen in Figure 2-8. The “o30: 99” and “u30: 99” means the model assessed a 99% probability that each of those objects belongs to the over 30% defoliation and under 30% defoliation classes, respectively. For the over 30% defoliation class, there were 22 TP, 5 FP, and 3 FN (Table 2-1). For the under 30% defoliation class, there were 27 TP, 1 FP, and 2 FN (Table 2-1). The precision and recall values for each class and the average precision (AP) and average recall (AR) can be seen in Table 2-2. Overall, the model identified and labeled 49 of the 60 test images correctly (81.67%).

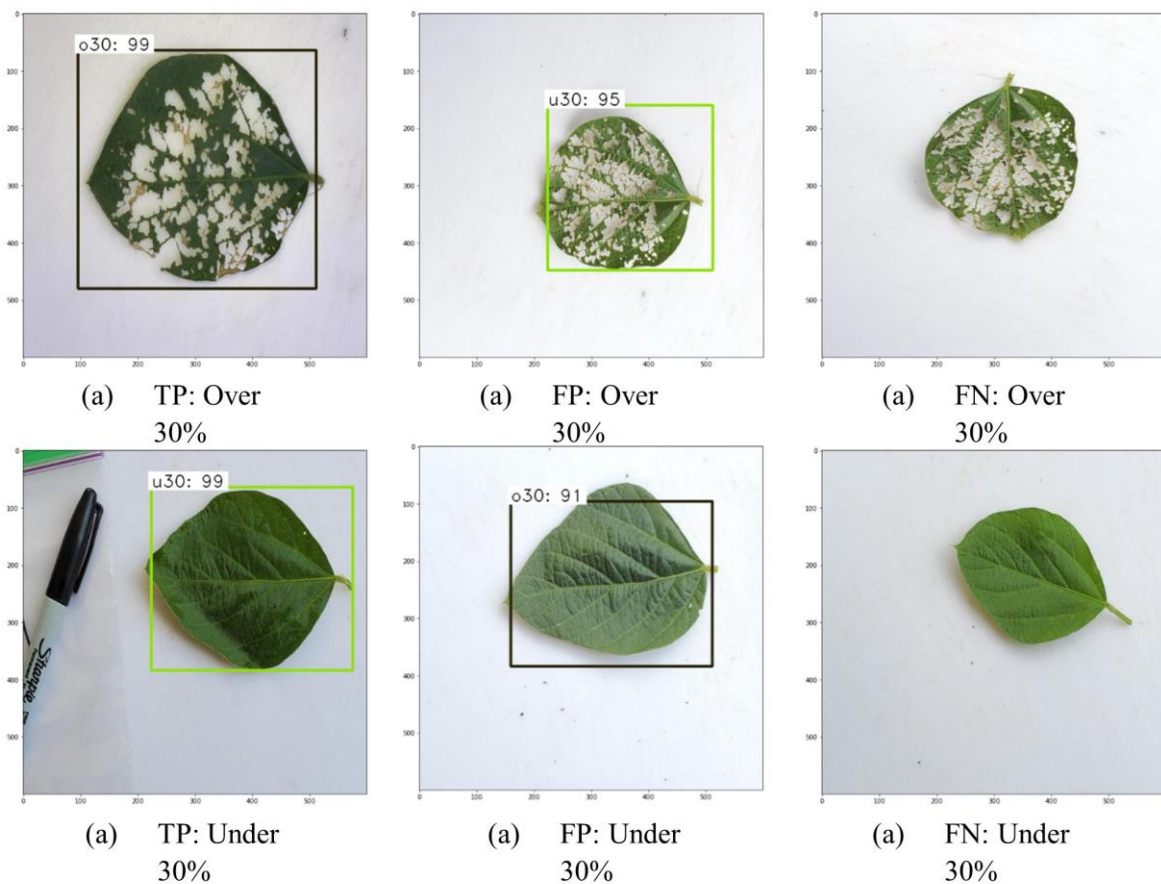


Figure 2-8 Examples of results from the model. (a-c) represent the three types of results from the over 30% classification: TP, FP, and FN. (d-f) represents the three types from the under 30% classification.

Table 2-1 The resulting true positive, false positive, and false negative values for the test dataset. There are a total of 30 images in each class.

	True Positives	False Positives	False Negatives
Over 30% Defoliation	22	5	3
Under 30% Defoliation	27	1	2

Table 2-2 Precision and recall for the over 30% defoliation class, under 30% defoliation class, the average precision (AP), and the average recall (AR).

	Over 30% Defoliation	Under 30% Defoliation	Average
Precision	81.48%	96.43%	88.96%
Recall	88.00%	93.10%	90.55%

2.3.3 Sample of Results

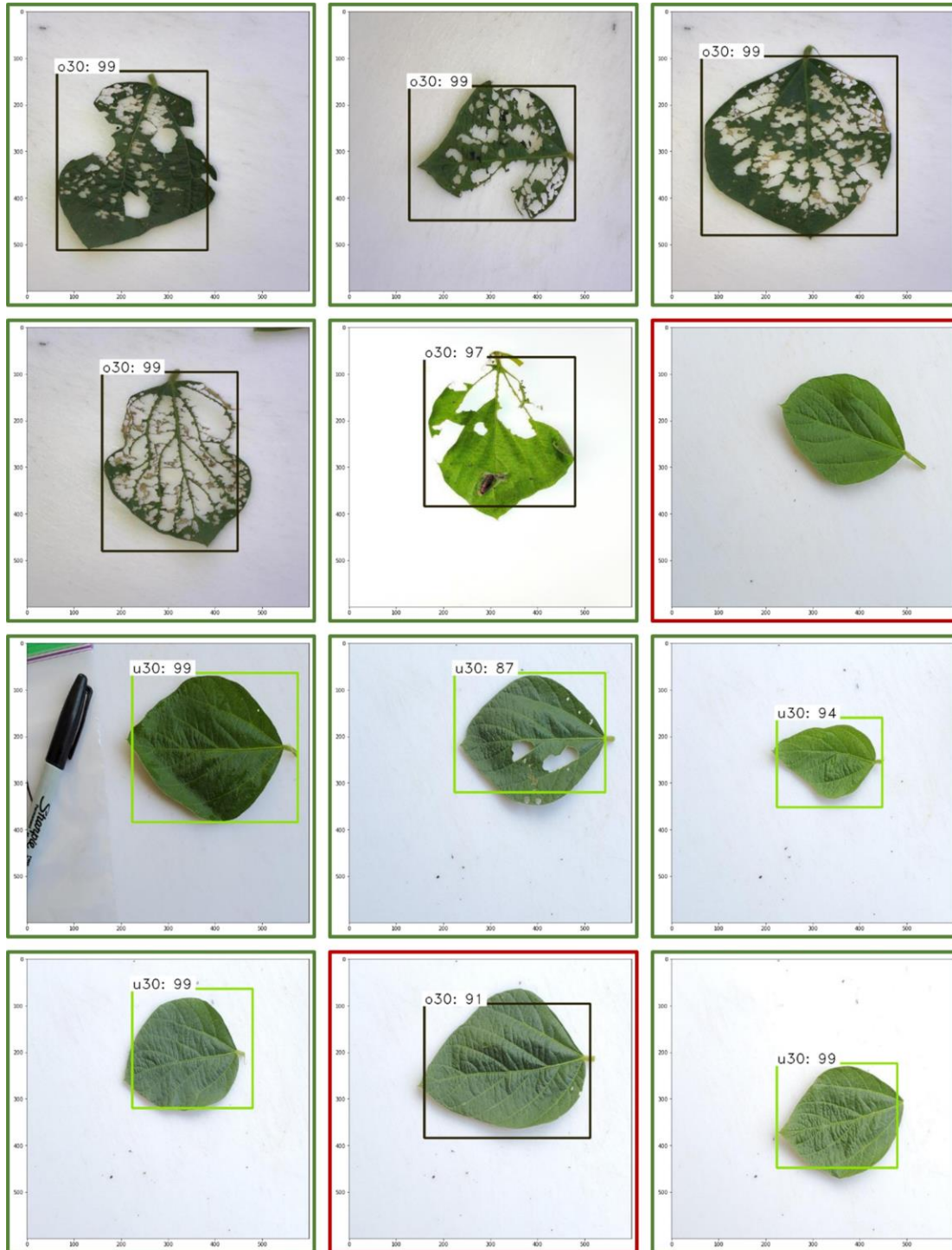


Figure 2-9 Results from 12 out of the 60 test images. A green border represents a correct classification and detection, while a red border represents an incorrect classification or detection.

2.4 Discussion and Conclusion

The current study aimed to evaluate the practicality and accuracy of detecting and labeling soybean leaflets based on their specific defoliation level as either over or under 30% defoliated. This was conducted using a modified Faster R-CNN with a VGG16 feature extraction network on a leaflet database collected with a smartphone. Using this architecture, average precision and average recall values from the two classes were 88.96% and 90.55%, respectively. The over 30% defoliation class obtained a precision and recall value of 81.48% and 88.00%, respectively, while the under 30% defoliation class obtained a precision score of 96.43% and recall score of 93.10%. The discrepancy in these scores (14.95% in the recall scores) indicates the model has a more difficult time correctly detecting and classifying the over 30% defoliation leaflets. This is likely due to the higher number of unique edge patterns present in a highly defoliated leaf compared to a non-defoliated leaf. In other words, non-defoliated leaflets are more uniform in shape and features than highly defoliated leaflets. This concept allows the detection and classification accuracy to be much higher.

A method to potentially improve the model's performance would be to obtain more training data, especially for the over 30% defoliation dataset. This acquisition of training data should not necessarily be identical to previous data. Images with various backgrounds, lighting conditions, and leaf orientation should prove beneficial in improving the model's performance for further applications. For example, if this model is to be implemented in the field, training data of leaves should be collected with a human hand holding the leaf with either the canopy or background scenery within the image. This increase in data should lead to better overall detection and evaluation metrics through the model being exposed to a greater number of unique defoliated leaflets. The dataset used in this study only consisted of 39 unique over 30% defoliated leaflets.

Flipping and transposing the images allowed for more training data but obtaining a more significant number could improve model performance.

Future work should be conducted to employ various feature extraction architectures and compare their performances. Some examples of these architectures include VGG19, VGG20, ResNet50, ResNet101, etc. Implementing this concept in real-time on a smartphone to get rapid detection and defoliation classification should also be a potential future direction of this work. Obtaining this information in situ could prove valuable for researchers and soybean growers alike. Having this technology easily accessible could lead to accurate and precise detection of defoliation within the canopy. This method would also reduce human error and lead to more accurate assessments of insect infestation rates. Lastly, future work should be conducted on identifying which defoliators caused the damage included within the leaf based on the pattern. This would be valuable for the reason that different insects are managed in different ways. This would make the overall assessment of the defoliation more accurate, leading to more accurate management practices.

2.5 Chapter 2 References

- Albawi, S., Abed Mohammed, T., & ALZAWI, S. (2017). *Understanding of a Convolutional Neural Network*. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to End Learning for Self-Driving Cars. *ArXiv:1604.07316 [Cs]*. <http://arxiv.org/abs/1604.07316>
- Bradshaw, C. J. A., Leroy, B., Bellard, C., Roiz, D., Albert, C., Fournier, A., Barbet-Massin, M., Salles, J.-M., Simard, F., & Courchamp, F. (2016). Massive yet grossly underestimated global costs of invasive insects. *Nature Communications*, 7(1), 12986. <https://doi.org/10.1038/ncomms12986>
- Briske, D. D., & Richards, J. H. (1995). Plant responses to defoliation: A physiological, morphological and demographic evaluation. *Wildland Plants: Physiological Ecology and Developmental Morphology*, 635–710.
- Clément, A., Verfaillie, T., Lormel, C., & Jaloux, B. (2015). A new colour vision system to quantify automatically foliar discolouration caused by insect pests feeding on leaf cells. *Biosystems Engineering*, 133, 128–140. <https://doi.org/10.1016/j.biosystemseng.2015.03.007>
- da Silva, L. A., Bressan, P. O., Gonçalves, D. N., Freitas, D. M., Machado, B. B., & Gonçalves, W. N. (2019). Estimating soybean leaf defoliation using convolutional neural networks and synthetic images. *Computers and Electronics in Agriculture*, 156, 360–368. <https://doi.org/10.1016/j.compag.2018.11.040>

- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition* (p. 255). <https://doi.org/10.1109/CVPR.2009.5206848>
- Easlon, H. M., & Bloom, A. J. (2014). Easy Leaf Area: Automated digital image analysis for rapid and accurate measurement of leaf area. *Applications in Plant Sciences*, 2(7), 1400033. <https://doi.org/10.3732/apps.1400033>
- FAO (Ed.). (2017). *The future of food and agriculture: Trends and challenges*. Food and Agriculture Organization of the United Nations.
- Girshick, R. (2015). Fast_R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. <https://doi.org/10.1109/CVPR.2014.81>
- Hirasawa, T., Aoyama, K., Tanimoto, T., Ishihara, S., Shichijo, S., Ozawa, T., Ohnishi, T., Fujishiro, M., Matsuo, K., Fujisaki, J., & Tada, T. (2018). Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. *Gastric Cancer*, 21(4), 653–660. <https://doi.org/10.1007/s10120-018-0793-2>
- Igathinathane, C., Prakash, V. S. S., Padma, U., Babu, G. R., & Womac, A. R. (2006). Interactive computer software development for leaf area measurement. *Computers and Electronics in Agriculture*, 51(1–2), 1–16. <https://doi.org/10.1016/j.compag.2005.10.003>

- Kogan, M., Turnipseed, S. G., Shepard, M., De Oliveira, E. B., & Borgo, A. (1977). Pilot insect pest management program for soybean in southern Brazil. *Journal of Economic Entomology*, 70(5), 659–663.
- Kvet, J., & Marshal, J. K. (1971). Assessment of leaf area and other assimilating plant. *Plant Photosynthetic Production*.
- LabelImg. Available online: <https://github.com/tzutalin/labelImg> (accessed on 13 March 2021).
- Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2019). *Exploring Strategies for Training Deep Neural Networks*. 10(1), 40.
- Lawrence, S., Giles, C. L., Ah Chung Tsoi, & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1), 98–113. <https://doi.org/10.1109/72.554195>
- Liu, Y., Cen, C., Che, Y., Ke, R., Ma, Y., & Ma, Y. (2020). Detection of Maize Tassels from UAV RGB Imagery with Faster R-CNN. *Remote Sensing*, 12(2), 338. <https://doi.org/10.3390/rs12020338>
- Machado, B. B., Orue, J. P. M., Arruda, M. S., Santos, C. V., Sarath, D. S., Goncalves, W. N., Silva, G. G., Pistori, H., Roel, A. R., & Rodrigues-Jr, J. F. (2016). BioLeaf: A professional mobile application to measure foliar damage caused by insect herbivory. *Computers and Electronics in Agriculture*, 129, 44–55. <https://doi.org/10.1016/j.compag.2016.09.007>
- Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>

- Pimentel, D. (2005). 'Environmental and Economic Costs of the Application of Pesticides Primarily in the United States.' *Environment, Development and Sustainability*, 7(2), 229–252. <https://doi.org/10.1007/s10668-005-7314-2>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Ren, Y., Zhu, C., & Xiao, S. (2018). Small object detection in optical remote sensing images via modified faster R-CNN. *Applied Sciences*, 8(5), 813.
- Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556 [Cs]*. <http://arxiv.org/abs/1409.1556>
- 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*, 108(50), 20260–20264. <https://doi.org/10.1073/pnas.1116437108>

Chapter 3 Abstract

Invasive insect-induced defoliation is a significant concern for maximizing potential soybean yields. The prevalence of UAV collected imagery and deep learning provides an opportunity to improve upon defoliation detection and assessment. This study uses images of soybeans captured via UAV at the canopy level to test the performance of a modified Faster Region-based Convolutional Network (Faster R-CNN) with a visual geometry group neural network (VGG16). The proposed model generated an output image with an applied bounding box(s). The average precision and recall over both classes were 26.16% and 65.00%, respectively. In future work, further improvements to the model's performance could be made by enlarging the training dataset to include more defoliation examples.

Chapter 3 - Detection of Insect-Induced Soybean Defoliation from UAV RGB Imagery with Faster R-CNN

3.1 Introduction

Demand for food and other agricultural commodities is estimated to rise by 50 percent by 2050 (FAO, 2017). This surge in demand is due to population growth, shifts towards increasing urbanization, and rises in per capita income (FAO, 2017; Tilman et al., 2011). To combat this looming concern, improved crop management strategies, specifically invasive insect control, are vital to improving productivity.

Annually, pest-induced crop production losses total 20 to 40 percent (FAO, 2017) and cost the global economy an estimated US\$70 billion (Bradshaw et al., 2016; da Silva et al., 2019). The primary concern with invasive insects is herbivory, especially defoliation, of the canopy. Defoliation lowers the plant's photosynthetic and energy production potential and, consequently, its final production capacity (Briske & Richards, 1995). For this reason, it is paramount to monitor defoliation levels within the canopy in case preventative actions must be implemented. For soybeans in the vegetative state, prevention measures become necessary when defoliation levels reach an average of 30% throughout the canopy (Kogan et al., 1977).

Quantifying the insect-induced defoliation is necessary for farmers, crop scouts, and entomologists for enhanced integrated pest management (IPM) practices. The monitoring of defoliation is usually performed manually and is considered tedious and time-expensive. Typically, leaf defoliation is visually approximated by an expert using a guide that compares leaf samples to example leaves with known defoliation levels (Kogan et al., 1977). The grid counting method is another commonly used technique. This practice involves an expert positioning the leaf

on a grid and counting the number of squares to assess the level of defoliation (Kvet & Marshal, 1971). Defoliation levels are habitually overestimated due to human error, which has led to the over-application of insecticides (Machado et al., 2016). Consequently, this also harms public health (Pimentel, 2005).

Leaf area meters can also assess defoliation levels. However, these meters only quantify the leaf area and do not directly quantify defoliation level (Clément et al., 2015; Easlon & Bloom, 2014; Igathinathane et al., 2006). To assess defoliation levels, measurements must be collected before and after defoliation, making these techniques impractical to implement in real-time. Smartphones have recently become a viable option making data acquisition more manageable and timely. Mechado et al. (2016) constructed a mobile application called BioLeaf to estimate insect-induced defoliation. This novel approach accepts input soybean leaf images and estimates the defoliation level by progressing through several steps, including image thresholding to reduce noise and employing Bezier curves to recreate the original leaf border. It then calculates the defoliation level in relation to leaf area as a percent defoliated. Although these methods improve upon previous work regarding speed and accuracy, they still require expert intervention.

When computers become the primary medium for object detection tasks, complex algorithms and models must be employed. These models typically have multiple layers and are referred to as Deep Neural Networks (Larochelle et al., 2019). One of the most popular deep neural networks is the Convolutional Neural Network (CNN) (Albawi et al., 2017). CNN's have paved the way for tasks such as face recognition (Lawrence et al., 1997), self-driving cars (Bojarski et al., 2016), and cancer detection (Hirasawa et al., 2018). Da Silva et al. (2018) constructed a fully automated method to estimate leaf defoliation level utilizing a convolutional neural network (CNN). Their model reconstructed the damaged leaf and assigned estimated percent defoliation.

Although this approach further improved defoliation assessment speed and reliability, it required that the images first be preprocessed and converted into a binary color scale.

Unmanned aerial vehicles (UAVs) hold great promise of value for agriculture production and research. UAVs have many advantages over ground-based sensors and human subjects. These advantages include superior maneuverability, wide field of view, and data collection resulting in virtually no damage to the studied crops. With the rise in UAV technology adoption, object detection practices have naturally become one of the most fruitful applications. UAV-based object detection has allowed for tassel counting in maize (Kumar et al., 2019), crop disease identification (Ha et al., 2017), tree species identification (Santos et al., 2019), and has also proven helpful for many other agricultural applications.

This study aims to utilize the VGG16 (Visual Geometric Group – 16) (Simonyan & Zisserman, 2015) as the feature extraction network in Faster R-CNN (Region-based Convolution Neural Network) (Ren et al., 2017) to fully automate the detection of insect-induced soybean defoliation from UAV-collected imagery.

The remainder of this paper is organized as follows: In section 3.2, the proposed novel methods and materials are introduced. This section also contains information about image acquisition, labeling, and model description. Results are included in Section 3.3. Lastly, Section 3.4 concludes the paper and discusses potential directions for future work.

3.2 Materials and Methods

3.2.1 Image Acquisition and Labeling

Image datasets were collected on July 22nd and 23rd, 2020, from experimental fields near Manhattan, Kansas, United States (39.115837, -96.635032), as shown in Figure 3-2. In total, 357 images of mid-vegetative stage soybean leaflets were captured with a DJI Mavic Pro UAV equipped with a 12-megapixel camera at a flying height of approximately 1 meter. Each original image was 4000x3000 pixels in size. These raw images are too large to distinguish defoliated leaves adequately; therefore, each image must be manually resized to approximately 600x600 pixels. An example of this resizing can be seen in Figure 3-1.

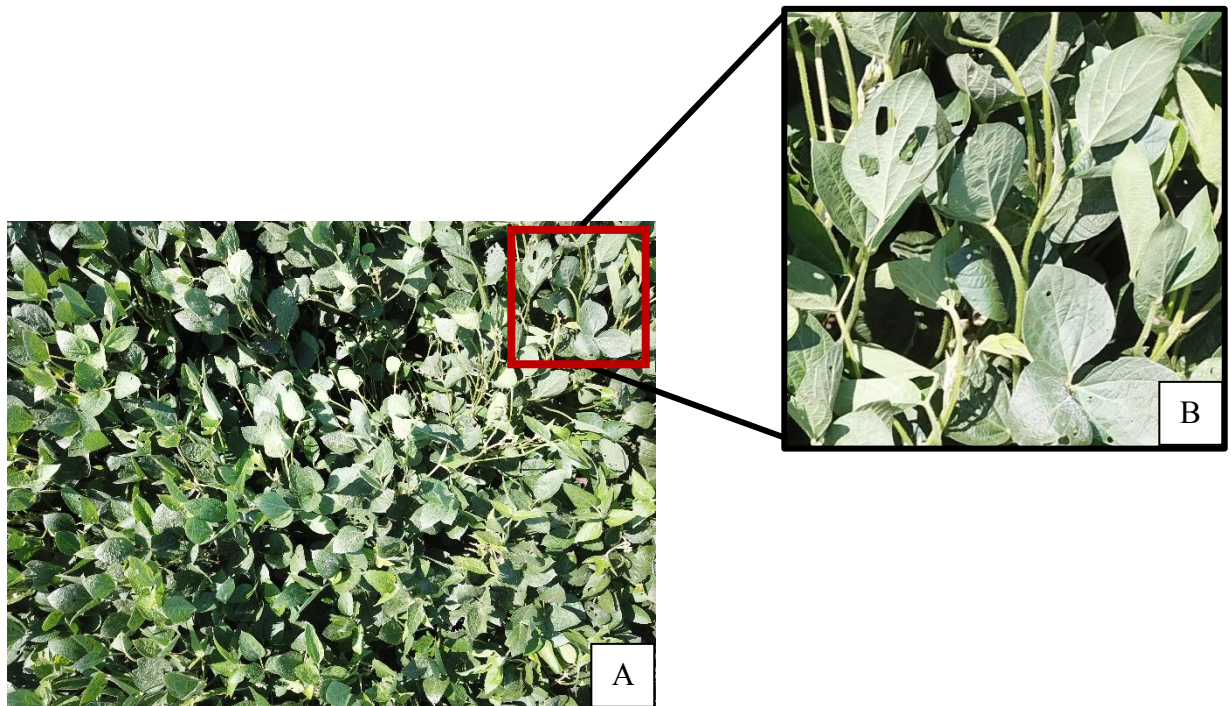


Figure 3-1 Example of a raw UAV image downsized to approximately 600x600 pixels to contain defoliation.

The graphical image annotation tool LabelImg (LabelImg, n.d.) was used to generate the ground truth bounding boxes in the images, as shown in Figure 3-3. The annotated images were then randomly divided into train and test datasets. There would be 200 training images and 40 test images in total. The training set and the test set were separated according to the ratio of ~5:1.

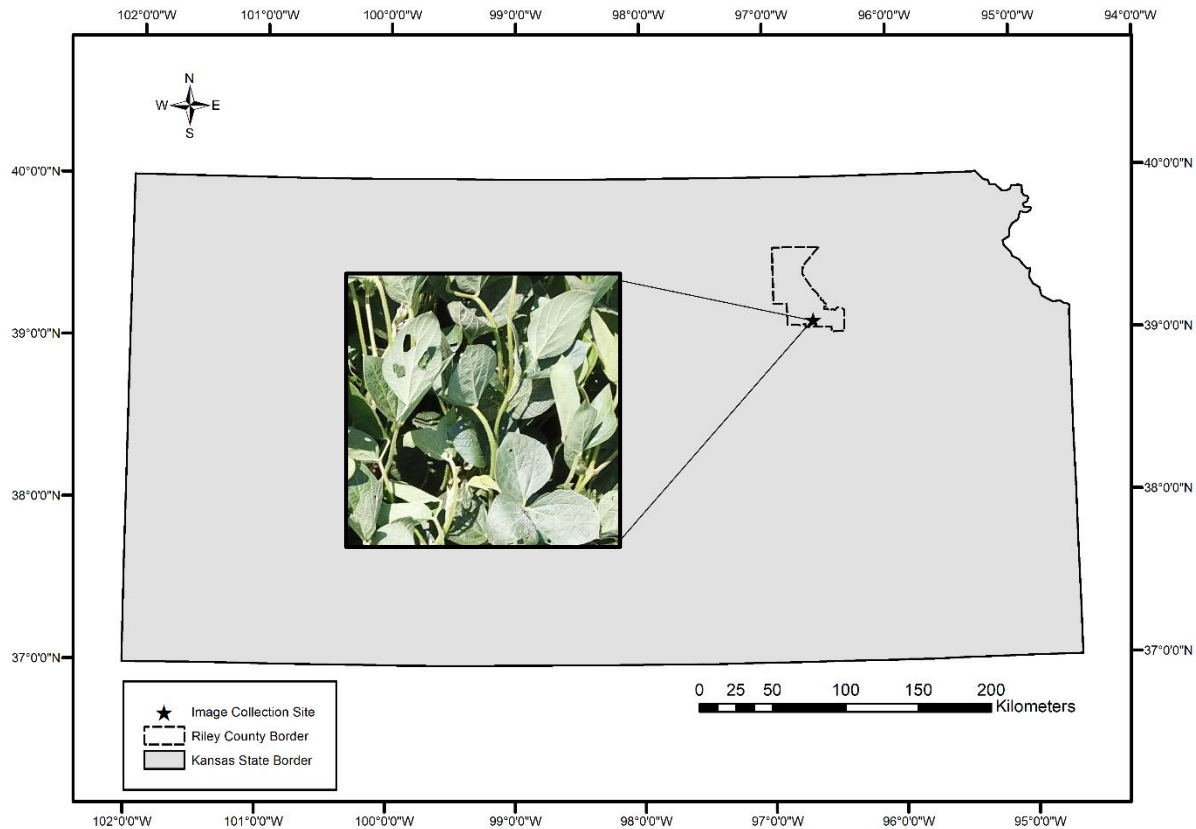


Figure 3-2 Image datasets were collected at an experiment site southwest of Manhattan, Kansas, United States. Images were collected using a DJI Mavic Pro, with a flying height of approximately 1 meter above the canopy. Acquisition dates were July 22nd and 23rd, 2020.



Figure 3-3 Example of a bounding box interactively drawn using the LabellImage software.

3.2.2 *Faster R-CNN Description*

Faster R-CNN allows for an end-to-end training algorithm for object detection. The general structure is shown in Figure 3-4. Feature maps are generated for each annotated image by inputting the image into a convolutional neural network (CNN). The region proposal network (RPN) determines if an object is foreground or background. These region proposals from the RPN are then processed by the region of interest (RoI) pooling layer. The model then splits into two fully connected layers (FC). The first is a boundary box regressor. This FC layer corrects the bounding box's position to fit the detected object more closely. The second FC layer uses a softmax classification to determine which category the detected object belongs.

Previous R-CNN architectures (R-CNN and Fast R-CNN) (Girshick, 2015; Girshick et al., 2014) used a selective search method to predict the presence of an object. Selective search applied approximately 2000 proposals to each image, and each proposal had to be passed through a CNN,

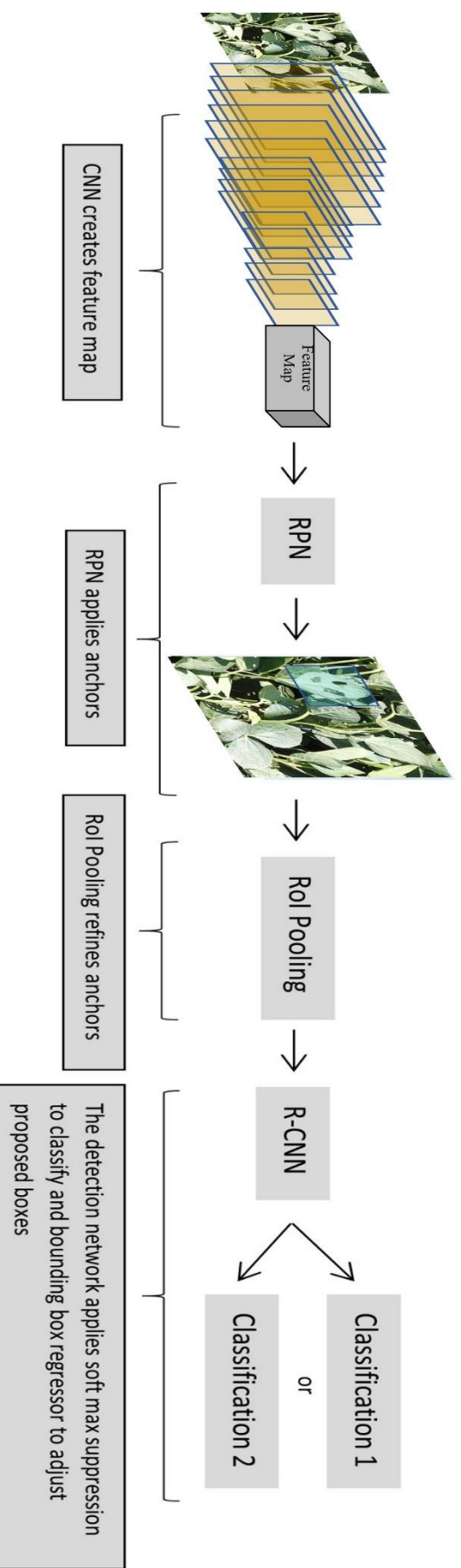


Figure 3-4 Diagram illustrating the Faster R-CNN model. The CNN (VGG16) generates a feature map through five convolutional layers. The region proposal network (RPN) determines if the object in question is foreground or background. Lastly, region of interest (RoI) pooling occurs, and the region proposals are corrected and classified.

which became very computationally taxing. Faster R-CNN is superior to previous architectures by employing a region proposal network (RPN). This RPN allows each input image to pass through the CNN to create a feature map and then predict features. The RPN uses user-defined ground truth boxes and applies its bounding boxes, called anchors (Figure 3-5), with different aspect ratios. Using the ground truth boxes and the applied anchors, it calculates the intersection over union (IoU). IoU simply measures how much overlap is present between the applied anchors and ground truth boxes. Anchors with an IoU greater than .7 are considered foreground (an object), and anchors with an IoU of less than .3 are considered background (non-object).

3.2.3 Anchor Size Justification

For Faster R-CNN, the anchor size's default parameters are $[128^2, 256^2, 512^2]$. It was found that some of the soybean leaflets required smaller anchors. Therefore, anchor size was adjusted to $[64^2, 128^2, 256^2]$. This practice was inspired by previous work showing that utilization of smaller anchor sizes can improve detection results when the dataset contains objects much tinier than the smallest anchor size (Y. Ren et al., 2018).

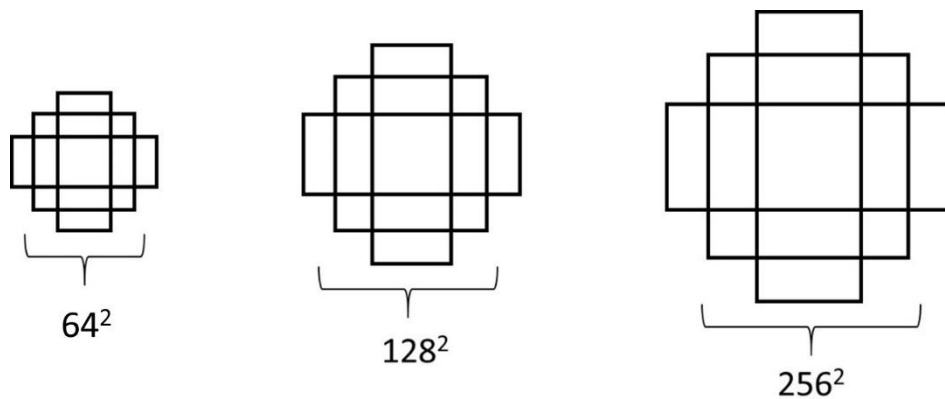


Figure 3-5 Examples of anchors applied through the region proposal network (RPN). All nine of the anchor are applied to the proposed region.

3.2.4 Convolutional Neural Network Architecture

The Visual Geometry Group first conceived the visual geometry group network (VGGNet) at Oxford University (Simonyan & Zisserman, 2015). In this paper, a state-of-the-art Faster R-CNN with the VGG16 architecture was used for the object detection network. In 2014, the VGG16 network architecture placed 2nd in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) annual competition. However, it performed better than the first place finisher (GoogLeNet) to detect multiple objects (Deng et al., 2009). The original VGG16 network is pre-trained with the ImageNet dataset and then retrained with the soybean canopy dataset. The default pixel dimension for the input image is 224 x 224 for the VGG16 model. This parameter was changed to 600 x 600 pixels in this experiment to fit the canopy dataset better. The VGG16 model used for feature map extraction in Faster R-CNN in this study can be seen in Figure 3-6.

3.2.5 Training

The model was implemented using the Keras framework on an Ubuntu 20.04 LTS (64-bit) workstation with an Intel Xeon 10 core Silver 4210 CPU and an 8 gigabyte GeForce RTX 2080 Super with 3072 CUDA cores GPU. Weight parameters from ImageNet's pre-trained model were used to reduce the training time. A concept known as transfer learning (Pan & Yang, 2010). The model was trained through two separate training runs of 75 epochs for a total of 150 epochs. The epoch number is a parameter that determines the number of times the training process works through the entire dataset. Epoch length (also known as batch size) was set to 300. This parameter determines how many of the dataset images are fed into the model at once. There were 200 training images in the dataset. The iteration number was set to zero because it was deemed unnecessary to

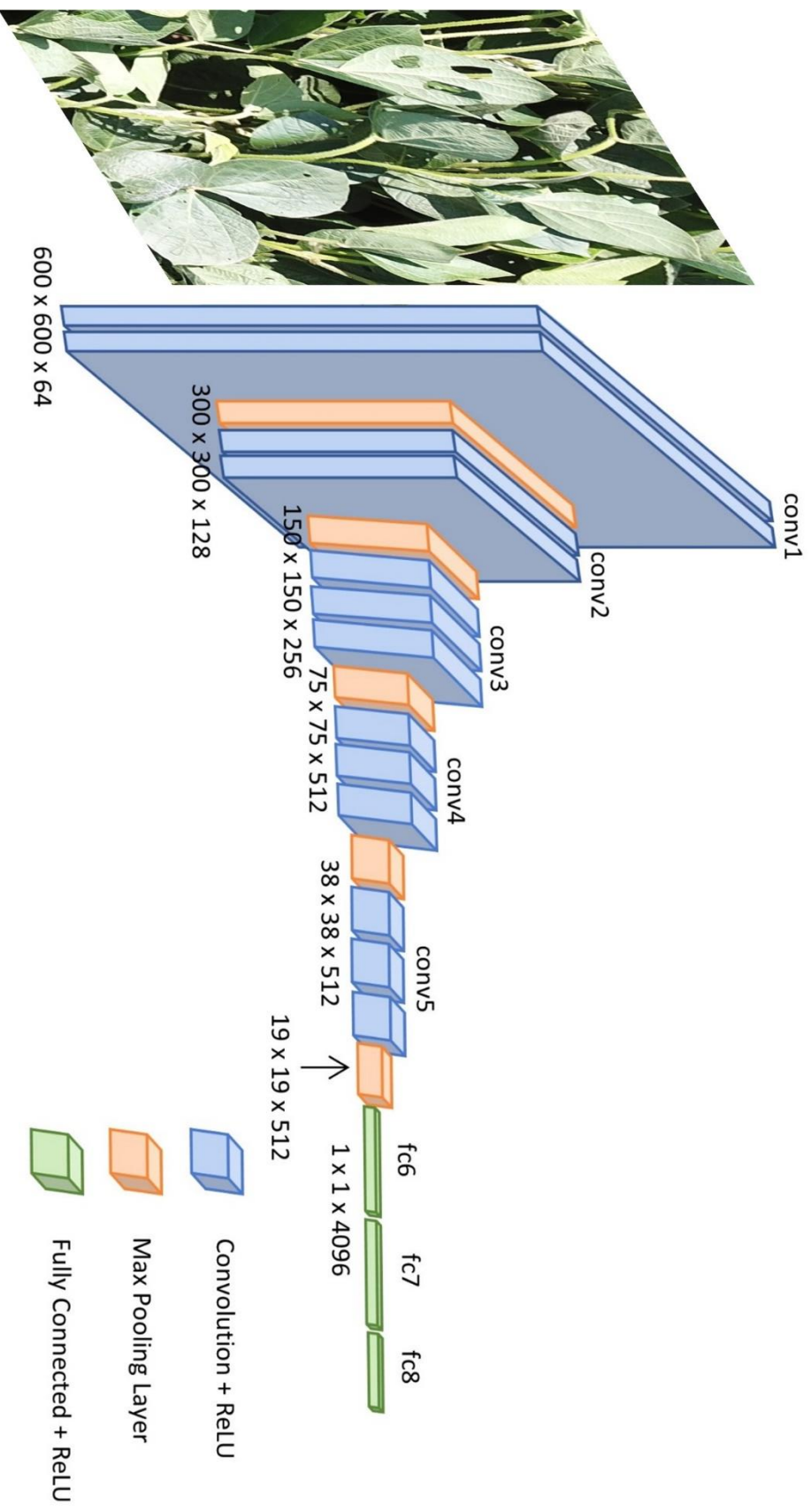


Figure 3-6 Diagram of the VGG16 architecture for feature extraction.

to iterate the training process for this current study. In total, it took the model two hours and seventeen minutes to train to completion.

Classification and location regression loss functions for individual images are referenced from Liu et al., 2020 and S. Ren et al., 2017. The functions are as follows:

Equation 3-1

$$L(\{p_i, t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \alpha \frac{1}{N_{loc}} \sum_i p_i^* L_{loc}(t_i, t_i^*)$$

Here N_{cls} , N_{loc} , and α represent loss parameters. The index of an anchor in a mini-batch is represented by i .

Equation 3-2

$$L_{cls}(p_i, p_i^*) = -\log [p_i p_i^* + (1 - p_i)(1 - p_i^*)]$$

p_i is the predicted probability of anchor i being an object. p_i^* is zero if the anchor is negative and is one if it is positive.

Equation 3-3

$$L_{loc}(t_i, t_i^*) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L1}(t_i - t_i^*)$$

Equation 3-4

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Equation 3-5

$$\begin{cases} t_x = \frac{(x - x_a)}{w_a}, & t_y = \frac{(y - y_a)}{h_a} \\ t_w = \log\left(\frac{w}{w_a}\right), & t_h = \log\left(\frac{h}{h_a}\right) \\ t_x^* = \frac{(x^* - x_a)}{w_a}, & t_y^* = \frac{(y^* - y_a)}{h_a} \\ t_w^* = \log\left(\frac{w^*}{w_a}\right), & t_h^* = \log\left(\frac{h^*}{h_a}\right) \end{cases}$$

x, y are coordinates of the center point of the ground truth bounding box; w, h is the width and height of the ground truth bounding box; and x, x_a , and x^* correspond to the predicted bounding box, the applied anchor box, and the ground-truth bounding box, respectively.

3.2.6 Model Evaluation

The evaluation methods are based on the model's ability and accuracy in detecting and labeling the objects of interest. For each input image, the object detection model applied a set of regression boxes with an attached confidence score ranging from zero to one. Each predicted bounding box possessed the same format as the ground truth box: {Xmin, Ymin, Xmax, Ymax}.

Intersection over Union (IoU) is a popular evaluation metric used to measure an object detection model's accuracy. The formula for IoU is:

Equation 3-6

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

The detection model applies the predicted bounding box to the image, and the IoU is calculated. If the ground truth bounding box and the predicted bounding box match up perfectly,

the IoU is equal to one. For this experiment, the IoU threshold is set to 0.5, meaning if the IoU value exceeds the threshold, the detected object is considered correct. Correctly applied bounding boxes are marked as a true positive (TP), while bounding boxes less than the threshold are supposed false positives (FP). Average Precision (AP) is the average precision over all detected object categories. The precision of a given classification is given as the ratio of true positives and the total number of predicted positives. The formula to find precision is as follows:

Equation 3-7

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

The recall of classification, also known as the true positive rate, is the ratio of true positives and the total number of ground truth positives. The formula to find recall is as follows:

Equation 3-8

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

A high accuracy indicates a low number of false positives, and a high recall indicates most of the targets can be detected via the model.

3.3 Results

The total loss drop curve from training is shown below in Figure 3-7. The loss dropped drastically in the first 30 epochs and then leveled off after 100 epochs at a loss value of approximately 0.2.



Figure 3-7 The total loss curve with VGG16 as the feature map extraction in a Faster R-CNN model.

2.3.2 Precision and Recall

Detection results are classified in one of three ways: true positive (TP), false positive (FP), or false negative (FN). Examples of each of these can be seen in Figure 3-8. The “Def: 80” located



Figure 3-8 Model result with examples of instances of true positive (green box), false positive (red box), and false negative (yellow box).

on the top of the applied bounding boxes means the model assessed an 80% probability that each detected region contains defoliation. Out of the 40 randomly selected test images, there were 39 TP, 116 FP, and 21 FN (Table 3-1). The precision and recall values for each class can be seen in Table 3-2.

Table 3-1 The resulting true positive, false positive, and false negative values for the test dataset. There were a total of 40 test images in the dataset.

	True Positives	False Positives	False Negatives
Defoliated Leaves	39	116	21

Table 3-2 Precision and recall values for the test dataset.

	Defoliated Leaves
Precision	25.16%
Recall	65.00%

3.3.3 Sample of Results



Figure 3-9 Results from 9 out of the 40 test images. The blue boxes are the applied bounding boxes.

3.4 Discussion and Conclusion

The current study aimed to evaluate the practicality and accuracy of detecting defoliation from canopy-level images. This was conducted using a modified Faster R-CNN with a VGG16 feature extraction network on a canopy level database collected with a DJI Mavic Pro UAV with a flying height of approximately 1 meter. Using the Faster R-CNN/VGG16 architecture, precision and recall values were 25.16% and 65.00%, respectively. Looking into the resulting precision and recall values, it could be easy to assume that no detection occurred. However, looking at the visual results it can be seen that the model achieved some level of detection of defoliation. However, it proved to be somewhat erroneous by applying many false positives. This likely is a result of the complex nature of the canopy images. Many shadows, high background noise levels, and many different soybean leaves' orientations make object detection a much more difficult task to accomplish accurately. All of these factors may have influenced the model's ability to distinguish between background noise and defoliated leaves. It is worth noting that the primary objective of this study was to determine if the introduced methods were a viable option to detect defoliation in the soybean canopy. The results of this study show that detection is possible. This model, in its current state, cannot explain 100% of the defoliation present within the image. However, it could be used as a first detector to determine if defoliation is present in hard-to-reach locations of a given field. This method could effectively be used as a surveillance application for soybean growers.

With these applications in mind, there is obvious room for improvement. One method to improve the model's performance includes obtaining more training data. This increased level of training data should also include varying levels of defoliation, lighting conditions, and defoliation levels. This study utilized 200 training images with 258 instances of defoliation. Increasing the number would allow the model to be exposed to a broader range of defoliation types and potentially

improve the accuracy of the outputs. Improved technology could also improve this study's practicality, especially when it comes to the UAV camera payload. This study utilized a standard stock camera on the DJI Mavic Pro. This required the images to be collected approximately 1 meter above the canopy, causing “wind whip” from the UAV’s rotors. Equipping the UAV with a 4x zoom camera would allow the pilot to fly the UAV 4 meters above the canopy, collect imagery that appeared to be collected at 1 meter while reducing or even eliminating the “wind whip.” This would allow the collection of canopy images to appear more natural.

Future work should be conducted to employ various feature extraction architectures and compare their performances. Some examples include VGG-19, VGG-20, ResNet50, ResNet101, etc. This would allow for more variety in methods to determine which architecture yielded the most accurate results.

3.5 References

- Albawi, S., Abed Mohammed, T., & ALZAWI, S. (2017). *Understanding of a Convolutional Neural Network*. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to End Learning for Self-Driving Cars. *ArXiv:1604.07316 [Cs]*. <http://arxiv.org/abs/1604.07316>
- Bradshaw, C. J. A., Leroy, B., Bellard, C., Roiz, D., Albert, C., Fournier, A., Barbet-Massin, M., Salles, J.-M., Simard, F., & Courchamp, F. (2016). Massive yet grossly underestimated global costs of invasive insects. *Nature Communications*, 7(1), 12986. <https://doi.org/10.1038/ncomms12986>
- Briske, D. D., & Richards, J. H. (1995). Plant responses to defoliation: A physiological, morphological and demographic evaluation. *Wildland Plants: Physiological Ecology and Developmental Morphology*, 635–710.
- Clément, A., Verfaillie, T., Lormel, C., & Jaloux, B. (2015). A new colour vision system to quantify automatically foliar discolouration caused by insect pests feeding on leaf cells. *Biosystems Engineering*, 133, 128–140. <https://doi.org/10.1016/j.biosystemseng.2015.03.007>
- da Silva, L. A., Bressan, P. O., Gonçalves, D. N., Freitas, D. M., Machado, B. B., & Gonçalves, W. N. (2019). Estimating soybean leaf defoliation using convolutional neural networks and synthetic images. *Computers and Electronics in Agriculture*, 156, 360–368. <https://doi.org/10.1016/j.compag.2018.11.040>

- Easlon, H. M., & Bloom, A. J. (2014). Easy Leaf Area: Automated digital image analysis for rapid and accurate measurement of leaf area. *Applications in Plant Sciences*, 2(7), 1400033. <https://doi.org/10.3732/apps.1400033>
- FAO (Ed.). (2017). *The future of food and agriculture: Trends and challenges*. Food and Agriculture Organization of the United Nations.
- Ha, J. G., Moon, H., Kwak, J. T., Hassan, S. I., Dang, M., Lee, O. N., & Park, H. Y. (2017). Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *Journal of Applied Remote Sensing*, 11(04), 1. <https://doi.org/10.1117/1.JRS.11.042621>
- Hirasawa, T., Aoyama, K., Tanimoto, T., Ishihara, S., Shichijo, S., Ozawa, T., Ohnishi, T., Fujishiro, M., Matsuo, K., Fujisaki, J., & Tada, T. (2018). Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. *Gastric Cancer*, 21(4), 653–660. <https://doi.org/10.1007/s10120-018-0793-2>
- Igathinathane, C., Prakash, V. S. S., Padma, U., Babu, G. R., & Womac, A. R. (2006). Interactive computer software development for leaf area measurement. *Computers and Electronics in Agriculture*, 51(1–2), 1–16. <https://doi.org/10.1016/j.compag.2005.10.003>
- Kogan, M., Turnipseed, S. G., Shepard, M., De Oliveira, E. B., & Borgo, A. (1977). Pilot insect pest management program for soybean in southern Brazil. *Journal of Economic Entomology*, 70(5), 659–663.
- Kumar, A., Taparia, M., Rajalakshmi, P., Desai, U. B., Naik, B., & Guo, W. (2019). UAV Based Remote Sensing for Tassel Detection and Growth Stage Estimation of Maize Crop using F-RCNN. *Computer Vision Problems in Plant Phenotyping*, 3.

- Kvet, J., & Marshal, J. K. (1971). Assessment of leaf area and other assimilating plant. *Plant Photosynthetic Production*.
- Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2019). *Exploring Strategies for Training Deep Neural Networks*. 10(1), 40.
- Lawrence, S., Giles, C. L., Ah Chung Tsoi, & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1), 98–113. <https://doi.org/10.1109/72.554195>
- Machado, B. B., Orue, J. P. M., Arruda, M. S., Santos, C. V., Sarath, D. S., Goncalves, W. N., Silva, G. G., Pistori, H., Roel, A. R., & Rodrigues-Jr, J. F. (2016). BioLeaf: A professional mobile application to measure foliar damage caused by insect herbivory. *Computers and Electronics in Agriculture*, 129, 44–55. <https://doi.org/10.1016/j.compag.2016.09.007>
- Pimentel, D. (2005). ‘Environmental and Economic Costs of the Application of Pesticides Primarily in the United States.’ *Environment, Development and Sustainability*, 7(2), 229–252. <https://doi.org/10.1007/s10668-005-7314-2>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Santos, A. A. dos, Marcato Junior, J., Araújo, M. S., Di Martini, D. R., Tetila, E. C., Siqueira, H. L., Aoki, C., Eltner, A., Matsubara, E. T., Pistori, H., Feitosa, R. Q., Liesenberg, V., & Gonçalves, W. N. (2019). Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs. *Sensors*, 19(16), 3595. <https://doi.org/10.3390/s19163595>

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556 [Cs]*. <http://arxiv.org/abs/1409.1556>

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*, *108*(50), 20260–20264. <https://doi.org/10.1073/pnas.1116437108>

Chapter 4 – Conclusion and Synthesis

4.1 – Manuscript 1

4.1.1 - Summary of Findings

This study aimed to evaluate the practicality and accuracy of detecting and labeling soybean leaflets based on their specific defoliation level. This was conducted using a modified Faster R-CNN with a VGG16 feature extraction network on a leaflet database collected with a smartphone. Using this architecture, average precision and average recall values from the two classes were 88.96% and 90.55%, respectively. The over 30% defoliation class obtained a precision and recall value of 81.48% and 88.00%, respectively, while the under 30% defoliation class obtained a precision score of 96.43% and recall score of 93.10%. The discrepancy in these scores (14.95% in the recall scores) indicates the model has a more difficult time correctly detecting and classifying the over 30% defoliation leaflets. Intuitively, this makes sense due to the higher number of unique edge patterns present in a highly defoliated leaf compared to a non-defoliated leaf. In other words, non-defoliated leaflets are more uniform in shape and features than highly defoliated leaflets. This concept allows the detection rate to be much higher.

4.1.2 - Implications for Crop Pest Management

Quantifying insect-induced defoliation is necessary for farmers, crop scouts, and entomologists for enhanced integrated pest management (IPM) practices. The monitoring of defoliation is usually performed manually and is considered tedious and time-expensive. Typically, leaf defoliation is visually approximated by an expert using a guide that compares leaf samples to example leaves with known defoliation levels or even by a knowledgeable estimate

with no reference. Therefore, defoliation levels are habitually overestimated due to human error, which has led to the over-application of insecticides.

This study hoped to bridge the gap between accuracy and efficiency. The methods used hope to allow users to identify and quantify defoliation on a soybean leaflet, based on a selected threshold, accurately and promptly. This study applied the threshold at 30% defoliation; however, this threshold could be adjusted to any value through the retraining of the model. Reducing error could lead to more accurate and reliable defoliation assessments, leading to more accurate and timely insecticide applications. Although the results from this model appear to be promising, there still needs to be work conducted to improve upon the accuracy and reliability of the model.

4.1.3 - Suggestions for Future Work

There are several avenues to take to further improve upon the obtained results. One method to improve the model's performance includes obtaining more training data, especially for the over 30% defoliation dataset. This increase in data could lead to better overall detection and evaluation metrics through the model being exposed to a more significant number of unique defoliated leaflets. The dataset used in this study only consisted of 39 unique over 30% defoliated leaflets. Flipping and transposing the images allowed for more training data but obtaining a more significant number of individual leaflets could improve model performance. Each defoliated leaf is unique, and there is potential to be a wide range of what a 31% defoliated leaf could look like compared to, say, a 50% defoliated leaf. Including more examples of highly defoliated leaves should only increase the model's accuracy.

Future work could be conducted to employ various feature extraction architectures and compare their performances. Some examples include VGG-19, VGG-20, ResNet50, ResNet101, etc.. Another interesting possibility would be implementing this concept in real-time on a smartphone to get rapid detection and defoliation classification. This could be accomplished by training on the You Only Look Once (YOLO) architecture, allowing for real-time detection from a smartphone or video camera. Obtaining this information in situ could prove valuable for researchers and soybean growers alike. Lastly, comparing this model's defoliation assessment performance against human assessment could be an interesting question worth exploring. It would be worth investigating how human error compares to the model's error.

4.2 – Manuscript 2

4.2.1 - Summary of Findings

The current study aimed to evaluate the practicality and accuracy of detecting defoliation from canopy-level images. This was conducted using a modified Faster R-CNN with a VGG16 feature extraction network on a canopy level database collected with a DJI Mavic Pro UAV with a flying height of approximately 1 meter. Using the Faster R-CNN/VGG16 architecture, precision and recall values were 25.16% and 65.00%, respectively. Looking deeper into the resulting precision and recall values, it can be concluded that the model did a relatively decent job at detecting where there is, in fact, defoliation present. However, it proved to be somewhat erroneous by applying many false positives. This likely is a result of the complex nature of the canopy images. Many shadows, high background noise levels, and many different soybean leaves' orientations make object detection a much more difficult task to accomplish accurately. The

results obtained from this study indicate that further improvements in detection ability need to be made before practical applications can be implemented.

4.2.2 - Implications for Crop Pest Management

As stated earlier, quantifying insect-induced defoliation is necessary for farmers, crop scouts, and entomologists for enhanced integrated pest management (IPM) practices. The monitoring defoliation strategies are usually performed manually and are considered tedious and time-expensive. A trained expert typically performs these tasks, so there is some required knowledge or training level. Even so, defoliation estimates are often overestimated, leading to excessive or increased levels of insecticides.

This study looked to build upon detection practices and increase the accuracy and efficiency of monitoring insect-induced defoliation. This study has laid the groundwork to continue integrating UAV imagery, machine learning, and insect monitoring. The methods utilized in this study do not seem to be adequate for production utilization and continued work needs to be conducted to improve upon results obtained through this method.

4.2.3 - Suggestions for Future Work

One method to improve the model's performance includes obtaining more training data. This study utilized 200 training images with 236 instances of defoliation used and captured with ground truth bounding boxes. Increasing the number would allow the model to be exposed to a broader range of defoliation types and improve the accuracy of the outputs. Improved technology could also improve this study's practicality, especially when it comes to the UAV camera payload. This study utilized a standard stock camera on the DJI Mavic Pro. This required the images to be

collected approximately 1 meter above the canopy, causing some “wind whip” from the UAV’s rotors. Equipping the UAV with a 4x zoom camera would allow the pilot to fly the UAV 4 meters above the canopy, collect imagery that appeared to be collected at 1 meter while reducing or even eliminating the “wind whip.” This would allow the collection of canopy images that appeared more natural.

Future work could be conducted to employ various feature extraction architectures and compare their performances. Some examples include VGG-19, VGG-20, ResNet50, ResNet101, etc. This would allow for more variety in methods to determine which architecture yielded the most accurate results.

4.3 – Synthesis

The two studies discussed in this thesis have overlapping methods, however very different findings, and potential applications. In Chapter 2, a few comparisons with previous methods can be made. Compared to the guide book and grid counting methods (see page 16 in Chapter 1), this model is superior in the fact that it can reduce the potential human error. Reducing this error could have profound impacts on the accuracy and timing of insecticide applications. In regard to the BioLeaf application (see page 17, Chapter 1), the proposed model is also superior in its application because BioLeaf still requires some user input such as drawing missing leaf edges. BioLeaf also takes about a minute in a half to analyze a leaf when factoring in reconstructing some of the leaf edges. This model could give you instantaneous results drastically cutting down on data acquisition time. This model could also be developed to not only determine defoliation levels but also detect feeding patterns and determine which types of insects the defoliation belongs to. An area the guide book and grid counting methods are limited due to the requirement of the user to know the feeding

patterns to assess insect type. This could further improve the insecticide application because different species of insects require different management practices.

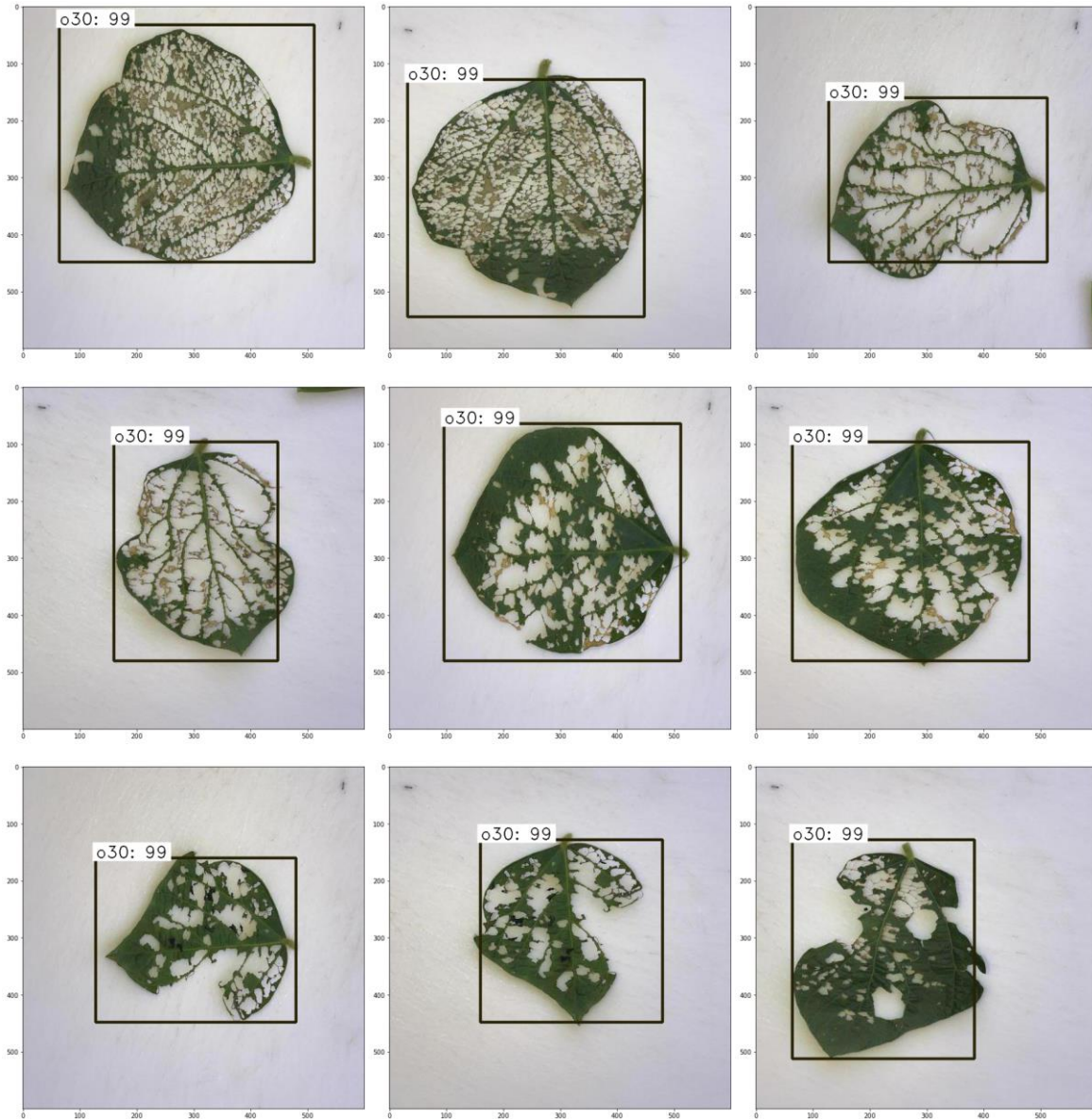
The method discussed in Chapter 3 would be a superior alternative for the surveillance of defoliation from human-based approaches. Instead of an individual having to walk throughout the field, a UAV could be deployed to determine if defoliation is present in difficult to reach regions. This would drastically cut down on time spent in a field. It would allow the user to quickly fly a drone to several locations, collect acceptable amounts of imagery, then run the collected data through the model to determine if further action is required. This could be a valuable tool for any grower, crop scout, or scientist. At this point in the literature, there aren't any alternatives to this method other than human-based detection. Making this a novel tool for detection and a lot of room for improvements and advanced applications. The simple fact that it could cut down on time spent in a field and detect defoliation from an individual standing at the edge of the field is incredibly valuable.

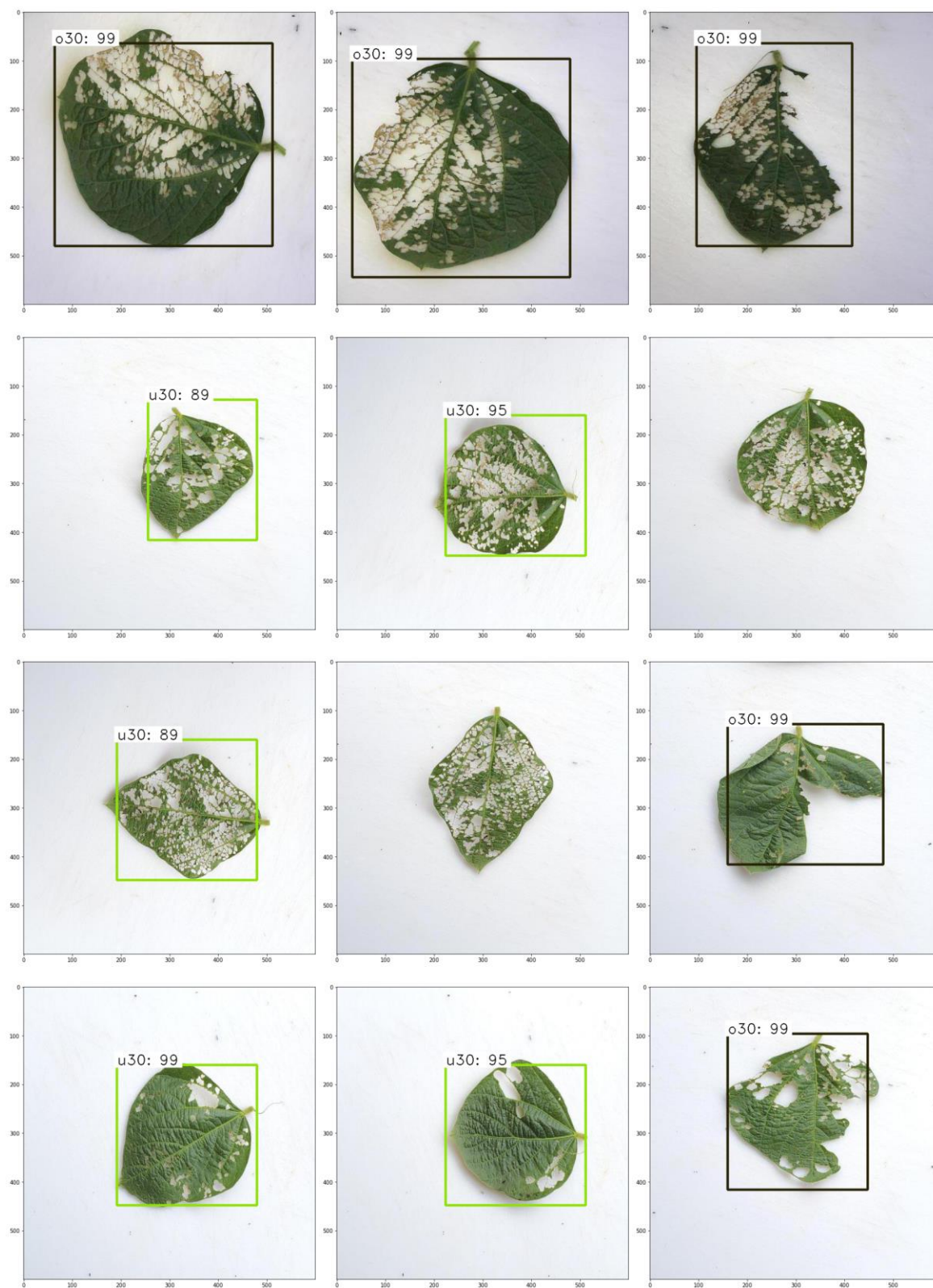
Between the two studies included in this thesis, there were significant differences in the datasets used. Chapter 2 involved a dataset of leaflets with mostly plain white backgrounds and very little to no “noise” contained within the image. This led to very clean and easily identifiable leaflets. The characteristics of this training and test data may also limit the applications of this model in its current state. More data should be used for training including images with varying backgrounds. For example, leaflets with different backgrounds such as a soybean canopy or the horizon in the background could be included to improve the model's overall robustness. In contrast, Chapter 3 included UAV imagery. These images were much more complex and contained a much higher level of “noise”. These UAV images contain background shadows, soil, stems, both sides of leaves, and leaves at different angles and orientations. All of these factors lead to an overall

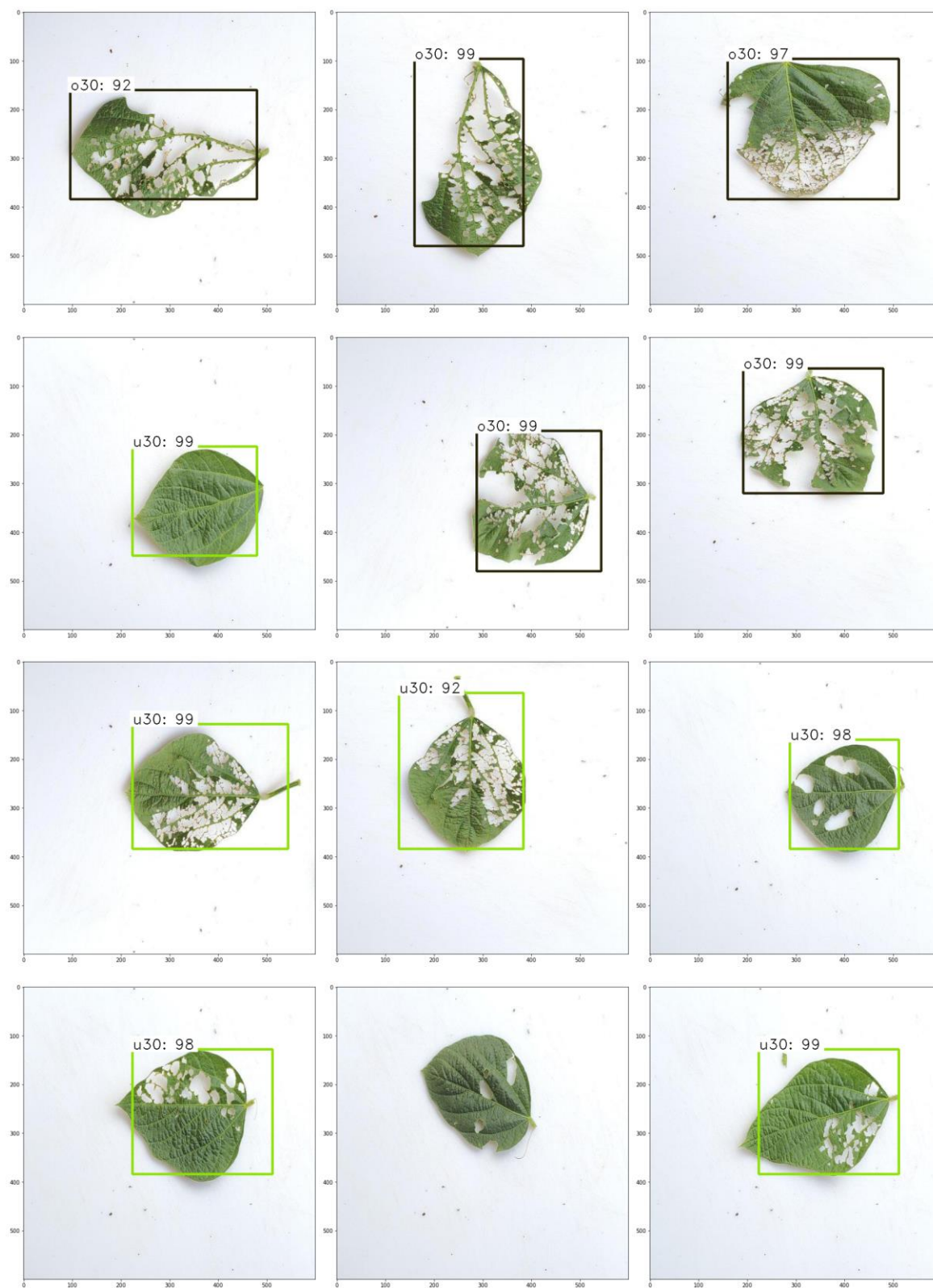
increase in visual complexity to this dataset. These different levels of image complexity are likely the result of the differing levels of accuracy between the two studies.

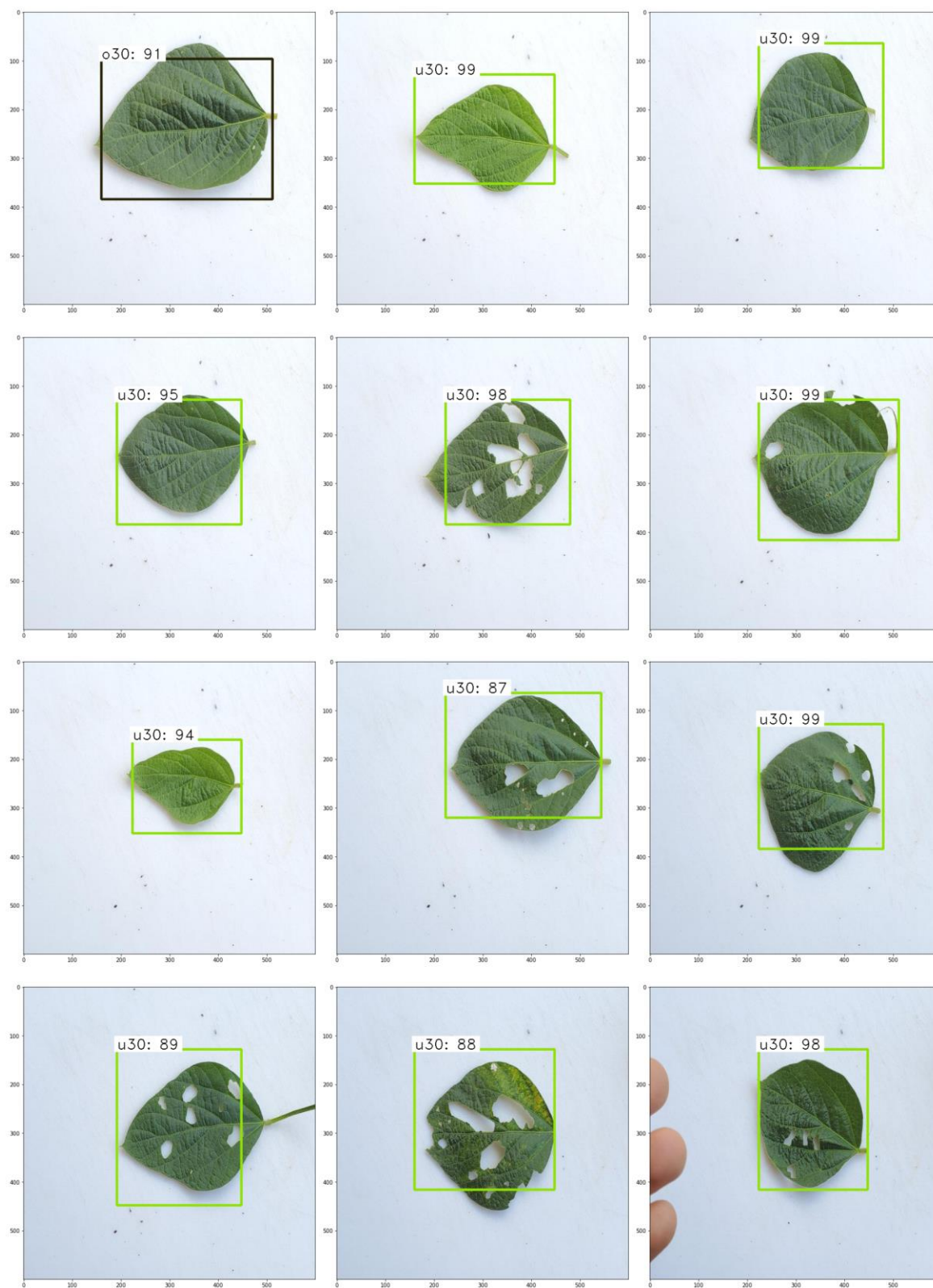
Appendix A - Result Images

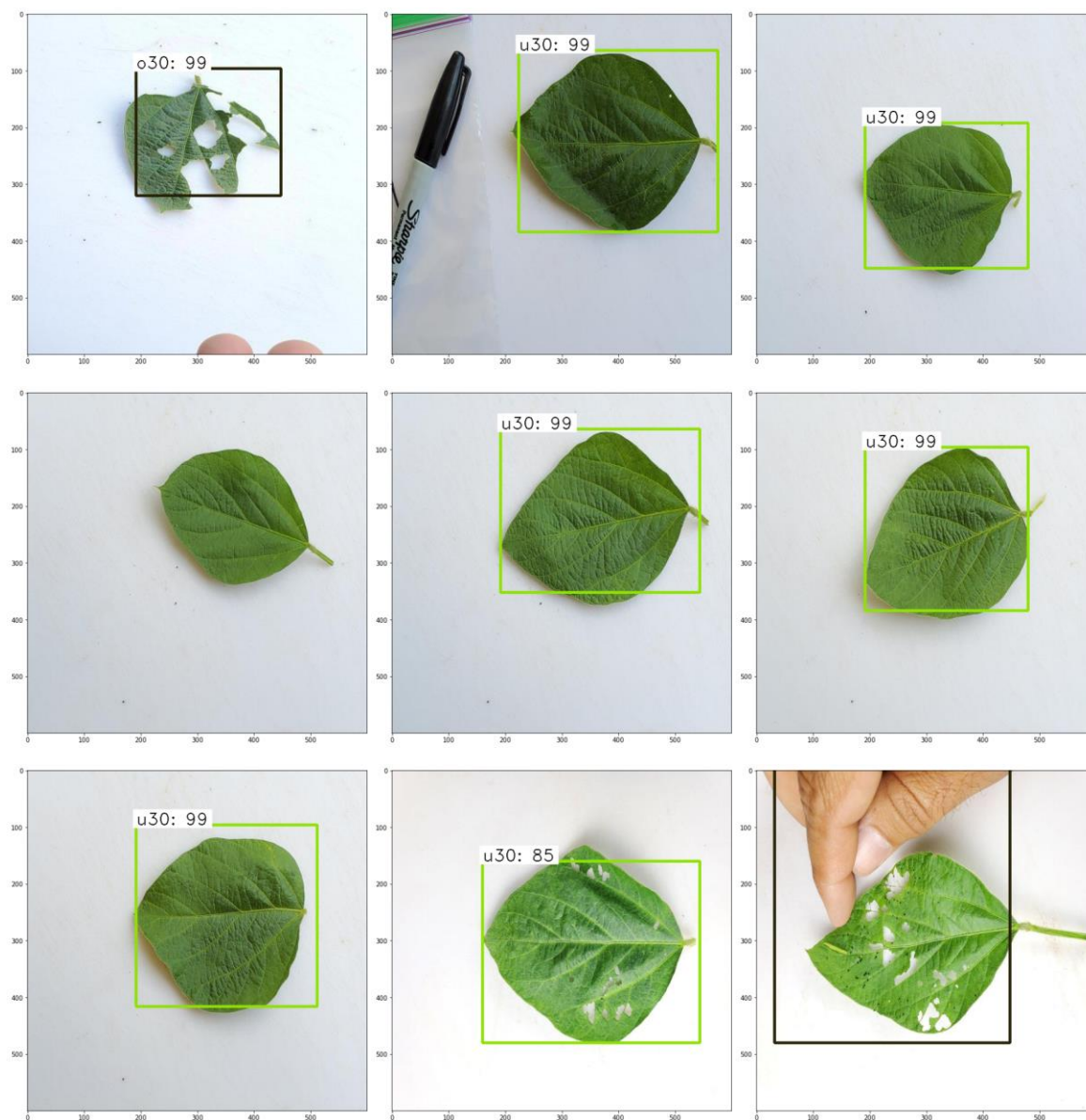
Study One – Leaflet Dataset

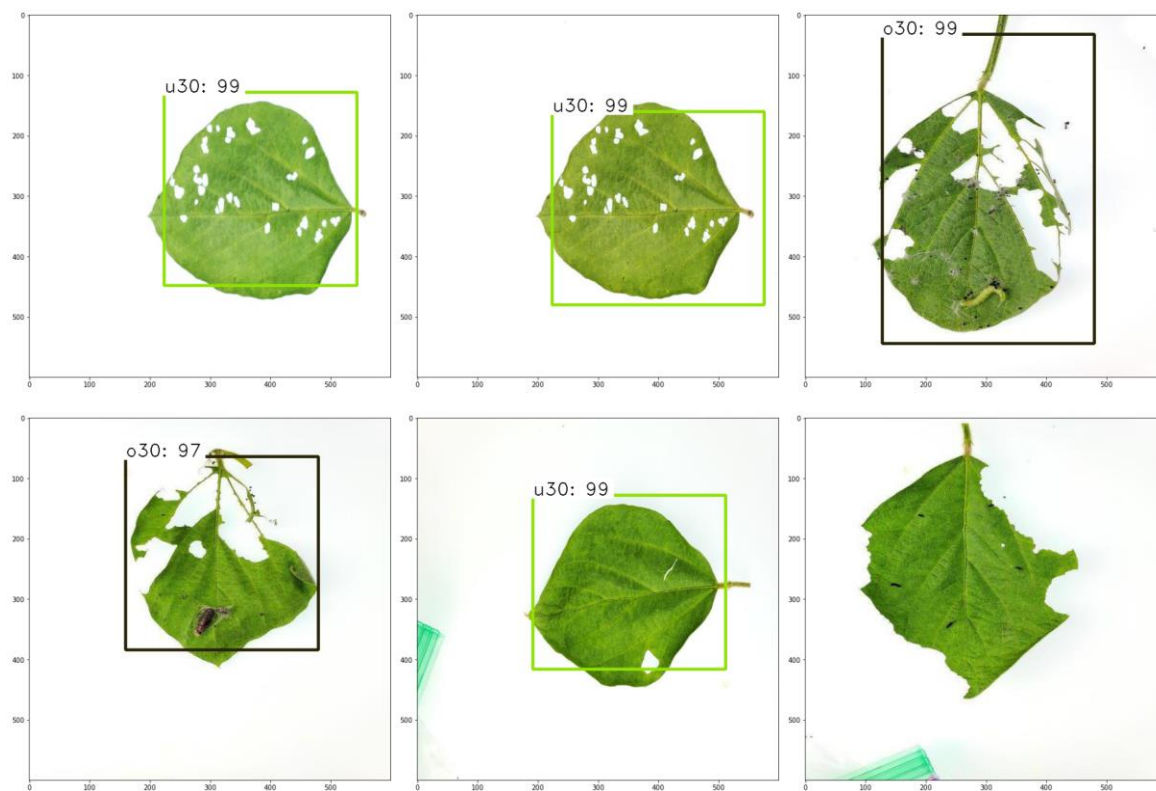










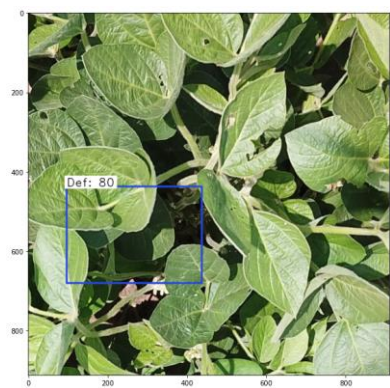
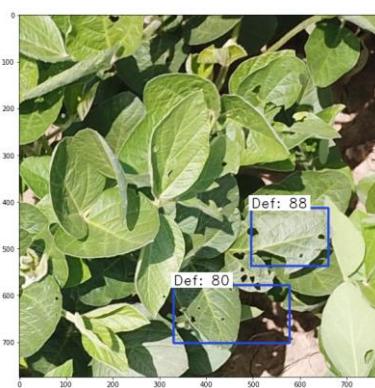
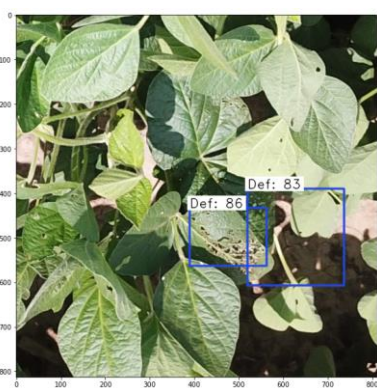
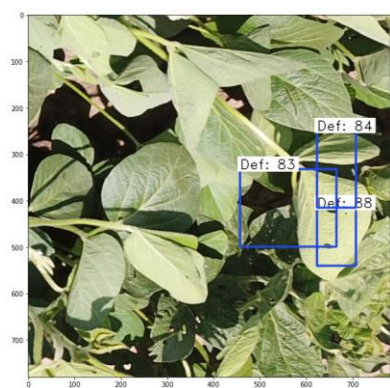


Study Two – Canopy Dataset









Appendix B - Codes

Image Resize and Rotate Code

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Wed Feb 24 15:57:00 2021
```

```
@author: Hayden
```

```
"""
```

```
import os, PIL
```

```
from PIL import Image
```

```
PATH = "F:\\GradSchoolThesis\\Leaflet_Data\\Over_30\\Over30_4\\"
```

```
Copy_to_path="F:\\GradSchoolThesis\\Leaflet_Data\\Over_30\\Over30_5\\"
```

```
for filename in os.listdir(PATH):
```

```
    img = Image.open(os.path.join(PATH, filename)) # images are color images
```

```
    img = img.resize((600,600), Image.ANTIALIAS)
```

```
    img = img.transpose(Image.ROTATE_90)
```

```
    #img = img.transpose(PIL.Image.FLIP_LEFT_RIGHT)
```

```
    img.save(Copy_to_path+filename)
```

Train and test code originally obtained here:

https://github.com/RockyXu66/Faster_RCNN_for_Open_Images_Dataset_Keras