

AUTOMATED WEED DETECTION USING MACHINE LEARNING
TECHNIQUES ON UAS-ACQUIRED IMAGERY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Aaron J. Etienne

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Dharmendra Saraswat, Chair

Department of Agricultural and Biological Engineering

Dr. Dennis Buckmaster

Department of Agricultural and Biological Engineering

Dr. Bryan Young

Department of Botany and Plant Pathology

Approved by:

Dr. Nathan Mosier

Head of the Departmental Graduate Program

To my parents for believing in me and pushing me to get through tough times.

ACKNOWLEDGMENTS

I'd like to acknowledge Dr. Saraswat for his support and expertise. I'd like to acknowledge Ben Hancock for always making time to help me with tough, technical questions in a kind and calm manner. Your help saved me from countless hours of frustration. I would like to acknowledge my colleagues in ABE. They have helped me through tough times and given me help and advice when I needed it most. I would like to acknowledge Jaspar Saadi-Klein for his help in installing UAS sensors and helping during data collection. I would like to acknowledge Dr. Bryan Young and his graduate students for providing their expertise in weed science and allowing me to conduct research over their trial plots. I would like to acknowledge Aanis Ahmad and Hardi Sura for their help in data collection and processing. I would also like to acknowledge Jeff Boyer, Jason Adams, Jim Beaty, and Gary Tragesser for their help with experimental setup and allowing me to conduct research on their farms. Lastly, I would like to acknowledge Purdue University for awarding me with the Ross Fellowship and the Indiana Corn Marketing Council for awarding me the Gary Lamie Research Assistantship.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
SYMBOLS	xiii
ABBREVIATIONS	xiv
GLOSSARY	xvi
ABSTRACT	xvii
1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	6
1.2.1 Overall Goal	7
1.2.2 Research Objectives	7
2 THEORETICAL BACKGROUND AND REVIEW OF LITERATURE	8
2.1 Unmanned Aerial Systems	8
2.2 Background on Machine Learning Methods	11
2.2.1 Machine Learning Methods for Weed Classification and Detection	19
2.3 Ground-Based Weed Detection and Application Systems	21
2.4 Large Scale Map Creation	27
2.5 UAS-Based Spraying Systems	27
2.6 Factors influencing weed growth and development	28
2.6.1 Weeds Common to the Midwest and Pertinent Study Area	29
3 METHODOLOGY	33
3.1 Methods used in dataset creation	33
3.1.1 Choosing Research Area for Data Acquisition	34
3.1.2 Data Collection Mission Setup	36

	Page
3.1.3 Analysis of UAS acquired imagery	43
3.2 Labelling and Annotation of Weeds Dataset	45
3.2.1 Methods for Choosing Imagery for Manual Annotation	45
3.2.2 Hardware and software setup used for deep learning network training	50
3.3 Object Detection Network Training Methods	50
3.3.1 Initial Network Training using NVIDIA DIGITS	50
3.3.2 Initial training methods with YOLOv3 on OpenImages Dataset	53
3.3.3 Methods for multi-class weed detection with YOLOv3	56
3.3.4 Methods for Two-class UAS-based Image Set Training	58
3.3.5 Methods for 2019 Data Collection and Pre-plant Weed Image Set Addition for Training with YOLOv3	60
3.3.6 Methods for 2019 Early Season Corn Image Set Creation and Training with YOLOv3	61
3.3.7 Additions to 2019 Early Season Corn Image Set and Configu- ration Changes for Training with YOLOv3	64
4 RESULTS AND DISCUSSION	67
4.1 Results of Data Collection	67
4.2 Manually Annotated Dataset Results	68
4.3 Object Detection Network Training Results	76
4.3.1 Initial Results with NVIDIA DIGITS	76
4.3.2 Initial YOLOv3 Results with OpenImages Dataset	77
4.3.3 Results of Multi-class Detection with YOLOv3	78
4.3.4 Results of Two-class UAS-based Image Set Network Training on with YOLOv3	81
4.3.5 2019 Pre-plant Weed Image Addition Network Training Results with YOLOv3	82
4.3.6 Early Season Corn Image Set Training Results with YOLOv3	83
4.3.7 Training Results for 2019 Early Season Corn Image Set Addi- tions with YOLOv3	86

	Page
4.4 Discussion	92
4.4.1 Discussion on Data Collection	92
4.4.2 Discussion on Annotated Training set Creation	94
4.4.3 Discussion on Weed Detection Network Training Results	96
5 SUMMARY AND RECOMMENDATIONS	104
5.1 Summary	104
5.1.1 Image acquisition	104
5.1.2 Annotated Dataset Creation	105
5.1.3 Weed Detection Network Training	105
5.2 Recommendations and Future Work	106
REFERENCES	108
APPENDIX	118

LIST OF TABLES

Table	Page
3.1 Specifications of the RGB, Multispectral, and Thermal Sensors used for this Study.	33
4.1 Total Flights and Flight Specifications for 2018 and 2019 Growing Season Data Acquisition.	67
A.1 DJI Matrice 600 Specifications.	118
A.2 DJI Mavic Pro Specifications.	119
A.3 DJI Mavic Pro Camera Specifications.	119
A.4 Flir Duo Pro R Thermal and RGB Sensor Specifications.	120
A.5 Slantrange 3PX Multispectral Sensor Specifications.	121
A.6 Gilbreth Cluster Specifications (Information Technology at Purdue University, 2019).	122

LIST OF FIGURES

Figure	Page
2.1 Comparing Spatial Resolution and Height of Image Capture in a Satellite Derived Image. (Jensen, 2005)	10
2.2 Left: Plot of Average Intersection over Union (IoU) between Anchors and Ground Truth Boxes, with Varying Number of Clusters (Anchors). Right: Comparing Anchor Shapes of COCO and VOC 2007 Image Set Trainings. (Hui, 2019).	12
2.3 The Workflow of Feature Learning and Classification in a Network, from an Input Image (Saha, 2018).	12
2.4 An Example of a Traditional Convolutional Neural Network Architecture (Saha, 2018).	13
2.5 A Regular 3-layer Neural Network vs. a ConVent Architecture (Barker and Prasanna, 2016).	14
2.6 An Example Faster-Regional Convolutional Neural Network Architecture as Developed by Ren et al. (2017).	15
2.7 An Example Single Shot Detector Architecture as Developed by Zhao et al. (2018).	16
2.8 A Feature Pyramid Network’s Bottom-up Pathway is Laterally Connected to the Top-down Pathway through “Residual Connections” in the YOLO Architecture (Lin et al., 2016).	18
2.9 Xu et al. (2018)’s Smart Sprayer System Configuration.	22
2.10 Partel et al. (2019)’s Experimental YOLO Network Architecture, Visualizing Input Layer and Feature Mapping Parameters.	24
2.11 Partel et al. (2019)’s UTV-based Intelligent Herbicide Spraying System Setup.	25
2.12 Gonzalez-de-Soto et al. (2016)’s Unmanned, Herbicide-applcator Robot, Based off of New-Holland Tractor.	26
2.13 Setting up Yamaha R-MAX for Custom Herbicide Application in Vinyard, Sonoma Valley, CA (Yamaha Motors USA, 2016).	28
2.14 Identifying Features of Monocot Weeds (Welch-Keesey, 2011).	30

Figure	Page
2.15 Identifying Features of Dicot Weeds (PSU Turfgrass, 2017).	31
3.1 Visual Comparisons of the Multispectral and Thermal/RGB Sensors used in This Research.	34
3.2 Research Plot Locations 1:PPAC 2:DPAC 3: TPAC.	35
3.3 PPAC Trial Plot.	35
3.4 DPAC Trial Plot.	35
3.5 TPAC Trial Plot.	35
3.6 Ground Control Point Boundary Map Created from RTK Point Collection.	37
3.7 TopCon HiPer V™ RTK Base Station and Receiver and GCP Measurement.	37
3.8 2018 Data Acquisition Dates.	38
3.9 UAS-based Image of the Spectral Targets used During Data Collection.	39
3.10 2019 Data Acquisition Dates.	40
3.11 2019 TPAC Corn Plot Location.	40
3.12 2019 ACRE Pre-plant and Corn Plot Locations.	41
3.13 Geographic Location of Agronomy Center for Research and Education (ACRE) .	42
3.14 RGB Orthomosaic Map of TPAC Plot, created June 18 th , 2018 of V4 Growth Stage Soybeans.	44
3.15 Thermal Orthomosaic Map of TPAC Plot, created June 18 th , 2018 of V4 Growth Stage Soybeans.	44
3.16 NDVI Orthomosaic of TPAC Plot, created June 18 th , 2018 of V4 Growth Stage Soybeans.	44
3.17 Flowchart of Training Image Set Creation for Object Detection Network Training.	47
3.18 NVIDIA DIGITS Graphical User Interface (GUI) Dataset Creation Options (Barker and Prasanna, 2016).	51
3.19 NVIDIA DIGITS GUI Object Detection Network Model Creation Options (Barker and Prasanna, 2016).	52
3.20 Bash Command for Parsing Plant Subset from OpenImagesv4 Database (Vittorio, 2018) .	55

Figure	Page
3.21 Bash Command for Training YOLOv3 Network on Labelled Plant Subset from OpenImagesv4.	56
3.22 Bash Command for YOLOv3 Anchor Calculation, used in Network Configuration File (Monhatkin, 2019).	60
3.23 Anchor Calculation Result of Labelled 2019 Early Season Corn Dataset. . .	62
3.24 Predicted Bounding Box Calculation, based on Annotated Ground Truth Box (Redmon and Farhadi, 2016).	63
4.1 Manual Labelling Process for DetectNet Training Set Creation (Training Image Set 1).	69
4.2 Manual Labelling Process for OpenImages Plant Training Set Creation (Training Image Set 2).	70
4.3 Manual Labelling Process for Multiple Weed Classes Training Set Creation (Training Image Set 3).	71
4.4 Manual Labelling Process for 2018 Two-Class Early Season Image Set Creation (Training Image Set 4).	72
4.5 Manual Labelling Process for 2019 Pre-Plant Image Addition (Training Image Set 5).	73
4.6 Manual Labelling Process for 2019 Two-class Early Season Image Set Creation (Training Image Set 6).	74
4.7 Manual Labelling Process for 2019 Early Season Image Set Addition (Training Image Set 7).	75
4.8 An Example of the Weed-free Negative Samples Added to Training Image Set 7.	76
4.9 Results of DetectNet training with Training Image Set 1 Over 300 Epochs. . .	77
4.10 Detection Results of Training Image Set 2 on Testing set, with Final YOLOv3 Training Weights.	78
4.11 Giant Ragweed Class Detection Results from YOLOv3 Network Training on Training Image Set 3.	79
4.12 Foxtail Class Detection Results from YOLOv3 Network Training on Training Image Set 3.	80
4.13 Cocklebur Class Detection Results from YOLOv3 Network Training on Training Image Set 3.	81

Figure	Page
4.14 Monocot and Dicot Detection Results from YOLOv3 Network Training on Training Image Set 4.	82
4.15 Average mAP and Loss Score Results from YOLOv3 Network Training on Training Image Set 5.	83
4.16 Average mAP and Loss Score Results from YOLOv3 Network Training on Training Image Set 6.	85
4.17 Dicot Detection Test Set Result from YOLOv3 Training on Training Image Set 6.	86
4.18 Average mAP and Loss Score Results from YOLOv3 Training on Training Image Set 7.	88
4.19 Average mAP and Loss Score Results- 9,000-17,000 Iterations, from YOLOv3 Training on Training Image Set 7 Using the Gilbreth Cluster.	90
4.20 Average mAP and Loss Score Results- 17,000-20,000 Iterations, from YOLOv3 Network Training on Training Image Set 7, using the Gilbreth Cluster.	91
4.21 Dicot Detection Result from YOLOv3 Network Training on Training Image Set 7, using the Gilbreth Cluster.	92
4.22 Example of heavy weed infestation at TPAC field site, July 6 th , 2018.	95
4.23 Weed Class Detection on Test Set Image as Result of DetectNet Training on Training Image Set 1.	97
4.24 Plant Class Detection on Test Set Image as Result of YOLOv3 Network Training on Training Image Set 2.	98
4.25 Problem Detecting Multiple Weed Class Instances in Test Set Image after YOLOv3 Network Training on Training Image Set 2.	99

SYMBOLS

m/s meters per second

ABBREVIATIONS

ESWM	Early Season Weed Management
CTWR	Critical Time for Weed Removal
SSWM	Site Specific Weed Management
ESSWM	Early Season Site-Specific Weed Management
PA	Precision Agriculture
GPA	Gallons Per Acre
GMO	Genetically Modified Organism
VRA	Variable Rate Application
ML	Machine Learning
DCNN	Deep Convolutional Neural Network
UTV	Utility Terrain Vehicle
NDVI	Normalized Differential Vegetation Index
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System
DSP	Digital Signal Processor
IPSO	Improved Particle Swarm Optimum
GPU	Graphics Processing Unit
CPU	Computer Processing Unit
FPS	Frames Per Second
USB	Universal Serial Bus
CNN	Convolutional Neural Network
FCN	Fully Convolutional Neural Network
FPN	Feature Pyramid Network
SSD	Single Shot Detector

FPS	Frames Per Second
YOLO	You Only Look Once
DIGITS	Deep Learning GPU Training System
IoU	Intersection over Union
TP	True Positive
FP	False Positive
FN	False Negative

GLOSSARY

Dicotyledon	flowering plants which have two cotyledons or embryonic leaves
Monocotyledon	flowering plants which have one cotyledons or embryonic leaf

ABSTRACT

M.Sc, Purdue University, August 2019. Automated Weed Detection Using Machine Learning Techniques on UAS-Acquired Imagery. Major Professor: Dharmendra Saraswat.

Current methods of broadcast herbicide application cause a negative environmental and economic impact. Computer vision methods, specifically those related to object detection, have been reported to aid in site-specific weed management procedures to target apply herbicide on per-weed basis within a field. However, a major challenge to developing a weed detection system is the requirement for properly annotated training data to differentiate between weeds and crops under field conditions. This research involved creating an annotated database of weeds by using UAS-acquired imagery from corn and soybean research plots located in North-central Indiana. A total of 27,828 RGB; 108,398 multispectral; and 23,628 thermal images, were acquired using FLIR Duo Pro R sensor that was attached to a DJI Matrice 600 Pro UAS. An annotated database of 306 RGB images, organized into dicot (broadleaf) and monocot (grass like) weed classes, was used for network training. Two Deep Learning networks namely, DetectNet and You Only Look Once version 3 (YOLO ver3) were subjected to eight training stages using seven annotated image sets. The precision for weed detection ranged between 4.22-45.13% for dicot and 3.63-65.37% for monocot weed detection. This research has demonstrated a need for creating a large annotated weed database for improving precision of deep learning algorithms through better training of the network.

1. INTRODUCTION

1.1 Background

Precision agriculture (PA) practices have allowed farmers to reduce both cost and amount of seed used, cultivation applied, labor needed, and machinery hours spent (Leonard, 2015). Currently, there is very little precision in broadcast herbicide application. Applying herbicide in bulk, over the entire field, regardless if there is a weed present or not, has a detrimental financial and environmental impact (Lingenfelter and Hartwig, 2007). The ability to not only detect but control the growth of weeds in the early stages of plant development is crucial to a healthy and profitable harvest. Having an effective management strategy in the early season helps prevent weed infestation from spreading to other areas of the field.

Weed infestation occurs throughout the growing season in crops. Weeds compete with crops for nutrients, moisture, sunlight, as well as physically attacking them by wrapping around their stalk or leaf blades (Birch et al., 1989). These issues attribute to yield loss at harvest. In order to control weeds before they have the chance to produce seed and spread further infestation throughout the field, weeds should be removed in soybeans between the V1 to V3 growth stage. In corn, weeds should be eradicated by the V4 growth stage (Gower et al., 2003). This is referred to as an early season weed management (ESWM) strategy. Weeds are controlled before they reach a height of 6 to 8 inches in soybeans and 4 to 5 inches in corn. This is referred to as the “critical time for weed removal” (CTWR) (Knezevic et al., 2003). After the V4 growth stage in corn and V3 growth stage in soybeans, studies have shown that insignificant to no loss of yield happened, due to new crop emergence, at these stages (Gower et al., 2003; Carey and Kells, 1995; Knake and Slife, 1962). After these growth stages, the crop has a competitive advantage on any new emerging weeds due

to size, canopy cover, and the amount of time left before crop maturity. Even with advances in weed management strategies and herbicide application practices, weed infestation continues to account for over 10 percent of yield loss in Midwest fields ([Hartzler, 2009](#)). Globally, the amount of yield loss is reported to be over 40 percent ([Oerke, 2006](#)). With 14.5 billion bushels of corn and 4.5 billion bushels of soybean produced annually in the United States, a 10 percent yield loss means 1.9 billion less bushels for traditional row-crops alone ([Sterk, 2018](#)). The current market price for corn and soybeans (June 18th, 2019) are \$4.49/bushel and \$9.12/bushel respectively ([Chicago Mercantile Exchange Group, 2019](#)). In dollar terms, the average loss per year is \$10.51 billion on just corn and soybeans, in the United States alone. Reducing harvest loss by just two percent in U.S.-grown corn and soybean fields would save farmers an average of \$2.1 billion annually. While there have been several advances in weed control over the last hundred years, there is obvious economic incentive for further research and development. Advances that have been made to current day and will continue to evolve include technological, mechanical, chemical, and biological ([Abbas et al., 2018](#)).

Among the most substantial technological, as well as mechanical, improvement in weed control was the self-propelled sprayer. The first self-propelled sprayer was invented by Ray Hagie in 1947 ([Scott, 2018](#)). This invention allowed for a substantial increase in the land area that herbicide could now be applied. The original self-propelled sprayer and the improvements made to it, up to current day, has revolutionized the way farmers and applicators apply not just herbicides, but all liquid agricultural chemicals in general ([Scott, 2018](#)). With the evolution of mechanical, chemical, and technological innovations in weed control, it was possible to eliminate weeds after the crop had emerged. Improved sprayer technologies, along with the invention of post-emergence herbicides have allowed for application well beyond when the crop had started to sprout.

Another technological advancement that enabled precision agriculture (PA) practices, was the introduction of GPS guidance for tractors and combines in the early

1990s ([Whelan and Taylor, 2013](#)). This guidance system, although coarse in accuracy by today's standards, gave accurate enough latitude and longitude coordinates of the implement's location to allow farmers to see a map of their fields and store data about planting, fertilizing, pesticide application, and harvest for each field. From this data, farmers could now create maps corresponding with this information in-field. Analysis of these maps allowed detection of problem areas in the field. For instance, an area that yielded 50 bushels less of corn than the field average could indicate disease or poor stand due to weed pressure. A farmer could reference that area in the herbicide application map and realize there was a problem with the sprayer at that time, or that herbicide-weed contact was sub-par. Areas within the field could also be broken down into management zones, based on the amount of material that needs to be applied to a given area, to produce the best yield. This is also referred to as a prescription map. Many prescription maps utilized Geographic Information Systems (GIS) to interpolate attribute data, such as yield, seeding population rate, and gallons per acre (GPA) applied, into a spatially variable map capable of inference by a farmer. This was done by fusing geo-referenced (GPS) data with attribute data and performing a spatial query, or retrieving spatial information from a geodatabase system ([Strickland et al., 1999](#)).

With this prescription map, farmers had the need for a technology capable of automatically varying input material in seeding, fertilization, and pesticide application tasks. This technology was called variable-rate application (VRA). More specifically, this was map-based VRA ([Strickland et al., 1999](#)). Map-based VRA interprets the prescription map loaded on the tractor's processor, via a dedicated software, converts that information to a form understood by the implement's controller (an attached planter or combine harvester, or the boom on a self-propelled sprayer) and sent as a command to be followed by the variable-rate drive system on the implement (what allows the implement to physically vary the rate applied to the field surface) ([Miles et al., 2002](#)). Later on, a sensor-based VRA method was developed that no longer needed prescription maps to vary application rate. As the name implies, this method applies

sensors to measure properties pertinent to the application task, such as soil properties and specific crop attributes (Miles et al., 2002). Sensor types can be proximal (measuring property without physically touching the object) and intrusive (gaining measurement from physically sampling the object). The workflow of a sensor-based VRA system first takes reading from the attached sensor, sends that information to the on-board processor, converts the measurements into a variable that the controller can understand, outputs the controller command into the implement’s variable-rate drive system, and causes the implement to physically vary output (Miles et al., 2002). Both sensor and map-type VRA systems have been used in herbicide application. Trimble’s WeedSeeker® is a sensor-based VRA system that measures reflected light to differentiate between weeds and bare soil (Miles et al., 2002). However, since this product cannot distinguish between crops and weeds, it can only be used inter-row or when no crops have been planted. Another sensor-based VRA system for target herbicide application is Blue River’s See and Spray™ system. Unlike WeedSeeker®, See and Spray™ is capable of distinguishing between weed and crop intra-row and can accurately spray the weed without damaging the crop (Golden, 2017). Advancements in proximal-sensors and processing power make it plausible to create an open-source version of these products, that incorporates a sensor-based VRA system.

In addition to technological advances, genetic advances in weed control include glyphosate tolerant soybean and corn seed, which were the first genetically modified organisms (GMOs) to incorporate herbicide resistance. They were called Roundup Ready® and developed by Monsanto in the ten years preceding their commercial launches in 1996 and 1998 respectively. These products allowed for post-emergence application of glyphosate, a non-selective herbicide only used for pre-emergence, to be applied post-emergence in both corn and soybeans (Dill, 2005). As a result, Roundup® became much more widely applied within the United States. Since the inception of the genetically modified crops, glyphosate use has increased fifteen times, with 1.6 billion kilograms applied within the United States in just the last ten decade (Benbrook, 2016). Over-application of glyphosate can cause detrimental

side-effects in contamination to groundwater from agricultural run-off, and can be toxic for beneficial organisms like earth worms, butterflies, bees, fish, and microbes found within the soil (Myers et al., 2016). Another GMO seed advancement was Monsanto’s dicamba and glyphosate resistant soybean variety, termed Roundup Ready 2 Xtend®. This soybean variety was commercially introduced in 2016 and allowed for control of marestail and waterhemp weeds resistant to other types of herbicides. It also allowed for better control of broadleaf weeds in soybeans.

One method that aims at reducing herbicide application rate per acre, while precisely targeting weeds is site specific weed management (SSWM). SSWM blends the precision agriculture approach of utilizing technology to observe, measure, and respond to intra-field variability (Leonard, 2015) similar to the 4R nutrient management approach.

Similar to how PA focuses on in-field variability, the 4Rs nutrient stewardship guidelines aim to apply a precision management approach to fertilizer application. The four Rs are right source, right rate, right time, and right place (Liu et al., 2015). Much like nutrient stewardship, the 4R method can also be applied to herbicide application. Having the right source of chemical, applied at the right rate for effective coverage and uptake, only where a weed is present at the right place in the field, and at the right time during the season and weather condition. Two of these areas that need immediate improvement are right rate and right place. As mechanical, technological, and managerial strategies have evolved to how agriculture is understood today, so must herbicide application and management practices evolve. It is possible to target apply herbicide to an individual weed, through the use of object detection. Object detection is a computer vision method capable of classifying an object within a given image, as well as locate all instances of that object. As the following sections will detail, the ability to train an object detector to detect weeds at an individual level, in a complex environment such as a row-crop field is quite challenging.

1.2 Problem Statement

Weed detection is a challenging task within tilled and no tilled row-crop fields. These fields present a complex and challenging environment for computer vision tasks in several ways. Weeds and crops have similar spectral characteristics (similar color), they also share physical similarities early in the growing season, soil conditions can vary heavily within a small area, and stalks and debris in no-till or minimal till fields can cause problems with false detection. In addition, differing weather conditions can affect how well a weed detection system can discriminate weeds from non-weeds. This includes shadow and glare caused by sunlight, image blurriness caused by clouds and/or sensor settings, snow or rain obscuring objects within the image, and vibration in the camera system caused by wind. Computational time required by a computer vision system to detect a weed instance will affect the speed at which a weed remediation operation can occur. Application of this remediation, be it chemical destruction, physical destruction, or a system that incorporates both, will also determine the structure of computer vision system used. One Specific method of weed identification might be more suited to a scenario than another. For example, in an area of heavy weed infestation, where the speed of an operation can be slower, a weed identification method that is more computationally intense but highly accurate may be preferred to a method that is computationally light, allowing for faster operation but sacrifices accuracy.

Autonomous detection of these noxious weeds requires the implementation of computer vision methods, primarily image classification and object detection. Object detection is preferred in this instance, as it both locates the desired object within the image and classifies what it is (Li, 2018). However, object detection requires extensive labelling and annotation of images to allow the network being trained to learn the objects and their corresponding locations. A large, variable image set is required to be manually labelled in this fashion.

A wide variety of remote sensing devices are commercially available with varying cost, spatial and spectral qualities, and application basis. Specific application, desired sensor type for the study, availability, and desired resolution comparative to budget were all factors influencing sensor selection for this study.

1.2.1 Overall Goal

The overall goal of this research is to gauge network training performance of deep convolutional neural network (DCNN) models for detection of dicot (broadleaf) and monocot (grass type) weeds in corn and soybean fields.

1.2.2 Research Objectives

1. To compile a database of UAS acquired images using RGB, thermal, and multi-spectral sensors for providing critical training data for machine learning algorithms.
2. To classify an RGB image dataset of weeds obtained from multi-location corn and soybean test plots into monocot and dicot classes and manually-annotate for training of object detection models.
3. To evaluate the performance of DetectNet and YOLOv3 object detection models for identifying monocot and dicot weeds in corn and soybean fields.

2. THEORETICAL BACKGROUND AND REVIEW OF LITERATURE

Definitions used in remote sensing that are pertinent to this thesis research, advantages of Unmanned Aerial Systems over satellite and manned aircraft, and autonomous applications of Site-Specific Weed Management will be discussed in this chapter. For various machine learning techniques, theoretical framework, model structure and components, and applications for weed identification will also be discussed.

2.1 Unmanned Aerial Systems

An unmanned aerial system (UAS) offers the advantage of autonomous flight and data collection (Milioto et al., 2017). Its ability to fly at a low altitude, which allows for very-high resolution imagery, makes it advantageous over satellite-based options. Flying closer to the target area allows for better spatial resolution, or being able to utilize more pixels in the construction of the digital image (Milioto et al., 2017). An unmanned aerial system also provide opportunities for high temporal resolution, the ability to repeat a given measurement over a series of time, required to track physical and biological changes within the field over a time period (Manfreda et al., 2018). Examples of such changes include tracking crop and weed growth, disease or insect infestation, and problem areas within a field.

High spatial resolution, ability for low altitude flight, and high definition, with live video and imagery make an unmanned aerial system a suitable vehicle for agricultural weed identification. The ability to detect weeds present in the field, in real-time, allows for accurate ground truth data. Ground truthing is the act of linking information from data to features and materials on the ground (Manfreda et al., 2018). Unmanned aerial systems provide several advantages over traditionally acquired satellite imagery.

Satellite imagery provide inadequate spatial and temporal resolutions necessary for highly accurate measurements over an extended period or entire growing season in certain remote sensing and precision agriculture applications ([Torres-Sánchez et al., 2014](#)). While using satellite imagery for later growth stage crops have obtained decent results, early season growth is dependent on a much more accurate spatial resolution ([Torres-Sánchez et al., 2014](#)).

UAS offer the highly accurate, down to sub centimeter, accuracy required to accurately map early season crop growth. Researchers have been using unmanned aerial systems for agricultural applications longer than two decades. During this time, UAS have gone from a tool for simple field scouting to providing insightful data and information on everything from vegetation health to stand count and biomass estimation ([Foreign Agricultural Service, 1994](#)) Rules and regulations governing the use of UAS and specific needs has defined the global adoption of this technology in agriculture.

Countries like Japan have been using unmanned helicopters for high volume spraying since 1995. However, the system described for spraying in Japan is not legal in the United States. Due to the weight restriction of an unmanned aerial vehicle (UAV), must be between 0.5 and 55.0 pounds fully loaded. It is only recently that UAS-based sprayers have been introduced in the American market ([Federal Aviation Administration, 2018](#)). UAS give an affordable alternative to airborne and satellite-based systems.

UAS offer an advantage to manned aircraft-based sensors from a cost and timing standpoint. Due to the relative ease of setup and quick deployment time of a UAS, they are able to get airborne and complete the mission much more quickly than a manned aircraft ([Watts et al., 2012](#)). This can be a crucial factor during Midwest summers, when it can go from sunny and ideal flying conditions to cloudy or storming within the same hour. In addition to timing manned flights around the weather, there is a broader problem of having to schedule their data collection in advance. Pilot availability and competency, demand for services, mechanical issues of manned

aircraft, and many other factors could determine commercial provider's availability to fly farmer's fields. The cost of UAS and accompanying sensors have consistently decreased, thereby becoming a viable option for ownership by individuals, agricultural cooperatives, and researchers (Watts et al., 2010). Another advantage of UAS is their capability of low altitude flight, as low as 2-3 feet off of the ground in certain use cases. This is a much lower altitude than manned aircraft are capable of flying. This allows for a much more accurate spatial resolution, the number of pixels used in the formation of a digital image (Davidson, 2016). The lower the altitude that an image is taken, the higher the spatial resolution of that image. However, less area will be covered within the image as the altitude at which it is taken at decreases. Users have to deal with a trade-off of spatial resolution to flying height at which an image is captured (Figure 2.1).

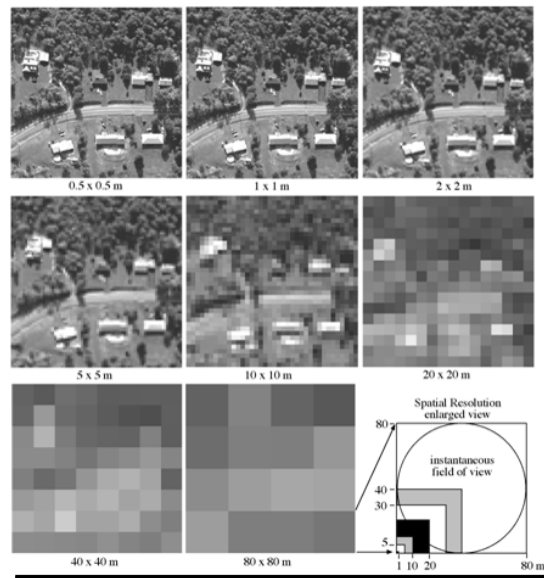


Figure 2.1. Comparing Spatial Resolution and Height of Image Capture in a Satellite Derived Image. (Jensen, 2005)

Looking at Site-Specific Weed Management (SSWM) from a remote-sensing perspective gives insight to the potential to decrease weed control costs through reductions in both manpower and use of herbicide (Hassan-Esfahani et al., 2017). This

method also aids in the development of site-specific herbicide application maps. Performing these tasks with small sensors mounted on unmanned aerial vehicles (UAVs) allows for weed management and herbicide application practices that are affordable, easy to implement, and environmentally friendly (Shi et al., 2016). In order to implement SSWM at field-scale, it is necessary to combine software and hardware to create a weed control system Shi et al. (2016). At the root of this system is a technique called machine learning (ML). According to Christopher bishop, one of the foremost experts on ML, machine learning “is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence” (Bishop, 2006). Rapid advances in this field over the last 20 years have made ML more approachable by non-computer scientists and has been implemented across several different fields, such as medicine, automotive, construction, and recently in agriculture.

2.2 Background on Machine Learning Methods

Machine learning and computer vision advances have made it possible to create and train a model capable of detecting specific instances of weed species within a field scene. A method that has benefited from these advances is object detection. Object detection is defined as the process of finding instances of real-world objects, like cars, people, buildings, etc. in images or videos (MathWorks, 2018). The main function of an object detection system is detecting all known occurrences of a specified category of object within the image (Barker and Prasanna, 2016). Object detection networks such as region-based, fully convolutional, and single shot detectors implement feature extraction by calculating the intersection over union (IoU) of the annotated vs. predicted bounding box to recognize instances of a certain object or category (shown in Figure 2.2).

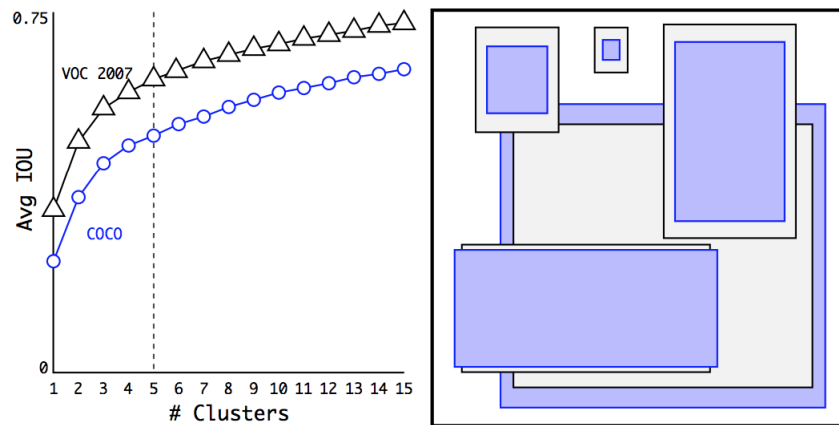


Figure 2.2. Left: Plot of Average Intersection over Union (IoU) between Anchors and Ground Truth Boxes, with Varying Number of Clusters (Anchors). Right: Comparing Anchor Shapes of COCO and VOC 2007 Image Set Trainings. (Hui, 2019).

Thus, it's clear that object detection requires supervised learning, in order to train and make predictions. A feature learning process that an object detection network undergoes to classify a located object according to its class label is shown in Figure 2.3.

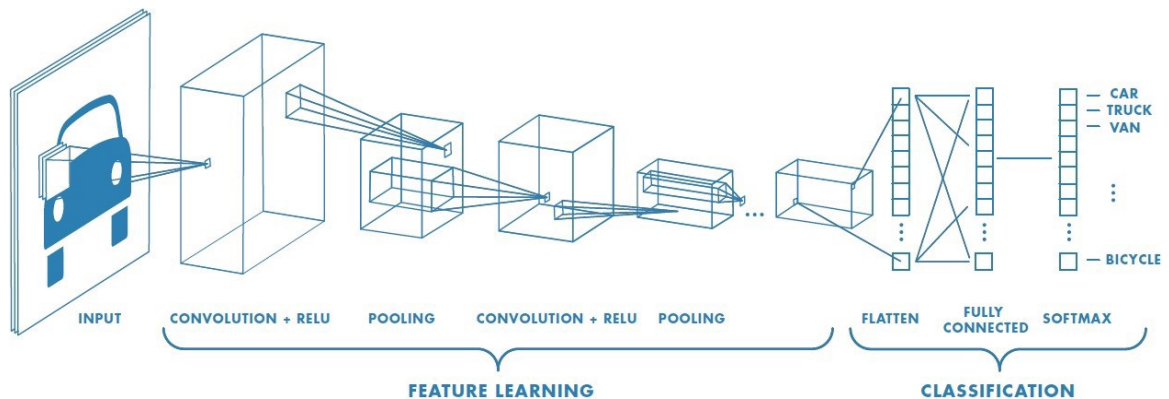


Figure 2.3. The Workflow of Feature Learning and Classification in a Network, from an Input Image (Saha, 2018).

The downside of object detection is that model training requires large, labelled image datasets. As of now, it means hand-labelling all instances of the desired object with that object's category, a bounding box that must be hand drawn around the corners of an object instance, and a way for the model's architecture to read and understand the bounding box points and categorical labels (Barker and Prasanna, 2016). Convolutional neural networks (CNNs) like the one shown in Figure 2.4 are most popular in object detection tasks.

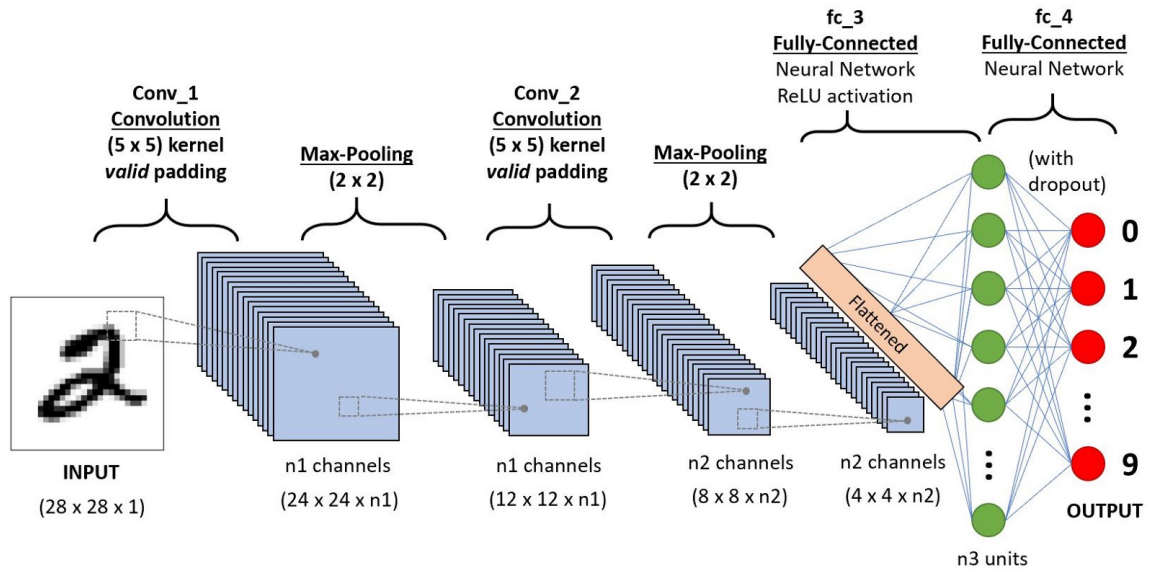


Figure 2.4. An Example of a Traditional Convolutional Neural Network Architecture (Saha, 2018).

A convolutional neural network is comprised of neurons which invoke “learnable weights and biases” (Li, 2018). Every neuron will receive some input and perform matrix multiplication. A single unique score function is expressed by the network, inputting raw image pixels and outputting class scores. What differentiates CNNs from ordinary neural networks is that their architectures, or the overall processing workflow of the network, makes an explicit inference that files inputted to the network are solely imagery. This inference makes it so the user can input the desired properties,

such as image size and file type into the network configuration (Li, 2018). The CNN architecture arranges neurons in 3 dimensions: width (how many layers wide the network is specified to be), height (how many layers tall the network is specified to be), and depth (how many layers deep the network is specified to be), shown in Figure 2.5.

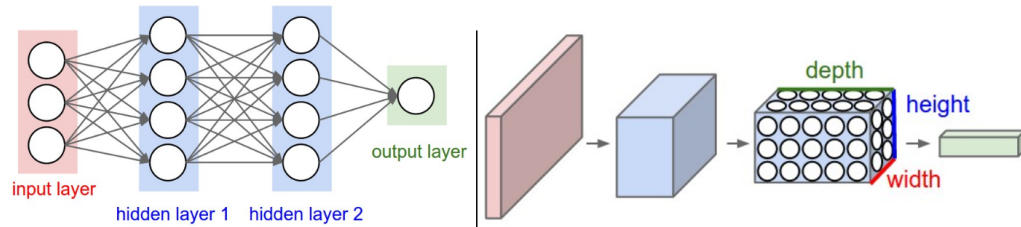


Figure 2.5. A Regular 3-layer Neural Network vs. a ConVent Architecture (Barker and Prasanna, 2016).

Another type of object detection is a region proposal-based network (RPN). In an RPN framework, an entire image is first scanned and then a particular region of interest is engaged. The scan is performed by a sliding window approach that inserts a CNN to predict bounding boxes straight from locations in the uppermost layer of the feature map after obtaining confidence levels of the underlying object categories (Zhao et al., 2018).

Ren et al. (2017) combined a deep fully convolutional network capable of proposing regions with a detection module that implements these proposed regions. They called the network Faster R-CNN, an innovative type of region-based proposal network. RPNs input an image and output sets of rectangular object proposals (Ren et al., 2017). The Faster R-CNN approach put a small network into a sliding window that runs over a feature pyramid layer (Ren et al., 2017). Using a concept of anchors, or maximum number of predicted region proposals per sliding window location, the authors denoted them as k . Anchors were defined as k reference boxes in which region proposals were relatively parameterized to. The anchors were centered to each sliding

window and had an associated scale and aspect ratio (Ren et al., 2017). Within this study, “three scales and three aspect ratios were used, giving 9 total anchors per position”. The same number of anchor boxes have been implemented in an object detection method named You Only Look Once (YOLO), that has been a methodology of interest for weed detection research (Redmon and Farhadi, 2018) A high level R-CNN workflow is shown in Figure 2.6

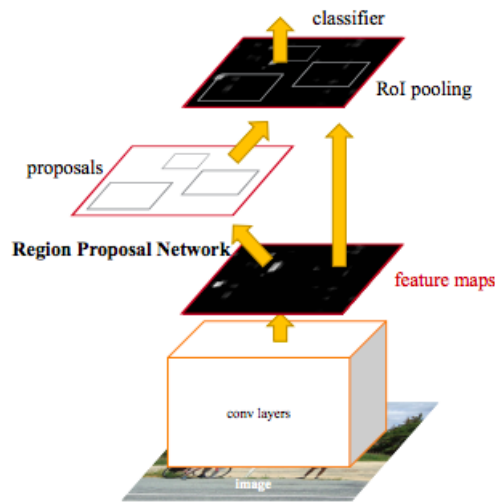


Figure 2.6. An Example Faster-Regional Convolutional Neural Network Architecture as Developed by Ren et al. (2017).

Unlike an RPN, the single shot detection method of object detection does not use a sliding window approach for bounding box prediction. A single shot multibox detector, also referred to as an SSD, is a relatively new method of object detection that is capable of real time performance. Single shot refers to object localization and classification tasks being done with a single forward pass of the network (Liu et al., 2015). Multibox is the name given to the bounding box technique developed by (Liu et al., 2015). The SSD architecture builds off of the VGG16 network architecture, a region-based or R-CNN. In this architecture, “auxiliary convolutional layers were added to VGG16 after the sixth convolutional layer”, to allow for feature extraction

at various scales and gradually decrease the size of the input to each successive layer (Forson, 2017). The architecture of an SSD is shown in Figure 2.7.

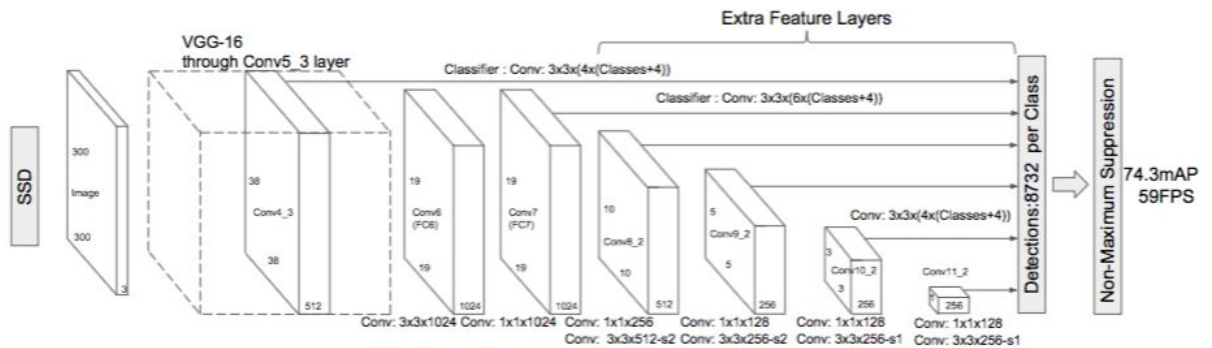


Figure 2.7. An Example Single Shot Detector Architecture as Developed by Zhao et al. (2018).

YOLO (You Only Look Once) is a unified object detection system capable of running in real time (Redmon and Farhadi, 2018). It utilizes a single CNN to predict multiple bounding boxes and their corresponding class probabilities (Redmon et al., 2015). Unlike RPNs or other sliding-window approaches, YOLO is capable of visualizing the entire image while training and testing and inherently encodes contextual class information and their corresponding appearances (Redmon et al., 2015). The YOLO network architecture is defined in a configuration or *.cfg* file. In order to understand how YOLO network training works, the following hyperparameters must be defined.

Batch size refers to the number of training samples, in this case images, used in each training iteration. An iteration is also referred to as an epoch in different network architectures. A subdivision is how many “mini-batches” the initial batch is split. For example, if the batch size is 64 and the subdivision size is 8, the initial batch size is divided by the subdivision ($64/8$) and 8 images per this smaller batch will be processed by the GPU until all 64 images have been processed. Then the next iteration will start.

While subdivisions serve the purpose of reducing GPU memory needed during processing, they also increase the amount of training time. It is recommended to reduce the amount of subdivisions used, if GPU memory capacity can handle a higher load of images ([Redmon and Farhadi, 2018](#)). Image detail information, height; width; and channels, are important to have correct before beginning training. Specifying image height and width will resize all images within the training set into those pixel values. This is beneficial as all images within the training set do not need to be resized beforehand.

Channel refers to the spectral wavelengths, or bands that the image pixels fall into. For example, a color image has three channels, red, green, and blue. The camera used to capture the image has a specified band range for red, for green, and for blue. Multispectral images are broken down into these individual bands, but color images combine the three.

Learning rate, also referred to as step size, controls the speed in which a neural network is able to learn a problem ([Brownlee, 2019](#)). Network weights are produced by this training, which have to be calculated using empirical optimization methods, like stochastic gradient decent, instead of traditional statistical methods such as confidence interval, variance, and clustering that generalize problem distributions.

Stochastic gradient decent “estimates the error gradient for the current state of the model” ([Brownlee, 2019](#)) through picking images from the training set at random. Model weights are updated by distributing errors backwards through the network’s layers, since error is computed at the output ([Deep AI, Incorporated, 2017](#)). This is referred to as backpropagation.

Feature scaling, also called data normalization, is “method used to standardize the range of independent variables or features of data” ([Nadahalli, 2014](#)). As most datasets contain a highly varying magnitude of features and most machine learning algorithms use Euclidean distance between two points in their computations, high magnitude features are given much more weight in distance calculations than low magnitude features. To mediate this problem, all features need to be brought to the

same level of magnitude. Feature scaling methods include standardization; shown in equation (2.2), mean normalization; shown in equation (2.2), mix-max scaling; shown in equation (2.3), and unit vector; shown in equation (2.4) (Asaithambi, 2017).

$$x' = \frac{x - \bar{x}}{\sigma} \quad (2.1)$$

$$x' = \frac{x - \text{mean}(x)}{\text{max}(x) - \text{min}(x)} \quad (2.2)$$

$$x' = \frac{x - \text{min}(x)}{\text{max}(x) - \text{min}(x)} \quad (2.3)$$

$$x' = \frac{x}{||x||} \quad (2.4)$$

The YOLO detection network utilizes a feature pyramid network (FPN), shown in Figure 2.8, to create multiple feature map layers at three different scales. FPN data flow is structured in a bottom-up and top-down pathway. The bottom-up pathway is the traditional method that CNNs use to perform feature extraction.

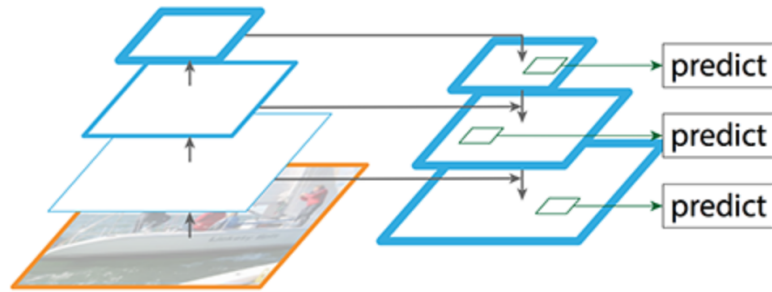


Figure 2.8. A Feature Pyramid Network’s Bottom-up Pathway is Laterally Connected to the Top-down Pathway through “Residual Connections” in the YOLO Architecture (Lin et al., 2016).

Spatial resolution decreases as the pyramid layers go from bottom to top. However, the semantic value of each ascending layer increases, as more high-level structures are detected (Hui, 2018). The top down-pathway constructs higher resolution layers, drawing from semantically weaker features. Due to the imprecise locations of objects within these reconstructed layers, lateral connections are added between these layers and their corresponding feature maps to allow the detector to better predict object locations. These connections are commonly referred to as skip or residual connections.

According to the creators of YOLO Redmon and Farhadi (2018), "the initial convolutional layers of the network perform this feature extraction, while trailing fully-connected layers output class probabilities and predicted bounding box coordinates".

Max_batches refers to the maximum number of iterations that the network will train on. Number of classes are the categories of objects within the image that has been annotated. During network training in YOLO, this is denoted by the *.names* file. This file lists the names of the classes to be trained on. Another often overlooked hyperparameter in the *.cfg* file is steps. The steps hyperparameter specifies the number of iterations at which scale will be applied. Both scale and steps values determine how the learning rate will be changed over increasing numbers of iterations during the training process (Mohnatkin, 2017).

The performance of each ML model discussed, as well as all others, is dependent upon the dataset on which it is learning from. For detection within row crop fields, the images in the dataset used to train a particular model must match the complex environmental conditions found in the field. While images taken under staged conditions in a lab produce highly accurate, theoretical detection (Lee et al., 2015), (Wang et al., 2016), they do not yield good results when tested on realistic field conditions

2.2.1 Machine Learning Methods for Weed Classification and Detection

Pérez-Ortiz et al. (2015) compared the accuracy of various classification methods at discriminating between crop, soil, and weeds in a sunflower crop. An image or-

thomosaic was created for 30, 60, and 100 meter UAS-collected images. Each were subdivided into 1000x1000 pixel subimages. Unsupervised k-means, semi-supervised linear-SVM (support vector machine) approximation, and supervised kernel SVM classification methods were evaluated on their mean average error (MAE) and tested on a 32-frame ground-truth set at each corresponding altitude (Pérez-Ortiz et al., 2015). At 30 meters, MAE was lowest with the semi supervised method, at 4.57%. At 60 meters, the supervised method had the lowest MAE, with 6.5%. At 100 meters, the supervised method also had the lowest MAE, with 26.5% (Pérez-Ortiz et al., 2015). The best MAE scoring approach, the semi-supervised method at 30 meters, was tested on binarized ground truth frames showing only weed presence. The semi-supervised classifier obtained an accuracy of 75% Pérez-Ortiz et al. (2015).

dos Santos Ferreira et al. (2017) created a software to classify weeds and distinguish between grass and broadleaf weeds in soybean field images. 400 UAS-based RGB images were segmented into single-object instances and manually annotated according to their class and resized to 256x356 pixels. Classes were defined as soil, soybean, grass weeds, and broadleaf weeds. A total of 15,336 segments were created, 3249 of soil, 7376 of soybean, 3520 of grass, and 1191 of broadleaf weeds (dos Santos Ferreira et al., 2017). A balanced image set, with 1125 images for each class (4500 total) was split into 3000 training, 500 validation, and 1000 test images. The unbalanced image set used 15,000 of total collected images, with 70% of the images used for training, 10% for validation, and 20% for testing. The test set was created to visually compare network weight results against ground truth imagery. Thus, it was a physical experiment that mimicked validation tests by the network. This ratio was also implemented within individual classes (dos Santos Ferreira et al., 2017). A modified CaffeNet model was trained over 7500 iterations for the balanced image set and 15,000 iterations for the unbalanced set. In the balanced image set, weighted average precision of the classification results for the four classes were as follows: soil 100%, soybean 99.5%, grass 99.7%, and broadleaf 98.3%. In the unbalanced image set, the following weighted average precision of classification results are given: soil

100%, soybean 99.5%, grass 99.7%, and broadleaf 98.3% (dos Santos Ferreira et al., 2017). However, the authors point out that these highly accurate classification results would be unlikely in a “real field application” as all research was done in a controlled environment (dos Santos Ferreira et al., 2017).

Yu et al. (2019) trained, tested and compared three deep convolutional neural networks (DCNNs) for detecting broadleaf weeds in bermudagrass. Training was done under multiple weed class and single weed class scenarios. Each original image was split into 9, 640x360 pixel images. For single-class dataset creation, the three training sets were set as follows: 6000 positive images (with weeds) and 6000 negative images (without weeds), 7632 positive and 8500 negative images, and 6206 positive and 7000 negative images. Each set had a total of 500 positive and 500 negative validation images. For multi-class training, a total of 18,000 positive and 18,000 negative images were taken, with 6000 positive and negative images for each class. The NVIDIA Deep Learning GPU Training System (DIGITS) was used for training the three ML algorithms, utilizing the Convolutional Architecture for Fast Feature Embedding (Caffe) (Yu et al., 2019). All network training was done over 30 epochs, with a learning rate of 0.1. Results of single-class network training showed VGGNet to have near perfect classification results, ranging from 98.2 to 100%. This was nearly twice as accurate as GoogleNet, whose most accurate result was 57%. VGGNet also performed best in multi-class network training, with a range in accuracy of 91 to 97% (Yu et al., 2019).

Deep convolutional neural networks are the foundation of autonomous smart-sprayer systems. Translating the trained detection network results to physical herbicide application is a complex process that will be explained in section 2.3.

2.3 Ground-Based Weed Detection and Application Systems

The end goal of a weed detection system is to apply it to a physical machine capable of eliminating weeds present in a field. There have been a handful of studies

demonstrating such a system, with varying methods of weed detection and elimination. Xu et al. (2018) developed an “automated weed mapping and variable-rate herbicide spraying system” (Xu et al., 2018) for implementation in row crop fields. A complex segmentation algorithm was performed on images input from a high resolution, grayscale camera. The study combined the OTSU binarization method, the maximum computed variance between clusters, with a particle swarm algorithm (PSO) that searched for the optimum value in complex space to determine maximum variance (Xu et al., 2018). This novel method was termed “Improved particle swarm optimization” or IPSO. This complex segmentation was used to produce a weed distribution map. Figure 2.9 shows Xu et al. (2018)’s smart sprayer system.



Figure 2.9. Xu et al. (2018)’s Smart Sprayer System Configuration.

A variable-rate herbicide control system was designed to “collect and process the weed information and determine herbicide application rates”. A digital signal processor (DSP) controller received decision signals from a Wi-Fi module and pulsed

the nozzle solenoids on and off as a weed was detected. The authors claim that their control system could “simultaneously monitor liquid level, pressure, and flow rate of the sprayer” [Xu et al. \(2018\)](#). A total of 30 images were collected from in-field experiments and segmented using OTSU, PSO, and IPSO respectively. The standard error rate of segmentation results from the OTSU, PSO, and IPSO methods, tested the 30 acquired images ranged from 7.1 to 0.1%. The standard error rate used by [Xu et al. \(2018\)](#) is shown in Equation 2.5 below.

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (T_i - TU_i)^2}{N}} = \sqrt{\frac{\sum_{i=1}^N E_i^2}{N}} \quad (2.5)$$

Within Equation 2.5, “n was the number of optimizations and N was the total number of images (N = 30). T_i is the threshold value of different algorithms, and TU_i is the threshold value of the OTSU method. E_i is the difference of T_i and TU_i ” ([Xu et al., 2018](#)).

[Partel et al. \(2019\)](#) developed and tested a ground-based detection and spraying system. The computational system on the vehicle included an NVIDIA™Jetson TX2 and an 8 GB NVIDIA™GTX 1070 Ti GPU. Images were acquired from three Logitech c920 webcams. The three cameras acquired images at 640 x 640 resolution that were merged to form a 1920x480 pixel image and then resized to a 1024x256 pixel image ([Partel et al., 2019](#)). The YOLOv3 object detection network was trained on 1000 manually labelled images of “targets and non-targets” for each “target-position” within the images and three experiments were performed. The first experiment was trained on the Tiny YOLOv3 weights, using images of artificial weeds as “targets” and artificial plants as “non-targets”. The second experiment featured the same network but training it on a *Portulaca oleracea* (common purslane) weed as a labelled target, and a pepper plant labelled as non-target. The same images collected for the second experiment were used to train a standard YOLOv3 network in experiment three. The standard YOLO network was capable of running at 24 fps on the GTX 1070 Ti GPU, but ran at a tenth of that speed, 2.4 fps, on the Jetson TX2. The Tiny YOLOv3 network was used on the TX2 to achieve a frame rate of 22 fps. While this tiny

network was faster than the standard, it sacrificed accuracy for speed. Experimental YOLOv3 architecture utilized is shown in Figure 2.10.

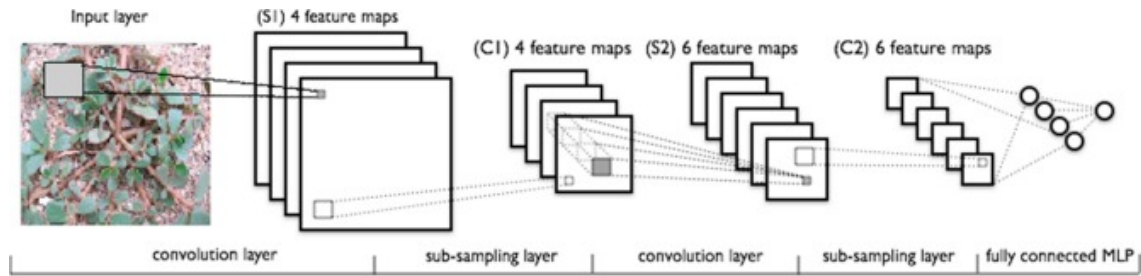


Figure 2.10. Partel et al. (2019)’s Experimental YOLO Network Architecture, Visualizing Input Layer and Feature Mapping Parameters.

The object detection system on Jetson TX2 communicated with the sprayer system through USB connection between the Jetson TX2 and an Arduino controller. Once a weed was detected, its bounding box’s center coordinates were utilized to calculate spray nozzle position and the height of the targeted weed. Once a detection was made, a 12-byte string of all “target-calculated values” on the frame was sent to be processed by the Arduino controller, triggering each nozzle individually (Partel et al., 2019). As a target got detected through the computer vision system, latitude and longitude values were recorded by the on-vehicle RTK GPS system. An angle was calculated from the activated sprayer’s orientation, based on the previously recorded GPS data and image coordinates to create an “absolute position for every target”. Each absolute target position was stored in a text file for future weed map creation. This setup resulted in spraying accuracy of 91% by utilizing the GTX 1070 Ti GPU. Partel et al. (2019) defined sprayer accuracy as the average of overall herbicide coverage per weed area. If every weed sprayed had 100% of the top leaves covered (similar to broadcast application targets weeds), the sprayer accuracy would also be 100%. The smart spraying system developed is shown in Figure 2.11.

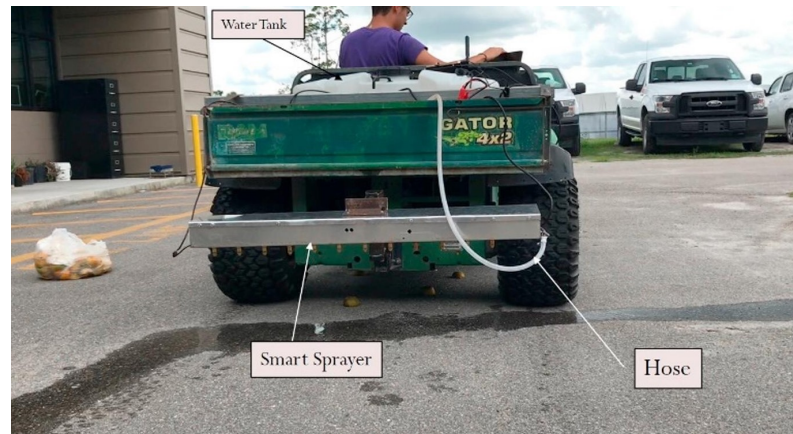


Figure 2.11. [Partel et al. \(2019\)](#)'s UTV-based Intelligent Herbicide Spraying System Setup.

While [Partel et al. \(2019\)](#)'s UTV smart sprayer attachment autonomously targeted and sprayed weeds, [Gonzalez-de-Soto et al. \(2016\)](#) designed, developed, and tested an unmanned, robotic patch-spraying system for herbicide application in cereals on a site-specific basis. A medium-sized New-Holland Tractor was customized to create the unmanned ground vehicle. The cab of the tractor was removed and replaced with a receptacle for computers and subsystems. A laser system was attached to the front of the vehicle for obstacle detection. The unmanned ground robot smart spraying system developed by [Gonzalez-de-Soto et al. \(2016\)](#) is shown in Figure 2.12.



Figure 2.12. [Gonzalez-de-Soto et al. \(2016\)](#)'s Unmanned, Herbicide-applicator Robot, Based off of New-Holland Tractor.

In this study, [Gonzalez-de-Soto et al. \(2016\)](#) tested a “WDS” or weed detection system that was remotely controlled and a system that was based entirely on-board the vehicle. During demonstration on a wheat field, two “aerial robots” collected images of the crop that were sequentially processed by a computer, on-board the unmanned ground vehicle, to create a weed control method (WCM) and relay that information to the actuation controller. When tested in a wheat field, with 16.5 x 6 meter weed patches randomly sewn into the field, the external WDS resulted in more than 95% of the weed patches within the field being sprayed. The vehicle-based weed detection system yielded an accuracy of 99.5% spray coverage. According to [Gonzalez-de-Soto et al. \(2016\)](#), the difference in accuracy between the two Weed Detection Systems was caused by a poorer position accuracy of the external WDS positioning system than by the GPS used in the vehicle-based WDS.

2.4 Large Scale Map Creation

([Sa et al., 2018](#)) developed a crop/weed segmentation and mapping framework to process UAS-based multispectral imagery on a whole field basis. To overcome challenges of using field scale, mosaiced images into general classifiers, a sliding window technique was used. This technique was performed on a subsection of the mosaic and then placed that section back on the map. The size of the maps created in this study were approximately four acres (16,500 square meters). Eight orthomosaic maps were created in total using a RedEdge-M camera and a Parrot Sequoia camera.

Out of both tested camera sets, one model performed the best for crop detection with 96% accuracy for background and crop detection, but only 62% accuracy for weed detection. Another model performed most accurately; with an 84% accuracy on background detection, 87% accuracy on crop detection, and 79% accuracy on weed detection. Overall conclusion was that though crop and background detection was detected with high accuracy, weed detection was still sup-par.

2.5 UAS-Based Spraying Systems

UAS-based sprayer systems from major companies such as DJI™ and Yamaha™ have been on the market for the last several years. The Yamaha R-MAX unmanned helicopter was developed in the mid-1990s. However, it was not until 2015 that the Federal Aviation Administration approved it for operation in the United States. As of May 2019, this unmanned helicopter has only been tested on vineyards in California's Sonoma Valley. Due to its weight being in excess of the 55-pound commercial UAV limit (physically attached to the aircraft itself, a special license and permission is required from the FAA to fly the R-MAX. A Yamaha R-MAX unmanned helicopter spraying system is shown in [Figure 2.13](#). As the FAA comes out with official rules and regulations for UAS herbicide application, it is likely that [Yamaha Motors USA \(2016\)](#) will expand their custom application business with the R-MAX to more tra-

ditional agricultural crops. It is also likely that they will expand operation to other parts of the United States, specifically Midwest crop production.



Figure 2.13. Setting up Yamaha R-MAX for Custom Herbicide Application in Vinyard, Sonoma Valley, CA ([Yamaha Motors USA, 2016](#)).

Without an efficient weed management strategy, investment in cutting edge technology like UAS sprayer systems is a wasted expenditure. In order to develop this strategy, it is important to know the factors that influence weed growth and development.

2.6 Factors influencing weed growth and development

Weed growth can be influenced by competition with the crop and other weeds present in the field, soil fertility, soil moisture, soil reaction to certain weed species, and climate factors ([Zimdahl, 2007](#)). The weed growth cycle includes seed, germination, growth, and death. Weed life cycles can be annual, completing their life in one growing season, biennial, requiring two seasons, or perennial, living more than two years. Weeds can be grouped into four major categories: grasses, sedges, lilies, and broadleaves ([Lingenfelter and Hartwig, 2007](#)).

Becoming familiar with the weed growth cycle allows for correct timing of herbicide application. This is critical for an effective weed management program. With the onset of glyphosate and other chemically-resistant weeds becoming more common and the new herbicides coming on to the market to control them, there are at least eight distinct application timings in row crop agriculture. This includes early preplant application, pre-emergence application, early-post application, late-post application, drop nozzle application, and harvest aid application ([Matthews et al., 2014](#)). It is also important to become familiar with the type of weeds that are common to the area application is undertaken in. Knowing common weeds to the Midwest and specific study area gives the user a better idea of what to look for in-field and developing a strategy to target these specific weeds.

2.6.1 Weeds Common to the Midwest and Pertinent Study Area

Before experimental research can be done on automating the detection of weeds, the type of weeds we can expect to find in the area being studied as well as their physical characteristics need to be distinguished. Due to the vast number of different species present within each major weed class, this section will focus on only grass-type weeds and broadleaf weeds ([Welch-Keesey, 2011](#)).

Monocot Weeds

Monocot weeds feature only a single cotyledon, or embryonic leaf. Hence the term monocot[yledon]. They are also referred to as grass type weeds. Monocot weed leaves have two main parts, a sheath which surrounds the stem and a blade which grows away from the stem ([Welch-Keesey, 2011](#)). Between these two parts lies the collar, an important area for identifying grasses. Other identifying features of grasses include the ligules, a growth standing up from the collar, auricles, protrusions that extend from the collar and wrap around the stem, and a presence or absence of hairs, wavy or straight edges, color, and the distinction of midrib ([Welch-Keesey, 2011](#)).

Other identifying features of a monocot weed is the presence of stolons and rhizomes. Stolons are stems that grow at the soil surface, or directly below the ground, which form adventitious roots at the nodes and new plants at the buds ([Woolley et al., 1979](#)). Conversely, rhizomes are modified subterranean plant stems that send out roots and shoots from their nodes ([Woolley et al., 1979](#)). The basic structure of monocot or grass-type weed is shown in Figure 2.14.

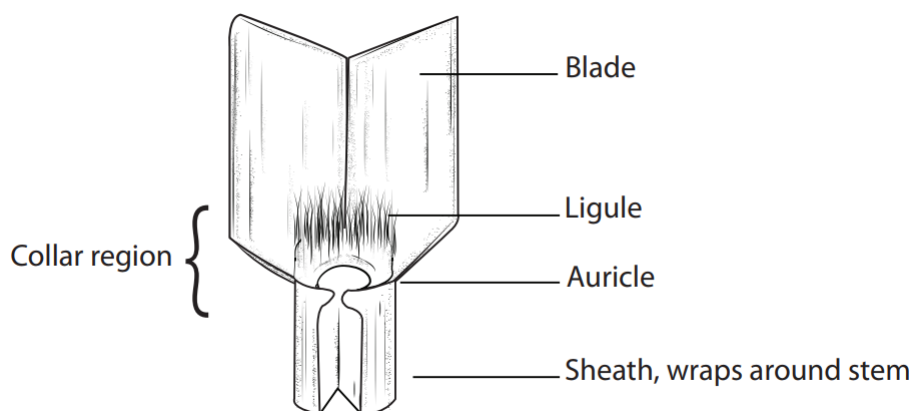


Figure 2.14. Identifying Features of Monocot Weeds ([Welch-Keesey, 2011](#)).

Monocot weeds common to the Midwest are crabgrass, foxtail, and quackgrass.

Broadleaf weeds are often more prevalent in Midwestern row-crop fields and are typically harder to control than monocot weeds.

Dicot Weeds

Dicot weeds have wide leaves and a central vein leading to smaller veins branching out. They feature paired cotyledons, or seed leaves that will typically appear around germination, hence the term dicot[elydon] ([PAN Germany, 2005](#)). Dicot weeds come in three different types, including basal leaves, opposite (also called whorled) leaves,

and alternate leaves. Identifying features of a dicot, also called broadleaf, weed is shown in Figure 2.15.

Basal leaf species common to the Midwest include broadleaf plantain and dandelion. Opposite (also called whorled) leaved broadleaves common to the midwest include Common chickweed, honeyvine milkweed, purslane, common ragweed, and giant ragweed fall into this category. The final type of broadleaf weeds belong to the alternate-leaf class. Types of alternate leaf weeds common to the midwest are Canada thistle, common lambsquarters, field bindweed, pokeweed, redroot pigweed, and velvetleaf. Identifying features of a dicot weed is shown in Figure 2.15.

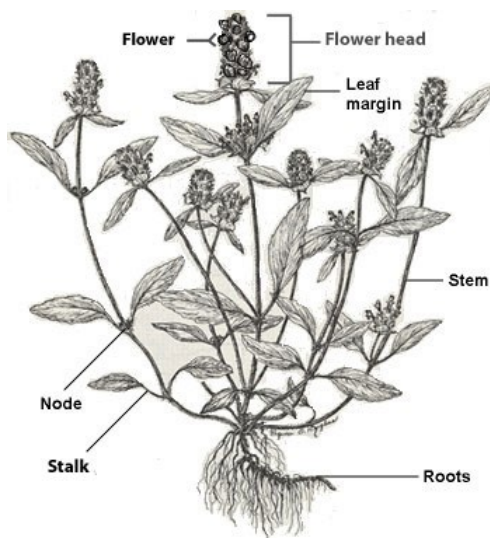


Figure 2.15. Identifying Features of Dicot Weeds ([PSU Turfgrass, 2017](#)).

These traits and physical characteristics are an important aspect of weed identification. The difficulty of training an object detection network to recognize certain weed types is that the network must learn these properties from the training images and annotations provided. Much like how humans are trained to recognize a type of weed by studying several, varying examples, a neural network must also learn recognition in this fashion in order to spot a weed instance in-field.

The studies and research reviewed in this chapter has shown that machine-learning based methods have the potential to develop a simple but accurate weed detection process that can be implemented with a small UAS and attached camera systems ([Sanchez-Lopez et al., 2017](#)).

3. METHODOLOGY

3.1 Methods used in dataset creation

Three sensors were selected for use in the research. They included the Flir Duo Pro R (Flir Systems Inc., Wilsonville, Oregon), Slantrange 3PX (Slantrange Inc., San Diego, California), and Micasense RedEdge 3 (MicaSense, Inc., Seattle, Washington). Detailed specifications of the sensors are provided in Table 3.1. Each sensor is shown in Figure 3.1 for comparison.

Table 3.1.
Specifications of the RGB, Multispectral, and Thermal Sensors used for this Study.

Specification	Micasense	Slantrange	Flir RGB/T
Spatial Resolution at 30 meters	2 cm	1.8 cm	1.5 cm/ 3 cm
Spectral Range, Bandwidth (nm)			410-680 (RGB spectrum)
Blue	475, 20	550, 40	750-1350 (Thermal)
Green	560, 20	650, 40	
Red	668, 10	710, 30	
NIR	840, 40	850, 100	
Spectral Resolution	5 bands	4 bands	
Cost (USD)	\$5,500	\$5,250	\$7,599
Sensor Size (mm)	4.8 x 6.2 mm	5.76 x 4.53 mm	10.88 x 8.704 mm/ 7.4 x 5.5 mm

Two unmanned aerial systems (UAS) were chosen for this study. The first UAS, a DJI Mavic Pro (SZ DJI Technology Co., Ltd., Shenzhen, China), was used for initial field scouting and pre-planting data acquisition. It has a built in 4K color (referred

to as RGB throughout the rest of the chapters) sensor and gimbal that had turned this small, foldable UAS, an ideal preliminary data collection tool. The second UAS, a DJI Matrice 600 Pro was purchased along with a Gremsey T3 gimbal for attaching the three sensors mentioned in the beginning. The cost of the Mavic pro was \$1,200 USD, while the cost of the Matrice 600 Pro was \$4,000 USD. The cost of the Gremsey T3 gimbal was \$1,500 USD. Detailed specifications about these two UAS has been provided in Figures A.1 and A.2 in the appendix.



Figure 3.1. Visual Comparisons of the Multispectral and Thermal/RGB Sensors used in This Research.

The sensors were attached to UAS for collecting imagery in corn and soybean fields at the three different research sites located in the State of Indiana. Data was collected throughout the 2018 and the first month of the 2019 growing season.

3.1.1 Choosing Research Area for Data Acquisition

Experimental research plots for this study were chosen for their weed density and diversification throughout the growing season. Areas chosen also needed to be in different areas of Indiana, to get a sample of weeds present in the state. This study focused on three research locations, Pinney Purdue Agricultural Center (PPAC) (fig.

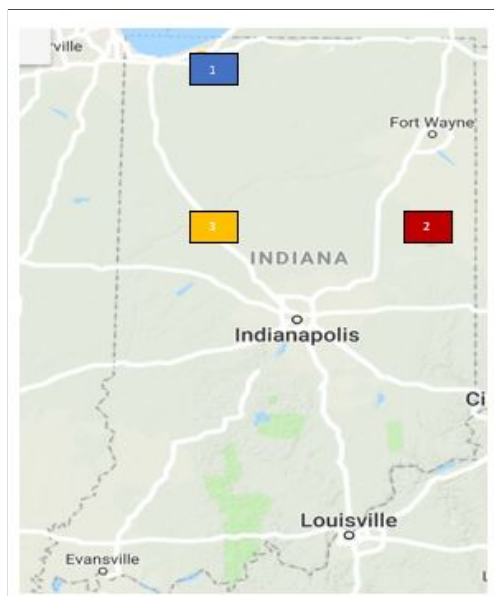


Figure 3.2. Research Plot Locations 1:PPAC 2:DPAC 3: TPAC.

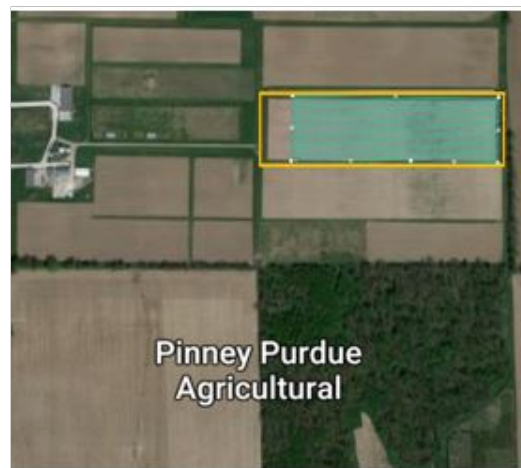


Figure 3.3. PPAC Trial Plot.



Figure 3.4. DPAC Trial Plot.



Figure 3.5. TPAC Trial Plot.

3.3), Davis Purdue Agricultural Center (DPAC) (fig. 3.4), and Throckmorton Purdue Agricultural Center (TPAC) (fig. 3.5).

At PPAC, the most commonly found weeds within the research plot where data was collected were giant ragweed (*Ambrosia trifida*), velvetleaf (*Abutilon theophrasti*), common lambs-quarters (*Chenopodium album* L.), and redroot pigweed (*Amaranthus retroflexus* L.). Monocot weeds at PPAC were predominately giant foxtail (*Setaria faberi*). At DPAC, the most commonly found weeds within the research plot were tall waterhemp (*Amaranthus tuberculatus*), velvetleaf, and prickly sida (*Sida spinosa*). At TPAC, the most commonly occurring weeds found in the research plots were giant ragweed (*Ambrosia trifida*), and redroot pigweed (*Amaranthus retroflexus*). Monocot weeds commonly found at TPAC were green foxtail (*Setaria viridis*), and panicum (*Poaceae Panicum virgatum*).

3.1.2 Data Collection Mission Setup

Each of these trial plot locations were scouted with a DJI Mavic Pro UAS, well before planting began. Video and RGB images were captured of the field sites, from the built-in camera on the Mavic Pro. With real time video capability on the Mavic Pro, potential problems within the field site, such as large trees and busy roads in close proximity to the field boundaries were able to be inspected during flight. Images from the flight mission were stitched together, to create an orthomosaic. This enabled the creation of a high-resolution base map of each field site.

Ground control points (GCPs) were placed around the boundary of each experimental plot. These points were created from 1x1checkerboard-patterned linoleum squares, placed on a heavy ceramic tile. They were spaced equidistant around the plot perimeter. The latitude and longitude position of GCPs were acquired with a TopCon HiPer VTMRTK base station and receiver (Topcon positioning systems, inc., Livermore, CA)([TopCon, 2018](#)) and was used in image stitching to improve GPS accuracy. These measurements were within a 2-3 centimeter accuracy of the GCP's physical location in-field. The GCP boundary map and TopCon HiPer VTMRTK base station and receiver used to locate the GCPs are presented in Figures [3.6](#) and [3.7](#).



Figure 3.6. Ground Control Point Boundary Map Created from RTK Point Collection.



Figure 3.7. TopCon HiPer V™ RTK Base Station and Receiver and GCP Measurement.

At TPAC, the soybean research site was planted on May 27th, 2018. Planting began on the TPAC corn research site on April 27th, 2018. Planting began at the PPAC research site, consisting of both corn and soybean plots, on May 28th, 2018. Planting began at the DPAC research site, also consisting of both corn and soybean plots, on May 17th, 2018. A timeline of data acquisition dates, over the 2018 growing season is shown in Figure 3.8.

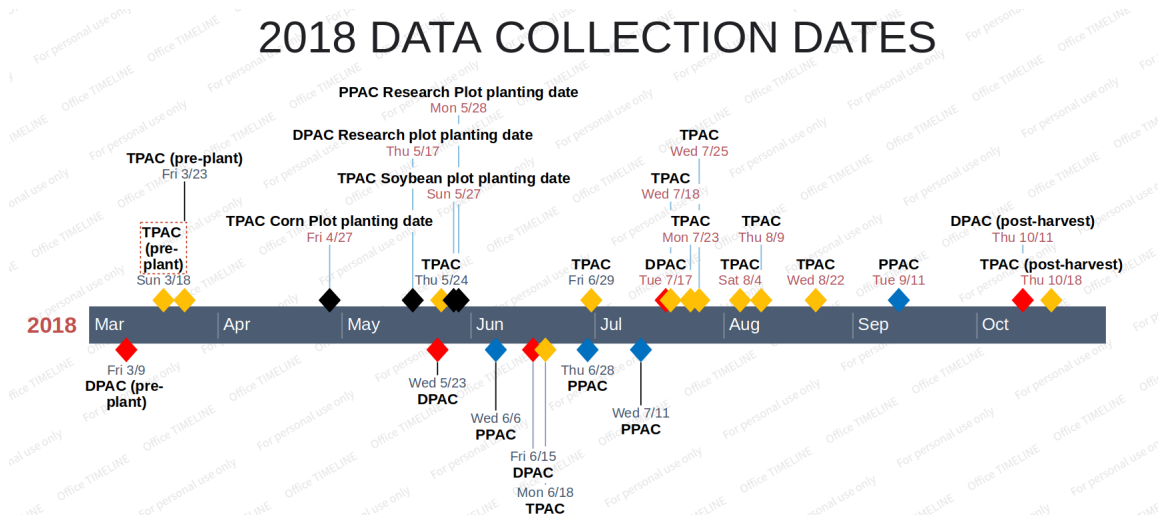


Figure 3.8. 2018 Data Acquisition Dates.

Data collection missions were flown within 30 minutes of solar noon, or when the sun is at its highest point in the sky. This practice is an important step in radiometric calibration, or reducing the effects of radiation scattered on the atmosphere. Sunlight's spectral composition is directly affected by factors such as weather conditions (cloudy, haze, fog), time of day, and solar position ([Assmann et al., 2018](#)). A board of known reflectance was used to calibrate relative measurements obtained from the multispectral, thermal, and RGB sensors. This calibration was deemed essential to attain temporal accuracy across different data acquisition dates and sites. Calibration images were taken of a SpectralonTM panel with known reflectance value before and after each flight. In addition, black, white, red, gray, and green spectral targets were also placed in line, next to the SpectralonTM panel. During several data collection dates, a SpectraVistaTM mass spectrometer was used to measure the radiance of these targets, as well as the radiance of the crop and soil background. This data was then downloaded, converted to reflectance, and used at a later image processing step.



Figure 3.9. UAS-based Image of the Spectral Targets used During Data Collection.

Data collection was also done during the first month of the 2019 growing season, at the Agronomy Center for Research and Education (ACRE) and TPAC. Due to the high amount of rain received during the Spring of 2019, planting at these locations did not take place until late May. Planting at the TPAC corn plot began on May 24th, 2019. Planting at the ACRE corn plot began on May 28th, 2019. The timeline of 2019 data acquisition dates is shown in Figure [3.10](#) below.

2019 DATA COLLECTION DATES

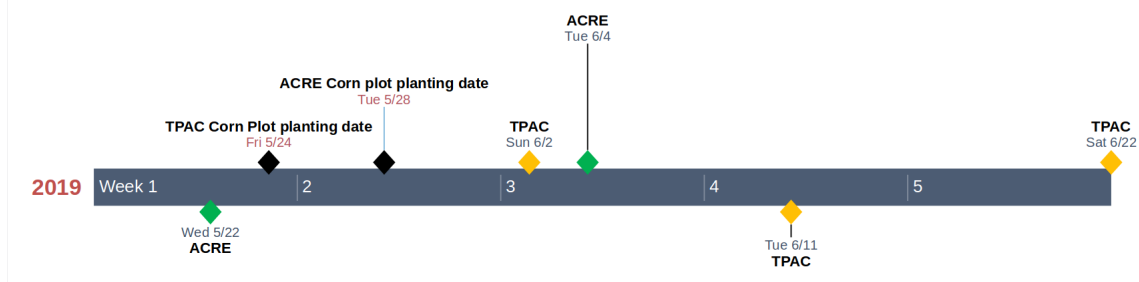


Figure 3.10. 2019 Data Acquisition Dates.

As research plot fields were rotated between corn and soybeans between the 2018 and 2019 growing seasons at TPAC, the location of the corn plot flown was slightly different than in 2018. The 2019 corn plot location, used for data collection, is shown in Figure 3.11 below.



Figure 3.11. 2019 TPAC Corn Plot Location.

Data collection at ACRE was done to increase the amount of images for the 2019 growing season. Late planting and the conclusion of research dictated the removal of DPAC and PPAC from the flights conducted in 2019, as neither location had planted plots at the time data collection needed to be done. The map shown in Figure 3.12 below shows the two plot locations flown at ACRE. The first location, flown May 22nd, 2019 was done during pre-plant. The second plot location, flown June 4th, 2019, had V1 growth stage corn.

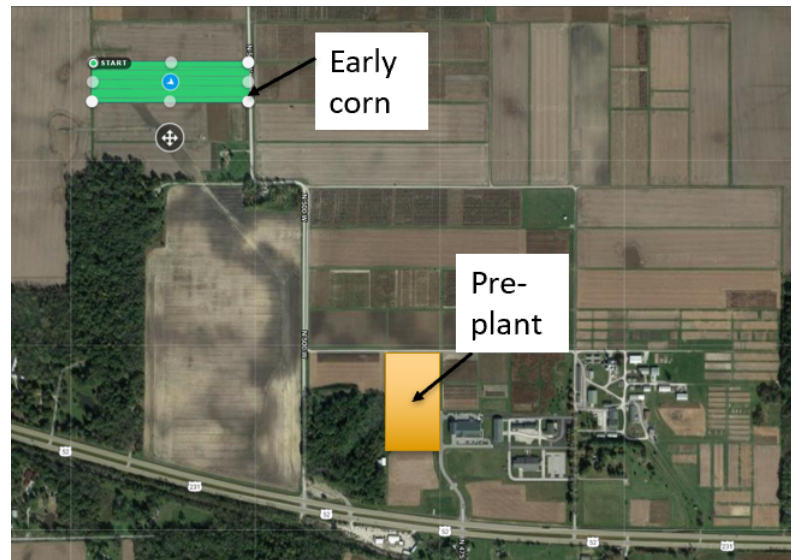


Figure 3.12. 2019 ACRE Pre-plant and Corn Plot Locations.

ACRE is located at 4540 US-52, West Lafayette, IN 47906, as shown in Figure 3.13 below. The coordinates of the pre-plant field flown are 40.471077, -86.996117. The coordinates of the early season corn field flown are 40.477930, -87.003301.

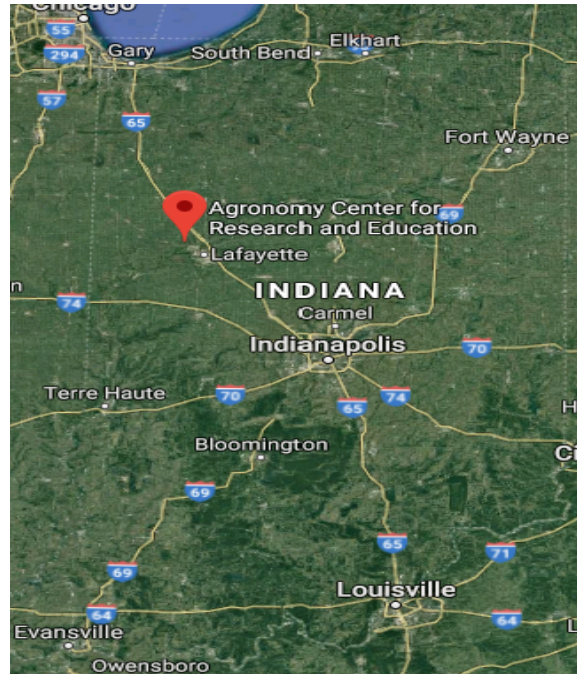


Figure 3.13. Geographic Location of Agronomy Center for Research and Education (ACRE) .

Data collection missions were set up using the DroneDeployTM mobile application. DroneDeployTM is a flight automation software for unmanned aerial systems that allows the user to set a predetermined flight path, speed, and percentage values of side and front lap. Both the side and front lap values were kept as 75% meaning thereby that each consecutive image captured would overlap 75% with the image taken beside of it (from the previous flight line) and in front of it. Flight elevation was set at 30 meters at ground level (AGL) for the majority of data collection missions. For later stage corn (V6 and later), altitude was increased to 50 meters due to crop height, weed density, and canopy cover. Flight speed was kept constant at 3 meters per second. The SlantrangeTM 3PX multispectral sensor and the FlirTM Duo Pro R sensors were attached to the UAS to capture imagery during flight. Before these sensors were available, the MicasenseTM RedEdge 3 was used for early data collection. Each camera was set to trigger at one second intervals. At the end of the flight, SD

cards were removed from the sensors, data was downloaded on a laptop and analyzed for errors, skips, and distortion. After data collection, image analysis, and machine learning methods were applied.

3.1.3 Analysis of UAS acquired imagery

Images acquired before missions were flown, to calibrate the sensor GPS, were removed from mission imagery to reduce geometric errors during processing. During the early part of the 2018 growing season, between V3 and V5 growth stages RGB, multispectral, and thermal images, respectively were stitched together using Pix4DTM image stitching software (Pix4D S.A., Prilly, Switzerland) to create an orthomosaic map of the research areas. Mosaics were created for each of the three image types for the TPAC soybean, DPAC, and PPAC research areas. Mosaics maps of the TPAC corn research site were not created, as imagery collected after v6 growth stage made it difficult to stitch. To create mosaic images in Pix4D, image key points were computed, an aerial triangulation and bundle block adjustment procedure was performed for geometric correction, a point cloud array was created from automatic tie points, and then a 3-D textured mesh was created from the point cloud (Pix4D, 2017). These steps allowed for creation of a digital surface model, that lead for orthorectification and building of a final mosaic. Using multispectral imagery collected with the Slantrange 3PX, a normalized differential vegetation index, NDVI, was created with the near infrared (NIR) and red bands to measure vegetation stress (Rouse et al., 1974). RGB, thermal, and NDVI orthomosaic maps of the TPAC soybean research site are shown in Figure 3.14.



Figure 3.14. RGB Orthomosaic Map of TPAC Plot, created June 18th, 2018 of V4 Growth Stage Soybeans.

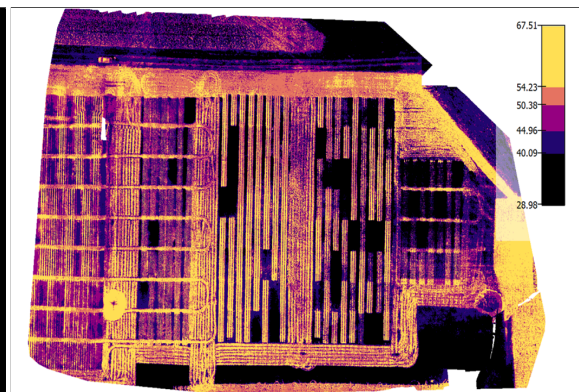


Figure 3.15. Thermal Orthomosaic Map of TPAC Plot, created June 18th, 2018 of V4 Growth Stage Soybeans.

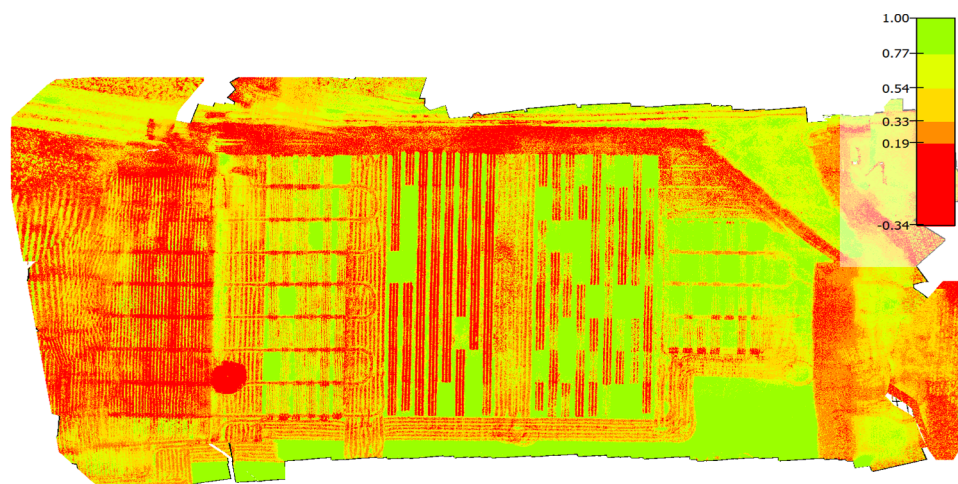


Figure 3.16. NDVI Orthomosaic of TPAC Plot, created June 18th, 2018 of V4 Growth Stage Soybeans.

3.2 Labelling and Annotation of Weeds Dataset

While plot mosaic maps were useful for ground truthing and field layout purposes, training an object detection model required a set of individual, labelled images. This was accomplished by manual annotation and bounding box creation for hundreds of individual images. Images chosen for manual annotation were acquired during the 2018 and early 2019 growing season, within corn and soybean crops. Only RGB imagery was used for creating a labelled dataset for weed detection model training. The RGB camera on the Flir Duo R Pro offered a larger sensor size (10.9 x 8.7 mm), a wider field of view (FoV) (56°x 45°), higher spatial resolution (1.5cm at 30 meters), and higher pixel resolution (4000x3000), than other sensors used in this research (shown in Figures A.3 and A.4). Another reason for focusing on RGB imagery is that high resolution, UAS-based, RGB cameras are more affordable and more commonly used by farmers than multispectral or thermal UAS-based cameras (Margaritoff, 2018).

After a suitable RGB image had been selected for labeling, a bounding box was drawn around each instance of a weed, and an annotation was given of the weed's common name. This was done using Alps Labelling Tool, an extension of the ImageJ™ open source photo editing software (Alp's Labeling Tool, 2017). A training set of these images was created, along with their labels, in a fashion that could be read in and used by Python.

3.2.1 Methods for Choosing Imagery for Manual Annotation

In the context of this study, the desired object to be labelled and bounded is the specific weed type, either monocot or dicot, present in the UAS collected imagery. A total of 24 data collection flights were conducted throughout the 2018 growing season and four flights were conducted during the first month of the 2019 growing season. Of the 25 flights where RGB imagery was collected, an average of 1113 images were acquired per flight. Only multispectral imagery was collected with the MicanSense Rededge 3 on May 23rd and 31st, 2018 and July 23rd, 2018. The May,

2018 data collection with this sensor was done before receiving the Flir Duo Pro R. The July collection only using the Micasense RedEdge 3 was done to compare image stitching results to the Slantrange 3PX and Flir Duo Pro R cameras. There was difficulty stitching imagery with all three of these cameras, as the corn plot flown at TPAC, in late July, was too homogeneous for digital surface model creation (DSM). The Flir Duo Pro R camera was set to capture one image every second and the average flight time of data collection missions were approximately 24.5 minutes. The Flir Duo Pro R camera captures RGB and thermal imagery and stores both in single JPG image format. For image annotation, only RGB images were extracted, as Visual JPG 4000x3000 pixel images. While a total of 27,828 RGB images were acquired during data collection, only two percent of these images were annotated for network training dataset creation. The reason for this is that only early season, post-harvest, and pre-plant, UAS-based RGB imagery was chosen for manual annotation. Individual weed examples during later growth stages of the crop could not be used for annotation because there were no instances of individual weeds to be labelled. All of the later growth stage imagery had areas of heavy weed infestation, where weeds were clumped together. These images might be useful for other researchers and future research strives, but the focus of this research was on individual weed identification. A flowchart of manually annotated, training image set creation is shown in figure 3.17. Each of these sets will be denoted as “*Training Image Set #*” in order to better explain the methods for training set creation.

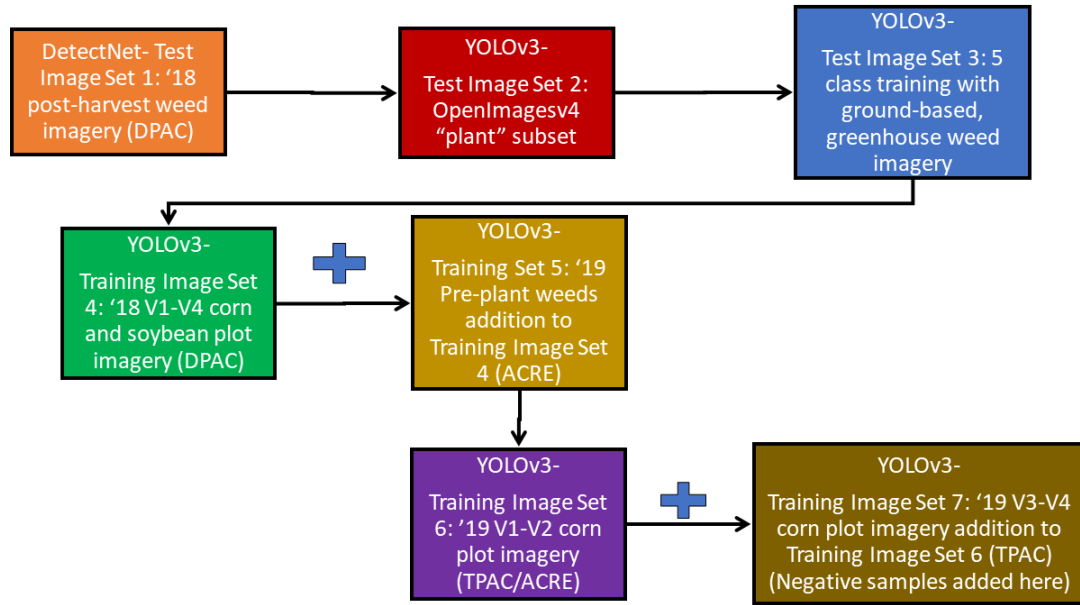


Figure 3.17. Flowchart of Training Image Set Creation for Object Detection Network Training.

For the creation of *Training Image Set 1* for Detectnet training in NVIDIA DIG-ITS, discussed in 3.3.1, post-harvest imagery from DPAC (taken October 11th) and TPAC (taken October 18th) fields was chosen for manual labelling. Choosing these images made it easier to find and label weeds, without an excessive amount of background noise interfering. Image selection was based on weed variation and intensity within the harvested field. Average weed diameter was 4 inches and average height was 3 inches. Background features, such as straw, stover, and grasses, needed to be minimal within the image, for ease of annotation and better training results. A mix of post-harvest images from each research plot location were selected, to add variability to the dataset.

For the creation of *Training Image Set 2*, a proof of concept was done for initial training with YOLOv3. Images were parsed, along with their corresponding labels, from the OpenImages online database, discussed in 3.3.2. The parsed images belonged

to the *plant* class subset within the larger OpenImages database. Plant diameter and height varied in this imagery, as uploaded images included different and random scenes. Because this initial training with YOLOv3 was done as a proof of concept, class labels were changed from *plant* to *weed*. Succeeding image set classes were labelled based on individual weed species and two main weed classes.

For creation of *Training Image Set 3*, multi-class annotated image set creation was done for training, discussed in 3.3.3. For this creation, giant ragweed, cocklebur, foxtail, corn, and nitrogen deficient corn were chosen as weed and crop class labels. Corn and nitrogen deficient corn were added to this set for variation. Dicot weeds (giant ragweed and cocklebur) had an average diameter of 4.5 inches and an average height of 3.5 inches. Monocot weeds (foxtail) had an average diameter of 6 inches and an average height of 8 inches. As this image set was made for a controlled network training experiment, lab-based conditions were applied for growth and image acquisition. Weeds and corn in this imagery were grown in the Purdue greenhouse. In addition, the lighting, angle and height of images taken were controlled. Training for this control group was compared to training performance of succeeding UAS-based image sets, discussed in 4.3.4. Due to the difficulty and amount of time needed to label individual weeds by their unique species, the weeds in the following UAS-based image data sets were classified based on two major types i.e. monocot and dicot. These images were acquired during April and May 2018, as well as April and May 2019 with a handheld camera, discussed in 3.3.3.

For creation of *Training Image Set 4*, UAS-based weed imagery was acquired during the early growth stages of corn and soybeans in 2018. Training on this image set was done in 3.3.4. Images used to create this dataset were collected on June 15th, 2018, at the Davis Purdue Agricultural Center (DPAC) in Farmland, Indiana. The corn growth stage in these images were between V2 and V4. The soybean growth stage in these images were also between V2 and V4. Average diameter of dicot weeds in this imagery was 5 inches and the average height was 3.5 inches. The average diameter of monocot weeds in this imagery was 6.5 inches and the average height

was 7 inches. In order to accurately label the larger-sized UAS-based imagery, a new labelling tool had to be used, capable of zooming in on a particular region and clearly showing the weeds present therein. Research and testing was performed on several, open-source labelling tools, including OpenLabeling, BBox-Label-Tool, Boobs (an enhanced zoom labelling tool), and Labelbox to identify a labeling tool that provides the speed needed to label up to hundred of weeds within an image, as was needed for the UAS-based image set. It was found that none of these labelling tools were able to provide the speed needed to label up to hundreds of objects within an image, as would be required in the UAS-based image set. A labelling tool was finally found, LabelIMG, which provided adequate labelling speed and user-friendly zoom and pan features. LabelIMG, written by Darren Tzutalin, is a “graphical image annotation tool written in Python and using Qt for its graphical interface” (Tzutalin, 2019). Due to the amount of time required to complete manual annotation, factors such as speed and ease of use become increasingly important in the overall network training workflow.

For creation of *Training Image Set 5*, 2019 pre-plant weed imagery added to *Training Image Set 4*. The average diameter of dicots in this imagery was 4.5 inches and the average height was 3 inches. The average diameter of monocot weeds was 5.5 inches and average height was 6 inches. This bolstered image set then underwent network training in 3.3.5. Images labelled within this training set were acquired at the ACRE test plot on Wednesday May 22nd, 2019.

For creation of *Training Image Set 6*, images of VE-V1 growth stage corn plots were labelled and added to the overall dataset. This was done to compare the differences between network performance on a multicrop image set, such as *Training Image Sets 4 & 5* to an image set from a single crop plot. These images were collected on June 2nd, 2019 and June 4th, 2019 at TPAC and ACRE respectively. Average diameter of dicot weeds in this imagery was 1.5 inches. Average height was less than 1 inch. There were no monocot weeds in this imagery.

For creation of *Training Image Set 7*, V2-V4 growth stage corn plot imagery was labelled. These images were acquired on June 11th and June 21st 2019, at TPAC and ACRE respectively. These images were added to Training Image Set 6 and used in network training in 3.3.7. Within these images, corn had fully emerged and was easier to decipher between weeds and crop. Average diameter of dicot weeds in this imagery was 3 inches and average height was 2 inches. For monocot weeds, average diameter was 4 inches and average height was 4.5 inches. Corn in this imagery ranged between the v2 and v3 growth stages. For every monocot and dicot labelled image, a weed-free image showing only corn was added to the training set. Reasoning for this will be discussed in detail in 3.3.7.

3.2.2 Hardware and software setup used for deep learning network training

An Alienware R3 laptop computer with a 2.8GHz Core i7-7700HQ processor, with 32 GB of RAM was used to train the Detectnet and Darknet networks, used by NVIDIA DIGITS and YoloV3 respectively. A six gigabyte NVIDIA GTX 1050 graphical processing unit (GPU) was installed on the laptop to enable deep learning network training. The Ubuntu operating system, version 18.04 was installed on the laptop. Ubuntu was disk partitioned to 200 GB, as it was set up to be dual booted alongside the Windows operating system already available in the computer.

3.3 Object Detection Network Training Methods

3.3.1 Initial Network Training using NVIDIA DIGITS

Once the training and validation datasets had been created, they were loaded into the NVIDIA Deep Learning GPU Training System (DIGITS) (Barker and Prasanna, 2016). DIGITS is an online platform that simplifies deep learning tasks such as data structuring and management, neural network design and customization, real time

performance monitoring and visualization, and model selection from the browser to deployment (Barker and Prasanna, 2016). The post-harvest image set was used to train a DetectNet model. DetectNet is a Caffe-based framework, created specifically for object detection. Every image was obtained at a height of 30 meters. Image and label paths were loaded into DIGITS API platform, under the “Object Detection Training Set” creation tab. On the object detection training set creation page, the following options were given:

Object Detection Dataset Options
 Images can be stored in any of the supported file formats (.png/.jpg/.jpeg/.bmp/.ppm).
Training image folder
 /home/ai/Desktop/DPAC_post_images_for_obj/Train/Images
 Label files are expected to have the .txt extension. For example if an image file is named foo.png the corresponding label file should be foo.txt.
Training label folder
 /home/ai/Desktop/DPAC_post_images_for_obj/Train/Labels
Validation image folder
 /home/ai/Desktop/DPAC_post_images_for_obj/Val/Images
Validation label folder
 /home/ai/Desktop/AgBot_data_for_ECE570/Val/Labels
Pad image (Width x Height)
 2000 x 2000
Resize image (Width x Height)
 1900 x 1200
Channel conversion
 RGB
Minimum box size (in pixels) for validation set
 25
Custom classes
 cocklebur, weed

Feature Encoding
 PNG (lossless)
Label Encoding
 None
Encoder batch size
 32
Number of encoder threads
 4
DB backend
 LMDB
Enforce same shape
 Yes
Group Name

Dataset Name
 DPAC_post_weeds
 Create

Figure 3.18. NVIDIA DIGITS Graphical User Interface (GUI) Dataset Creation Options (Barker and Prasanna, 2016).

The following network hyperparameters were set:

- Image padding: 2000 x 2000 pixels
- Image resize: 1900 x 1200 pixels
- Channel conversion: RGB
- Minimum Box Size: 25
- Classes: cocklebur, giant ragweed, and redroot pigweed

- Feature encoding: PNG
- Label encoding: none
- Batch size: 32
- Number of encoder threads: 4
- Database back end: LMDB (Lightning MemoryMapped Database Manager)
- Dataset name: *Post_harvest_test*

These steps were input into the *new image model* creation page shown in Figure 3.19.

New Image Model

The screenshot shows the 'New Image Model' creation interface in the NVIDIA DIGITS GUI. It is organized into four main panels:

- Select Dataset:** A list of datasets is shown, with 'DPAC_post_right' selected. Below the list, details for the selected dataset are provided, including the database backend (lmdb), entry count (38), feature shape (3, 1000, 1000), and label shape (1, 71, 16).
- Python Layers:** A section for specifying the model architecture, currently showing a 'Server-side file' input field and an unchecked 'Use client-side file' option.
- Solver Options:** A section for configuring the training process. Fields include 'Training epochs' (50), 'Snapshot interval (in epochs)' (1.0), 'Validation interval (in epochs)' (1.0), 'Random seed' (none), 'Batch size' (2), 'Batch Accumulation' (2), 'Solver type' (Adam (Adaptive Moment Estimation)), and 'Base Learning Rate' (0.0001). There is also a checkbox for 'Show advanced learning rate options'.
- Data Transformations:** A section for specifying data augmentation and normalization. Fields include 'Subtract Mean' (None) and 'Crop Size' (none).

Figure 3.19. NVIDIA DIGITS GUI Object Detection Network Model Creation Options (Barker and Prasanna, 2016).

For initial testing, only one custom class, “weed”, was specified. This was done to ensure that the DetectNet Network could differentiate between the weeds and post-harvest corn stover within the imagery.

The two main calculations used to measure network training effectiveness are recall and mAP. From NVIDIA’s documentation, precision, “is the ratio of the accurately identified objects to the total number of predicted objects. Recall is, “the ratio of the accurately identified objects to the total number of actual objects in the images. In this instance, mAP, or mean average precision, is “the product of the precision and recall” ([Barker and Prasanna, 2016](#)). Mean average precision measures how apt the network is to objects of interest. At the same time, it gives an idea of how well the trained network can avoid false alarms. One reason for this low accuracy is the limited number of images trained on.

3.3.2 Initial training methods with YOLOv3 on OpenImages Dataset

Another object detection model used in this research was the YOLO (You Only Look Once) single prediction network. The YOLO model used in this research was based off of the Darknet open-source neural network framework ([Redmon and Farhadi, 2018](#)). For training of the annotated weed dataset, a 13x13 grid detector was used. Nine anchor boxes were used per cell for bounding box prediction, which is fixed in the YOLOv3 model. A confidence score is given on the object certainty within a bounding box prediction, along with a probability distribution over all given classes, this is called the detection threshold. The threshold of detected objects was tested at 25% and 50%, in the validation and test image sets. For this particular network training, a 50% detection threshold yielded the best results, shown in [4.3.2](#). Depending on the image set that the network trains on, the detection threshold that yields the best results will vary. This research utilized the third version of YOLO, YOLOv3. The architecture of this version utilizes a 53 convolutional layer variant of Darknet, that was trained on ImageNet. These convolutional layers are used for feature extraction. For detection, another 53 layers are added, giving a total of 106 layers to the YOLOv3 architecture ([Redmon and Farhadi, 2018](#)). The same number of layers were used during YOLOv3 network training of Training Image Sets

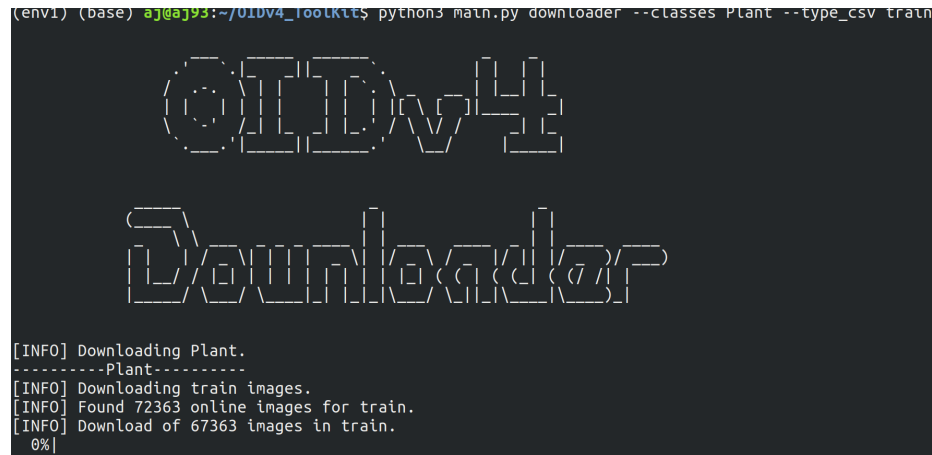
2-7, as training resulted in higher mean average precision than other YOLO network configurations with less layers. These configurations included *YOLOv3_tiny* with 30 total layers, *Darknet19*, which has 38 total layers, and *YOLO9000* which has 25 total layers. A higher mean average precision produces better detection results for the specified classes in the training image sets. This format was kept for all YOLOv3 network training.

The YOLOv3 network was trained using the OpenImages V4 dataset. This is an open source, online project in which people can submit their labelled imagery of different classes. The current version of Open Images contains “15.4 million bounding-boxes for 600 categories on 1.9 million images. This is the largest existing dataset with object location annotations” (Ferrari, 2018). Images, along with their label files, in the plant category were chosen to train network weights on. In order to do this, several steps had to be taken, from downloading the image sets and labels, to converting annotations to a format that YOLO can read. These steps were all done using the OIDv4 Toolkit Github repository by Angelo Vittorio (Vittorio, 2018). Using the *main.py* file in the repository, a command line script was written, shown in Figure 3.20, which downloaded the images and labels by class type and test, train, or validation sets.

Once the Plant image and label sets for train, test, and validation were downloaded, the label files were converted from the format given by Open Images (a CSV file) to a text document that could be interpreted by YOLO. This was done using the *OID_to_YOLO.py* file (Ferrari, 2018). After converting the labels to a usable format, the next step was to train weights. A data file first had to be written, in order to specify number of classes being trained on, the file path to the images and labels for training and validation, a path to the name of the classes text file, and a path to the folder storing the resulting training weights. This data file was named *darknet.data*. The configuration file for YOLOv3 was then edited to the desired hyperparameters for training:

- Batch size: 32

```
(env1) (base) aj@aj93:~/01bv4_toolkit$ python3 main.py downloader --classes Plant --type_csv train
```



```

  @Iv4
  Downloader

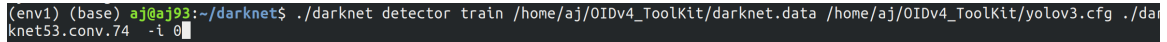
[INFO] Downloading Plant.
-----Plant-----
[INFO] Downloading train images.
[INFO] Found 72363 online images for train.
[INFO] Download of 67363 images in train.
0%|

```

Figure 3.20. Bash Command for Parsing Plant Subset from OpenImagesv4 Database ([Vittorio, 2018](#)) .

- Subdivisions: 16
- Classes: 1
- Input image width and height: 416 x 416
- Channels: 3 (for R,G,B)
- Learning rate: 0.001
- Max batches: 5000
- Label encoding: none
- Batch size: 32

The rest of the configuration file was left the same. Finally, a convolutional weight file, pre-trained on ImageNet, was used in the training command to help initialize transfer learning from a much larger image dataset, to the specific class of image desired for the network to learn.



```
(env1) (base) aj@aj93:~/darknet$ ./darknet detector train /home/aj/0IDv4_ToolKit/darknet.data /home/aj/0IDv4_ToolKit/yolov3.cfg ./darknet53.conv.74 -i 0
```

Figure 3.21. Bash Command for Training YOLOv3 Network on Labelled Plant Subset from OpenImagesv4.

Training was stopped at 5,000 iterations (also referred to as batches or epochs) because results no longer improved with further training. As YOLO training weights were set to be saved at every thousand iterations, results were also evaluated at the 3,000 and 4,000 iteration weights. It was found that the 5,000 iteration weights gave the best overall results based on percentage of images where “plants” were detected and where the highest overall accuracy was achieved. Once adequate network training results from this experiment were achieved, this research wanted to experiment with YOLOv3 network training for multi class weed detection.

3.3.3 Methods for multi-class weed detection with YOLOv3

After verification of results from physical training as a proof of concept, a larger, more robust training set was created. The images in this dataset were acquired with a Sony™WX350 18.2 megapixel digital camera. At minimum, 300 images of each of the five classes were chosen for inclusion in the new image training set.

A new image labelling software was implemented, which significantly sped up the annotation time of the image training set. This labelling software, *YOLO-mark*, was developed by a software engineer deeply involved in Darknet programming, Alex Mohnatkin (Monhatkin, 2019). This tool can be run from a simple Python script. The script outputs an interactive window which can be used to label all of the images within a designated folder. Labelling was done by drawing a rectangular bounding box around each desired training object within the image. Label annotations were based on categories, given from the *YOLO.names* file. A slider within the script window

was used to transition between these categories and choose the correct one for the object being labelled. The Python script translated the top right, top left, bottom right, and bottom left pixel locations of the drawn bounding boxes and corresponding class name into a *.label* file. While the *YOLO.names* file was edited to include all of the desired training classes beforehand, each class was trained under a sub/-folder of the corresponding images. For example, while labelling cocklebur, only images of cocklebur were put into the folder to be labelled. This was done to prevent errors in object labelling and enabled objects within the image to be labelled quickly and accurately. In addition to a faster annotation process, YOLO-mark offers direct integration into the network training process. All of the necessary files for object detection network training were produced, in the same directory. File paths were copied into the command line and ran. In order to ensure the network training process runs quickly, several other parameters were considered.

One such parameter to consider was the configuration file. This file, also referred to as the *.cfg* file, sets the architecture for how the network was trained. The following hyperparameters were chosen for the study:

- batch size- 64
- subdivisions- 32
- image height- 608
- width- 608
- channels- 3
- learning rate- 0.001
- max batches- 50,000
- feature scale- 0.1. 1
- number of classes- 5

During the next stage of network training, anchor box configurations were set to a range that is able to detect all of the class objects within the image, from the smallest to the largest labelled object. This allowed for all class objects to be detected, regardless of their size.

3.3.4 Methods for Two-class UAS-based Image Set Training

Once the YOLOv3 network had been trained on this larger, more complex image set, a UAS-based image set was then created. Images in this dataset were taken with the FlirTMDuo Pro R, discussed in section 3.1.

For initial training of the YOLOv3 network, using UAS-based imagery, early season data was selected for annotation and dataset creation. Images chosen for labelling for this initial UAS dataset were in a two acre soybean plot, surrounded on all sides by corn. The soybeans in these images were in the V1-V2 growth stage. The surrounding corn plots however, were planted prior to the soybean plot were in the V4 growth stage. A Flir Duo Pro R camera, attached to a DJI Matrice 600 Pro UAS was used to collect this imagery. Each RGB image had a 4000x3000 pixel resolution. Instead of labelling each weed by their species, it was decided to use two categories, dicot and monocot. Aside from allowing for quicker annotation, these two categories were chosen to provide a representation of weed types present in the field. It is possible that a different weed management strategy be used for a field with more monocots than dicots present, or visa versa.

Before the network could be trained, several parameters needed to be changed in the configuration file *YoloV3.cfg*. The following hyperparameters were modified for this training:

- batch size- 64
- subdivisions- 16
- image height- 416

- width- 416
- channels- 3
- learning rate- 0.001
- max batches- 50,000
- policy steps- 45,000, 48,000
- feature scale- 0.1, 1
- burn in- 1000
- number of classes- 5
- i_snapshot_iterations- 100
- anchors- 4, 4, 6, 6, 12, 6, 21, 8, 12, 15, 43, 11, 21, 25, 39, 44, 70, 86

Due to memory constraints on the laptop's GPU, image height and width had to be resized to 416 x 416 pixels. This was nearly ten times smaller than the original input images. The Darknet neural network was used to read a bounding box by the pixel location of its four corners within an image, allowing for the bounding box's width and height to be normalized by the width and height of the resized image. Specifically, the x-center, y-center, width, and height are float values which are relative to the width and height of an image. The values of these indices could range from 0.0 to 1.0 ([Mohnatkin, 2018](#)). In addition to resizing the images, anchor boxes were calculated based on the manually labelled bounding boxes, using the *calc_anchors* bash command. This command was built into AlexyAB's Darknet repository.

```

aj@aj-DAD:~/darknet$ ./darknet detector calc_anchors /home/aj/darknet/obj.data -num_of_clusters 9 -width 416 -height 416 -show
num_of_clusters = 9, width = 416, height = 416
read labels from 140 images
loaded      image: 140      box: 8609
all loaded.

calculating k-means++ ...

iterations = 47

avg IoU = 68.59 %

Saving anchors to the file: anchors.txt
anchors =  4,  4,  6,  6, 12,  6, 21,  8, 12, 15, 43, 11, 21, 25, 39, 44, 70, 86

```

Figure 3.22. Bash Command for YOLOv3 Anchor Calculation, used in Network Configuration File ([Monhatkin, 2019](#)).

In addition to mAP, other statistical analyses of the training results included Precision, Recall, and F1 score, shown in Equation 3.1.

$$\begin{aligned}
 \text{Precision} &= \frac{TP}{TP+FP} & TP &= \text{True positive} \\
 \text{Recall} &= \frac{TP}{TP+FN} & FP &= \text{False positive} \quad (\text{Hui, 2019}) \\
 F1 &= 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
 \end{aligned} \tag{3.1}$$

3.3.5 Methods for 2019 Data Collection and Pre-plant Weed Image Set Addition for Training with YOLOv3

Data was collected before the beginning of the 2019 growing season . Due to the high amount of rain received in the study area between April and June 2019, very few plots were planted. Because of this, pre-plant data was collected with weeds present in the field. The reason for this particular data collection was primarily to test the functionality of a newly installed camera gimbal. Small to medium-sized dicot weeds were mainly present in this field. Dicot weeds included *Packera glabella* (cressleaf groundsel), *Ambrosia trifida* (giant ragweed), *Conyza canadensis* (marestail), and *Amaranthus rudis* (common waterhemp). Few grass type weeds were present within the field. Weeds were again labelled as either dicot or monocot. This updated image

set was then trained on 20,000 iterations with the same configuration as the previous training.

In added images to the early season corn training set, corn was at the V2 growth stage and weeds were primarily very small dicot (less than two inches in diameter). Anchors were again calculated, to ensure accuracy with the updated data, and the learning rate was increased to 0.001 from 0.01 in the configuration file. During this training intersection over union (IoU) of the annotated vs. predicted bounding boxes was calculated to predict areas where a certain class of weed was present. The IoU is the ratio of the overlapping area of ground truth and predicted area to the total area (Hui, 2019). This is also known as the Jaccard index, for whom created this equation.

3.3.6 Methods for 2019 Early Season Corn Image Set Creation and Training with YOLOv3

During 2019 early season data collection missions, the Flir Duo Pro R camera was used to take 4000x3000 resolution RGB images at a height of 10 meters. Altitude was set at 10 meters for these flights because of the small height of the corn and small size of weeds present in the fields. As corn was the only crop planted during these data collection dates, corn plots were chosen to make up this dataset. Nearly all of the weeds present in these fields were boradleaf. Thus, testing should show nearly all dicot prediction results. For this training, the following changes were made to the *yolov3_weeds.cfg* file. Image size was increased to 512 x 512, to test whether resizing to a higher resolution would improve training accuracies. Learning rate was also increased from 0.001 to 0.0001. Anchors were also recalculated, using the *calc_anchors* command shown in Figure 3.23.

```

aj@aj-DAD:~/darknet$ ./darknet detector calc_anchors obj_early.data -num_of_clusters 9 -width 512 -height 512

num_of_clusters = 9, width = 512, height = 512
read labels from 78 images
loaded          image: 78          box: 6635
all loaded.

calculating k-means++ ...

iterations = 154

avg IoU = 71.67 %

Saving anchors to the file: anchors.txt
anchors = 7, 8, 9, 13, 16, 14, 11, 24, 20, 24, 20, 45, 34, 31, 42, 60, 55, 125

```

Figure 3.23. Anchor Calculation Result of Labelled 2019 Early Season Corn Dataset.

Anchors were calculated through a k-means clustering technique, shown in Equation 3.2.

$$\begin{aligned}
 b_x &= \sigma(t_x) + c_x \\
 b_y &= \sigma(t_y) + c_y \\
 b_w &= p_w e^{t_w} \\
 b_h &= p_h e^{t_h}
 \end{aligned} \tag{3.2}$$

$$\text{Pr}(\text{object}) * IOU(b, \text{object}) = \sigma(t_o)$$

Within this equation, t_x , t_y , t_w , and t_h are the bounding box coordinate predictions made by YOLO. C_x and C_y is the top left corner of the grid cell that the anchor lies within. P_w and P_h are the width and height of the anchor. C_x , C_y , P_w , and P_h are normalized by the width and height of the image in which the anchor is being predicted. b_x , b_y , b_w , and b_h are the x coordinate, y coordinate, width, and height of the predicted bounding box (Hui, 2019). Finally, $\sigma(t_o)$ outputs the box confidence score.

Anchor box calculation relative to annotated bounding box location is visualized in Figure 3.24.

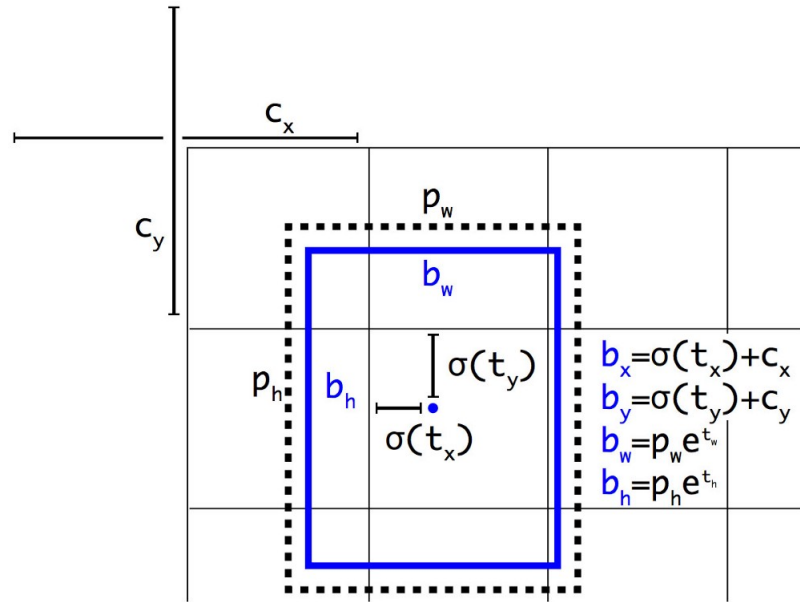


Figure 3.24. Predicted Bounding Box Calculation, based on Annotated Ground Truth Box (Redmon and Farhadi, 2016).

YOLOv3 employs nine anchor boxes per grid cell. The nine calculated anchors for the early season corn image set were 7x8, 9x13, 16x14, 11x24, 20x24, 20x45, 34x31, 42x60, and 55x125. These calculated anchor boxes are quite small, compared to the resized image, but is consistent with the range of labelled bounding box sizes within the image set. Within the configuration file, certain YOLO layers are responsible for predicting a subset of these anchors. For example, the first YOLO layer predicts anchors 7, 8 and 9 because those are the largest boxes and this layer is set at the coarsest scale. The YOLO layer predicts anchor boxes 4,5, and 6, as these are the middle sized anchors and this layer is at a slightly finer scale. The third YOLO layer predicts anchors 1,2, and 3, as these anchors are the smallest of the bunch and this layer is set to the finest scale of the three (Redmon and Farhadi, 2018).

Once the configuration file had been updated, the darknet network was trained over 20,000 iterations, with the *darknet53.conv.74* weights pre-trained on Imagenet. This is an example of transfer learning from a much larger dataset. During initial dataset creation, initial 4000x3000 images were split 2x2, into 4 quadrants, to create 2000x1500 resized images. This made it easier to label weeds within the image, as well as reduce the detail loss during downsampling in half. Network training began on Thursday, June 13th 2019, at 7:00 PM EST. Training finished on Sunday June 16th, at 3:50 AM EST.

3.3.7 Additions to 2019 Early Season Corn Image Set and Configuration Changes for Training with YOLOv3

Before further training could be done, a significant amount of changes had to be made, in order to increase average precision, lower false positive rate, and introduce imagery with monocot weeds present to test network performance with two-class detection. Upon further reading into training the Darknet network on the YOLO model, there was mention of adding images of only objects not wanted to be detected. Each of these images needed a corresponding, empty text file to tell the network there were no labelled objects within. According to [Mohnatkin \(2017\)](#), there should be as many negative samples, i.e. images with empty bounding boxes of non-desirable objects, as the number of labelled images. In addition to adding negative samples to the new image set, every image was resized to a resolution 800x800 to closer match the maximum size that the network could be trained due to GPU memory. This split the original 4000x3000 image into 20 smaller images. In addition to training the network on a 1:1 image size, weeds within the images were much easier to label, images were quicker to annotate, and amount of weeds per image were significantly decreased as smaller sections of the original image were used. Even with splitting the original image into fourths, an average of 78 weeds were manually labelled per

image. Splitting the original image into twenty decreased that average to 26 weeds per image.

This network training newly introduced true positives, false positives, and false negatives at the 25% threshold. True positives in this instance is the total number of average true positives per validation image for each class. False positives are defined in the same manner. False negatives at the 25% threshold are the total amount of weeds that were not detected, which were labelled during manual annotation.

Access to Purdue University’s community supercomputer cluster allowed for network training at larger scale. The Gilbreth cluster utilized for this training is “Purdue’s newest Community Cluster and is optimized for communities running GPU intensive applications such as machine learning” ([Information Technology at Purdue University, 2019](#)). The Gilbreth cluster is comprised of “Dell compute nodes with Intel Xeon processors and NVIDIA Tesla GPUs” ([Information Technology at Purdue University, 2019](#)). A detailed overview of the compute nodes and GPU specifications can be found in Figure A.6. Before the addition 2019 early season corn training set could be trained on the Gilbreth cluster, several changes had to be made to the *YOLOv3.cfg* file. These changes are detailed below:

- batch size: 64
- subdivisions: 16
- angle 30
- random: 1

The angle hyperparameter allows for random rotation of an image over the training iterations by a specified degree, i.e. 30 degrees. This image augmentation improves network performance and reduces loss ([Monhatkin, 2019](#)). The random hyperparameter augments images by resizing them every few batches ([Monhatkin, 2019](#)). Resizing is resolution and timing is random, hence the name of the hyperparameter.

For two class weed detection network training with YOLOv3, the following image sizes were used:

- Experiment in [3.3.4](#): 4000x3000 pixels, resized to 416x416 during training
- Experiment in [3.3.5](#): 2500x1500 pixels
- Experiment in [3.3.6](#): 800x800 pixels.

4. RESULTS AND DISCUSSION

4.1 Results of Data Collection

A total of 28 data collection missions were flown throughout the 2018 and first month of the 2019 growing season. The date, location, UAS flown, sensors used for image acquisition, number of images collected by each sensor, and flight altitude for every mission are listed in Table 4.1.

Table 4.1.
Total Flights and Flight Specifications for 2018 and 2019 Growing Season
Data Acquisition.

Date	Location	Drone	Camera	# Images	Altitude
3/9/2018	DPAC	Mav Pro	RGB- MP		144 30m
3/18/2018	TPAC	Mav Pro	RGB- MP		456 30m
3/23/2018	TPAC	Mav Pro	RGB- MP		782 30m
5/23/2018	DPAC	M600	RedEdge		1350 30m
5/24/2018	TPAC	Mav Pro	RGB- MP		1468 30m
5/31/2018	TPAC	M600	RedEdge		1225 30m
6/6/2018	PPAC	M600	Flir DPR, SL 3PX	F: 2886, S: 3272	30m
6/15/2018	DPAC	M600	Flir DPR, SL 3PX	F: 548, S: 750	30m
6/18/2018	TPAC	M600	Flir DPR, SL 3PX	F: 1944, S: 2158	30m
6/28/2018	PPAC	M600	Flir DPR, SL 3PX	F: 156 , SL: 1308	30m
6/29/2018	TPAC	M600	Flir DPR, SL 3PX	F: 356 ,SL: 448	30m
7/11/2018	PPAC	M600	Flir DPR, SL 3PX	F: 3307, SL: 3795	30m
7/17/2018	DPAC	M600	Flir DPR, SL 3PX	F: 1442, SL: 1357	30m, 50m
7/18/2018	TPAC	M600	Flir DPR		1694 50m
7/23/2018	TPAC	M600	MRE3		6789 30m
7/25/2018	TPAC	M600	Flir DPR, SL 3PX	F: 1081 , S: 2667	50m
8/4/2018	TPAC	M600	Flir DPR		741 50m
8/9/2018	TPAC	M600	Flir DPR, SL 3PX	F: 1454, SL: 3609	30m, 50m
8/22/2018	TPAC	M600	Flir DPR, SL 3PX	F: 1820, SL: 2802	30m, 50m
9/11/2018	PPAC	M600	Flir DPR, SL 3PX	F: 658, SL: 2529	30m
10/11/2018	DPAC	M600	Flir DPR		45 10m
10/18/2018	TPAC	M600	Flir DPR, SL 3PX	F: 24, SL: 401	10m
5/22/2019	ACRE	M600	Flir DPR		147 10m
6/2/2019	TPAC	M600	Flir DPR		1560 5, 10, 30m
6/4/2019	ACRE	M600	Flir DPR		1262 10m
6/11/2019	TPAC	M600	Flir DPR		951 10m
6/21/2019	TPAC	M600	Flir DPR		1175 10m

The Slantrange 3PX multispectral camera saves each band to a stacked TIFF image. When processed with the Slantrange’s Slantview software, each image is broken down into 4 individual band images. Therefore, Slantrange images can be multiplied by 4 within the chart, to coincide with the amount of images acquired from the Micasense RedEdge3. As the Flir Duo Pro R stores thermal and RGB imagery as a single JPG image, which is then extracted to a thermal TIFF image and visual JPG image, the amount of RGB and thermal images taken are the same.

A total of 27,828 RGB images were collected during this period. A total of 23,628 thermal images were acquired during this period. A total of 108,398 single-band multispectral images were taken during this period.

4.2 Manually Annotated Dataset Results

After examining several open-source labelling tools, including OpenLabeling, BBox-Label-Tool, Boobs etc., a labelling tool was finally found, LabelIMG, which provided adequate labelling speed and user-friendly zoom and pan features. LabelIMG, written by Darren Tzutalin, is a “graphical image annotation tool written in Python and using Qt for its graphical interface” (Tzutalin, 2019). Factors such as amount of time required to complete manual annotation, speed and ease of use for network training workflow were mainly responsible for the selection of LabelIMG.

A total of 45 images were manually labelled “weed” as the class for annotated image set developed for DetectNet training in 3.3.1. An example of how these training set images were labelled is shown in Figure 4.1

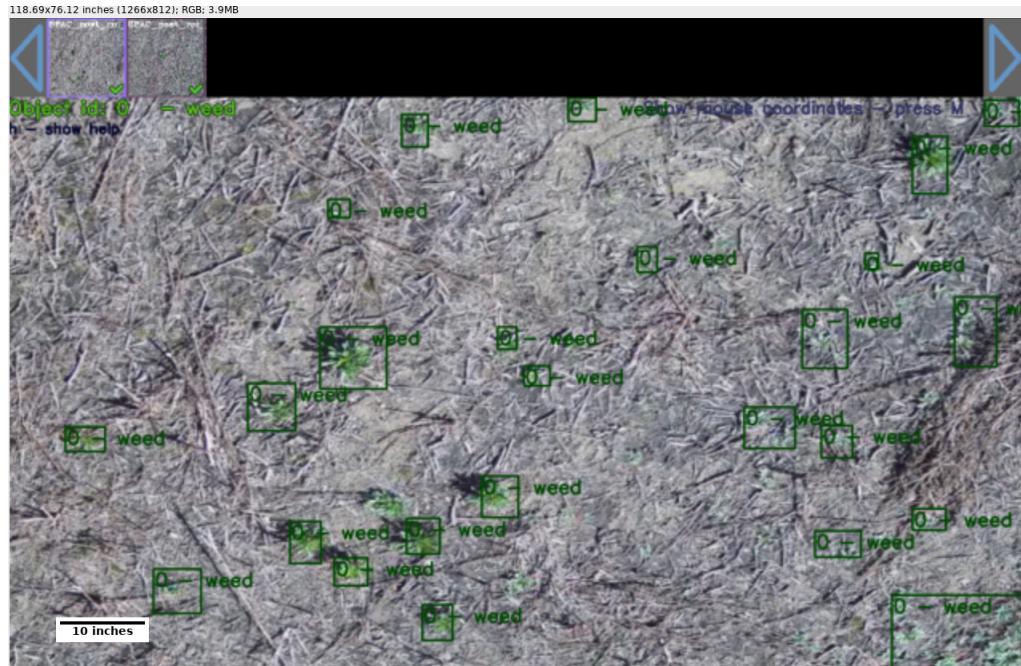


Figure 4.1. Manual Labelling Process for DetectNet Training Set Creation (Training Image Set 1).

For initial training on YOLOv3 with the parsed OpenImages plant subset, in 3.3.2, a total of 2,959 images, along with their corresponding labels were parsed from the OpenImages online database for annotated image set creation. An example of how images are labelled within the OpenImages plant subset is shown in Figure 4.2.

Figure 4.2. Manual Labelling Process for OpenImages Plant Training Set Creation (Training Image Set 2).

Results of mutli-class image set creation in 3.3.3 totaled 1549 images. The following shows the total number of labelled images for each class:

- giant ragweed: 317
- cocklebur: 319
- foxtail: 308
- corn: 303
- nitrogen deficient corn: 303

An example of image labelling in *Yolo_mark* for multi-class dataset creation is shown in Figure 4.3.

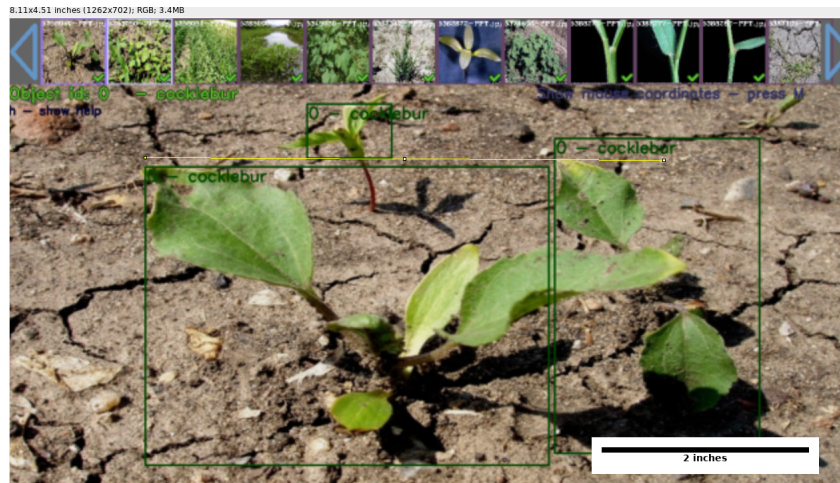


Figure 4.3. Manual Labelling Process for Multiple Weed Classes Training Set Creation (Training Image Set 3).

For two class labelling of UAS-based imagery in 3.3.4 a total of 100 images were manually annotated using the labelImg annotation tool. Within these images a total of 8638 weeds were labelled as either broadleaf or monocot. An average of 86 weeds were labelled in each image.

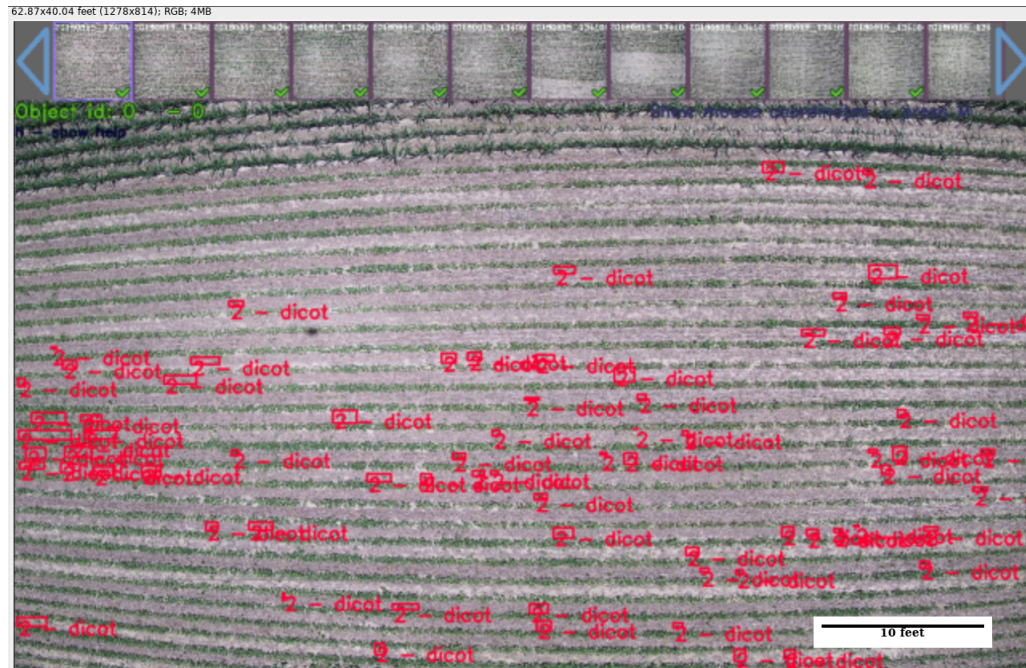


Figure 4.4. Manual Labelling Process for 2018 Two-Class Early Season Image Set Creation (Training Image Set 4).

After this training concluded, an additional annotated 40 images were added to the previous dataset from 2019 pre-plant data collection. A total of 1,801 dicot and monocot weeds were labelled in this addition. The updated dataset included a total of 208 images (100 from subsection 3.3.4; 68 from subsection 3.3.5, and 40 from the 2019 pre-plant data collection campaign) with 12,715 monocot and dicot weeds.

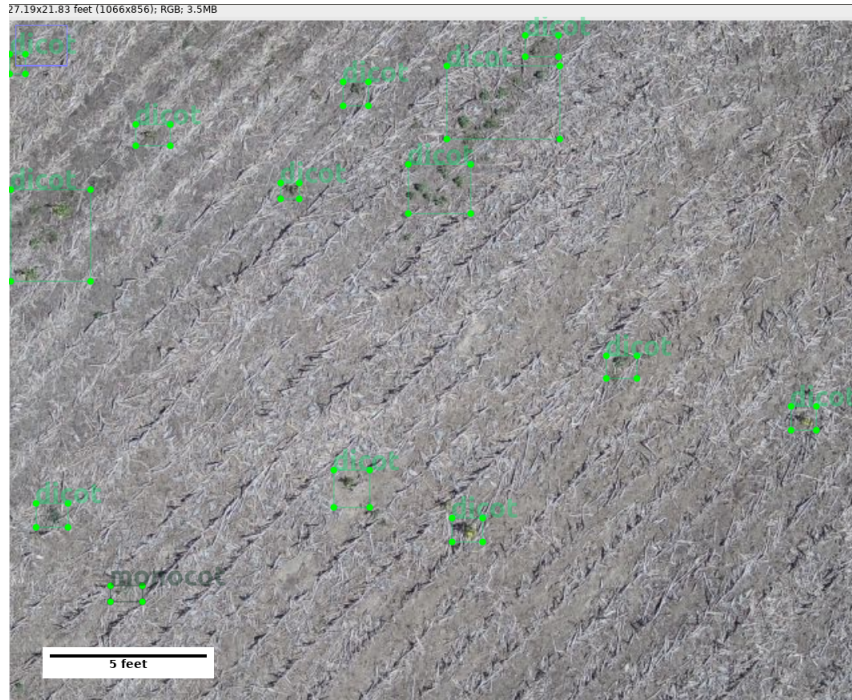


Figure 4.5. Manual Labelling Process for 2019 Pre-Plant Image Addition (Training Image Set 5).

Results of 2019 early-season corn annotated image set in 3.3.5 totaled 100 labelled images, with a total of 7,795 monocot and dicot bounded. Of these, 78 images were used for training and 22 were used for validation. This is slightly over the 80/20 threshold commonly used in training and validation dataset splits [Partel et al. \(2019\)](#) ([dos Santos Ferreira et al., 2017](#)). An example of labelling done for this image set is shown in Figure 4.6.

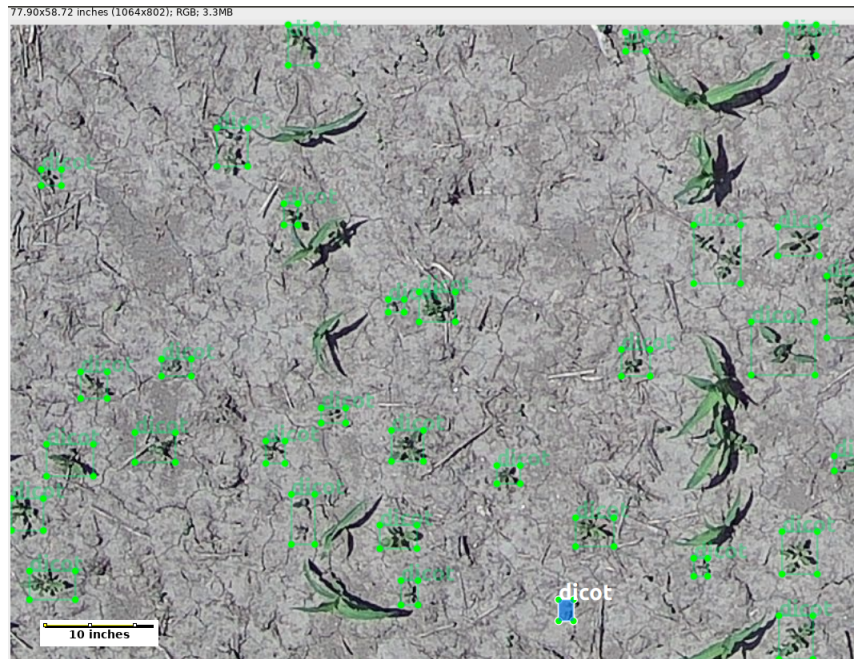


Figure 4.6. Manual Labelling Process for 2019 Two-class Early Season Image Set Creation (Training Image Set 6).

There were 66 additional images to the early season corn annotated image set creation, detailed in [3.3.7](#). A total of 5,147 instances of monocot and dicot weeds were labelled in this addition. An example of the result of manual labelling process for this image set is shown in [Figure 4.7](#).

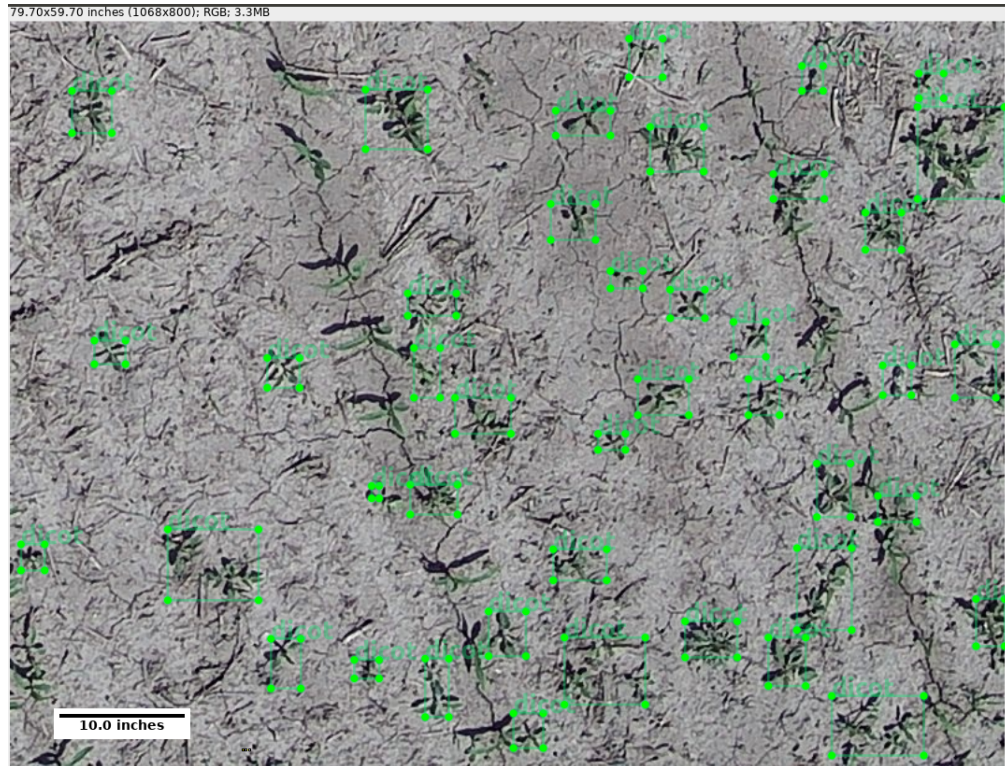


Figure 4.7. Manual Labelling Process for 2019 Early Season Image Set Addition (Training Image Set 7).

A total of 166 positive samples (labelled images) were added to this image set and 166 negative sample (non-labelled), weed-free corn images were added. A total of 262 of the 332 total images were used for training, while 70 images were set aside for validation. The training set comprised of 131 labelled and 131 non labelled images, while 35 labelled and 35 non labelled images comprised the validation set. An example of the negative samples used in this image set is shown in [Figure 4.8](#)

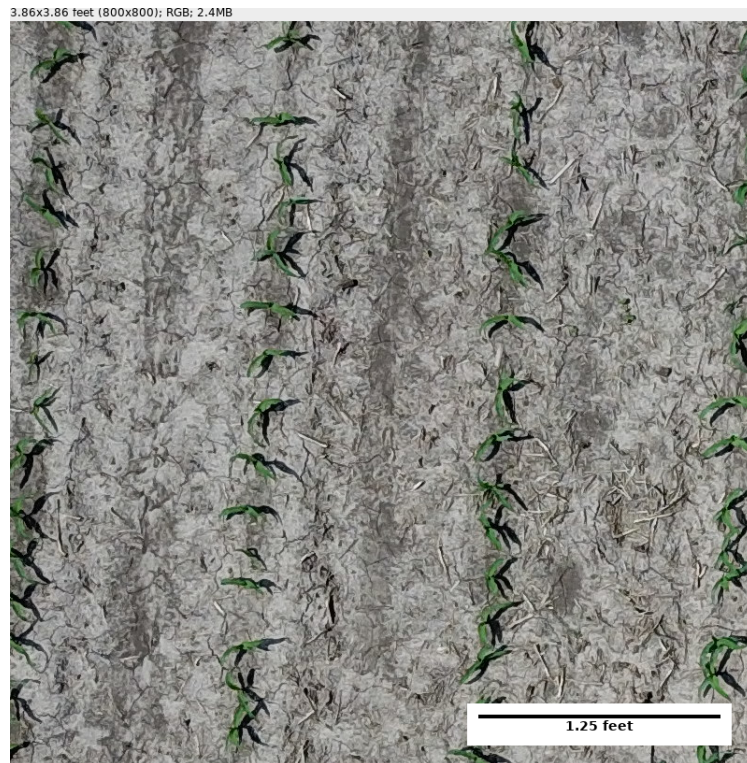


Figure 4.8. An Example of the Weed-free Negative Samples Added to Training Image Set 7.

4.3 Object Detection Network Training Results

4.3.1 Initial Results with NVIDIA DIGITS

After 300 epochs, or training steps, the detection accuracy was approximately 45 percent. These results are visualized on the graph in Figure 4.9. Each training sequence was split up into 50 epochs, for sake of time and memory usage.

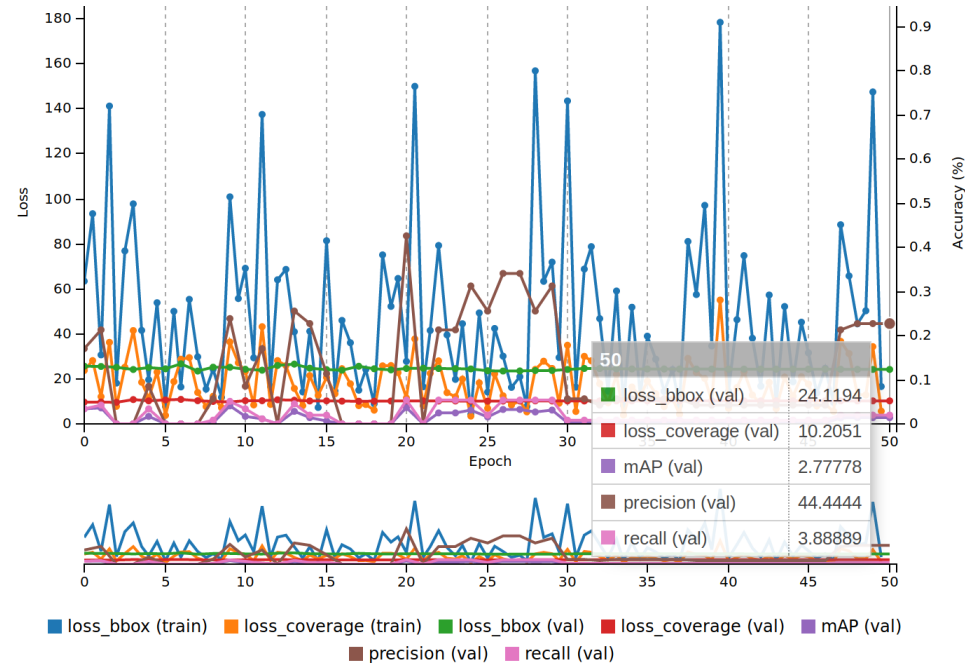


Figure 4.9. Results of DetectNet training with Training Image Set 1 Over 300 Epochs.

In addition to a 45 percent precision value, the mAP and recall values were quite low, at 12 percent and 8 percent respectively. The poor results can be explained by the fact that an earlier study reported a minimum cutoff for obtaining best DetectNet network performance requires a minimum of several hundred images for training (Warden, 2017). However, only 45 manually labelled images were used for this network training.

4.3.2 Initial YOLOv3 Results with OpenImages Dataset

After running the trained weights on the OpenImages training set for 5000 iterations, only 69 percent of the images tested from the test set successfully deployed a bounding box. The accuracy of the objects detected, in the images where detection worked, ranged from 54 to 98 percent (shown in Figure 4.10). This means that 31

percent of tested images weren't similar enough to the images used in the annotated training set, or there was a problem with the network over training (only able to detect a specific range of object sizes) or under trained (not able to predicted a certain range of object sizes).

```

Enter Image Path: /home/aj/01Dv4_ToolKit/01D/DataSet/test/Plant/fff23a84a63a1acd.jpg: Predicted in 0.165484 seconds.
Enter Image Path: /home/aj/01Dv4_ToolKit/01D/DataSet/test/Plant/ffdce0c23890ea75.jpg: Predicted in 0.050009 seconds.
Plant: 83%
Enter Image Path: /home/aj/01Dv4_ToolKit/01D/DataSet/test/Plant/ffcbacdb03acf225.jpg: Predicted in 0.049960 seconds.
Enter Image Path: /home/aj/01Dv4_ToolKit/01D/DataSet/test/Plant/ffb251259eb97e04.jpg: Predicted in 0.049915 seconds.
Plant: 54%
Enter Image Path: /home/aj/01Dv4_ToolKit/01D/DataSet/test/Plant/ffb57bb12eacea73.jpg: Predicted in 0.050043 seconds.
Plant: 93%
Enter Image Path: /home/aj/01Dv4_ToolKit/01D/DataSet/test/Plant/ffb7ccce9283da67.jpg: Predicted in 0.065184 seconds.
Plant: 86%

```

Figure 4.10. Detection Results of Training Image Set 2 on Testing set, with Final YOLOv3 Training Weights.

Once the YoloV3 was tested in this manner as a proof of concept, training and testing could be done with multiple weed classes.

4.3.3 Results of Multi-class Detection with YOLOv3

After training YOLOv3 on the OpenImages plant subset, a custom, annotated image set was created of single weed instances. For example, the giant ragweed class was composed of images with only giant ragweeds as the objects therein. Images across all classes featured a seedling or early stage corn or weed, depending on class. As images used for this multi-class training set had been previously acquired and were much faster to label than UAS-acquired imagery, it was decided to label and train with these ground-based images first.

After training on 50,000 iterations, average precision of each class is given below:

- giant ragweed: 78% ap
- cocklebur: 68% ap
- foxtail: 84% ap

- corn: 73% ap
- nitrogen deficient corn : 48% ap

While final network training weights were tested on a test set of images not used for training or validation sets, test images were of similar, small weeds for each class and no test image had more than 5 examples of a particular class of weed. Also, test set images only pertained a single weed class, as shown in Figure 4.11.



Figure 4.11. Giant Ragweed Class Detection Results from YOLOv3 Network Training on Training Image Set 3.

There were no images with multiple weed classes or a mixture of weeds and corn, weeds and nitrogen deficient corn, etc. These factors likely contributed to high class average precision.

Results of testing on the foxtail class is shown in Figure 4.12.

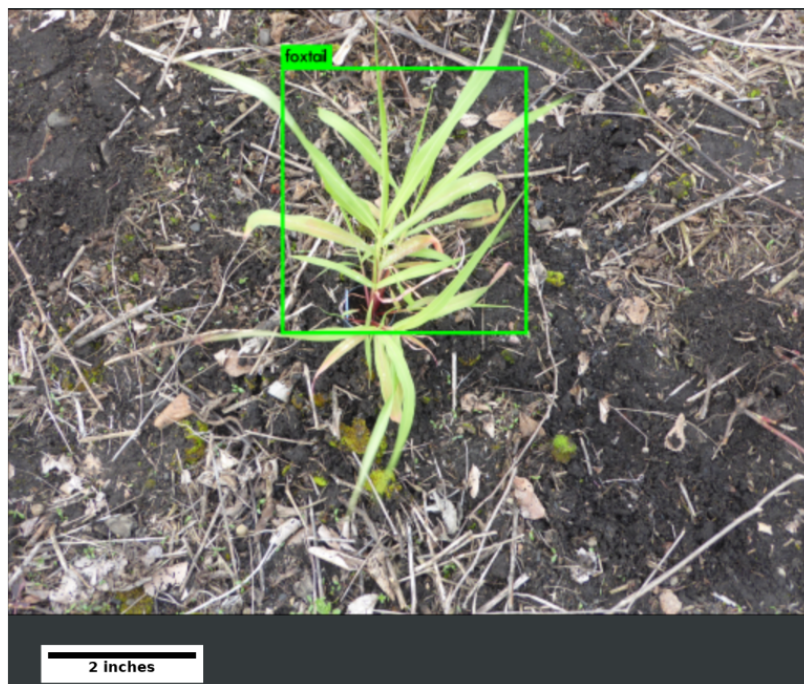


Figure 4.12. Foxtail Class Detection Results from YOLOv3 Network Training on Training Image Set 3.

Results of testing on the cocklebur class is shown in Figure 4.13.

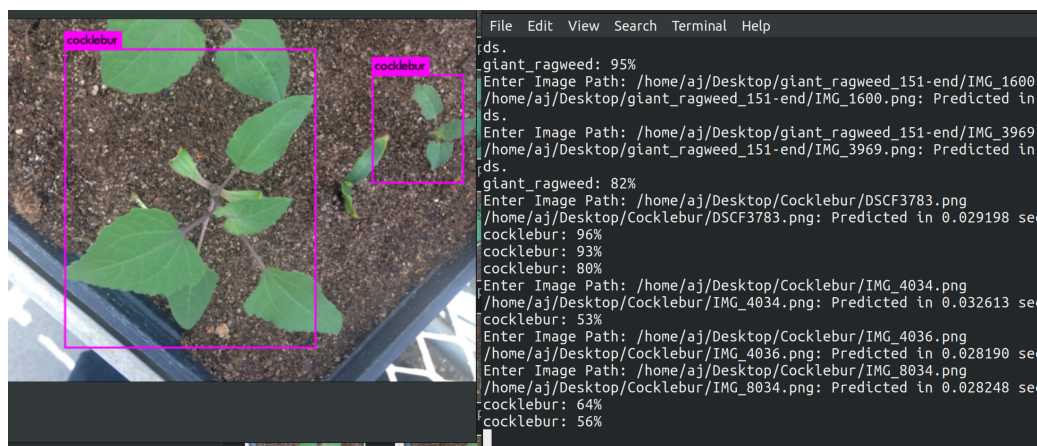


Figure 4.13. Cocklebur Class Detection Results from YOLOv3 Network Training on Training Image Set 3.

As shown in the following training on UAS-based imagery, average precision, along with other benchmark measures applied were much lower.

4.3.4 Results of Two-class UAS-based Image Set Network Training on with YOLOv3

After 20,000 iterations, this training resulted in an average loss of 4.7. Accuracy at detecting dicot was a meager 3.63 percent and monocot was 9.88 percent. A major contributor to this poor performance was the high amount of false positives for both classes. For dicot, there were an average 103 true positives to 499 false positives, over the validation set. For monocot, there were an average of 188 true positives to 224 false positives, over the validation set. The mean average precision of objects with an IoU over 50 percent (mAP@0.50) was poor at 6.76 percent. For a 25 percent threshold, the average precision was 23 percent, recall was 15 percent, and the F1-score was 0.18. Testing results with the final network training weights for this image set are shown in Figure 4.14 below.



Figure 4.14. Monocot and Dicot Detection Results from YOLOv3 Network Training on Training Image Set 4.

4.3.5 2019 Pre-plant Weed Image Addition Network Training Results with YOLOv3

The results of this training continued to be poor, with an average loss of 7.77 percent. Accuracy of detecting dicot weeds was 4.22 percent, while monocot accuracy was 9.32 percent. Again a high number of false positives in both classes was a major factor in poor network performance. For dicot, average true positives per validation set image were 133 and false positives were 618. For monocot, average true positives per validation set image were 101 and false positives were 199. The mean average precision of 6.8 percent at the 50 percent threshold. At a 25 percent threshold, precision was 22 percent, recall was 14 percent. and the overall F1-score was 0.17.

Average IoU at this threshold was 13.64 percent. At this training stage, live mAP tracking was newly discovered within [Monhatkin \(2019\)](#)'s repository and will be used during further network training result interpolation.

Results of this training are shown in Figure 4.15

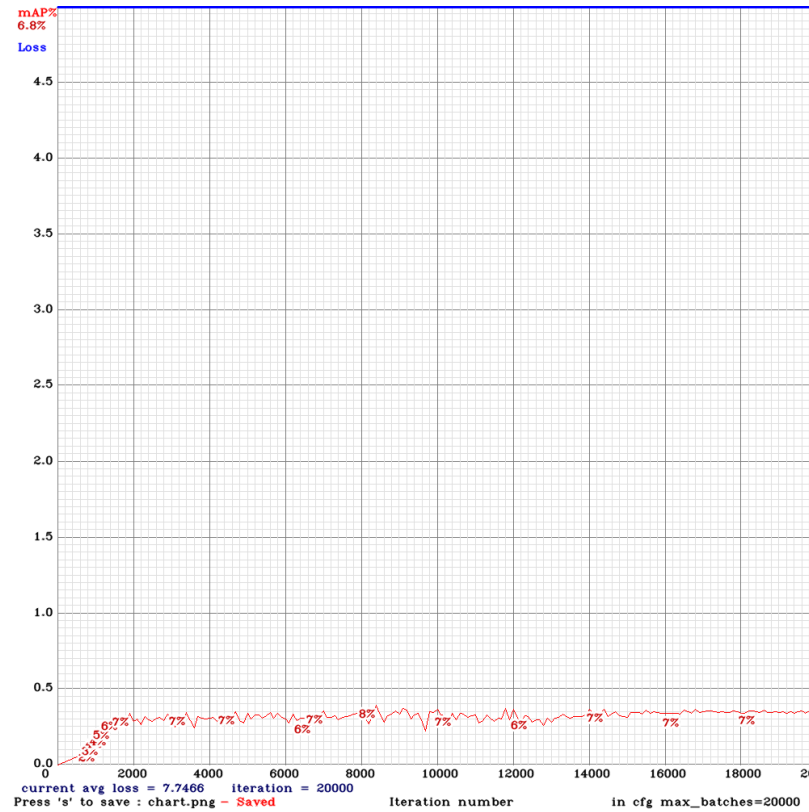


Figure 4.15. Average mAP and Loss Score Results from YOLOv3 Network Training on Training Image Set 5.

4.3.6 Early Season Corn Image Set Training Results with YOLOv3

Performance measures of the network training results, after 20,000 iterations, continued to be poor. mAP at a 50 percent IoU threshold was 10.21 percent. The average loss of this training was 3.9 percent.

As there were no monocot weeds present in this training set, monocot accuracy was zero percent. The average precision (ap) or accuracy, average true positives (TP) found per validation image, and average false positives (FP) of the dicot weed class is given below:

- ap- 20.42%
- TP- 15
- FP- 44

The following results were given for a 25 percent threshold:

- precision- 32 percent
- recall- 17 percent
- F1 score- 0.22

The average mAP and loss chart for early season corn network training results is shown in Figure [4.16](#) below.

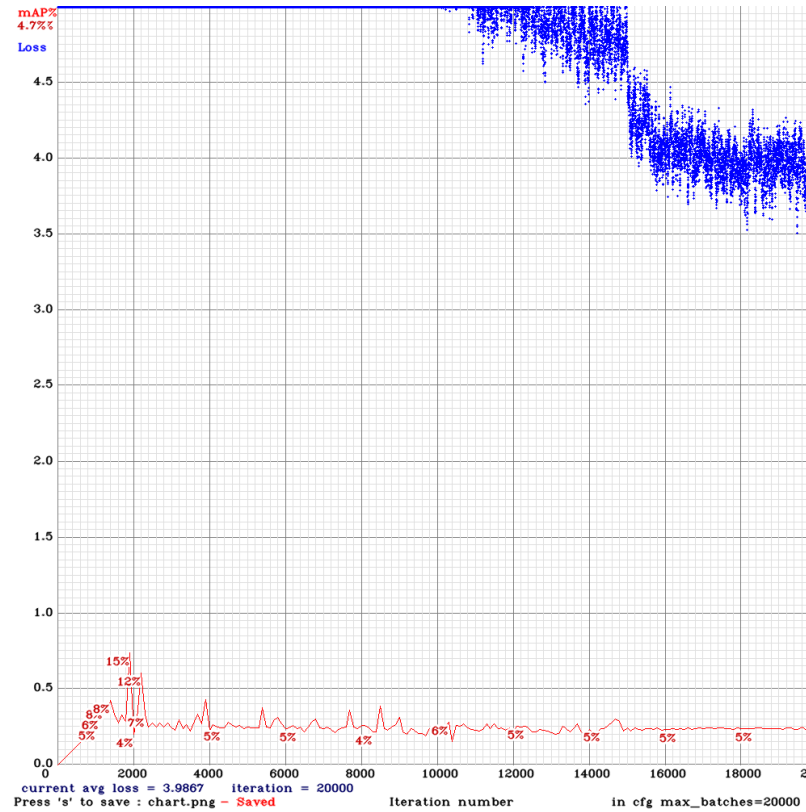


Figure 4.16. Average mAP and Loss Score Results from YOLOv3 Network Training on Training Image Set 6.

Precision in the context of this study meant the amount of weeds correctly identified (true positive) divided by the amount of weeds correctly identified plus the amount of weeds incorrectly identified (false positive). Recall meant the amount of weeds correctly identified divided by the amount of weeds correctly identified plus the amount of weeds not identified that should have been (false negative). The F1 score is calculated with two times the precision times recall divided by precision plus recall. In other words, it takes a weighted average of precision and recall to measure overall network performance. In terms of weed detection, the F1 score gives an idea on how well the trained network weights are able to correctly detect a weed's class and how accurately it is able to find every instance of a weed within an image. A

large contributing factor for this poor statistical performance was the high amount of false positives predicted. True positives were 178, while false positives were 374. False negatives were also quite high at 900. Average IoU at the 25 percent threshold was 20.53 percent. When the IoU threshold was increased to 50 percent, mAP was 4.68 percent. However, loss was slightly lower during this training, at 3.98 percent. The chart of average mAP and loss results for this network training is shown in Figure 4.16.

An example of testing results on the test image set is shown in Figure 4.17 below.

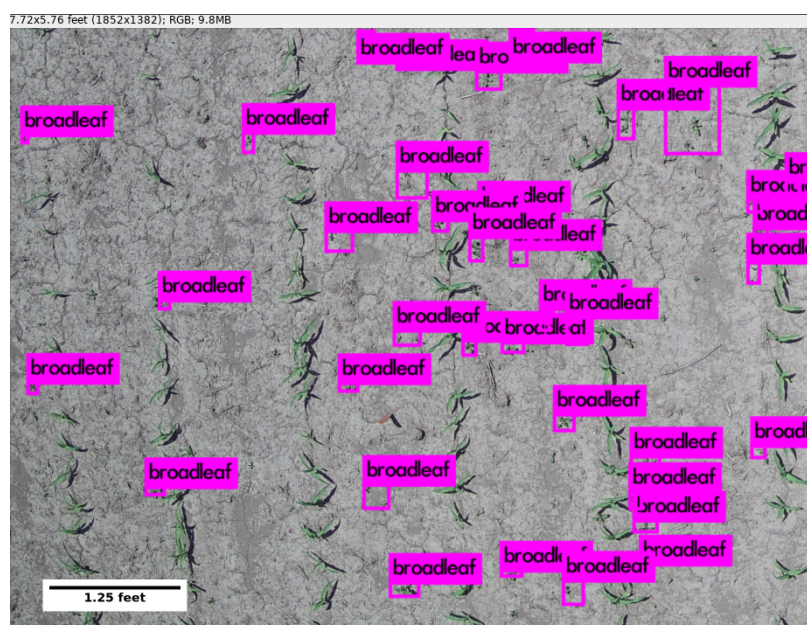


Figure 4.17. Dicot Detection Test Set Result from YOLOv3 Training on Training Image Set 6.

4.3.7 Training Results for 2019 Early Season Corn Image Set Additons with YOLOv3

Additional images, collected in early season 2019 corn plots, were added to the previous training set. The YOLOv3 network was again trained over 20,000 iterations.

Unlike the previous image set, several instances of monocot weeds were found while labelling this addition, thus the accuracy of monocot weeds will only apply to the additional 66 images. Results of this training are given below:

- avg. loss- 0.522%
- mAP at 50% IoU- 15.59%
- dicot ap- 31.38%
- monocot ap- 58.31%
- dicot TP- 410
- dicot FP- 99
- monocot TP- 167
- monocot FP- 38
- 25% threshold precision- 80%
- 25% threshold recall- 25%
- 25% threshold F1 score- 0.38
- 25% threshold TP- 577
- 25% threshold FP- 137
- 25% threshold FN- 1249

Results of additional 2019 early season corn training set is shown in [Figure 4.18](#).

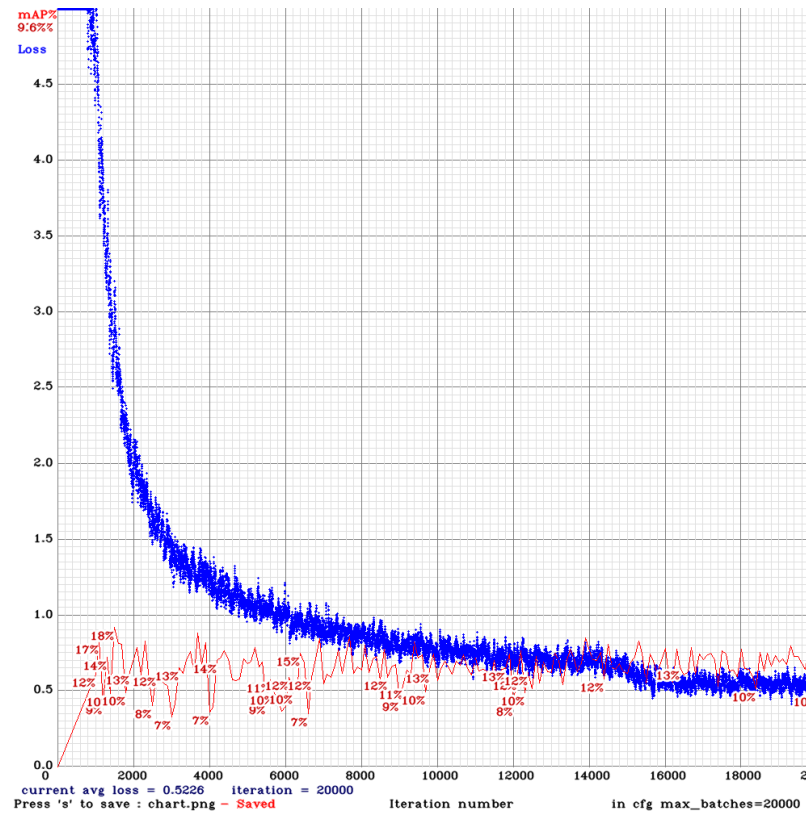


Figure 4.18. Average mAP and Loss Score Results from YOLOv3 Training on Training Image Set 7.

This image set was then trained over 20,000 iterations on the Purdue Gilbreth supercomputer cluster. The results of this training are given below:

- avg. loss- 0.98%
- dicot ap- 45.13%
- monocot ap- 65.37%
- mAP @ 50% IoU threshold- 22.57%
- precision @ 25% threshold- 86%
- recall @ 25% threshold- 47%

- F1 score @ 25% threshold- 0.55
- TP @ 25% threshold- 515
- FP @ 25% threshold- 268
- FN @ 25% threshold- 576
- avg IoU @ 25% threshold- 44.14%

Combined class true positives and false positives were only calculated at the 25 percent threshold for this training.

The results of Gilbreth cluster network training are shown in Figures [4.19](#) and [4.20](#).

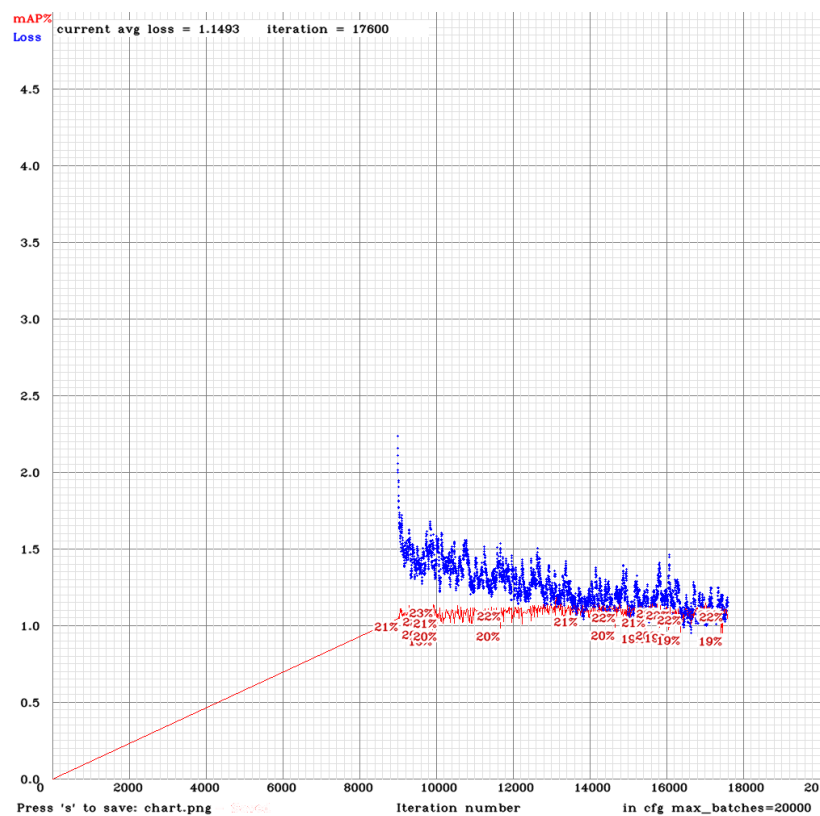


Figure 4.19. Aveage mAP and Loss Score Results- 9,000-17,000 Iterations, from YOLOv3 Training on Training Image Set 7 Using the Gilbreth Cluster.

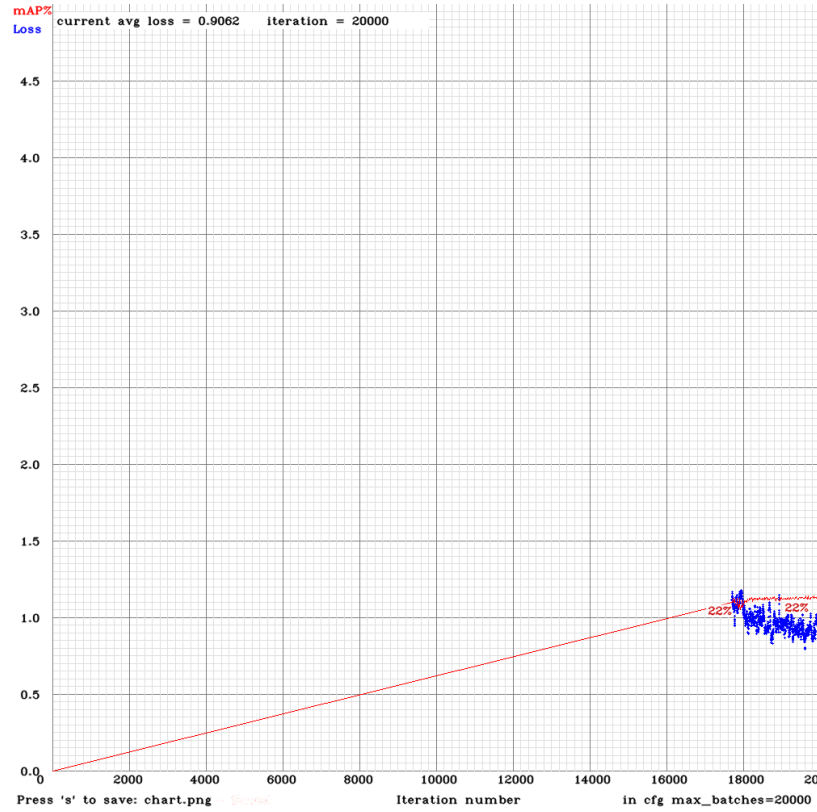


Figure 4.20. Aveage mAP and Loss Score Results- 17,000-20,000 Iterations, from YOLOv3 Network Training on Training Image Set 7, using the Gilbreth Cluster.

The same test set was used to test the accuracy of the final network weights, trained on the cluster. Dicot class prediction accuracy on a test set image, from the final YOLO weights is shown in Figure 4.21.

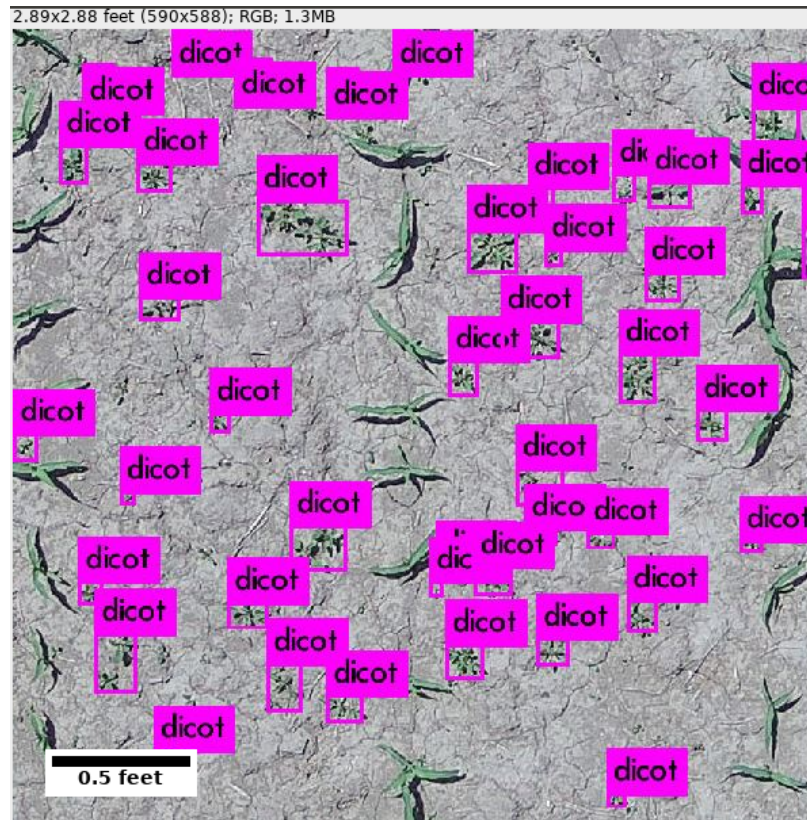


Figure 4.21. Dicot Detection Result from YOLOv3 Network Training on Training Image Set 7, using the Gilbreth Cluster.

Challenges and problems faced during image collection, annotated training set creation, and weed detection network training will be discussed in the next section.

4.4 Discussion

4.4.1 Discussion on Data Collection

There were several factors that influenced data collection timing and location. In order to collect reliable data, weather conditions during UAS flights needed to be clear, with little to no wind. Cloudiness, high wind, and precipitation all have a detrimental effect on the collected imagery. The UAS is also incapable of flying

in winds over 18 knots, or in any type of precipitation, as it is not water resistant. Distance to the PPAC and DPAC locations also contributed to the limited amount of data collection dates at those locations. The distance from Purdue University to DPAC was 208 miles round trip and 4.5 hours. Data collection done at DPAC required an entire dedicated day. PPAC was 151 miles and a 3.5-hour round trip from Purdue University, Data collection done at PPAC also required a entire dedicated day.

The research areas chosen for this study were in conjunction with the Dr. Bryan Young's Weed Science lab at Purdue University. TPAC, DPAC, and PPAC plot location data collection was done over Dr. Young's or Dr. Bill Johnson's experimental research plots. The preplant and early season corn collection at ACRE were not associated with the previous research plots mentioned, and were included in this research due to the late 2019 planting of DPAC and PPAC plots, flown during the 2018 growing season. As the areas selected were part of Purdue University Weed Science research, weed populations in may be considerably higher than in commercial corn and soybean fields throughout the north-central Indiana region.

In addition to challenges faced from weather and distance to research area, there were also mechanical issues with the original gimbal that the Flir Duo Pro R camera was attached to. For the first three data collection dates, the Gremsey T3 gimbal worked correctly, it held the camera firmly in place at nadir and did not list from locked position. As such, imagery collected with the attached camera was uniform in direction and free of distortion from vibration or shadow. However, during data collection at PPAC on July 28th, 2018, the Gremsey gimbal started to malfunction and cause the Flir Duo Pro R to move in the pan axis (left to right) from its locked position. This caused collected imagery to be positionally distorted as the camera moved out of its intended position. This movement also caused the tilt motor on the gimbal to be overexerted, as it had to hold the camera at nadir during continuous pan movement. This caused vibration distortion in several images. Attempts at fixing the pan movement problem included increasing the stiffness on the pan rotation motor in the Gremsey GTune application, extending the GNSS cable attached to the Flir

Duo Pro R, so as to not come in contact with the gimbal, re-balancing the camera with small, lead weights, and finally securing the gimbal's camera mount frame to the M600 UAS with zip ties. While none of these options completely eliminated the pan movement problem, zip tying the camera mount to the UAS frame proved to be the most effective at reducing movement. It was found that this gimbal had a malfunctioning pan motor. For 2019 data collection, a Yangda Sky Eyo Duo three-axis gimbal was mounted to the M600, in place of the Gremsey T3 gimbal. There were no problems with this new gimbal, for the four data collection missions performed in 2019.

These factors that influenced UAS-based image collection similarly affected the outcome of annotated weed detection network training datasets.

4.4.2 Discussion on Annotated Training set Creation

The major limiting factor to size of the annotated weed detection network training datasets was the amount of time required to manually label all of the instances of weeds within a particular image. This was especially prevalent in labelling of UAS-based imagery, where there were over 300 individual examples of weeds labelled in some images. Initial training set creation on early growing season 2018 imagery, discussed in 3.3.4 took over 40 hours to completely label. With the addition of image splitting in 3.3.5, more images could be added to the training set, with less time spent on manual labelling per image. It was found that the ideal image split size for this research was 800x800. This split the original 4000x3000 pixel image into 20 smaller, more manageable pieces. This also proved better for network training, as the amount of memory required was too much to train at any image size over 800x800 pixels without resizing, which leads to loss of detail during training.

Another factor that limited the size of annotated network training sets was the heavy weed infestation within the research areas that occurred after V6 growth stage in corn and soybeans. After this growth stage, occurring on or after June 18th, 2018

at TPAC, DPAC, and PPAC. This infestation made it difficult to label individual weed examples. While a small, initial annotated training set was made with images taken after the 2018 V6 growing stage, it was not added to the extent of the research covered in this thesis due to sub-par and widely varying detection. In testing the final weights of DetectNet and YOLOv3 network training with this mid-late growing season annotated image set, there were more false positive detections than true positive. This meant that crops were being detected as either monocot or dicot weeds more than the weed instances themselves. An example of this heavy weed infestation is shown in Figure 4.22.



Figure 4.22. Example of heavy weed infestation at TPAC field site, July 6th, 2018.

Aside from splitting the training set images into 800x800 pixels, adding negative samples to the training set, as discussed in section 4.2, helped reduce network training loss from 4.7 percent loss to 0.52 percent loss with the same 2019 early season corn training set.

Discussion on the results of training the various annotated image sets will be detailed in the next section.

4.4.3 Discussion on Weed Detection Network Training Results

The Initial detection results of DetectNet training in NVIDIA DIGITS shows that the larger monocot and dicot examples within the image are not getting detected. This is due to the maximum bounding box size in the DetectNet architecture not being large enough to fit these weeds inside. Another problem with this result is that the network seems to distort the color of the output image in a way that masks some of the greenness in the weeds. A possible reason for this is that hue, saturation, and value were utilized for image augmentation within the DetectNet configuration file, but might have caused testing images to be distorted as well, as shown in [Figure 4.23](#).

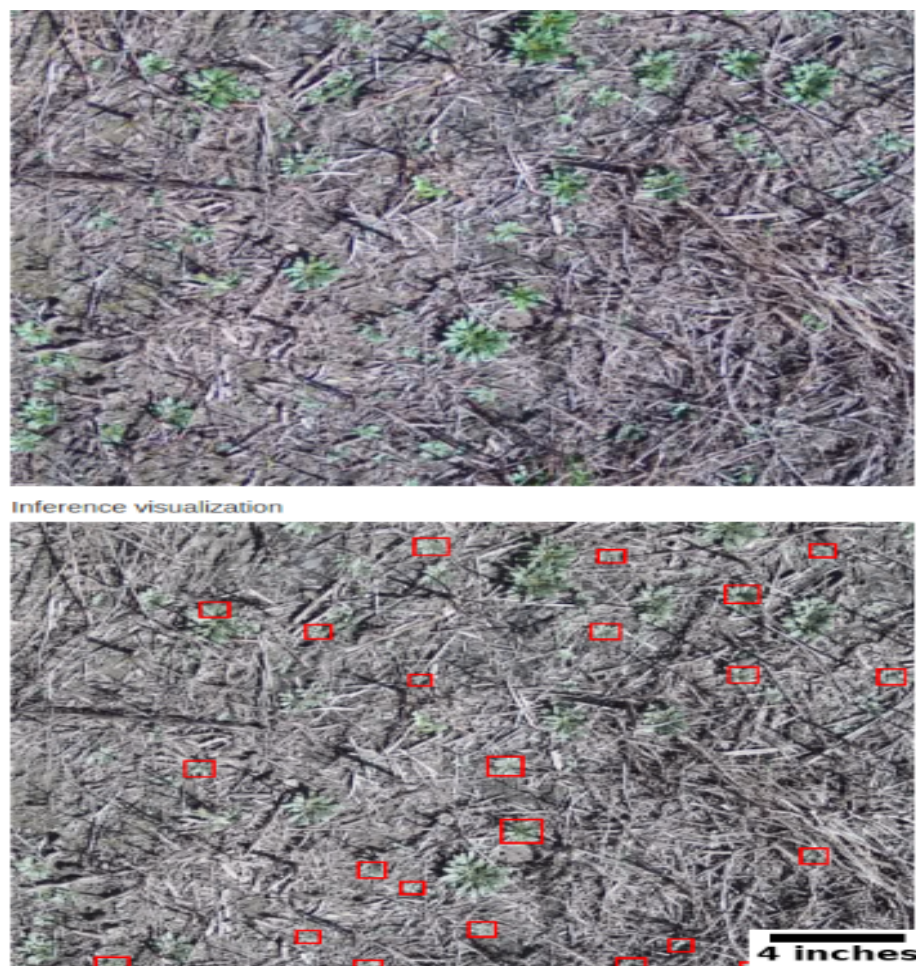


Figure 4.23. Weed Class Detection on Test Set Image as Result of DetectNet Training on Training Image Set 1.

During the initial YOLOV3 network testing, the final weights were tested on the same post-harvest images as used for DetectNet testing in DIGITS. The final network weights of YOLO network training produced a very similar result to Detectnet testing, around a 45 percent detection accuracy. However, the YOLOv3 network was not trained on the post-harvest images. YOLO detection network architecture allowed for larger size monocot and dicot weeds to be detected than what DetectNet allowed, as shown in Figure 4.24. This advantage gave better results on larger image sets, such as OpenImages and multi-class training.

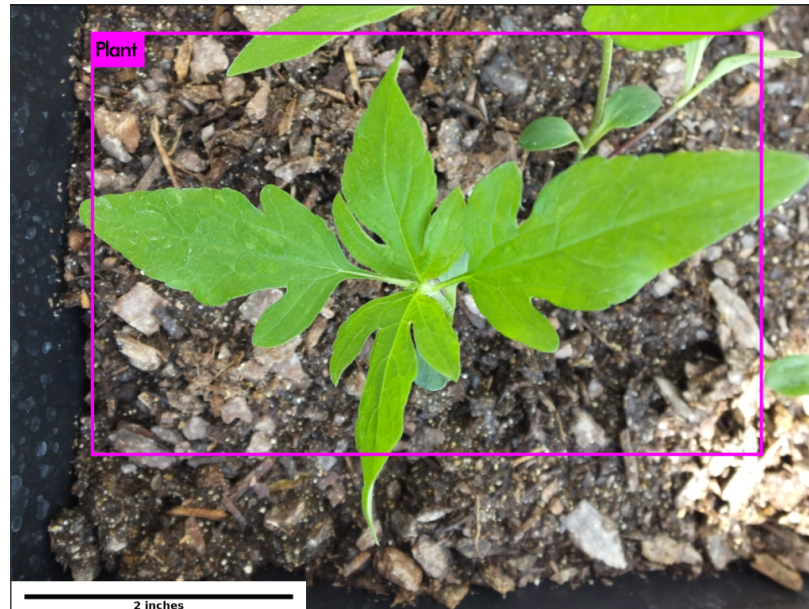


Figure 4.24. Plant Class Detection on Test Set Image as Result of YOLOv3 Network Training on Training Image Set 2.

When training the YOLOv3 network on multiple weed classes, a problem arose where not all instances of a particular class were detected within the image. Even after the training of 1200 labelled images, over 10,000 iterations, the YOLO network still had difficulty detecting multiple instances of the predicted weed class. This was especially prevalent when that instance was obscured or physically different than the other objects within the image, such as shown in Figure 4.25.

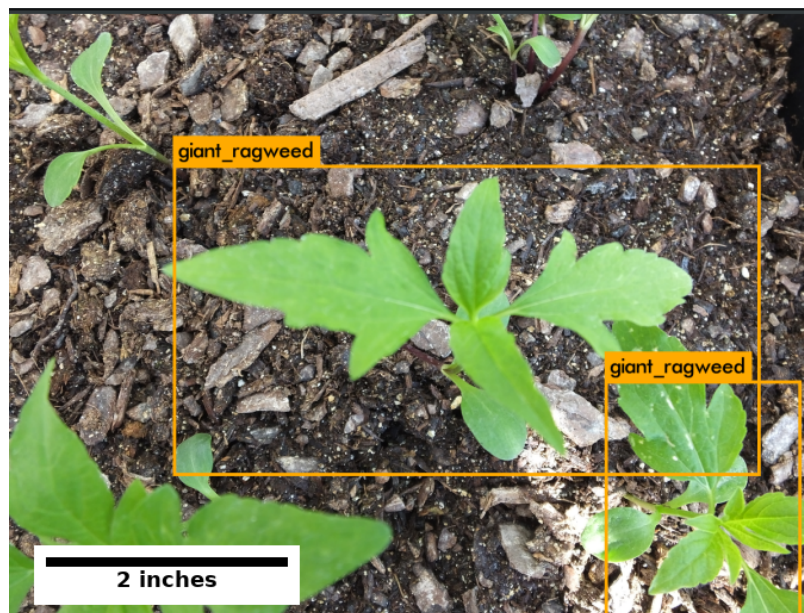


Figure 4.25. Problem Detecting Multiple Weed Class Instances in Test Set Image after YOLOv3 Network Training on Training Image Set 2.

A problem that was common within the literature and other studies undertaken, was the high number of false positives when training on image sets in a complex environment, such as a cornfield. In agronomic terms, this meant incorrectly detecting cash crop within the field, such as corn, soybeans, sugar beets, and others, as weed (Sa et al., 2018), (Pérez-Ortiz et al., 2015). Another case, likely more commonly occurring than incorrectly identifying crop as weed, was incorrectly identifying weed class, either monocot as dicot or visa versa. While this problem represents inaccuracies in the network training, it is remedied in the field through mixing herbicides in the sprayer tank. This is done to gain control of both dicot and monocot weed species in a single application. For example, in Roundup Ready 2 Xtend® soybeans, glyphosate and dicamba can be mixed in the sprayer tank to provide a more thorough coverage of monocot and dicot weed species.

There were several issues with two-class training of UAS-based image sets.

The initial UAS-based image set network training only produced a detection accuracy of 3.63 percent for dicot and 9.33 percent for monocot. As the annotated training set created for this training included monocot and dicot weeds interspersed in corn and soybean plot areas, it was possible that the similarity between labelled dicot weeds and soybeans were too much for the network to distinguish between. Likewise, labelled monocot weeds and early growth stage corn might have shared too many similarities. Reasoning for this assumption was the high number of false positives that the network training resulted.

With the addition of pre-plant weed imagery from 2019, the hypothesis was that a wider variety of weed images without crop or heavy background occlusion would allow the network to better distinguish between crop and weed. However, the training results continued to be poor, with less than a one percent improvement in dicot detection accuracy (4.22%) and a .01% decrease in monocot detection accuracy (9.32%). Also, with average loss staying relatively constant over the last several thousand iterations and average mAP score stagnating over the same period, it is safe to assume that increasing the amount of training iterations would not improve overall network performance. In the context of this research, the results of this network training showed it was clear that a comprehensive, highly variable training set would not produce desirable results (high class detection accuracy). The next experiment involved building a dataset tailored to a specific growth stage period. The next annotated dataset discussed was tailored to early season corn fields with small sized weeds, VE through V3 growth stage.

Results of 2019 early season corn image set training were slightly improved from the previously trained, UAS-based image set. While no monocot weeds were labelled in this training set, dicot weed detection accuracy was 20.42 percent. Although there continued to be a high number of average false positives resulting from this network training, the accuracy of dicot detection was substantially improved. It is important to decrease the amount of false positives as much as possible, as the final network training weight used in real-time detection could incorrectly identify the corn crop as

a weed. While applying herbicide to a tolerant crop, such as glyphosate to Roundup Ready® corn would not hurt the crop, it is still a waste of product. A contributing factor to this accuracy increase was that the training set was only comprised of early season corn imagery, with small dicot weeds. Despite low statistical significance in mAP score after training of the 2019 early stage corn image set, network testing on a unique test set, and reducing image size, proved to have a more accurate prediction of dicot weeds. This testing set also had a total of 100 labelled images, with early season corn and small, dicot weeds. Images in this test set were also taken at TPAC and ACRE, but of different fields than images used in the training set. For the next round of training, image size was reduced further, the amount of images in the dataset was increased, and images of corn without weeds were added to the dataset, in order to show the network what should not be predicted. However, the high number of false positives still leads to the question of how well the network is able to recognize the variation in some dicot weeds to the small corn crop. To test this, 66 additional images were added to this training set, which featured solely monocot weeds in early stage corn, as well as a mixture of monocot and dicot weeds.

Training on the bolstered 2019 early season corn dataset, with additional labelled images achieved improved results over previous training. The major reason for this improvement was the introduction of negative samples to the training set. While helping to reduce average loss, as detailed in the annotated training set creation discussion section, mAP score also increased by 5%, dicot detection accuracy increased 11%, to 31.38% and monocot weed detection was nearly 50% higher than initial UAS-based, two class training, at 58.31%. At a 25% threshold, combined precision of both classes was 80%. This was close to 50% higher than the precision at the same threshold from the original 2019 early corn training set. Another contributing factor to improved network training results was the significantly reduced number of average false positives. In the previous training, average false positives outnumbered average true positives 3:1. With this training, average true positives for both monocot and dicot classes outnumbered average true negatives 4:1. Introducing negative samples

to this training set helped reduce the number of average false positives by feeding the network explicit examples of which regions not to predict anchor boxes for. In other words, the negative samples provided instruction to the network on which objects (early season corn) not to predict within the image.

The 2019 early season corn addition image set was then trained on the Gilbreth supercomputer cluster, as discussed in 3.3.7. With access to training on two, twelve GB GPUs, as opposed to one, six GB GPU on the Alienware R3 laptop, batch size could be increased and subdivision size could be decreased. Previous training on the laptop needed the batch size set to 1 and subdivisions set to 64, in order to not exceed GPU memory when training the YOLOv3 network at 800x800 pixel image resolution. On the cluster, batch size could be set to 64 and subdivisions could be set to 16. The training time over 20,000 iterations took 108 hours to complete on the laptop and 62 hours to complete on the cluster. A possible reason why the cluster training wasn't nearly as fast as it theoretically should have been is that with the sheer amount of labelled objects (monocot or dicot weeds) in the annotated image set, a bottleneck could have occurred during training with anchor box prediction or feature extraction. Also, as the processing job on the Gilbreth cluster had to share GPU memory with other jobs running on the subcluster (Gilbreth has 5 sub clusters to choose from), it is possible that the weed detection network training job was paused or slowed when GPU space was low. After training over 20,000 iterations the Gilbreth cluster, with the 2019 early corn image set addition, detection accuracy increased 14% to 45.13% for dicot, and 8% to 65.37% with monocot. mAP score also increased 7% to 22.57%.

Detection of monocot and dicot weeds in corn and soybean fields is best utilized for early-post emergence herbicide application. Image acquisition for application on or before the V4 growth stage of corn allows the weeds to be large enough to accurately label and provides enough of a physical difference for the network to distinguish class and weed from crop. Likewise, image collection for herbicide application at V3 growth stage in soybeans differentiates features between soybeans and weed enough for accurate labelling and network training. Once weeds are eliminated at these

growth stages, the crop is capable of out-competing new weeds that emerge. The size of the crop compared to the new weeds on or after the post-applied growth stages, as well as the crop's canopy coverage hindering the amount of sunlight the weed is able to receive makes competition from new weeds an insignificant factor. From a nutrient uptake standpoint, the root structure of the corn and soybean plant after V4 and V3 growth stages respectively, is much larger and deeper than a newly emerging weed. This allows the plant to out-compete for nitrogen uptake, as well as other applied fertilizers, making the nutrient uptake by weeds a negligible factor.

A Github repository was created, *UAS-based-Weed-Detection*, featuring the labelled datasets discussed, as well as the modified YOLO configuration files and instructions for running the necessary bash commands for network training, testing, and analysis. The link to this repository is <https://github.com/AaronJames93/UAS-based-Weed-Detection>.

5. SUMMARY AND RECOMMENDATIONS

5.1 Summary

This research evaluated the accuracy of two deep convolutional neural networks at detecting dicot and monocot weeds in corn and soybean fields. Achievement of this evaluation happened in three stages, leading to the results of training a weed detection network. UAS-based RGB imagery was collected over the 2018 and first month of 2019 growing seasons, at three different research sites, throughout the state of Indiana. Four manually-annotated image sets were created to form a database for detection of monocot and dicot weeds in corn and soybean fields. One network training was done on the DetectNet object detection model and six trainings were undertaken with the YOLOv3 object detection model. Results of the network trainings were evaluated on a manually labelled validation set and tested on an unlabelled, unique test image set to gauge detection accuracy of monocot and dicot weeds in corn and soybean fields.

5.1.1 Image acquisition

UAS-based imagery was acquired in corn and soybean research plots from three Purdue agricultural research stations, throughout the 2018 and first month of 2019 growing season. A total of 25 flights were conducted with an RGB sensor during this time. In addition to UAS-acquired imagery, plant subset images were parsed from Google's OpenImagesv4 online, labelled image set for transfer learning. Handheld camera collection was also done for three different weed classes, as well as two classes of corn, to compare results to that of UAS-based image set network training.

5.1.2 Annotated Dataset Creation

Seven manually-labelled image sets were created for object detection network training. The first image set was created from 2018 post-harvest imagery, labelled as “weed” for training the DetectNet detection network in NVIDIA DIGITS. The second image set was parsed from the plant subset of OpenImagesv4, labels were converted to YOLO format and the “plant” class was replaced with “weed”. This image set was trained on the YOLOv3 detection network. The third image set created was comprised of five different classes, labelled “cocklebur”; “giant_ragweed”, “fox-tail”, “corn”, and “nitrogen_defecient_corn”. This image set was also trained on the YOLOv3 detection network. The fourth image set created constituted of 2018 early season corn and soybean plot images, with labelled “dicot and “monocot” weeds, and trained on the YOLOv3 detection network. The fifth image set created was made up of the fourth image set with added 2019 pre-plant images of labelled “dicot” and “monocot” weeds, and trained on the YOLOv3 detection network. The sixth image set created contained 2019 early season corn images of labelled “dicot” weeds. This image set was comprised of 2000x1500 split RGB images. The seventh image set created was composed of the sixth image set and additional 2019 early season corn images of labelled “monocot” and “dicot” weeds and trained on the YOLOv3 detection network. This dataset was comprised of 800x800 split RGB images.

5.1.3 Weed Detection Network Training

Detectnet training with Image Set 1 was done over 300 iterations, with a 45% precision value. YOLOv3 training with Image Set 2 was done over 5000 iterations with a prediction accuracy ranging from 54% to 98% in test images that successfully deployed bounding box predictions. YOLOv3 training on Image Set 3 was trained over 50,000 iterations and yielded an average precision of 78% for giant ragweed, 65% for cocklebur, 84% for foxtail, 73% for corn, and 48% for nitrogen deficient corn. YOLOv3 training on Image Set 4 was done over 20,000 iterations and yielded

an average precision of 3.63% for dicot and 9.88% for monocot. YOLOv3 network training on Image Set 5 was done over 20,000 iterations and resulted in an average precision of 4.22% for dicot and 9.32% for monocot. YOLOv3 training on Image Set 6 spanned 20,000 iterations and resulted in an average precision of 20.42% for dicot. YOLOv3 network training on Image Set 7 was ran over 20,000 iterations and resulted in an average precision of 31.38% for dicot and 58.31% for monocot. YOLOv3 network training for Image Set 7, on the cluster, was composed over 20,000 iterations and resulted in an average precision of 45.13% for dicot and 65.37% for monocot.

5.2 Recommendations and Future Work

Considering the evaluations, comparisons, and testing done on the eight network training results in this research, it is recommended that as many negative samples (non-labelled images) be added to an annotated training set as there are positive samples (labelled images). The negative samples should not contain any objects that are desired for detection. It is also recommended that an annotated image set be created for the specific crop detection is to be done in. For example, unique images sets should be created for corn and soybeans separately. This will cut down on the amount of false positives from the resulting network training. Another recommendation is to split a larger resolution image, such as a 4000x3000 pixel image, into several smaller parts with smaller resolution. For the context of this research, splitting the 4000x3000 pixel image into 20 800x800 sections proved to be advantageous in achieving improved training results. This also limits the amount of images needed to be collected. Finally, it is recommended that UAS-based image collection for labelled weed dataset creation occur in the early growth stages of corn and soybean fields, due to difficulty identifying weeds under canopy and finding individual weed examples to label in areas of heavy weed infestation. From an agronomic perspective, herbicide application should be done in the early stages of crop development to stem weed spread within the field. Early detection of weed invasion, paired with a quick, coordinated response is crucial

to completely eradicate or control the spread of weed species before they can become too widespread and difficult to remediate. It is also important to identify the weed's introductory source or pathway to the field, and understand why it has infested that particular field to begin with. Weed seeds disperse in four dimensions, length width, height, and time (Dekker, 1997). Gravity, wind, water, specialized structures and mechanisms, seed ejaculation, and animal movement are all factors in weed seed dispersal (Dekker, 1997). Some seeds can survive for decades within the soil and thus vigilance on weed seed sprouting is necessary.

Future work done on this research will involve creating a larger annotated image set of 2019 early season corn with monocot and dicot weed labels. Due to the late planting in 2019, a labelled early season soybean image set was unable to be created before research for this thesis was concluded. This image set will be created, trained, and compared against the results of early season corn image set training. The next step for this project is to move towards a real-time, UAS-based, weed detection system. This can be achieved by installing a microcomputer, such as a Raspberry PiTM or NVIDIA NanoTM to the UAS, interfacing with the camera collecting imagery, and installing a lighter memory version of YOLO, such as TinyYOLO to the microcomputer. Final trained weights (the results of completed network training) can be loaded onto the microcomputer and real time video detection testing can be started from the Linux operating system command line. To go a step further, this detector video stream can be broadcast to a base-station computer for map creation and real-time analysis. It is the hope of this graduate student that the open source object detection networks discussed in this thesis will one day be used for a real-time, spot-targeting, weed detection system that can be installed on a current self-propelled sprayer via an add-on attachment. In addition to this “bolt-on” intelligent computer vision system, current pulse-width-modulation (pwm) and quick-switch solenoids can allow sprayer nozzles on the boom of a self-propelled sprayer to be adapted to the rapid activation and shutoff of the valve that would be necessary with a spot spraying system.

REFERENCES

- Abbas, T., Zahir, A., Naveed, M., and Kremer, R. J. (2018). Limitations of existing weed control practices necessitate development of alternative techniques based on biological approaches. *Science Direct*, 147(9):239–280. <https://doi.org/10.1016/bs.agron.2017.10.005>.
- Alp’s Labeling Tool (2017). Alp’s labeling tool. Retrieved from <https://alpslabel.wordpress.com/2017/01/26/alt/>.
- Asaithambi, S. (2017). Why, How and When to Scale your Features. Retrieved from <https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e>.
- Assmann, J. J., Kerby, J. T., Cunliffe, A. M., and Myers-Smith, I. H. (2018). Vegetation monitoring using multispectral sensors – best practices and lessons learned from high latitudes. *bioRxiv*. <https://doi.org/10.1101/334730>.
- Barker, J. and Prasanna, S. (2016). Deep Learning for Object Detection with DIGITS. Retrieved from <https://devblogs.nvidia.com/deep-learning-object-detection-digits>.
- Benbrook, C. M. (2016). Trends in glyphosate herbicide use in the united states and globally. *Environmental Sciences Europe*, 28(3):1–15. <https://doi.org/10.1186/s12302-016-0070-0>.
- Birch, C., Cooper, I., Adkins, G., and Partridge, I. (1989). *Weed Management in Rainfed Agricultural Systems*. Dordrecht, Netherlands: Kluwer Academic.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Information science and statistics. New York, NY: Springer.

- Brownlee, J. (2019). How to Configure the Learning Rate Hyperparameter When Training Deep Learning Neural Networks. Retrieved from <https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/>.
- Carey, J. B. and Kells, J. J. (1995). Timing of Total Postemergence Herbicide Applications to Maximize Weed Control and Corn (*Zea mays*) Yield. *Weed Technology*, 9(2):356-361. <https://doi.org/10.1017/S0890037X00023472>.
- Chicago Mercantile Exchange Group (2019). Agricultural Products Futures. Retrieved from <https://www.cmegroup.com/trading/agricultural/>.
- Davidson, M. W. (2016). Spatial resolution in digital images. Retrieved from <https://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/spatialresolution/>.
- Deep AI, Incorporated (2017). Backpropagation. Retrieved from <https://deeptai.org/machine-learning-glossary-and-terms/backpropagation>.
- Dekker, J. (1997). Weed seed dispersal. Retrieved from <http://agron-www.agron.iastate.edu/~weeds/Ag317/bioeco/lifecycle/disperse.html>.
- Dill, G. M. (2005). Glyphosate-resistant crops: history, status and future. *Pest Management Science*, 61(3):219–224. <https://doi.org/10.1002/ps.1008>.
- dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., and Folhes, M. T. (2017). Weed Detection in Soybean Crops using ConvNets. *Computers and Electronics in Agriculture*, 143:314–324. <https://doi.org/10.1016/j.compag.2017.10.027>.
- Federal Aviation Administration (2018). FAA Drone Zone. Retrieved from <https://faadronezone.faa.gov/#/>.

- Ferrari, V. (2018). Announcing Open Images V4 and the 2018 Open Images Challenge. Retrieved from <https://ai.googleblog.com/2018/04/announcing-open-images-v4-and-eccv-2018.html>.
- Foreign Agricultural Service (1994). Control of water pollution from agriculture - FAO irrigation and drainage paper 55. Retrieved from <http://www.fao.org/3/w2598e/w2598e07.htm>.
- Forson, E. (2017). Understanding SSD MultiBox-Real-Time Object Detection In Deep Learning. Retrieved from <https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab>.
- Golden, K. (2017). Deere to Advance Machine Learning Capabilities in Acquisition of Blue River Technology. Retrieved from, <https://www.deere.com/en/our-company/news-and-announcements/news-releases/2017/corporate/2017sep06-blue-river-technology/>.
- Gonzalez-de-Soto, M., Emmi, L., Perez-Ruiz, M., Aguera, J., and de Santos, P. G. (2016). Autonomous systems for precise spraying - Evaluation of a robotised patch sprayer. *Biosystems Engineering*, 146:165–182. <https://doi.org/10.1016/j.biosystemseng.2015.12.018>.
- Gower, S. A., Loux, M. M., Cardina, J., Harrison, S. K., Sprankle, P. L., Probst, N. J., Bauman, T. T., Bugg, W., Curran, W. S., Currie, R. S., Harvey, R. G., Johnson, W. G., Kells, J. J., Owen, M. D. K., Regehr, D. L., Slack, C. H., Spaur, M., Sprague, C. L., Vangessel, M., and Young, B. G. (2003). Effect of Postemergence Glyphosate Application Timing on Weed Control and Grain Yield in Glyphosate-Resistant Corn: Results of a 2-Yr Multistate Study. *Weed Technology*, 17(4):821–828. Retrieved from <http://www.jstor.org/stable/3989768>.
- Hartzler, B. (2009). Managing weeds to protect crop yields. Retrieved from <https://crops.extension.iastate.edu/managing-weeds-protect-crop-yields>.

- Hassan-Esfahani, L., Ebtehaj, A. M., Torres-Rua, A., and McKee, M. (2017). Spatial Scale Gap Filling Using an Unmanned Aerial System: A Statistical Downscaling Method for Applications in Precision Agriculture. *Sensors* 2017. 17(9), 2106; <https://doi.org/10.3390/s17092106>.
- Hui, J. (2018). Understanding Feature Pyramid Networks for object detection (FPN). Retrieved from https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c.
- Hui, J. (2019). Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. Retrieved from https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088.
- Information Technology at Purdue University (2019). Overview of gilbreth community cluster. Retrieved from <https://www.rcac.purdue.edu/compute/gilbreth>.
- Jensen, J. R. (2005). *Introductory Digital Image Processing : a Remote Sensing Perspective (3rd ed.)*. Upper Saddle River, N.J.: Prentice Hall.
- Knake, E. L. and Slife, F. W. (1962). Competition of *Setaria faberii* with Corn and Soybeans. *Weeds*, 10:26–29. <https://doi.org/10.2307/4040555>.
- Knezevic, S. Z., Evans, S. P., and Mainz, M. (2003). Row Spacing Influences the Critical Timing for Weed Removal in Soybean (*Glycine max*) 1. *Weed Technology*, 17(4):666–673. Retrieved from <http://www.jstor.org/stable/3989745>.
- Lee, S. H., Chan, C. S., Wilkin, P., and Remagnino, P. (2015). Deep-plant: Plant identification with convolutional neural networks. *CoRR*, 1506.08425. <http://arxiv.org/abs/1506.08425>.
- Leonard, E. (2015). *Encyclopedia of Food Grains (2nd ed., Vol. 1)*. Maitland, Australia: Elsevier.
- Li, F. (2018). Convolutional Neural Networks for Visual Recognition. Retrieved from <http://cs231n.stanford.edu>.

- Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. (2016). Feature Pyramid Networks for Object Detection. *CoRR*, 162.031144. <https://arXiv.org/abs/1612.03144>.
- Lingenfelter, D. and Hartwig, N. L. (2007). Introduction to Weeds and Herbicides. Retrieved from <https://extension.psu.edu/introduction-to-weeds-and-herbicides>.
- Liu, G., Morgan, K., Li, Y., Zotarelli, L., DeValerio, J., and Wang, Q. (2015). What is 4R Nutrient Stewardship? Retrieved from <http://edis.ifas.ufl.edu/HS1264>.
- Manfreda, S., McCabe, M. F., Miller, P. E., Lucas, R., Pajuelo Madrigal, V., Mallinis, G., Ben Dor, E., Helman, D., Estes, L., Ciraolo, G., Mllerov, J., Tauro, F., De Lima, M. I., De Lima, J. L. M. P., Maltese, A., Frances, F., Caylor, K., Kohv, M., Perks, M., Ruiz-Prez, G., Su, Z., Vico, G., and Toth, B. (2018). On the use of unmanned aerial systems for environmental monitoring. *Remote Sensing*, 10(4). <https://doi.org/10.3390/rs10040641>.
- Margaritoff, M. (2018). Drones in Agriculture: How UAVs Make Farming More Efficient. Retrieved from <https://www.thedrive.com/tech/18456/drones-in-agriculture-how-uavs-make-farming-more-efficient>.
- MathWorks (2018). Object Detection. Retrieved from <https://www.mathworks.com/discovery/object-detection.html>.
- Matthews, G. A., Bateman, R., and Miller, P. (2014). *Pesticide Application Methods*. Hoboken, NJ: John Wiley & Sons, Ltd.
- Miles, G. E., Ess, D. R., Strickland, R. M., and Morgan, M. T. (2002). Agricultural systems management technologies for precision agriculture. *Precision Agriculture*, page 19. <https://doi.org/10.13031/2013.10370>.
- Milioto, A., Lottes, P., and Stachniss, C. (2017). Real-Time Blob-Wise Sugar Beets vs Weeds Classification for Monitoring Fields Using Convolutional Neural Networks.

- ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W3:41–48. <https://doi.org/10.5194/isprs-annals-IV-2-W3-41-2017>.
- Mohnatkin, A. (2017). Explanation of configuration file parameters. Retrieved from <https://github.com/AlexeyAB/darknet/issues/279>.
- Mohnatkin, A. (2018). Yolo-v3 and yolo-v2 for windows and linux. Retrieved from <https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects>.
- Monhatkin, A. (2019). Yolo mark labelling tool. Retrieved from https://github.com/AlexeyAB/Yolo_mark.
- Myers, J. P., Antoniou, M. N., Blumberg, B., Carroll, L., Colborn, T., Everett, L. G., Hansen, M., Landrigan, P. J., Lanphear, B. P., Mesnage, R., Vandenberg, L. N., vom Saal, F. S., Welshons, W. V., and Benbrook, C. M. (2016). Concerns over use of glyphosate-based herbicides and risks associated with exposures: a consensus statement. *Environmental Health*, 15(1):19. <https://doi.org/10.1186/s12940-016-0117-0>.
- Nadahalli, A. (2014). What is feature scaling? Retrieved from <https://www.quora.com/What-is-feature-scaling>.
- Oerke, E.-C. (2006). Crop losses to pests. *The Journal of Agricultural Science*, 144(1):31–34. <https://doi.org/10.1017/S0021859605005708>.
- Partel, V., Kakarla, S. C., and Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Computers and Electronics in Agriculture*, 157:339–350. <https://doi.org/10.1016/j.compag.2018.12.048>.
- Pérez-Ortiz, M., Pea, J., Gutierrez, P., Torres-Sánchez, J., Hervás-Martínez, C., and López-Granados, F. (2015). A semi-supervised system for weed mapping in sunflower

- crops using unmanned aerial vehicles and a crop row detection method. *Applied Soft Computing*, 37:533–544. <https://doi.org/10.1016/j.asoc.2015.08.027>.
- Pix4D (2017). Pix4dmapper: Professional drone mapping and photogrammetry software. Retrieved from <https://www.pix4d.com/product/pix4dmapper-photogrammetry-software>.
- PSU Turfgrass (2017). Quackgrass Identification. Retrieved from <https://plantscience.psu.edu/research/centers/turf/extension/plant-id/grasses/quackgrass>.
- Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. *CoRR*, 1506.02640. <http://arxiv.org/abs/1506.02640>.
- Redmon, J. and Farhadi, A. (2016). YOLO9000: better, faster, stronger. *CoRR*, 1612.08242. <http://arxiv.org/abs/1612.08242>.
- Redmon, J. and Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *CoRR*, 1804.02767. <http://arxiv.org/abs/1804.02767>.
- Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards RealTime 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>.
- Rouse, J., Haas, R., Scheel, J., and Deering, D. (1974). Monitoring Vegetation Systems in the Great Plains with ERTS. *Proceedings, 3rd Earth Resource Technology Satellite (ERTS) Symposium*. pp48-62. College Station, TX: NASA.
- Sa, I., Popovi, M., Khanna, R., Chen, Z., Lottes, P., Liebisch, F., Nieto, J., Stachniss, C., Walter, A., and Siegwart, R. (2018). WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network

- for Precision Farming. *Remote Sensing*, 10(9). <https://arxiv.org/abs/1808.00100>.
- Saha, S. (2018). A comprehensive guide to convolutional neural networks. Retrieved from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- Sanchez-Lopez, J. L., Molina, M., Bavle, H., Sampedro, C., Suárez Fernández, R. A., and Campoy, P. (2017). A Multi-Layered Component-Based Approach for the Development of Aerial Robotic Systems: The Aerostack Framework. *Journal of Intelligent & Robotic Systems*, 88(2):683 – 709. <https://doi.org/10.1007/s10846-017-0551-4>.
- Scott, J. (2018). Farm Sprayers Overview. Retrieved from <https://www.agriculture.com/machinery/spraying/farm-sprayers-overview>.
- Shi, Y., Thomasson, J. A., Murray, S. C., Pugh, N. A., Rooney, W. L., Shafian, S., Rajan, N., Rouze, G., Morgan, C. L. S., and Neely, H. L. (2016). Unmanned Aerial Vehicles for High-Throughput Phenotyping and Agronomic Research. *Plos One*, 11(7). <https://doi.org/10.1371/journal.pone.0159781>.
- Sterk, R. (2018). USDA lowers forecast for corn, soybean production. Retrieved from https://www.nass.usda.gov/Charts_and_Maps/Crop_Progress_&_Condition/2018/IL_2018.pdf.
- Strickland, R., Ess, D., and D. Parsons, S. (1999). Precision farming and precision pest management: The power of new crop production technologies. *Journal of Nematology*, 30:431–5. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2620313/>.
- TopCon (2018). HiPer V. Retrieved from <https://www.topconpositioning.com/gnss/integrated-gnss-receivers/hiper-v>.

- Torres-Sánchez, J., Pea-Barragn, J. M., De Castro, A., and Lpez-Granados, F. (2014). Multi-temporal mapping of the vegetation fraction in early-season wheat fields using images from UAV. *Computers and Electronics in Agriculture*, 103:104–113. <https://doi.org/10.1016/j.compag.2014.02.009>.
- Tzutalin, D. (2019). labeling. Retrieved from <https://github.com/tzutalin/labelImg>.
- Vittorio, A. (2018). Toolkit to download and visualize single or multiple classes from the huge open images v4 dataset. Retrieved from https://github.com/EscVM/OIDv4_ToolKit.
- Wang, Z., Sun, X., Zhang, Y., Ying, Z., and Ma, Y. (2016). Leaf recognition based on PCNN. *Neural Computing and Applications*, 27(4):899–908. <https://doi.org/10.1007/s00521-015-1904-1>.
- Warden, P. (2017). How many images do you need to train a neural network? Retrieved from <https://petewarden.com/2017/12/14/how-many-images-do-you-need-to-train-a-neural-network/>.
- Watts, A., Perry, J. H., Smith, S. E., Burgess, M., Wilkinson, B., Szantoi, Z., IFJU, P. G., and Percival, F. H. (2010). Small Unmanned Aircraft Systems for LowAltitude Aerial Surveys. *The Journal of Wildlife Management*, 74:1614 – 1619. <https://doi.org/10.1111/j.1937-2817.2010.tb01292.x>.
- Watts, A. C., Ambrosia, V. G., and Hinkley, E. A. (2012). Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing*, 4(6):1671–1692. <https://doi.org/10.3390/rs4061671>.
- Welch-Keesey, M. (2011). *Purdue Master Gardener Guide to Common Lawn and Garden Weeds*. West Lafayette, IN: Purdue University Extension.
- Whelan, B. and Taylor, J. (2013). *Precision Agriculture for Grain Production Systems*. Clayton, Australia: CSIRO Publishing.

- Woolley, E., Dibb, C., and Harvey, J. (1979). Weed control in grassland in Great Britain. *Conference du COLUMA: Journees d'Etudes sur le Desherbage*, 75. pp. 265-289. Paris, France: Committee for weed control.
- Xu, Y., Gao, Z., Khot, L., Meng, X., and Zhang, Q. (2018). A real-time weed mapping and precision herbicide spraying system for row crops. *Sensors*, 18(12). <https://doi.org/10.3390/s18124245>.
- Yamaha Motors USA (2016). Precision Agriculture With Yamaha RMax Helicopter. Retrieved from https://www.youtube.com/watch?time_continue=267&v=ydfPzqaNkuA.
- Yu, J., Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2019). Deep learning for image-based weed detection in turfgrass. *European Journal of Agronomy*, 104:78–84. <https://doi.org/10.1016/j.eja.2019.01.004>.
- Zhao, Z., Zheng, P., Xu, S., and Wu, X. (2018). Object Detection with Deep Learning: A Review. *CoRR*, 1807.05511. <http://arxiv.org/abs/1807.05511>.
- Zimdahl, R. L. (2007). *Fundamentals of Weed Science*. London, United Kingdom: Academic Press, an imprint of Elsevier.

APPENDIX

Table A.1.
DJI Matrice 600 Specifications.

DJI Matrice 600 Pro AIRCRAFT	
Diagonal Wheelbase	1133 mm
Dimensions	1668 mm × 1518 mm × 727 mm with propellers, frame arms and GPS mount unfolded (including landing gear) 437 mm × 402 mm × 553 mm with propellers, frame arms and GPS mount folded (excluding landing gear)
Package Dimensions	525 mm × 480 mm × 640 mm
Weight (with six TB47S batteries)	9.5 kg
Weight (with six TB48S batteries)	10 kg
Max Takeoff Weight Recommended	15.5 kg
Hovering Accuracy (P-GPS)	Vertical: ±0.5 m, Horizontal: ±1.5 m
Max Angular Velocity	Pitch: 300°/s, Yaw: 150°/s
Max Pitch Angle	25°
Max Wind Resistance	8 m/s
Max Ascent Speed	5 m/s
Max Descent Speed	3 m/s
Max Service Ceiling Above Sea Level	2170R propellers: 2500 m ; 2195 propellers: 4500 m
Max Speed	40 mph / 65 kph (no wind)
Hovering Time* (with six TB47S batteries)	No payload: 32 min, 6 kg payload: 16 min
Hovering Time* (with six TB48S batteries)	No payload: 38 min, 5.5 kg payload: 18 min
Flight Control System	A3 Pro
Propulsion System	Motor model: DJI 6010 Propeller model: DJI 2170R
Retractable Landing Gear	Standard
Operating Temperature	14° F to 104° F (-10° C to 40° C)
taken from: https://www.dji.com/matrice600-pro/info#specs	

Table A.2.
DJI Mavic Pro Specifications.

Aircraft	
Folded	H83mm x W83mm x L198mm
Diagonal Size (Propellers Excluded)	335 mm
Weight (Battery & Propellers Included)	1.62 lbs (734 g) (exclude gimbal cover)
Max Ascent Speed	16.4 ft/s (5 m/s) in Sport mode
Max Descent Speed	9.8 ft/s (3 m/s)
Max Speed	40 mph (65 kph) in Sport mode without wind
Max Service Ceiling Above Sea Level	16404 feet (5000 m)
Max Flight Time	27 minutes (no wind at a consistent 15.5 mph (25 kph))
Max Hovering Time	24 minutes (no wind)
Overall Flight Time	21 minutes (In normal flight, 15% remaining battery level)
Max Total Travel Distance (One Full Battery, No Wind)	8 mi (13 km, no wind)
Operating Temperature Range	32° to 104° F (0° to 40° C)
Satellite Positioning Systems	GPS / GLONASS
Hover Accuracy Range	
Vertical:	+/- 0.1 m (when Vision Positioning is active) or +/-0.5 m
Horizontal:	+/- 0.3 m (when Vision Positioning is active) or +/-1.5 m
Operating Frequency	FCC: 2.4-2.4835GHz; 5.150-5.250 GHz; 5.725-5.850 GHz
Transmitter Power (EIRP)	2.4GHz FCC:<=26 dBm; CE: <=20 dBm; SRRC:<=20 dBm; MIC:<=18 dBm
	5.2 GHz FCC:<=23 dBm
	5.8 GHz FCC:<=23 dBm; CE <=13 dBm; SRRC: <=23 dBm; MIC: -

(taken from: <https://www.dji.com/mavic/specs>)

Table A.3.
DJI Mavic Pro Camera Specifications.

Mavic Pro CAMERA	
Sensor:	1/2.3" (CMOS), Effective pixels:12.35 M (Total pixels:12.71M)
Lens:	FOV 78.8° 26 mm (35 mm format equivalent) f/2.2
Distortion:	< 1.5% Focus from 0.5 m to ∞
ISO Range:	video: 100-3200
photo:	100-1600
Electronic Shutter Speed:	8s -1/8000 s
Image Size:	4000×3000
Still Photography Modes:	Single shot
Burst shooting:	3/5/7 frames
Auto Exposure Bracketing:	3/5 bracketed frames at 0.7 EV Bias Interval
Video Recording Modes:	HD: 1280×720 24/25/30/48/50/60/120p
Max Video Bitrate:	60 Mbps
Supported File Systems:	FAT32 (≤ 32 GB); exFAT (> 32 GB)
Photo:	JPEG, DNG
Video:	MP4, MOV (MPEG-4 AVC/H.264)
Supported SD Cards:	Micro SD™
Max capacity:	128 GB. Class 10 or UHS-1 rating required
Operating Temperature Range:	32° to 104° F (0° to 40° C)

(taken from: <https://www.dji.com/mavic/specs>)

Table A.4.
Flir Duo Pro R Thermal and RGB Sensor Specifications.

Below are the specifications for the Flir Duo Pro R:	
Dimensions:	85 × 81.3 × 68.5 mm
Spectral Band:	7.5 – 13.5 μm
Thermal Frame Rate:	30 Hz
Thermal Imager:	Uncooled VOx Microbolometer
Thermal Lens:	19 mm: 32° x 26°
Thermal Sensitivity:	<50 mK (milliKelvin)
Thermal Sensor Resolution:	640 x 512
CONNECTIONS & COMMUNICATIONS	
Digital Video Output:	1080p60,
MAVLink Interface:	Yes
Remote Control:	Yes - PWM (3 channels), MAVLink
10 Pin Accessory Port:	Power in, Analog Video Out, PWM, MAVLink, Frame Sync Out
Micro - HDMI:	Digital Video Out
USB 3.0:	Power in, USB Mass Storage
ENVIRONMENTAL	
Operating Temperature Range:	-20°C to +50°C
Operational Altitude:	+38,000 feet
Storage Temperature Range:	-20°C to +60°C
IMAGING & OPTICAL	
Global Positioning System:	Yes (GPS, GLONASS)
Multiple Color Palettes:	Yes – Adjustable in App and via PWM
Visible Camera FOV:	56° x 45°
Imaging Modes:	IR-only, Vis-only, Picture-in-Picture (IR in Vis)
MSX Image Enhancement:	Yes
Other Sensors:	Accelerometer, Gyroscope, Magnetometer, Barometer
Visible Sensor Resolution:	4000 x 3000
PERFORMANCE	
Measurement Accuracy:	Accuracy is +/- 5 C or 5% of readings in the -25°C to +135°C Range
POWER	
Input Voltage:	5.5 - 26.0 VDC (10-pin JST Port) 5.0 VDC (USB-C Port)
Power Dissipation [avg]:	10 W
(taken from https://www.flir.com/products/duo-pro-r/?model=436-0332-62-00)	

Table A.5.
Slantrange 3PX Multispectral Sensor Specifications.

Slantrange 3PX specifications	
Spectral Channels (Reflectance Sensor):	4
Spectral Channels (Sunlight Sensor):	4
Processor Type:	Snapdragon 801
Processor Speed:	Quad-Core 2.26 GHz
On-Board RAM:	2 GB
Detector Type:	Si CMOS
Available Spectral Range:	410 - 950 nm
Band Positions:	550, 650, 720, 850 nm
Shutter Type:	Global
Positioning & Pointing:	GPS / IMU with EKF
GSD @ 120 m AGL:	4.8 cm
Recommended Image Overlap:	20%* (with Slantview Professional license)
160 Acre Survey @ 120 m AGL, 12 m/s:	25 minutes
Available Data Layers	All SlantView
In-Flight Data:	Setup, Status, Coverage
RTK GPS Option:	Available (Not used on this setup)
Onboard Data Storage:	64 GB SD card (4 hours)
Output Formats:	KML, SHP, GeoTIFF
Size Vegetation Sensor:	14.6 x 6.9 x 5.7 cm
Size Ambient Illumination Sensor:	8.1 x 3.5 x 1.7 cm
Weight:	400g
Power:	11 W @ 9.0 - 28.0 VDC
(Taken from https://www.slantrange.com/product-sensor/)	

Table A.6.
Gilbreth Cluster Specifications ([Information Technology at Purdue University, 2019](#)).

Gilbreth Detailed Hardware Specification

Gilbreth nodes have at least 192 GB of RAM, and 100 Gbps Infiniband interconnects.

Front-Ends	Number of Nodes	Cores per Node	Memory per Node	GPUs per node	Retires in
With GPU	2	20	96 GB	1 P100	2024
Sub-Cluster	Number of Nodes	Cores per Node	Memory per Node	GPUs per node	Retires in
A	4	20	256 GB	2 P100	2022
B	16	24	192 GB	2 P100	2023
C	3	20	768 GB	4 V100	2024
D	8	16	192 GB	2 P100	2024
E	16	16	192 GB	2 V100	2024