CONVOLUTIONAL NEURAL NETWORKS FOR REAL-TIME WEED IDENTIFICATION IN WILD BLUEBERRY PRODUCTION

by

Patrick J. Hennessy

Submitted in partial fulfilment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
December 2020

DEDICATION

This MSc thesis dissertation is dedicated to my loving parents, Karen and Michael Hennessy. Thank you for continually supporting me in my endeavours to better myself and the world around me.

Author

Patrick Hennessy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Yield-limiting weeds in wild blueberry fields, including hair fescue and sheep sorrel, are traditionally managed with uniform applications of herbicides. Spot applications of herbicides reduce the volume required for management. Convolutional Neural Networks (CNNs) were trained to identify hair fescue and sheep sorrel in images of wild blueberry fields. Six CNNs identified targets with a minimum F_1-score of 0.95 for hair fescue and 0.89 for sheep sorrel. Two CNNs were selected as viable for controlling applications from an eight-camera smart sprayer based on processing speeds above 9 frames per second and memory use below 6.4 GB. A graphical user interface was developed for monitoring CNNs and controlling hardware in real-time based on identification of target weeds. The results of this study indicate that CNNs are suitable for identifying hair fescue and sheep sorrel. Future research will involve using the output of the CNNs to automate spray applications, limiting herbicide use.

LIST OF ABBREVIATIONS AND SYMBOLS USED

ABBREVIATIONS

Ait. – Aiton

AP – Average Precision

C.I. – Confidence Interval

cm – Centimetre

CNN – Convolutional Neural Network

COCO – Common Objects in Context

CPU – Central Processing Unit

CUDA – Compute Unified Device Architecture

cuDNN – CUDA Deep Neural Network Library

DBE – Doug Bragg Enterprises Ltd.

F_1 – F_1-score

fn – False Negative

fp – False Positive

FOV – Field Of View

FPS – Frames Per Second

GHz – Gigahertz

GPU – Graphics Processing Unit

GUI – Graphical User Interface

ha – Hectare

Hedw. – Hedwig

IoU – Intersection over Union

IP – Internet Protocol

kg – Kilogram

L. – Linnaeus

m – Metre

mAP – Mean Average Precision

MHz – Megahertz

mm – Millimetre

n – Number of samples

p – p-value

Pourr. – Pourret

pr – Precision

PRN – Partial Residual Network

RAM – Random Access Memory

R-CNN – Region-based Convolutional Neural Network

re – Recall

ReLU – Rectified Linear Unit

R-FCN – Region-based Fully Convolutional Network

RGB – Red, Green, and Blue

RoI – Region of Interest

RPN – Region Proposal Network

s – second

s – Softmax value

SGD – Stochastic Gradient Descent

spp. – Several Species

TFLOPS – Trillion Floating Point Operations per Second

tn – True Negative

tp – True Positive

vRAM – Video Random Access Memory

W – Watt

YOLO – You Only Look Once

SYMBOLS

° – Degree

ACKNOWLEDGEMENTS

I would first like to thank my supervisor, Dr. Travis Esau, for his knowledgeable guidance, support, and patience through the course of this project. Your kind and constructive feedback has greatly improved my knowledge of the wild blueberry industry and my writing and researching skills. I would also like to thank my committee members Dr. Qamar Zaman, Dr. Arnold Schumann, and Dr. Kenny Corscadden. Each of you have provided me with excellent and valuable feedback on my research.

Although they are not committee members, Dr. Scott White and Dr. Aitazaz Farooque have provided excellent advice and guidance throughout the course of this project. Thank you, Dr. White, for your help in selecting appropriate fields for collecting training images and providing feedback on weed management methods. Dr. Farooque, thank you for allowing me to use the deep learning research computer at the University of Prince Edward Island and for your detailed feedback on chapter 2 of my thesis.

My fellow graduate students in the Advanced Mechanized Systems and Precision Agriculture research teams have contributed many hours of their time to help with data collection and provide feedback on my research. I would like to express my sincere gratitude to Craig MacEachern, Anup Kumar Das, Karen Esau, Hanmbal Khan, and Muhammad Saad for their help throughout my project. Additionally, I would like to thank my two colleagues at the University of Prince Edward Island, Hassan Afzaal and Nazar Hussain, for helping me continually access the deep learning server. Good luck with your research, everyone!

Thank you to the summer students who have worked in the Advanced Mechanized Systems and Precision Agriculture research teams over the past two years. Connor Mullins,

Jack Lynds, Derrick Ouma, Chloe Toombs, and Spencer Hauser each contributed many hours towards gathering and labelling training images and writing scripts to automate image sorting. You are all going to be excellent engineers.

Funding for this research was provided by the Natural Science and Engineering Research Council of Canada Discovery Grants Program (RGPIN-06295-2019). Additional supplies and guidance were generously provided by Doug Bragg Enterprises Ltd., who have continually supported wild blueberry research at Dalhousie University.

Finally, I would like to thank the wild blueberry growers and the Wild Blueberry Producers Association of Nova Scotia for allowing me to use their fields for image collection and testing. I am looking forward to continuing my work in the wild blueberry industry.

CHAPTER 1: INTRODUCTION

1.1 Literature Review

1.1.1 Wild Blueberry Cropping System

Wild blueberries (*Vaccinium angustifolium* Ait.) are a perennial, rhizomatous crop native to northeastern North America which are traditionally managed in a two-year cycle (Kinsman, 1993). Commercial fields are developed on deforested areas or abandoned farmland after trees and other vegetation are removed (Hall et al., 1979). The flower buds begin to grow from August to October in the first (sprout) year and lay dormant through the winter (Hall et al., 1979). Growth continues in the spring of the second (crop) year, and harvesting begins in August and September when approximately 90% of the berries are ripe (Farooque et al., 2014; Hall et al., 1979) (**Figure 1**). The berries are harvested using a mechanical tractor attachment which pulls the berries from the plants and drops them into bins (Farooque, 2015). Wild blueberries were harvested with a hand rake prior to the introduction of the first viable mechanical harvester by Doug Bragg in 1981 (Dale et al., 1994; Hall et al., 1983). After harvesting is complete, the branches are pruned by flail mowing or controlled burning (Hall et al., 1979). Better management practices, including development of the mechanical harvester, resulted in the commercial wild blueberry industry expanding in Canada (Farooque et al., 2014; Yarborough, 2004, 2012).When mechanized harvesting was introduced in 1984, the cost of picking was reduced by half and the quality of harvested berries improved (Kinsman, 1993). Farooque et al. (2014) determined that the picking efficiency of the wild blueberry harvesters produced by Doug Bragg Enterprises Ltd. (DBE) was more than 90%.

Figure 1: Wild blueberries prior to harvesting on August 8, 2019 in Debert, NS (45.4430°N, 63.4500°W). The larger bright blue berries are ripe, while the smaller green and red berries are not.

The wild blueberry crop is desired for its health benefits ranging from anti-aging and anti-inflammatory properties (Beattie et al., 2005) to high antioxidant content (Kay & Holub, 2002) which helps reduce the risk of cardiovascular disease and cancer (Lobo et al., 2010). The commercial wild blueberry industry is an economic driver in Nova Scotia, contributing over $100 million to the provincial economy in 2017, including $65.9 million in exports (Wild Blueberry Producers Association of Nova Scotia, 2018). Between 2007 and 2017, the field price of wild blueberries fell from $2.31 kg^{-1} to $0.44 kg^{-1} (Wild Blueberry Producers Association of Nova Scotia, 2018). Labour has become more expensive over this period, with the minimum wage in Nova Scotia increasing by more than 42% (Employment and Social Development Canada, 2018). The price has rebounded slightly to $0.99 kg^{-1} in 2018 and 2019 (Robinson, 2020), but growers continued to express concern over the low price (Esau, 2019). The surveyed growers also mentioned the price

of equipment and the education of their workers as keys issues affecting their businesses (Esau, 2019).

1.1.2 Weed Management Practices

Another major concern for growers is the presence of over 100 weed species including hair fescue (*Festuca filiformis* Pourr.) and sheep sorrel (*Rumex acetosella* L.) (**Figure 2**) in wild blueberry fields (McCully et al., 1991). These weeds inhibit wild blueberry growth and interfere with harvesting equipment further reducing collected yield. Management of weeds has traditionally occurred with a uniform application of liquid herbicide. This method is inefficient because weed instances in wild blueberry fields occur intermittently, not uniformly (McCully et al., 1991; White, 2019). Hair fescue and sheep sorrel have field uniformities of 25% and 63% respectively (White, 2019). Pronamide, which is currently used to manage hair fescue (White & Kumar, 2017), costs \$413.65 ha^{-1} for a uniform application (Esau et al., 2019). Sheep sorrel is traditionally managed with hexazinone (K. I. N. Jensen & Yarborough, 2004; Kennedy et al., 2010), which costs \$168.03 ha^{-1} to uniformly apply (Esau et al., 2019). The intermittent nature of these weeds presents an opportunity for increased herbicide application efficiency using a variable rate smart sprayer. Smart sprayers limit agrochemical application volume by only applying on areas of the field with weed cover (Esau et al., 2014, 2016, 2018; Hong et al., 2012; Partel et al., 2019; Rehman et al., 2018, 2019; Schieffer & Dillon, 2014). Commercial options available for other cropping systems such as See & Spray (Blue River Technologies, 2018), GreenSeeker (Trimble Inc., 2020a), WeedSeeker (Trimble Inc., 2020b), and AiCPlus (Agrifac Machinery B.V., 2016) are not suitable for wild blueberry management due to the highly variable topography of wild blueberry fields (**Figure 3**).

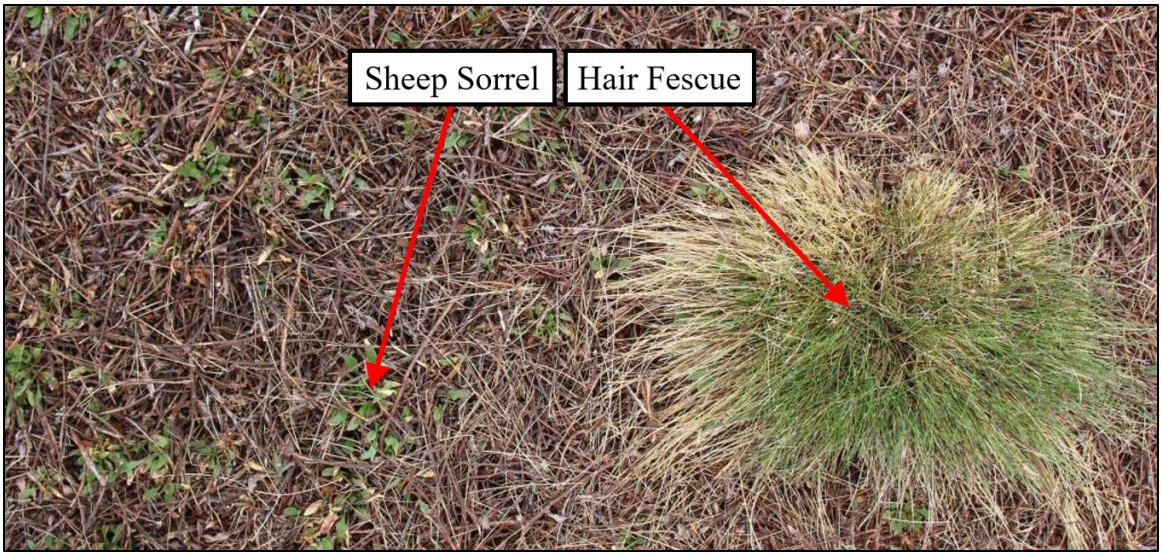

Figure 2: Sheep sorrel and hair fescue in a sprout-year Nova Scotia wild blueberry field during spring 2019. The wild blueberry plants have not started to regrow after flail mowing during the previous autumn.

Figure 3: A wild blueberry field on a series of hills in Murray Siding, NS (45.3654°N, 63.2118°W) on October 9th, 2020. The berries in this field have been harvested, but the plants have not yet been pruned. The leaves that were green in **Figure 1** have turned red with the change in season from summer to autumn.

Machine vision systems relying on green colour segmentation and colour co-occurrence matrices have previously been tested for use in wild blueberry fields (Esau et al., 2014, 2016, 2018; Rehman et al., 2018, 2019). Esau et al. (2014) developed a spot targeting system based on green colour segmentation to detect weeds in wild blueberry

fields. When used on a smart sprayer with cameras 1.2 m from the ground, the system resulted in 44.5% reduction in agrochemical usage compared to a basic sprayer (Esau et al., 2016). Further work with this smart sprayer resulted in herbicide savings of up to 78.5% (Esau et al., 2018). However, this system was limited by its inability to discriminate between different weeds of the same colour. This sprayer used 27 spray nozzles and 9 cameras equally spaced along a 13.7 m boom (Esau et al., 2018) (**Figure 4**). Colour co-occurrence matrices were used for real-time targeting of goldenrod (*Solidago* spp.) in wild blueberry fields (Rehman et al., 2019; Rehman et al., 2018). This method was effective but had to be designed specifically for goldenrod and was not easily scalable to other weeds. This sprayer included 8 nozzles and 4 cameras equally spaced along a 6.1 m boom (Rehman et al., 2019). The smart sprayers relying on imaging systems used in wild blueberry included graphical user interfaces (GUIs) for operators to control spray settings and view images from the cameras in real-time (Esau et al., 2014, 2018; Rehman et al., 2019).

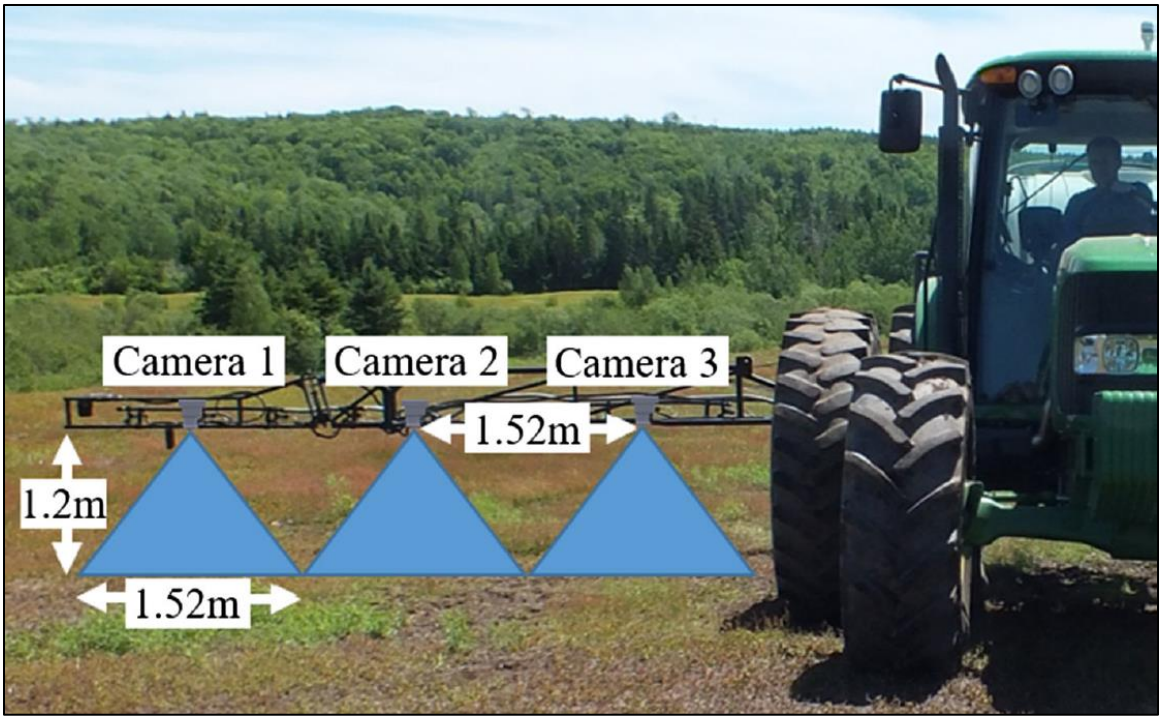

Figure 4: Smart sprayer developed by Esau et al. (2018) showing 3 of 9 cameras used in the image acquisition system.

1.1.3 Machine Learning and Neural Networks

Machine learning is the use of statistics by computers to estimate the solution to complex problems (Goodfellow et al., 2016). A neural network is a type of machine learning algorithm inspired by the way the human brain works. Neural networks are comprised of a series of connected nodes called neurons which are organized into groups of inputs, computations, and outputs (Schmidhuber, 2015). The computation nodes are not shown to users when in use and are usually referred to as hidden neurons. The hidden neurons transform data from the input neurons and classify it to an output neuron dependent on the result of the transformation (Goodfellow et al., 2016). Some neural networks have more than one group (layer) of hidden neurons and are referred to as deep neural networks (LeCun et al., 2015) (**Figure 5**). Data is also transformed when being transferred between

layers by a multiplier, called a weight, and an offset, called a bias. The weights and biases are initialized by the programmer but are modified by the network to optimize accuracy (LeCun et al., 2015). Weights and biases are stored as matrices, and an individual value is assigned for each transfer of information between nodes (Goodfellow et al., 2016). Neural networks obey the universal approximation theorem, which states that the solution to any problem can be approximated with a small degree of error given there is at least one hidden layer with non-linear functions (Hornik et al., 1989).

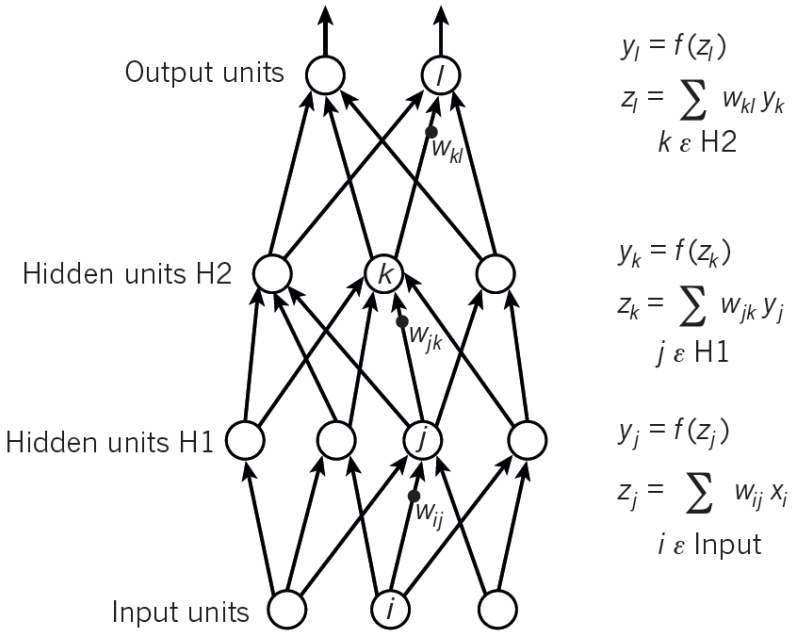

Figure 5: Example of a basic deep neural network (LeCun et al., 2015). An input layer of three neurons sends data to a four-neuron hidden layer, a three-neuron hidden layer, and finally an output layer with two neurons.

1.1.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) utilize multiple layers of neurons with linear and non-linear abstraction of data for image analysis (Goodfellow et al., 2016). CNNs were developed in the 1990s to allow computers at banks to analyse hand-written

digits on cheques (LeCun et al., 1998). This network architecture has resulted in faster and more accurate analysis of images than any other form of machine learning (Goodfellow et al., 2016; Krizhevsky et al., 2012; LeCun et al., 2015). Image processing using CNNs has been used in various aspects of agriculture since 2015 (Kamilaris & Prenafeta-Boldú, 2018). Innovative uses of this technology in agriculture have included livestock monitoring (Santoni et al., 2015; Wu et al., 2020; Yang et al., 2018), plant disease detection (Amara et al., 2017; Fuentes et al., 2017; Venkataramanan et al., 2019), wild blueberry ripeness detection (Schumann et al., 2019), and weed detection for strawberries (Sharpe et al., 2019), Florida vegetables (Sharpe et al., 2020), turfgrasses (Yu, Sharpe, et al., 2019b, 2019a), and ryegrass (Yu, Schumann, et al., 2019).

In CNNs, the most commonly used type of hidden layer is a convolutional layer, which contains three sub-layers of calculations (Goodfellow et al., 2016). These computations result in the computer finding recognizable patterns in the images regarding shape, texture, and colour (Zeiler & Fergus, 2013). The first sub-layer is a convolution operation, which transforms the input data by a linear function (Goodfellow et al., 2016). The second sub-layer transforms the data using a nonlinear function. This function is often a Rectified Linear Unit (ReLU), which raises the value of all negative numbers to zero, while all positive numbers are unchanged (Jarrett et al., 2009; Nair & Hinton, 2010). The third sublayer is a pooling function, which outputs an overview of a given node and the nodes nearby it (Goodfellow et al., 2016). A commonly used pooling function is max pooling, which outputs the maximum value of any node within a given area (Zhou & Chellappa, 1988). Other pooling functions output an average or a weighted average of the

nodes examined by the pooling function (Goodfellow et al., 2016). The pooling function helps to smooth noise from the input data (Boureau et al., 2010; Goodfellow et al., 2016).

Standard neural networks use fully connected layers of nodes, where each node in one layer receives information from every node in the layer before it and sends information to every node in the layer after it (Schmidhuber, 2015). The weights and biases are different between every node, and these values are stored in matrices large enough to hold every value (Goodfellow et al., 2016). With CNNs, the outputs of a small group of nodes are instead modified by a small filter, or kernel (Goodfellow et al., 2016). With black and white images, each pixel is assigned a value based on how dark it is (Goodfellow et al., 2016; Howard, 2018). With colour images, the image must first be separated into the red, green, and blue (RGB) colour coordinate system; each colour channel can then be assigned a value (J. R. Jensen, 2005). With most images, the values will be integers ranging from 0 to 255, as standard digital images have 8-bit colour (J. R. Jensen, 2005). When processing with neural networks, these values are sometimes normalized to floating-point numbers between 0 and 1 (Howard, 2018). The kernel is typically a 3 x 3 or 2 x 2 matrix of values which are initialized with random values and trained through backpropagation (Howard, 2018). Kernels in early layers are trained to isolate edges and simple patterns, while kernels in later layers are trained to isolate more complex objects (Goodfellow et al., 2016; Howard, 2018; Zeiler & Fergus, 2013). Using small kernels is more computationally efficient than fully connected layers (Goodfellow et al., 2016). Simonyan and Zisserman (2015) used a combination of kernels and fully connected layers to create VGGNet, a 19-layer convolutional neural network which placed second in the 2014 ImageNet classification competition (Goodfellow et al., 2016).

After transformation by the kernel, the calculated values (activations) are transformed by the ReLU and pooling function (Goodfellow et al., 2016; Howard, 2018). A final fully connected layer is used to determine the probability that the input image corresponds to each output class (Howard, 2018). The weights in this fully connected layer can sometimes produce negative values despite that the probability should be a positive number. To correct this, a softmax function is used (Elfadel & Wyatt Jr., 1994). One example of a softmax function begins by performing an exponential operation on the output (x). This value is then compared against the sum of all the exponentials to produce the softmax value (s) (Howard, 2018).

$$s(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}} \tag{1}$$

Exponential softmax functions inherently accentuate one result (Howard, 2018). Sometimes, an image should be classified in more than one category (Goodfellow et al., 2016). In this case, the neural network would need to use a different softmax function to produce high probabilities for multiple classes. An example of this is the sigmoid function (Howard, 2018).

$$s(x_i) = \frac{e^{x_i}}{1 + e^{x_i}} \tag{2}$$

The operations in neural networks are inspired by operations that happen in the human brain (Goodfellow et al., 2016). CNNs follow this design principle and incorporate elements of the human visual system. Hubel and Weisel (1968) discovered that the primary visual cortex, the part of the brain initially used in image analysis, responds most to simple shapes while more complex patterns are examined later. CNNs are designed similarly, with

early layers picking out edges and shapes, and later layers identifying specific objects (Zeiler & Fergus, 2013). Another area of the brain used later in the human visual system, the medial temporal lobe, is able to recognize objects regardless of minor transformations of the image such as lighting or viewing angle (Quiroga et al., 2005). CNNs can also recognize objects even with some distortion of the input data (Goodfellow et al., 2016).

Deep CNNs have been used to produce award-winning algorithms for image recognition tasks (Krizhevsky et al., 2012; LeCun et al., 1998; Simonyan & Zisserman, 2015; Szegedy et al., 2015). The number of layers in popular deep CNNs has varied from sixteen (Simonyan & Zisserman, 2015) to thirty (Ioffe & Szegedy, 2015). Deep networks have the ability to many learn features of varying levels of complexity (Zeiler & Fergus, 2013). In general, deeper networks produce more accurate image classifiers (Simonyan & Zisserman, 2015; Szegedy et al., 2015).

1.1.5 Object Detection with Convolutional Neural Networks

CNNs can also be used to detect one or multiple objects within the same image (Goodfellow et al., 2016). Uijlings et al. (2012) developed a selective search method which became the basis for object detection CNNs. Selective search starts by defining many small segments in an image, then recursively combines them with similar segments to form full objects (Uijlings et al., 2012). The selective search method was used in conjunction with a CNN to develop the region-based convolutional neural network (R-CNN) (Girshick et al., 2014). R-CNNs analyse an image by applying a CNN to each Region of Interest (RoI) proposed by the selective search method (Girshick et al., 2014). This is computationally expensive, and the Fast R-CNN model was developed to address this (Girshick, 2015). Fast R-CNN works by applying one CNN to the entire image, and using selective search to

determine the RoI afterward (Girshick, 2015). Faster R-CNN replaced selective search with a Region Proposal Network (RPN), a CNN which proposes the locations of possible objects (Ren et al., 2017). The weights determined by training the RPN are shared with the CNN used in Fast R-CNN to minimize computational load (Ren et al., 2017). Dai et al. (2016) built a region selection system into a CNN to create the region-based fully convolutional network (R-FCN), which produced more accurate identification up to 20 times faster than its predecessors.

The You Only Look Once (YOLO) model divides an image into segments and determines possible object regions in each segment with a confidence score (Redmon et al., 2016). Simultaneously, the probability of each segment belonging to a class is determined. The results are pooled together for the final object detection (**Figure 6**). This method allowed for real-time identification of objects, but with slightly less accuracy than Faster R-CNN (Redmon et al., 2016). YOLOv2 added batch normalization to help prevent overfitting, and increased the processing size of images from 448x488 to 608x608 so smaller objects could be detected (Redmon & Farhadi, 2016). YOLOv2 combines convolutional layers with 3x3 and 1x1 kernels to detect large and small objects (Redmon & Farhadi, 2016). YOLOv2 produced more accurate results than Faster R-CNN, and similar processing times to the original YOLO. YOLOv3 increased the network depth from 30 layers to 106 layers (Redmon & Farhadi, 2018). This decreased the processing speed compared to previous iterations of YOLO, but increased object identification accuracy. Redmon and Farhadi (2018) also created a "Tiny" version of YOLOv3 ("YOLOv3-Tiny") which reduced the number of layers to 23 to allow for faster processing at the expense of

accuracy. Wang et al. (2019a) modified YOLOv3-Tiny with elements of Partial Residual Networks (PRNs) to improve processing speed by 18.2% while maintaining accuracy.

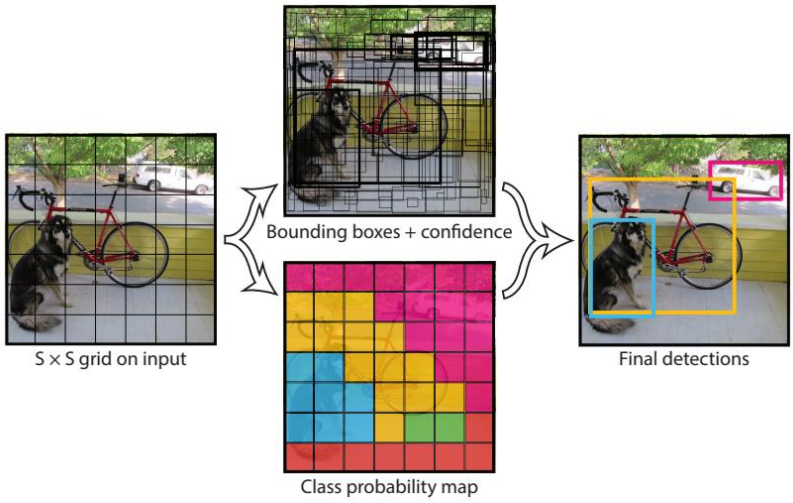

Figure 6: Object detection using YOLO (Redmon et al., 2016).

1.1.6 Backpropagation Training for Deep Neural Networks

The values used for computations between each neuron are determined through backpropagation (Rumelhart et al., 1986). Backpropagation involves feeding pre-classified training data through the network backwards, so the computation parameters can be trained through gradient descent (Cauchy, 1847). This process is repeated for many cycles (epochs) until the error produced by the neural network is minimized (Goodfellow et al., 2016). If too many epochs are used the network may over-fit to the data, which results in increased error on independent testing images (Bishop, 1995; Tetko et al., 1995).

To quickly train neural networks, graphics processing units (GPUs) are used instead of central processing units (CPUs) because they have more processing cores, which allows images to be processed faster through parallelization of the calculations and greater

memory bandwidth (Goodfellow et al., 2016). Steinkraus et al. (2005) were the first to use GPUs for machine learning, and found that an Nvidia GeForce 6800 Ultra GPU (Nvidia Corporation, Santa Clara, CA, USA) could train and test their neural network three times faster than an Intel Pentium 4 CPU (Intel Corporation, Santa Clara, CA, USA). Chellapilla et al. (2006) had similar results their neural network testing a GeForce 7800 Ultra and an ATI Radeon X800 GPU (Advanced Micro Devices Inc., Santa Clara, CA, USA) with a Pentium 4. Nvidia developed the CUDA (Compute Unified Device Architecture) programming language so GPUs could be used more easily and more efficiently for machine learning tasks (Harris, 2008). After the release of CUDA, Raina et al. (2009) trained neural networks up to 15 times faster with an Nvidia GTX 280 GPU than a CPU. Cireşan et al. (2010) determined that their neural network could be trained by a GTX 280 GPU 40 times faster than by an Intel Core2 Quad 9450 CPU.

1.1.7 Gradient Descent and Learning Rate

The learning rate defines the degree of change permitted in a neural network with each epoch of training (Howard, 2018). Small learning rates result in slow parameter training, while large learning rates can cause instability in training, resulting in the parameters not being defined correctly (Goodfellow et al., 2016). Training of machine learning algorithms often occurs using a variation of gradient descent (Cauchy, 1847; Goodfellow et al., 2016). One of the most common training methods is Stochastic Gradient Descent (SGD), which allows the learning rate to be adjusted during training (Bottou, 1998). This gives an advantage over traditional gradient descent, as the learning rate should be decreased over time for optimal results (Goodfellow et al., 2016). The method of momentum can be used to speed up learning (Polyak, 1964). Instead of adjusting the step

size using a single gradient, a series of many gradients are applied (Goodfellow et al., 2016; Polyak, 1964). Nesterov (1983) developed an accelerated momentum which has been applied for neural network training (Sutskever et al., 2013). Nesterov's (1983) momentum produces less error than Polyak's (1964) momentum, and trains the neural network in a comparable amount of time (Goodfellow et al., 2016; Sutskever et al., 2013).

The learning rate is one of the most difficult hyperparameters to choose correctly, so researchers often use an optimization algorithm to adjust the learning rate during training (Goodfellow et al., 2016). Most optimization algorithms result in faster network training than simply decreasing the learning rate over time (Howard, 2018). An early optimizer was the delta-bar-delta algorithm, which increased the learning rate if the partial derivative with respect to the weight did not change signs, and decreased the learning rate if the derivative did change signs. (Jacobs, 1988). The AdaGrad optimizer is a more recent method used for learning rate optimization (Duchi, 2011). AdaGrad works by dividing the learning rate by the square root of the sum of squares of all the previous gradients (Duchi, 2011). AdaGrad works well with sparse gradients, as it uses information from all previous gradients (Duchi, 2011). AdaGrad does not work well with some deep neural networks, as the learning rate can become very small resulting in a long training time (Goodfellow et al., 2016). RMSProp is another optimizer which adjusts AdaGrad by using a weighted average of the gradients for the calculation instead of treating all gradients equally, making it more reliable for deep learning (Hinton, 2012). The Adam optimizer adds SGD with momentum to RMSProp (Kingma & Ba, 2015). This results in Adam working well with sparse gradients, like AdaGrad, and deep neural networks, like RMSProp (Goodfellow et al.,

2016). Adam produced more accurate results than any other optimization algorithm when it was released (**Figure 7**) (Kingma & Ba, 2015).

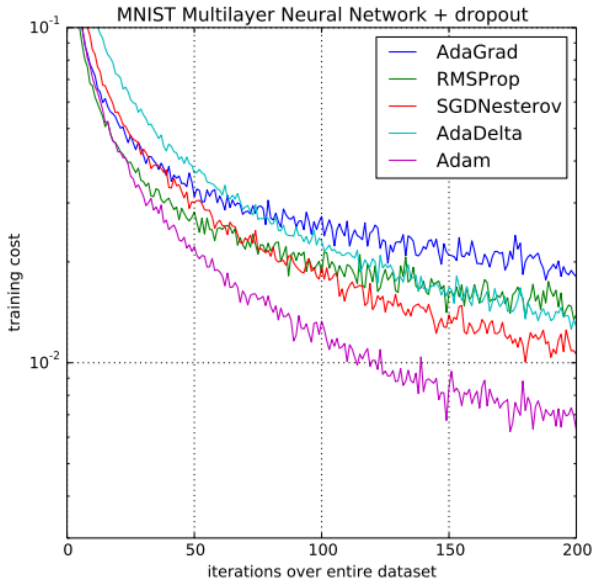

Figure 7: Comparison of training losses with different optimizers (Kingma & Ba, 2015).

1.1.8 Overfitting

Overfitting is a form of error in neural networks in which a trained network correctly classifies data used during training and validation but cannot classify new data with the same accuracy (Goodfellow et al., 2016). Overfitting can be caused if a network is trained for too many epochs, or if there are too few pieces of training data for a very large neural network (Goodfellow et al., 2016; Howard, 2018). If overfitting is the result of the number of training epochs, the network can be retrained with fewer epochs (Goodfellow et al., 2016; Howard, 2018). If overfitting is the result of the ratio of training images to network size, then the network size must be reduced so it can generalize with the available data (Goodfellow et al., 2016).

Dropout randomly removes nodes in a neural network during training and replaces them after a user-defined number of epochs (Srivastava et al., 2014). This improves the accuracy of the neural network by reducing overfitting (Srivastava et al., 2014). Dropout has few restrictions on the type of machine learning algorithm it is applied to, and requires less computational power than other methods (Goodfellow et al., 2016). A requirement for dropout to be effective is a large amount of training data (Goodfellow et al., 2016). Dropout was not as effective as Bayesian neural networks (Neal, 1996) with the same architecture, however, for classifying data in the Alternative Splicing Dataset which contains 3,665 pieces of data for modelling gene coding in proteins (Xiong et al., 2011).

Using larger training datasets is one of the most effective ways to improve machine learning algorithms (Goodfellow et al., 2016; Howard, 2018; LeCun et al., 2015). In cases where it is difficult or inconvenient to collect more training data, dataset augmentation can be used to artificially increase the amount of training data available (Goodfellow et al., 2016). For image processing, augmentation methods typically use affine transformations such as small rotations (5° or 10°), cropping and translating, and mirroring the image (Howard, 2018). Data augmentation is common in many convolutional neural network models (Devries & Taylor, 2017) such as LeNet (LeCun et al., 1998), AlexNet (Krizhevsky et al., 2012), and YOLOv3 (Redmon & Farhadi, 2018). When determining which augmentation method to use, the user must ensure that the data is modified in a realistic manner. For example, 180° rotation and horizontal mirroring would not be appropriate in some image classification tasks such as identification of cats and dogs (Goodfellow et al., 2016). Wang & Perez (2017) found that using generative adversarial networks (Goodfellow et al., 2014) or allowing a neural network to pick an augmentation method

resulted in slight improvements over traditional affine transformations, but required three times as much processing power.

1.2 Objectives

Hair fescue and sheep sorrel are problematic weeds which frequently grow in wild blueberry fields. To combat weed growth, a real-time smart sprayer using an advanced machine vision system is needed to detect the precise location of weeds for spraying. There are currently no commercially available systems that meet this specification which are viable for use in the wild blueberry industry. Researchers have had success getting fast and accurate results using CNNs in other machine vision applications (Kamilaris & Prenafeta-Boldú, 2018; LeCun et al., 2015). A requirement of CNNs is the large number of training images needed to properly train the network to detect the desired target. Deep learning CNNs have never been used for real-time herbicide application in wild blueberry production. As such, there are three main objectives to this research project:

1.) Develop an image dataset, train CNNs which can identify hair fescue and sheep sorrel, and determine a baseline number of images needed for accurately training.
2.) Verify the effectiveness of a trained CNN in a completely randomized field test using several cameras, target distances, and field sites.
3.) Determine processing requirements for using the CNNs to control herbicide application on a smart sprayer and develop a GUI for controlling spray applications with a CNN.

The CNN for recognizing these weeds will ideally control the smart sprayer to spray every target weed and apply the herbicides as efficiently as possible. Future work will involve using the trained CNN for spot-application of herbicide on hair fescue and sheep

sorrel in wild blueberry fields. Further development after this project will involve training the CNN to detect additional target weeds in wild blueberry fields for spot-application of herbicides. This will involve creating additional datasets of field images. This project will determine a baseline for the minimum number of images recommended for training a CNN in this application, which will limit the amount of labour and time needed for collecting training images.

CHAPTER 2: DEVELOPMENT OF HAIR FESCUE AND SHEEP SORREL IDENTIFICATION SYSTEMS USING CONVOLUTIONAL NEURAL NETWORKS

2.1 Introduction

Wild blueberries (*Vaccinium angustifolium* Ait.) are an economically important crop native to northeastern North America. Wild blueberry plants grow through naturally occurring rhizomes in the soils. Commercial fields are typically developed on abandoned farmland or deforested areas after the removal of trees and other vegetation (Hall et al., 1979). In 2016, there were more than 86,000 ha of fields in production in North America, yielding approximately 119 million kg of fruit (Yarborough, 2017). Wild blueberries contributed over $100 million to Nova Scotia's economy in 2017, including $65.9 million in exports (Wild Blueberry Producers Association of Nova Scotia, 2018). The crop is desired for its health benefits ranging from anti-aging and anti-inflammatory properties (Beattie et al., 2005) to high antioxidant content (Kay & Holub, 2002) which helps reduce the risk of cardiovascular disease and cancer (Lobo et al., 2010).

Infestations of hair fescue (*Festuca filiformis* Pourr.), sheep sorrel (*Rumex acetosella* L.), and over 100 other weed species (McCully et al., 1991) limit yield by competing with the wild blueberry plants for nutrients and interfering with harvesting equipment. In 2019, sheep sorrel and hair fescue were the first and fourth most common weeds in Nova Scotia wild blueberry fields, respectively (White, 2019). Weeds are typically managed by applying a uniform application of liquid herbicide using a boom sprayer fixed to an agricultural tractor. Hexazinone was used to manage a broad array of weeds in wild blueberry fields beginning in 1982 but is no longer used for hair fescue as it has developed resistance from repeated use (K. I. N. Jensen & Yarborough, 2004).

Hexazinone was shown to reduce instances of sheep sorrel in Nova Scotia wild blueberry fields (K. I. N. Jensen & Yarborough, 2004; Kennedy et al., 2010), but was ineffective in Maine (K. I. N. Jensen & Yarborough, 2004). Pronamide costing more than twice that of hexazinone (Esau et al., 2019), is currently used to manage hair fescue (White & Kumar, 2017). Other options for hair fescue management include glufosinate, sulfentrazone, and terbacil (White, 2018). A study by Hughes et al. (2016) resulted in pronamide reducing the number of sheep sorrel plants in three of four test sites, but only reducing the biomass in one of four sites. Sulfentrazone is currently being studied as a management option for sheep sorrel, with promising initial results (White, 2019).

Smart sprayers such as See & Spray (Blue River Technologies, 2018), GreenSeeker (Trimble Inc., 2020a) and WeedSeeker (Trimble Inc., 2020b) when used in other cropping systems such as cotton can detect and spray specific areas of fields that require application reducing the volume of agrochemical needed. Machine vision systems relying on imaging data from cameras have previously been researched for controlling smart sprayers in wild blueberry production (Esau et al., 2014, 2016, 2018; Rehman et al., 2018, 2019). A green colour segmentation algorithm was used to detect areas with weed cover, which resulted in herbicide savings of up to 78.5% (Esau et al., 2018). However, this system did not distinguish different species of green coloured weeds. Colour co-occurrence matrices were used to detect and spray goldenrod (*Solidago* spp.) in wild blueberry fields (Rehman et al., 2018, 2019). The algorithm developed by Rehman et al. (2018) was accurate, but inconvenient due to long processing times and because it had to be purpose-built for goldenrod. Convolutional neural networks (CNNs) are a recent processing technique which can classify entire images or objects within an image (LeCun et al., 2015). Image-

classification CNNs provide an inference about an entire image, while object-detection CNNs identify objects within images and label them with bounding boxes (Goodfellow et al., 2016). CNNs intelligently identify visual features and find patterns associated with the target with minimal input from the user, making them easily adaptable for new targets. They are trained to detect new targets through backpropagation, which involves repeatedly showing a computer many labelled pictures of the desired target (Rumelhart et al., 1986). This technology has been used in wild blueberry production for detecting fruit ripeness stages and estimating potential fruit yield (Schumann et al., 2019). In other cropping systems, CNNs have been effective for detecting weeds in strawberry fields (Sharpe et al., 2019), turfgrasses (Yu, Sharpe, et al., 2019b, 2019a), ryegrasses (Yu, Schumann, et al., 2019), and Florida vegetables (Sharpe et al., 2020). CNNs have also been used for detecting diseases on tomato (Fuentes et al., 2017; Venkataramanan et al., 2019), apple, strawberry, and various other plants (Venkataramanan et al., 2019). These CNNs used between 1,472 (Sharpe et al., 2019) and 40,800 (Yu, Schumann, et al., 2019) labelled images for training and validation. Given that there are more than 100 unique weed species in Nova Scotia wild blueberry fields, it would be best to train CNNs to identify more than just fescue and sheep sorrel. It is impractical to collect 40,800 images for weed detection in wild blueberry without significant resources. Spring applications of herbicide must occur before new blueberry plant growth, and autumn applications of herbicide must occur after plant defoliation but before the soil is frozen (Government of New Brunswick, 2020). This leaves only a few weeks during each spray timing interval for image collection.

This study evaluated the effectiveness of three object-detection CNNs and three image-classification CNNs for identifying hair fescue and sheep sorrel in images of wild

blueberry fields. The object-detection CNNs used were YOLOv3, YOLOv3-Tiny (Redmon & Farhadi, 2018), and YOLOv3-Tiny-PRN (C.-Y. Wang et al., 2019). YOLOv3 and YOLOv3-Tiny are leading object-detection CNNs designed on the Darknet framework (Redmon, 2016) which processed 416x416 images at 35 and 220 frames per second (FPS) respectively on an Nvidia Titan X graphics processing unit (GPU, Nvidia Corporation, Santa Clara, CA, USA), while achieving mean average precision (mAP, Szegedy et al., 2015) scores of 55.3% and 31.0% (Redmon, 2018; Redmon & Farhadi, 2018) on the COCO dataset (Lin et al., 2014). YOLOv3-Tiny-PRN added elements from Partial Residual Networks to improve the inference speed of YOLOv3-Tiny by 18.2% and maintain its accuracy on the COCO dataset (C.-Y. Wang et al., 2019). The image-classification CNNs used in this study were Darknet Reference (Redmon, 2016), EfficientNet-B0 (Tan & Le, 2019), and MobileNetV2 (Sandler et al., 2018). Darknet Reference is an image classifier which achieved a Top-1 accuracy (Szegedy et al., 2015) of 61.1% on the ImageNet dataset (Deng et al., 2009) and processed 224x224 images at 345 FPS on a Titan X GPU (Redmon, 2016). EfficientNet-B0 and MobileNetV2 achieved higher Top-1 scores (77.1% (Tan & Le, 2019) and 74.7% (Sandler et al., 2018), respectively) than Darknet Reference at the same resolution. The internal resolution of the CNNs can be increased for identifying small visual features, as was performed by Schumann et al. (2019), but large increases can result in a limiting return in accuracy improvements (Tan & Le, 2019). This technology can be used to discriminate between weed species in real-time, providing a level of control not present in previous wild blueberry smart sprayers (Esau et al., 2014, 2016, 2018; Rehman et al., 2018, 2019). Using a CNN to control herbicide spray has the potential to greatly

increase application efficiency, leading to significant cost-savings for wild blueberry growers.

2.2 Materials and Methods

2.2.1 Image Datasets

Images used for this study were collected from 58 fields in northern and central Nova Scotia during the 2019 field season (**Figure 8**). Eight digital cameras with resolutions ranging from 4000x3000 to 6000x4000 pixels captured colour pictures of wild blueberry fields containing hair fescue and sheep sorrel. All pictures were captured with the camera lens pointed downward and without zoom. The photographers were instructed to capture approximately 70% of images from waist height (0.99 ± 0.09 m), 15% from knee height (0.52 ± 0.04 m), and 15% from chest height (1.35 ± 0.07 m) to allow the CNNs to recognize targets at a range of distances. A total of 4,200 images with hair fescue, 2,365 images with sheep sorrel, 2,337 with both weeds, and 2,041 with neither weed were collected from April 24 to May 17, 2019 (spring), during the herbicide application timing interval. An additional 2,442 images with hair fescue and 1,506 images with sheep sorrel were collected from November 13 to December 17, 2019 (autumn), during the herbicide application timing interval.

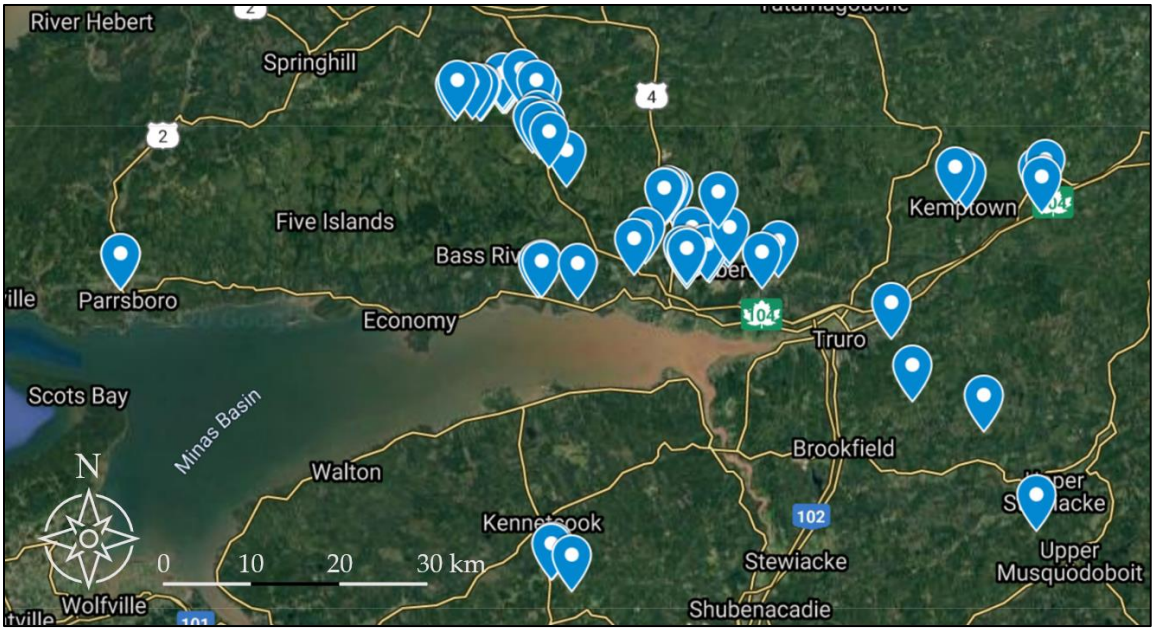

Figure 8: Map of image collection sites in Nova Scotia, Canada. Images were collected in 58 different fields to ensure variability of the dataset (Google, 2020).

The images were scaled and cropped to 1280x720 pixels (720p) using IrfanView (v4.52, IrfanView, Wiener Neustadt, Austria) for target labeling, training, validation, and testing processes. The hair fescue and sheep sorrel datasets were randomly subdivided into three datasets for training, validation, and independent testing of the object-detection CNNs in the Darknet framework (Redmon, 2016). For hair fescue, 3,780 images were used for training, 420 images were used for validation, and 250 images were used for testing. Training and validation images were randomly selected from the spring dataset, while the testing images were randomly selected from the autumn dataset. The training, validation, and testing datasets for sheep sorrel consisted of 960, 120, and 120 images from spring. Instances of hair fescue and sheep sorrel in the cropped images were labeled using custom-built software developed by Schumann et al. (2019). One bounding box was created for each hair fescue tuft or sheep sorrel plant whenever possible. Weeds were densely packed in some images, making it impossible to determine where one instance ended, and another

started. Hair fescue tufts would overlap one another, and it was unclear which sheep sorrel leaves belonged a common plant. In these cases, a box was drawn around the entire region encompassing the densely packed weeds.

For image classification CNNs, 2,000 images were randomly selected from each image set (hair fescue, sheep sorrel, both weeds, neither weed) from spring. Image-classification CNNs for hair fescue identification were trained to put images into one of two classes: “hair fescue” and “not hair fescue”. The datasets containing images with hair fescue and images with both weeds were labelled as “hair fescue”, while the datasets containing images with sheep sorrel and neither weed were labelled as “not hair fescue”. The same methodology was used for classifying images as “sheep sorrel” or “not sheep sorrel”, with the hair fescue dataset replaced by the sheep sorrel dataset and vice-versa. Randomly selected subsets containing 10% of the images were used for validation, while the remaining images were used for training. The testing datasets consisted of 700 randomly selected images from autumn.

2.2.2 Convolutional Neural Network Training and Validation

The networks were trained and validated on a custom-built Ubuntu 16.04 (Canonical Ltd., London, UK) computer with an Nvidia RTX 2080 Ti GPU. The object-detection networks were trained for 5,000 iterations each, with trained weights being saved every 50 iterations. The initial learning rate was set at 0.001, the default learning rate for each network, and was decreased by a factor of 10 after 2,000, 3,000, and 4,000 iterations. A general rule-of-thumb states that object detectors require 2,000 iterations per class to train (Redmon et al., 2020; Schumann et al., 2019), so the learning rate was decreased after 2,000 iterations to avoid overfitting. The default batch size, 64, for all networks was used

for training. The batch was subdivided into 16 for YOLOv3-Tiny and YOLOv3-Tiny-PRN, and 64 for YOLOv3 so the CNNs would fit within the GPU memory limits while training. All other hyperparameters were left identical to the settings used by Redmon and Farhadi (2018) and Wang et al. (2019a) in each network. After training was complete, the weight file achieving the highest average precision (AP) score was used for analysis. The image-classification networks were trained for 10,000 iterations each the default learning rates of 0.1 for Darknet Reference and MobileNetV2, and 0.256 for EfficientNet-B0. The batch size of 64 was subdivided by 16 for Darknet Reference, 16 for MobileNetV2, and 64 for EfficientNet-B0. All other parameters for each CNN were left as defined by their respective authors. The weight file achieving the highest Top-1 score was used for analysis. The network configuration files used for EfficientNet-B0 and MobileNetV2 were converted to the Darknet framework by Wang et al. (2019b). The Darknet framework requires image resolutions to be a multiple of 32, so the internal network resolutions were set to 1280x736 pixels for training and four different resolutions (1280x736, 1024x576, 960x544, and 864x480) for testing.

All networks were evaluated on three additional performance metrics with the detection threshold of object-detectors set at 0.25: precision, recall, and F_1-score (Sokolova & Lapalme, 2009). Precision, recall, and F_1-score are functions of true positive detections (*tp*), false positive detections (*fp*), and false negative detections (*fn*) of targets. Precision is ratio of true positives to all detections:

$$Pr = \frac{tp}{tp + fp} \tag{3}$$

Recall is the ratio of true positives to all actual targets:

$$Re = \frac{tp}{tp + fn} \tag{4}$$

F_1-score is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{Pr \cdot Re}{Pr + Re} \tag{5}$$

Object-detection networks were first tested on their ability to detect all instances of hair fescue or sheep sorrel in each validation image. A detection was counted as a true positive if the Intersection-over-Union (IoU, Rahman & Wang, 2016), the overlap between the ground-truth box and the detection box, was greater than 50%. However, a machine vision system on a smart sprayer would require one or more detections in an image to trigger a spray event. The area sprayed due to a single detection would be wide enough to cover all target weeds in an image, even if only one is detected. A second set of tests were performed on object-detection networks to measure their ability to detect one or more instances of the target weed in a picture, regardless of how many total targets were in each image. A result was considered a true positive if the network detected at least one instance of the target weed in an image containing the desired target. The second set of tests were performed at two detection thresholds, 0.15 and 0.25, and the same four resolutions. The networks were evaluated on precision, recall, and F_1-score.

Finally, all networks for hair fescue detection were retrained with progressively smaller datasets to examine the effect that the number of training images had on identification accuracy. Five training datasets containing 3,780, 2,835, 1,890, 945, and 472 images were used for this test. All networks were trained for the same number of iterations, at the same learning rate, and with the same batch sizes as in the initial tests. The F_1-scores

of all networks at 1280x736 resolution were compared to determine a preliminary baseline for the number of images needed to train a CNN for weed detection.

2.3 Results and Discussion

2.3.1 Fescue and Sheep Sorrel Targeting with Object-Detection Networks

YOLOv3, YOLOv3-Tiny, and YOLOv3-Tiny-PRN were successfully trained to detect hair fescue and sheep sorrel (**Figure 9**). The largest difference between validation and testing AP scores on ground-truth images was 2.99 percentage points with YOLOv3 at 1280x736 resolution, indicating that the trained networks did not overfit to the validation images (**Table 1**). The highest validation AP score for hair fescue detection (75.83%) was achieved with YOLOv3 at 1280x736 resolution, although the difference in AP score was within 1% for YOLOv3 and YOLOv3-Tiny networks with resolutions from 960x544 to 1280x736. The highest F_1-score (0.62) was achieved by YOLOv3 at 1280x736 resolution and YOLOv3-Tiny at 1024x576 resolution. YOLOv3-Tiny-PRN yielded the lowest AP and F_1-scores at each resolution. The precision of all networks at all resolutions was 0.90 or greater, indicating minimal false-positive detections. The recall varied from 0.34 to 0.46, indicating that more than half of the hair fescue plants were not detected at a threshold of 0.25. This was lower than in other agricultural applications, including Sharpe et al. (2020), which saw recall values ranging from 0.59 to 0.99 for detection of vegetation in vegetable row middles. This may be a result of the light-coloured soil in the images used by Sharpe et al. (2020) creating a larger contrast between the targets and the background compared to the images used in this study. Given that a machine vision system on a smart sprayer would require one or more detections in an image to trigger a spray event, the current level of detection accuracy may be acceptable. Alternatively, the detection threshold could be

lowered to decrease the overall number of true and false positive detections, thereby increasing the recall at the expense of precision.

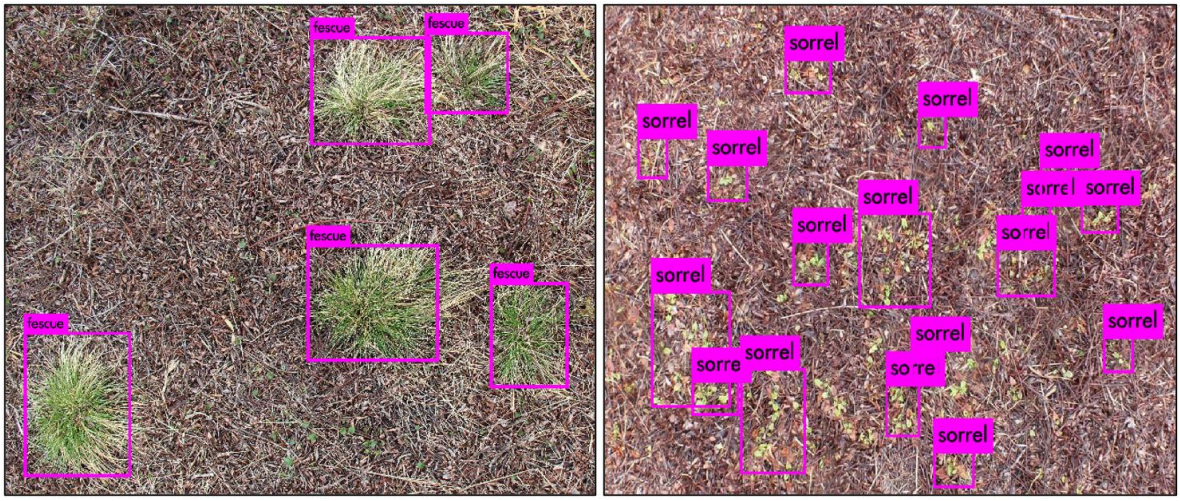

Figure 9: Examples of hair fescue (L) and sheep sorrel (R) detections on images captured in wild blueberry fields using YOLOv3-Tiny at 1280x736 resolution. The base hair fescue picture was captured on the morning of April 24, 2019 in Murray Siding, NS (45.3674°N, 63.2124°W), while the base sheep sorrel picture was captured on the morning of April 25, 2019 in Kemptown, NS (45.4980°N, 63.1023°W). Both pictures contain the respective target weeds with wild blueberry plants after flail mowing. The hair fescue detection picture also contains some sheep sorrel.

Table 1: Object-detection results for hair fescue plants in images of wild blueberry fields.

Network	Resolution	Validation				Testing
		Precision	Recall	F_1-Score	AP_{50} (%)	AP_{50} (%)
YOLOv3	1280x736	0.94	0.46	0.62	75.83	72.84
	1024x576	0.94	0.44	0.60	75.56	73.87
	960x544	0.94	0.43	0.59	75.13	72.81
	864x480	0.95	0.39	0.55	71.68	70.61
YOLOv3-Tiny	1280x736	0.91	0.45	0.60	74.83	75.18
	1024x576	0.92	0.46	0.62	75.46	76.57
	960x544	0.91	0.46	0.61	75.27	75.63
	864x480	0.9	0.45	0.60	71.40	74.06
YOLOv3-Tiny-PRN	1280x736	0.92	0.44	0.59	71.40	73.26
	1024x576	0.94	0.39	0.56	66.40	63.91
	960x544	0.94	0.38	0.54	61.38	61.04
	864x480	0.94	0.34	0.50	55.29	54.89

Detection accuracy for sheep sorrel was considerably lower than the accuracy for hair fescue. The peak validation AP score (60.81%) and F_1-score (0.51) were achieved by YOLOv3 at 1280x736 resolution (**Table 2**). The largest difference between validation and testing AP scores was 4.85 percentage points, indicating that the trained networks did not overfit to the validation images. The precision and recall metrics were lower for sheep sorrel than fescue for every test, with a mean change of 0.15 and 0.22, respectively. The peak recall (0.37) was achieved by YOLOv3 at 1280x736 resolution, with no other resolutions achieving above 0.29 for any network. YOLOv3-Tiny-PRN had lower scores than the other two networks, with the AP score decreasing by an average of 35.82 percentage points compared to YOLOv3 and 31.64 percentage points compared to YOLOv3-Tiny. The labelling strategy may have affected the sheep sorrel detection accuracy. The training bounding boxes were drawn to encompass each entire sheep sorrel plant, but this resulted in background pixels encompassing more than 50% of the bounding

box area in some cases. Additionally, the network may be drawing a box around a group of sheep sorrel leaves that is inconsistent with the box drawn in the labelling software. If the IoU between the ground-truth and detected boxes is less than 50%, the result would be considered a false negative and a false positive instead of a true positive. Labelling each individual leaf instead of each plant may improve the results. Alternatively, semantic image segmentation methods like the one used by He et al. (2015) could be effective as well. However, only one instance of sheep sorrel needs to be detected per image to trigger a spray application, so the current CNN may be viable for this application.

Table 2: Object-detection results for sheep sorrel plants in images of wild blueberry fields.

Network	Resolution	Validation				Testing
		Precision	Recall	F_1-Score	AP_{50} (%)	AP_{50} (%)
YOLOv3	1280x736	0.79	0.37	0.51	60.18	62.98
	1024x576	0.82	0.28	0.42	52.20	57.05
	960x544	0.83	0.26	0.40	49.15	53.04
	864x480	0.83	0.21	0.34	42.06	46.63
YOLOv3-Tiny	1280x736	0.79	0.29	0.42	54.02	57.00
	1024x576	0.81	0.24	0.37	48.29	52.32
	960x544	0.80	0.23	0.35	45.16	49.46
	864x480	0.78	0.19	0.31	39.40	42.91
YOLOv3-Tiny-PRN	1280x736	0.80	0.13	0.22	26.80	29.41
	1024x576	0.74	0.07	0.13	14.26	15.28
	960x544	0.72	0.06	0.11	11.54	12.26
	864x480	0.69	0.04	0.08	7.73	7.98

Given the high precision and low recall scores for detection of each weed, the networks' ability to detect at least one target weed per image was tested at a threshold of 0.15 in addition to the original threshold of 0.25. Through lowering the threshold, average recall and F_1-scores in fescue detection were improved by 0.08 and 0.03, respectively at the expense of 0.02 from precision. In hair fescue detection, the peak F_1-score (0.98) was

achieved by YOLOv3-Tiny-PRN at 1280x736 and 1024x736 and a threshold of 0.15, although no network at any resolution or threshold had an F_1-score of less than 0.90. (**Table 3**). Precision varied between 0.93 and 0.99, with the latter being achieved by YOLOv3-Tiny-PRN at the three lower resolutions. YOLOv3-Tiny at 864x480 produced a recall score of 1.00, and all other networks produced a recall of 0.93 or greater at a threshold of 0.15 and 0.83 or greater at a threshold of 0.25. All three object-detection networks at the tested resolutions are highly effective at detection of at least one hair fescue plant per image.

Table 3: Results for correct detection of at least one hair fescue plant per image.

Network	Resolution	Threshold = 0.15			Threshold = 0.25		
		Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
YOLOv3	1280x736	0.97	0.93	0.95	0.98	0.88	0.93
	1024x576	0.96	0.94	0.95	0.98	0.88	0.93
	960x544	0.96	0.95	0.95	0.98	0.88	0.93
	864x480	0.96	0.96	0.96	0.98	0.86	0.91
YOLOv3-Tiny	1280x736	0.97	0.97	0.97	0.97	0.91	0.94
	1024x576	0.96	0.97	0.96	0.97	0.93	0.95
	960x544	0.95	0.97	0.96	0.97	0.92	0.94
	864x480	0.93	1.00	0.96	0.96	0.94	0.95
YOLOv3-Tiny-PRN	1280x736	0.97	0.99	0.98	0.98	0.88	0.93
	1024x576	0.97	0.98	0.98	0.99	0.87	0.93
	960x544	0.97	0.96	0.97	0.99	0.86	0.92
	864x480	0.97	0.95	0.96	0.99	0.83	0.90

Similar to hair fescue, detection of at least one sheep sorrel plant produced greatly improved results compared to detection of all sheep sorrel plants (**Table 4**). Lowering the detection threshold yielded variable results on the F_1-scores. YOLOv3-Tiny-PRN improved at all four resolutions by an average of 0.06 while YOLOv3-Tiny performed better by an average of 0.01 at the original 0.25 threshold. The F_1-score for YOLOv3 showed no change at the two middle resolutions, increased by 0.01 at 864x480, and

decreased by 0.02 at 1280x736. The peak F_1-score (0.91) was achieved with YOLOv3 at 1280x736 and a threshold of 0.25, while the minimum F_1-score (0.68) was produced with YOLOv3-Tiny-PRN at 860x480 and a threshold of 0.25. All three networks except YOLOv3-Tiny-PRN at the three lowest resolutions had F_1-scores of 0.87 or higher at both thresholds.

Table 4: Results for correct detection of at least one sheep sorrel plant per image.

Network	Resolution	Threshold = 0.15			Threshold = 0.25		
		Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
YOLOv3	1280x736	0.83	0.96	0.89	0.88	0.94	0.91
	1024x576	0.86	0.94	0.90	0.91	0.89	0.90
	960x544	0.87	0.91	0.89	0.91	0.87	0.89
	864x480	0.89	0.88	0.88	0.92	0.83	0.87
YOLOv3-Tiny	1280x736	0.81	0.97	0.88	0.87	0.93	0.90
	1024x576	0.82	0.94	0.88	0.88	0.91	0.89
	960x544	0.82	0.94	0.88	0.89	0.90	0.89
	864x480	0.82	0.93	0.87	0.89	0.87	0.88
YOLOv3-Tiny-PRN	1280x736	0.84	0.89	0.86	0.88	0.78	0.82
	1024x576	0.82	0.79	0.80	0.88	0.65	0.75
	960x544	0.82	0.76	0.79	0.88	0.64	0.74
	864x480	0.82	0.69	0.75	0.86	0.56	0.68

Inference speed is also an important consideration for using CNNs to control spray applications. YOLOv3, YOLOv3-Tiny, YOLOv3-Tiny-PRN process 416x416 images at 46, 330, and 400 FPS respectively using a GTX 1080 Ti GPU (C.-Y. Wang et al., 2020). Sprayers used in the wild blueberry industry typically range up to 36.6 m in width, meaning that multiple cameras will be required to automate spray applications. The prototype research sprayer developed by Esau et al. (2018) used 9 cameras for capturing images along the spray boom. Assuming the single-image inference time scales proportionally with the number of cameras and one camera is required per nozzle, using YOLOv3 at 416x416

resolution would result in an inference speed of 5.11 FPS per camera in a 9-camera system. The higher resolution images used in fescue and sheep sorrel detection will also take longer to process. Additional testing will be required to confirm, but a framerate of 5.11 FPS on a GTX 1080 Ti GPU is not ideal for scaling to real-time spray applications. Using this GPU in a sprayer would be inconvenient because it is only available for desktop computers and has a 600 W system power requirement. Mobile processing hardware will take longer to process images, so using YOLOv3 for real-time spray may not be feasible. YOLOv3-Tiny and YOLOv3-Tiny-PRN process images 7.2 and 8.7 times faster respectively than YOLOv3. Considering their superior processing speed, YOLOv3-Tiny and YOLOv3-Tiny-PRN are more feasible options than YOLOv3 for real-time spray applications.

2.3.2 Classification of Images Containing Hair Fescue and Sheep Sorrel

The Darknet Reference network at 1280x736 resolution achieved the highest Top-1 accuracy for hair fescue classification with 96.25% on the validation dataset (**Table 5**). All networks at all resolutions scored above 90% except EfficientNet-B0 at 864x480 on the validation dataset. MobileNetV2 was the most consistent across different resolutions, with only a 1.13 percentage point difference between its best (95.63%) and worst (94.50%) Top-1 scores. Recall was higher than precision for each network at every resolution except MobileNetV2 and EfficientNet-B0 at 1280x736, while the recall of Darknet-Reference at this resolution was only slightly (0.01) higher than the precision. These results indicate that lower resolutions yield more false positive hair fescue classifications than false negatives. Given the high accuracy of these networks at lower resolutions, future work should involve further reducing the resolution to determine the minimum image size for accurate classification.

Table 5: Classification results for images with or without hair fescue.

Network	Resolution	Validation				Testing
		Precision	Recall	F_1-Score	Top-1 (%)	Top-1 (%)
Darknet Reference	1280x736	0.96	0.97	0.96	96.25	96.29
	1024x576	0.91	0.98	0.94	93.88	95.29
	960x544	0.93	0.98	0.95	95.25	96.14
	864x480	0.86	0.99	0.92	91.38	93.00
EfficientNet-B0	1280x736	0.96	0.95	0.96	95.50	95.71
	1024x576	0.90	0.96	0.93	92.75	97.29
	960x544	0.86	0.96	0.91	90.25	96.14
	864x480	0.83	0.95	0.88	87.38	95.57
MobileNetV2	1280x736	0.98	0.93	0.96	95.63	95.43
	1024x576	0.94	0.98	0.96	95.88	97.29
	960x544	0.95	0.97	0.96	95.75	98.14
	864x480	0.91	0.99	0.95	94.50	97.28

The best Top-1 validation accuracy for sheep sorrel classification (95.25%) was achieved by Darknet Reference at 1024x576 and 864x480 and by MobileNetV2 at 864x480 (**Table 6**). Darknet-Reference was the most consistent on the validation dataset, producing an F_1-score of 0.95 at all resolutions. An interesting observation is that the validation Top-1 scores of MobileNetV2 increased as the resolution was decreased. Decreasing the resolution results in the finer details of the sheep sorrel plants being removed, indicating that MobileNetV2 may be classifying the images on the colour and round overall shape. The results show that this works well when the network must choose between two very different classes but MobileNetV2 may struggle if it must differentiate between sheep sorrel and another weed with similar features. Another observation is that EfficientNet-B0 consistently performed better on the testing dataset than the validation dataset by an average of 10.34 percentage points. Darknet Reference and MobileNetV2 performed worse on the testing dataset by 1.78 and 2.94 percentage points, respectively. Like with fescue,

the accuracy of the CNNs at lower resolutions indicates that further testing should involve determine the minimum image size for accurate classification. Additionally, multiclass CNNs should be trained and tested to determine if they can distinguish between sheep sorrel and other similarly shaped weeds.

Table 6: Classification results for images with or without sheep sorrel.

Network	Resolution	Validation				Testing
		Precision	Recall	F_1-Score	Top-1 (%)	Top-1 (%)
Darknet Reference	1280x736	0.96	0.93	0.95	94.75	94.14
	1024x576	0.93	0.98	0.95	95.25	92.42
	960x544	0.93	0.97	0.95	95.00	92.42
	864x480	0.94	0.96	0.95	95.25	94.14
EfficientNet-B0	1280x736	0.97	0.87	0.92	92.13	96.57
	1024x576	0.98	0.74	0.84	86.25	98.00
	960x544	0.97	0.76	0.85	87.38	96.57
	864x480	0.97	0.63	0.76	80.88	96.86
MobileNetV2	1280x736	1.00	0.75	0.86	87.63	82.29
	1024x576	0.98	0.86	0.91	92.13	89.71
	960x544	0.96	0.91	0.94	93.88	92.71
	864x480	0.97	0.93	0.95	95.25	92.43

2.3.3 Effect of Training Dataset Size on Detection Accuracy

The CNNs were retrained for hair fescue identification with progressively smaller datasets and their F_1-scores were compared to understand the minimum requirements for training dataset size (**Figure 10**). For object-detection networks, the F_1-score for detecting at least one weed per image at a threshold of 0.15 was used. All networks except Darknet Reference consistently produced F_1-scores above 0.90 for all training dataset sizes from 472 to 3,780 images. F_1-scores for Darknet reference varied from 0.92 to 0.77. Interestingly, Darknet was consistently on the worst performing networks in this test, despite being one of the best when trained with 7,200 images. The results from the other

networks indicate that datasets of approximately 500 images can be used to train all tested CNNs except Darknet Reference for hair fescue detection in wild blueberry fields. This test should be performed with other datasets to confirm whether this is always the case for other weeds. If the results are consistent across other targets, this can greatly decrease the amount of time spent collecting images for training.

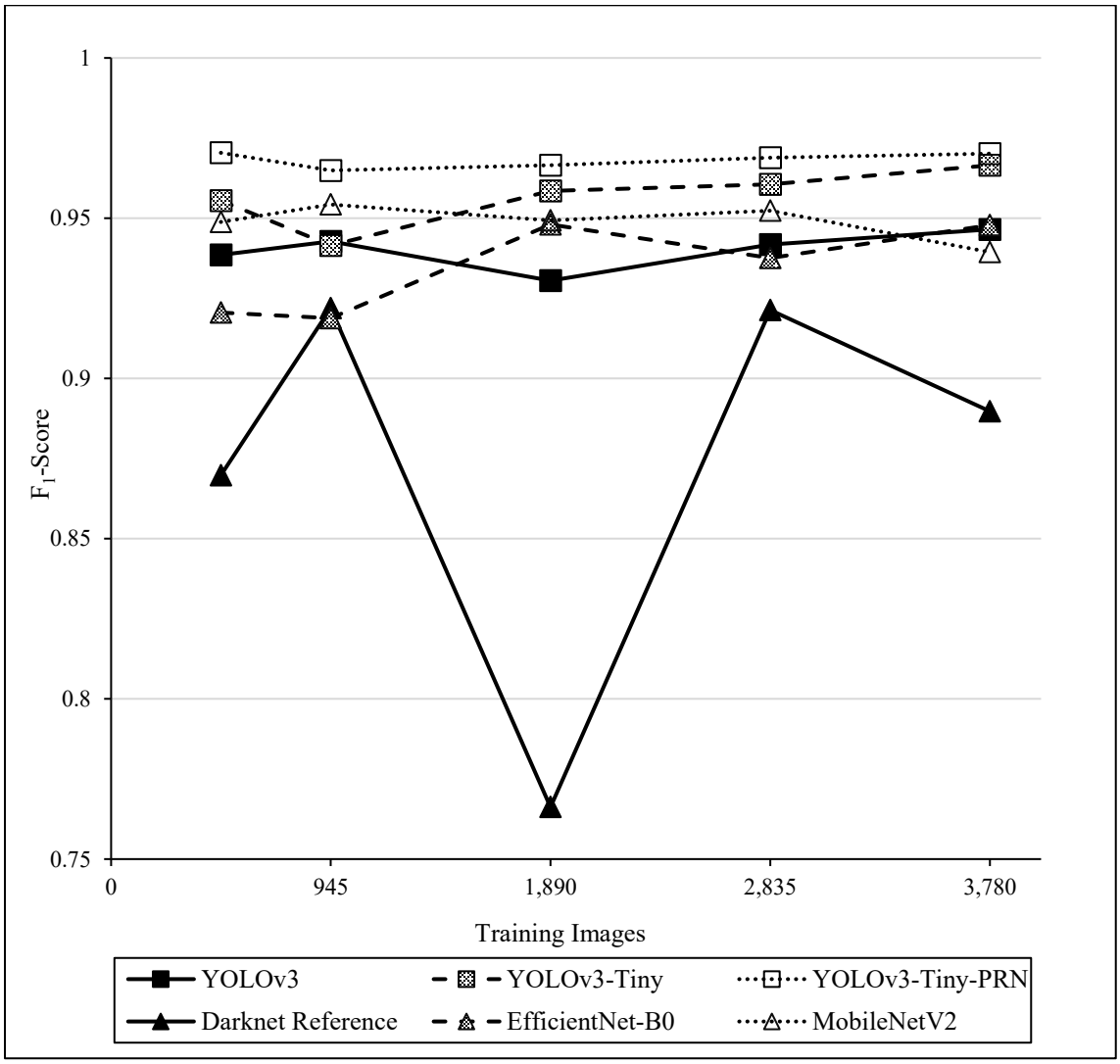

Figure 10: F_1-scores scores for object-detection CNNs at 1280x736 resolution when trained with different dataset sizes.

2.4 Conclusions

Limitations with previous machine vision systems for real-time spraying in wild blueberry were their inability to discriminate between weed species of the same colour or were not practical for scaling to other targets. Convolutional neural networks provide a promising solution to these problems, as they can successfully discriminate between different weeds and can be repurposed for other targets without manual feature extraction. The results of this study indicate that YOLOv3-Tiny is more suited than YOLOv3 and YOLOv3-Tiny-PRN due to its high level of accuracy in detection of both weeds and fast processing speed. For detection of at least one target weed per image, all three networks produced F_1-scores of at least 0.95 for hair fescue at 1280x736 resolution after lowering the detection threshold from the default 0.25 to 0.15. The results of changing the threshold were variable for sheep sorrel detection, with only small changes in the F_1-score. Object-detection results for sheep sorrel may be improved by labelling each individual sheep sorrel leaf or by selection regions of weed cover using semantic segmentation methods. Among image-classification networks, Darknet Reference produced the best overall results with F_1-scores of 0.92 or greater at all resolutions for classification of both weeds when trained with 7,200 images. MobileNetV2 and EfficientNet-B0 were more consistent than Darknet Reference with smaller dataset sizes, with F_1-scores above 0.92 with all training dataset sizes. Future work will involve testing the processing speed of all networks on a mobile GPU to determine if they can process images quickly enough for use in a real-time herbicide smart sprayer. The CNNs will also be tested on a moving sprayer, as the motion blur may limit their effectiveness. Using a CNN as part of a machine vision system in a smart sprayer could reduce herbicide use, resulting in major cost-savings for wild blueberry

growers. Field scouting software relying on CNNs to detect weeds could also be implemented into a smartphone application to aid growers with management decisions.

CHAPTER 3: FIELD EVALUATION OF HAIR FESCUE AND SHEEP SORREL DETECTION SYSTEMS

3.1 Introduction

Wild blueberries (*Vaccinium angustifolium* Ait.) are a perennial crop native to northeastern North America. The plants spread through rhizomes (Hall et al., 1979) and grow to a stem height of 5 to 30 cm (Farooque et al., 2014). Wild blueberries are grown in a two-year cycle, during which the plants are pruned by flail mowing or burning after the harvesting period in August and September of the second (crop) year. During the first (sprout) year after pruning, plant growth begins, and berry buds begin to regrow in August (Hall et al., 1979). The plants lay dormant through the winter, and growth continues during the crop year (Hall et al., 1979). Harvest begins when approximately 90% of the berries are ripe (Farooque et al., 2014). Wild blueberries were harvested with a hand rake prior to the introduction of a viable mechanical harvester by Doug Bragg in 1981 (Dale et al., 1994; Hall et al., 1983). Better management practices, including development of the mechanical harvester, resulted in the wild blueberry industry expanding in Canada (Farooque et al., 2014; Yarborough, 2004, 2012).

Weeds are a major yield limiting factor in wild blueberry production (K. I. N. Jensen & Yarborough, 2004; Yarborough, 2006; Yarborough & Bhowmik, 1993). Weeds in organic crops can be managed using tillage, crop rotation, and hand weeding (Klonsky, 2012). Tillage and crop rotation are not viable for the wild blueberry industry due to the perennial and rhizomatous nature of the crop (Hall et al., 1979). Hand weeding is prohibitively expensive due to labour costs (Fennimore et al., 2014), so a uniform application of liquid herbicides is typically performed to manage weeds in wild blueberry fields (Jensen & Yarborough, 2004; McCully et al., 1991). Hair fescue (*Festuca filiformis*

Pourr.) was the fourth most common weed in Nova Scotia wild blueberry fields in 2019 with a frequency and uniformity of 68% and 25% respectively (White, 2019). Hair fescue grows in tufts and is characterized by thin, green and tan coloured blades (**Figure 11**). The field uniformity of 25% indicates that hair fescue grows in patches of wild blueberry fields, rather than uniformly throughout each field. Hexazinone was commonly used to manage hair fescue until resistance developed from repeated applications (K. I. N. Jensen & Yarborough, 2004). Hair fescue in wild blueberry fields is currently best managed with pronamide (White & Kumar, 2017), which costs more than double the cost of hexazinone to spray on wild blueberry fields (Esau et al., 2019). Sheep sorrel (*Rumex acetosella* L.) was the most common weed in Nova Scotia wild blueberry fields in 2019 with a frequency and uniformity of 98% and 63% respectively (White, 2019). It is visually recognizable by its small, round arrow-shaped leaves which are green or red in colour (**Figure 11**). The measured uniformity of 63% stipulates that there are substantial sections of the fields which do not contain sheep sorrel. Pollen from sheep sorrel plants may increase the likelihood of botrytis blight (*Botrytis cinerea*) disease on wild blueberry leaves (Hughes et al., 2016), which can spread to the fruit buds if left unmanaged (Agriculture and Agri-Food Canada, 2016). Sheep sorrel has been managed using hexazinone (K. I. N. Jensen & Yarborough, 2004; Kennedy et al., 2010) and pronamide (Hughes et al., 2016) with mixed results. Spring applications of sulfentrazone have shown a reduction in sheep sorrel seedlings, indicating that this may be an effective future management option (White, 2019). Due to the intermittent nature of hair fescue and sheep sorrel in wild blueberry fields, spot application using a smart sprayer would reduce the overall volume of herbicide needed to manage these weeds.

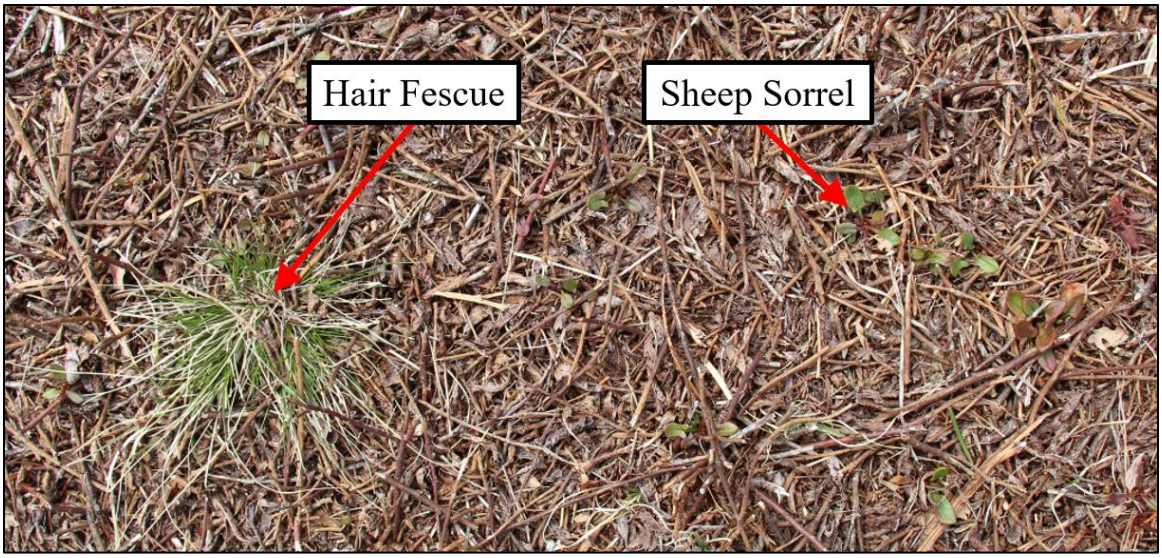

Figure 11: Hair fescue and sheep sorrel growing in a spout-year wild blueberry field during spring 2019 in Nova Scotia. The hair fescue tuft is characterized by its thin, green blades, while the sheep sorrel plants are characterized by their green and red arrow-shaped leaves.

Smart sprayers limit agrochemical application volume by only applying on areas of the field with weed cover (Esau et al., 2014, 2016, 2018; Hong et al., 2012; Rehman et al., 2018, 2019; Schieffer & Dillon, 2014). Esau et al. (2014) developed a spot targeting system based on green colour segmentation to detect weeds in wild blueberry fields. When used on a smart sprayer with cameras 1.2 m from the ground, the system resulted in 44.5% reduction in agrochemical usage compared to a basic sprayer (Esau et al., 2016). Further work with this smart sprayer resulted in herbicide savings of up to 78.5% (Esau et al., 2018). However, this system was limited by its inability to discriminate between different weeds of the same colour. Commercial smart sprayers GreenSeeker (Trimble Inc., 2020a) and WeedSeeker (Trimble Inc., 2020b) are available in other cropping systems and also work based on green colour detection. Colour co-occurrence matrices were used for real-time targeting of goldenrod (*Solidago* spp.) in wild blueberry fields (Rehman et al., 2019; Rehman et al., 2018). This method was effective but had to be designed specifically for goldenrod and was not easily scalable to other weeds.

Convolutional neural networks (CNNs) are an advanced form of machine vision which can successfully classify whole images or objects within an image (LeCun et al., 2015). They are trained using backpropagation (Rumelhart et al., 1986) with thousands of labelled pictures. CNNs intelligently identify visual features and find patterns associated with the target with minimal input from the user, making them easily adaptable for new targets. Images are typically processed at resolutions from 224x224 (Redmon, 2016; Sandler et al., 2018; Tan & Le, 2019) to 608x608 (Redmon & Farhadi, 2018; C.-Y. Wang et al., 2020), but this can be increased to improve clarity of visual features (Schumann et al., 2019; Tan & Le, 2019). CNNs have been used in agriculture for detecting weeds (Blue River Technologies, 2018; Sharpe et al., 2019; Yu, Schumann, et al., 2019; Yu, Sharpe, et al., 2019a, 2019b), detecting plant diseases (Fuentes et al., 2017; Venkataramanan, et al., 2019), monitoring plant growth and ripeness (Schumann et al., 2019; Tian et al., 2019), and monitoring livestock (Wu et al., 2020; Yang et al., 2018). Chapter 2 trained six CNNs using the Darknet framework (Redmon et al., 2020) to identify hair fescue and sheep sorrel in images of wild blueberry fields. The study concluded that the YOLOv3-Tiny CNN (Redmon & Farhadi, 2018) was highly effective for detecting these weeds in wild blueberry fields, and was a promising option for controlling spray applications from a smart sprayer.

This study used YOLOv3-Tiny to evaluate images captured at distances of 0.57 m to 1.29 m from target weeds with three digital colour cameras in wild blueberry fields in Nova Scotia, Canada. Images were captured in three commercial wild blueberry fields on three dates in May 2020 during the herbicide application timing interval. The results of this study provide valuable information for selecting cameras and spray boom height on smart sprayers. Using a CNN to control herbicide applications from a smart sprayer can reduce

the volume of agrochemical needed for field management, resulting in cost-savings for growers.

3.2 Materials and Methods

3.2.1 Image Collection at Varying Target Distances with Three Cameras

Images were captured on May 6, May 14, and May 25, 2020 in sprout year wild blueberry fields. The three dates were selected to correspond with the range of dates wild blueberry growers typically apply herbicides in the spring. The weather was overcast on May 6, sunny on May 14, and partly cloudy on May 25. Images were collected from three fields in Debert (45.4265°N, 63.4826°W), Folly Mountain (45.4829°N, 63.5755°W), and Portapique (45.4054°N, 63.6706°W), Nova Scotia from 9:00 am to 3:00 pm each day. Test sites were selected in each field by walking an inverted "W" pattern (Hughes et al., 2016; McCully et al., 1991; Thomas, 1985). The starting point was selected by walking 10 paces along the edge of the field, then 10 paces into the field, perpendicular to the original direction. Test plots were marked with flags along the "W" and randomly determined intervals of 5 to 10 paces created using Minitab 19 (Minitab, LLC, State College, PA, USA). Images were captured at each test plot at one of three possible heights, 0.57 m, 0.98 m, or 1.29 m, to account for variations in sprayer boom height. The 0.98 m and 1.29 m image heights did not vary significantly from image capture heights in chapter 2 (0.99 ± 0.09 m, 1.35 ± 0.07 m), while the 0.57 m height was significantly different (0.52 ± 0.04 m). The selected height for each test plot was also randomly determined with Minitab 19. This process was repeated until there were at least three target and three non-target plots at each height in each field for hair fescue and sheep sorrel (**Table 7**). A total of 83 plots were used: 29 at the Debert field, 23 at the Folly Mountain field, and 31 at the Portapique field.

Table 7: Number of target and non-target images collected for hair fescue and sheep sorrel. The images were captured three separate times on May 6, May 14, and May 25, 2020.

Field	Target	Height			Total
		0.57 m	0.98 m	1.29 m	
Debert	Hair Fescue	5	5	5	15
	Not Hair Fescue	4	4	6	14
	Sheep Sorrel	3	3	3	9
	Not Sheep Sorrel	6	6	8	20
Folly Mountain	Hair Fescue	3	5	3	11
	Not Hair Fescue	5	3	4	12
	Sheep Sorrel	4	4	4	12
	Not Sheep Sorrel	4	4	3	11
Portapique	Hair Fescue	6	4	7	17
	Not Hair Fescue	6	4	4	14
	Sheep Sorrel	8	4	5	17
	Not Sheep Sorrel	4	4	6	14

Three cameras were used to take images at each test plot. A Logitech c920 USB 2.0 webcam (Logitech International S.A., Lausanne, Switzerland) was mounted to a tripod with an extension arm with the camera lens pointed toward the ground (**Figure 12**). The camera was connected to USB 3.1 port on an MSI Workstation laptop (WS65 9TM-1410CA, Micro-Star International Co., Ltd, New Taipei, Taiwan) through a 2 m USB 3.0 extension cable (AmazonBasics HL-007250, Amazon.com, Inc., Seattle, WA, USA), and images were saved using Logitech Capture at 1920x1080 resolution. All settings in Logitech Capture except sharpness were left at their default values. Sharpness was reduced from 128 to 95 to prevent image tearing and artifacts (**Figure 13**). The autofocus sporadically changed the focus setting, so a manual focus of infinity was used. This camera was selected for being a low-cost consumer camera which easily interfaces with Windows (Microsoft Corporation, Redmond, WA, USA) and had been previously used in a smart sprayer (Partel et al., 2019). Eight Logitech c920 cameras were tested to make sure that the

image glitches and sporadic autofocus changes were not limited to a single faulty model. All eight Logitech c920 cameras exhibited similar behaviour. A Canon T6 DSLR camera (EOS Rebel T6, Canon Inc., Tokyo, Japan) and an LG G6 smartphone (G6-H873, LG Electronics Inc., Seoul, South Korea) were used to capture images in the same orientation with their respective lenses placed directly next to the lens of the Logitech camera (**Figure 12**). These cameras were selected for their subjectively clearer images and greater colour depth. The Canon T6 and LG G6 captured images at resolutions of 5184x3456 and 4160x3120, respectively. Images were captured with the Canon T6 and LG G6 using autofocus and without zoom.

Figure 12: Field experimental setup showing image capture in the Portapique field on May 25, 2020. A tripod with an extension arm was used to hold the Logitech c920 camera at a position of 0.98 m between the lens and the ground while the laptop captured the image (L). The Canon T6 camera was held with the lens directly beside the Logitech c920 at the same height for image capture (R).

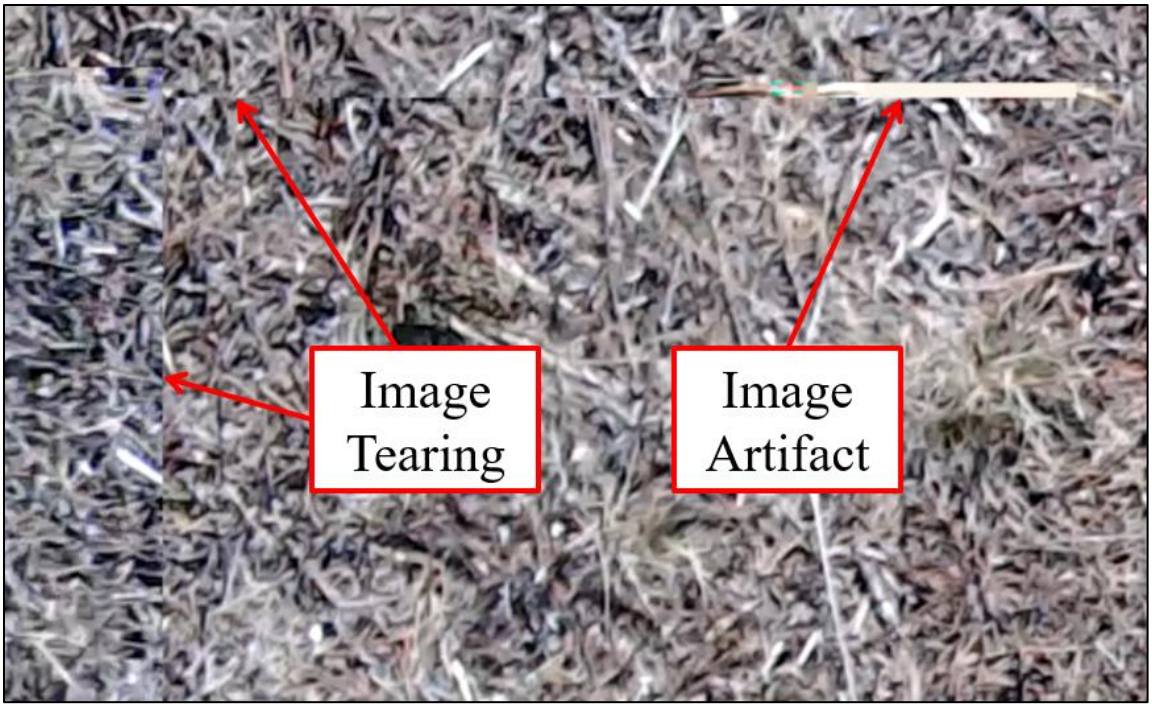

Figure 13: Examples of image tearing and artifacts seen in the Logitech c920 images when the default sharpness setting was used.

3.2.2 Target Dimension Measurements

A sample of hair fescue and sheep sorrel dimensions were measured in the Debert field on May 1, 2020 at the sampling plots along the "W" pattern. Measurements were recorded from up to three instances of each weed at each test location, pending availability, resulting in 25 measurements for hair fescue and 24 measurements for sheep sorrel. A ruler was held above the plants to record the length and width of each weed. The length was defined as the largest horizontal dimension of the weed and the width was defined as the widest measurement perpendicular to the length measurement (**Figure 14**). To understand the field of view (FOV) of the images, a measuring tape was placed on the ground underneath the Logitech c920 camera to measure the physical image size at each of the

three heights. Understanding the sizes of these weeds will help with interpretation of the detection results from the CNN.

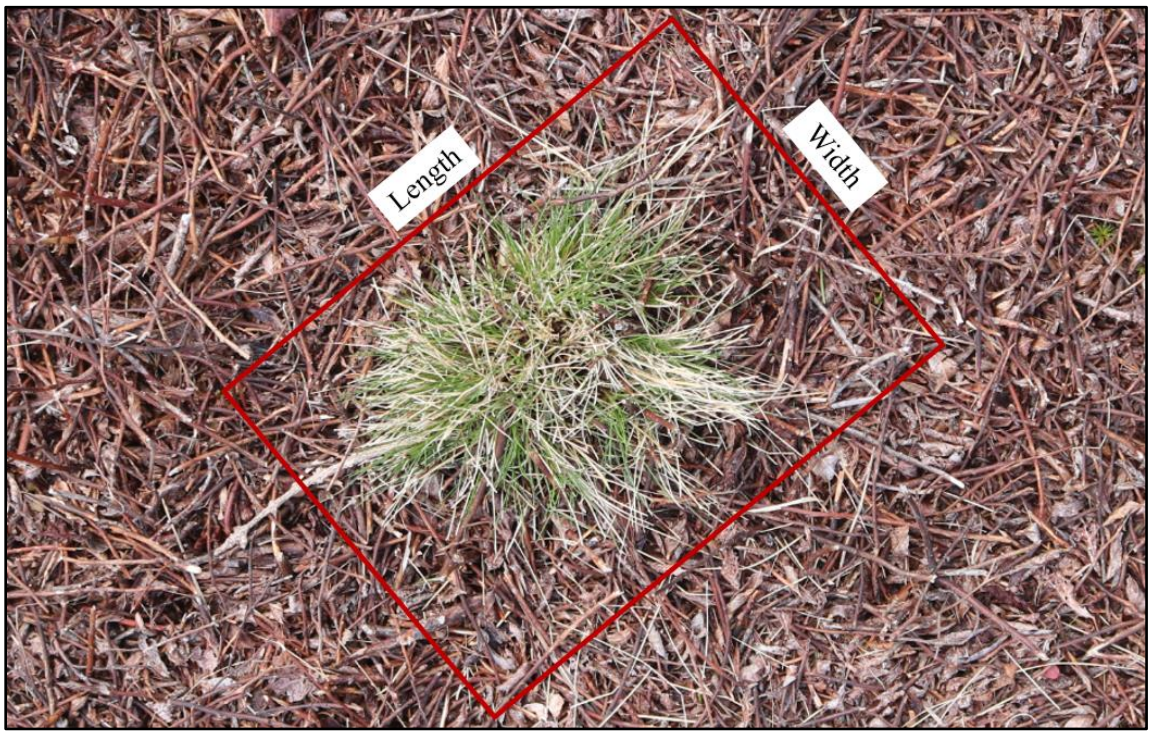

Figure 14: Example of the measurement method for a hair fescue tuft, with the red bounding box showing the length and width of the tuft. The base image was captured on April 26, 2019 with the Canon T6 camera at a sprout-year field in Debert, NS (45.4381°N, 63.4534°W). The image also shows branches and leaves from wild blueberry plants after mechanical flail mowing.

3.2.3 Image Processing

The field images were organized by date, field, camera, height, and targets on the MSI Workstation Laptop with an Intel Core i9-9980H central processing unit (CPU, Intel Corporation, Santa Clara, California, USA) and an Nvidia RTX Quadro 5000 graphics processing unit (GPU, Nvidia Corporation, Santa Clara, California, USA) for analysis. The YOLOv3-Tiny CNN running on the Darknet framework with the trained weights from chapter 2 was used to detect hair fescue and sheep sorrel in the field images. The images

were initially run as entire batches to determine the optimal image resolution and detection threshold for detection of the two weeds. The internal network resolutions tested were 1280x736 and 864x480, with an initial threshold of 0.15. The results of each resolution and threshold combination were evaluated on their effectiveness of detecting at least one target weed per image using the precision, recall, and F_1-score metrics (Sokolova & Lapalme, 2009). These scores are calculated based on the number of true positive (*tp*), false positive (*fp*), and false negative (*fn*) detection of targets. Precision is ratio of true positives to all detections:

$$Pr = \frac{tp}{tp + fp} \tag{3}$$

Recall is the ratio of true positives to all actual targets:

$$Re = \frac{tp}{tp + fn} \tag{4}$$

F_1-score is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{Pr \cdot Re}{Pr + Re} \tag{5}$$

The precision was observed to be higher than the recall in all cases at the threshold of 0.15, so two lower thresholds, 0.10 and 0.05, were also tested. The lowering the threshold increases the number of true and false positive detections, thus lowering precision and increasing recall. A resolution of 1280x736 and threshold of 0.05 produced the highest F_1-score for both weeds and was used for the rest of the analysis.

3.2.4 Experimental Design

Field images were tested for detecting hair fescue and sheep sorrel with the YOLOv3-Tiny CNN using the Darknet framework and trained weights from chapter 2. The F_1-score for detection of at least one target weed per image was calculated for each combination of date, field, camera, and lens height. An analysis of variance was done using Minitab 19 to determine the significant main and interaction effects. The main effects of camera selection, image height, and field, and the three-way interaction between these effects were significant for hair fescue. Further analysis for hair fescue was done in a 3x3 factorial arrangement of lens height (0.57 m, 0.98 m, 1.29 m) and camera (Canon T6, LG G6, Logitech c920) in a randomized complete block design for each field. The mean F_1-score for interaction effect of camera and lens height in each field was calculated and the mean comparisons generated with Tukey's pairwise comparison in Minitab 19. The analysis for sheep sorrel was done in a 3x2 factorial arrangement with the same three lens heights but omitted the results from the Logitech c920 camera due to insufficient data from this camera.

3.3 Results and Discussion

3.3.1 Determination of Optimal Resolution and Threshold

F_1-scores for detection of at least one hair fescue plant per image varied between 0.80 and 0.81 when changing the threshold and resolution (**Table 8**). The highest precision score (0.85) was produced at 1280x736 resolution at the threshold of 0.15, while the lowest precision score (0.78) was produced with both resolutions at the 0.05 threshold. Recall peaked (0.85) at 1280x736 resolution and 0.05 threshold, although it only decreased by 0.01 when the resolution was changed to 864x480. The general trend for both target weeds was that decreasing the threshold increased the recall at the expense of precision. The key

factor for sheep sorrel detection was maximizing the resolution. The networks were trained at 1280x736 resolution, which may have influenced the results. The F_1-scores for sheep sorrel were higher at 1280x736 than 864x480 by an average of 0.07, with the peak F_1-score (0.67) being produced at 0.05 threshold. A resolution of 1280x736 and a threshold of 0.05 was determined to be the optimal parameter combination for both weeds. The peak F_1-scores in this test were lower than the F_1-scores produced in chapter 2 for hair fescue (0.97) and sheep sorrel (0.90). The effects of camera selection are examined in this study, but another factor which influenced results may have been the training dataset. The images were collected by personnel walking through fields and manually scouting for target weeds. The personnel could have been more inclined to walk towards larger, more visible weeds when creating the training dataset. The "W" sampling method used in this study has less bias and should produce a better representation of the hair fescue and sheep sorrel present in wild blueberry fields used.

Table 8: Precision, recall, and F_1-scores for detection of at least one hair fescue or sheep sorrel target per image at varying detection threshold and resolution.

Threshold	Resolution	Hair Fescue			Sheep Sorrel		
		Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
0.05	864x480	0.78	0.84	0.81	0.72	0.57	0.63
	1280x736	0.78	0.85	0.81	0.68	0.66	0.67
0.10	864x480	0.81	0.79	0.80	0.73	0.48	0.58
	1280x736	0.84	0.79	0.81	0.71	0.59	0.65
0.15	864x480	0.85	0.75	0.80	0.74	0.40	0.52
	1280x736	0.89	0.74	0.80	0.76	0.53	0.63

3.3.2 Measurement of Hair Fescue and Sheep Sorrel Targets

The mean length and width of hair fescue tufts at the Debert site were 54.6 ± 15.9 mm and 42.6 ± 13.3 mm, respectively (**Table 9**). The mean length and width of sheep sorrel

were much smaller, at 11.0 ± 1.7 mm and 6.0 ± 0.9 mm, respectively. The 95% C.I. for the hair fescue measurements were much larger than for the sheep sorrel measurements, indicating more variability in the size of hair fescue tufts. The physical measurement represented by an individual pixel varied from 0.60 mm to 1.35 mm for 1280x736 images, and 0.89 mm to 2.00 mm for 864x480 images (**Table 10**). At both resolutions, the finer features of sheep sorrel may not be clear at higher image heights. With pixel sizes of 1.35 mm and 2.00 mm, the average width of a sheep sorrel leaf in an image was 4 and 3 pixels, respectively. Higher resolutions may be necessary for accurate detection of sheep sorrel.

Table 9: Mean measurements of hair fescue and sheep sorrel plants at the Debert Site on May 1, 2020.

Target Weed	Dimension	Mean Measurement (mm)
Hair Fescue	Length	54.6 ± 15.9
	Width	42.6 ± 13.3
Sheep Sorrel	Length	11.0 ± 1.7
	Width	6.0 ± 0.9

Table 10: FOV of the Logitech c920 camera at each tested image height. The size of the field represented by each pixel at 1280x736 and 864x480 resolution was calculated based on the FOV.

Height (m)	FOV (m)		Pixel Size, 1280x736 (mm)	Pixel Size, 864x480 (mm)
	Length	Width		
0.57	0.77	0.43	0.60	0.89
0.98	1.30	0.73	1.02	1.51
1.29	1.72	0.97	1.35	2.00

3.3.3 Effects of Camera Selection and Image Height

For hair fescue detection, the main effects of camera selection, image height, and field selection, the two-way interaction effect between image height and field, and the three-way interaction effect between camera selection, image height, and field were

significant ($p < 0.05$). The main effect of day, the two-way interaction effects between day and camera, day and height, day and field, camera and height, camera and field, and the three-way interaction between day, camera, and height were not significant. The Tukey's pairwise comparison for interaction between image height and camera selection showed that the best option for the Debert field was the LG G6 at 0.98 m (F_1-score = 0.972) (**Table 11**). The Canon T6 at 0.98 m (F_1-score = 0.819) was the second-best option for this field, but was not significantly different from any lesser performing combinations. In the Folly Mountain field, the best detection results came from images captured at 0.98 m with the Canon T6 and LG G6 cameras (F_1-score = 0.972). The images from the LG G6 at 1.29 m were produced the best detection in the Portapique field (F_1-score = 0.918), but they were not significantly different from detection results from all other combinations in this field. The only scenario where camera selection produced significantly different results was in the Debert field at a height of 0.98 m. The LG G6 performed the best (F_1-score = 0.972), followed by the Canon T6 (F_1-score = 0.819), and the Logitech c920 (F_1-score = 0.626). The LG G6 and Logitech c920 cameras varied significantly from each other, but not from the Canon T6. Approximately 70% of the images in the original training dataset were captured at 0.99 ± 0.09 m, which may have contributed to the high level of accuracy at the 0.98 m height.

Table 11: Effect of lens height and camera selection on mean F_1-score for detection of hair fescue plants using the YOLOv3-Tiny weights trained in chapter 2. Test images were captured in Nova Scotia on May 6, May 14, and May 25, 2020.

Height (m)	Camera	Field					
		Debert		Folly Mountain		Portapique	
0.57	Canon T6	0.742	CD	0.697	CD	0.839	ABCD
0.57	LG G6	0.667	CD	0.724	CD	0.839	ABCD
0.57	Logitech c920	0.608	D	0.750	BCD	0.773	ABCD
0.98	Canon T6	0.819	ABCD	0.972	A	0.830	ABCD
0.98	LG G6	0.972	A	0.972	A	0.766	ABCD
0.98	Logitech c920	0.626	D	0.966	AB	0.783	ABCD
1.29	Canon T6	0.800	ABCD	0.724	CD	0.864	ABCD
1.29	LG G6	0.811	ABCD	0.789	ABCD	0.918	ABC
1.29	Logitech c920	0.817	ABCD	0.761	ABCD	0.890	ABCD

During calculation of F_1-scores for sheep sorrel detection, 19 of 27 combinations of height, date, and field for images captured with the Logitech c920 did not return a result. There were zero true positive detections in these scenarios, thus creating precision and recall scores of zero, which resulted in the F_1-scores being incalculable due to a division by zero. The maximum average F_1-score from images captured with the Logitech c920 camera was 0.444 (**Table 12**). The features in images captured with the Logitech c920 were blurrier due to the reduced sharpness, and there was less contrast between colours (**Figure 15**). Preprocessing images from the Logitech c920 to accentuate green hues may improve results for sheep sorrel and hair fescue detection but would likely have a negative impact on processing speed. The Canon T6 and LG G6 cameras were used to collect training images, while the Logitech c920 was not used. This may have resulted in the CNN being more biased towards detecting weed instances in the Canon T6 and LG G6 images.

Table 12: Sheep sorrel detection results from YOLOv3-Tiny on images captured with the Logitech c920 camera at three wild blueberry fields in Nova Scotia.

Date	Field	Height (m)	F_1-Score
06-May	Portapique	0.57	0.222
		1.29	0.444
14-May	Debert	0.57	0.286
		1.29	0.333
	Folly Mountain	0.78	0.333
	Portapique	0.57	0.222
		1.29	0.444
25-May	Debert	0.57	0.333

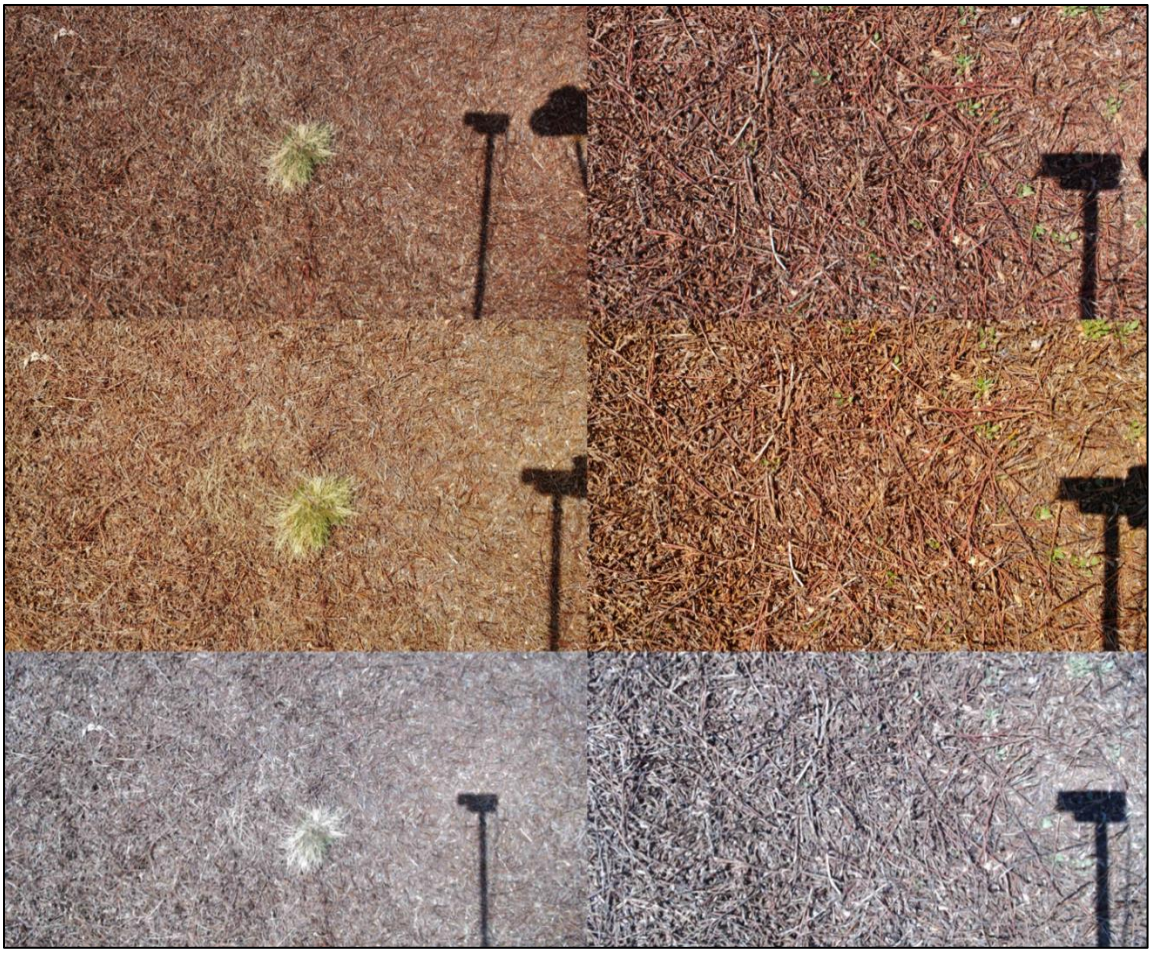

Figure 15: Sample of field images captured in the Folly Mountain field on May 6th, 2020. Pictures in the left column were captured at a height of 0.98 m and have a hair fescue tuft in the centre of the image. Pictures in the right column were captured at 0.57 m and have sheep sorrel leaves dispersed throughout the images. Images in the top row were captured with the Canon T6 camera, the middle row were captured with the LG G6 smartphone, and the bottom row were captured with the Logitech c920.

Significant effects ($p < 0.05$) for sheep sorrel detection were the main effects of image height and field, and the two-way interaction between field and image height. The main effect of camera, the two-way interaction effects between day and camera, day and height, day and field, camera and height, camera and field, and all three-way interaction effects were not significant. The best height for capturing images in the Debert field was 0.98 m, with images from both cameras producing an F_1-score of 0.833 (**Table 13**). The

other combinations of camera and image height did not vary significantly. In the Folly Mountain field, the best detection results were produced with images captured at 0.57 m with both cameras (F_1-score = 0.944, 0.933), but other combinations were not significantly different. Similar results were produced with images from the Portapique field, with images captured at 0.57 m using both cameras producing the best results (F_1-score = 0.884, 830), while the results at other heights were not significantly different. The small size of the sheep sorrel leaves may be contributing to the reduced accuracy in images captured from higher positions.

Table 13: Effect of lens height and camera selection on mean F_1-score for detection of sheep sorrel plants using the YOLOv3-Tiny weights trained in chapter 2. Test images were captured in Nova Scotia on May 6, May 14, and May 25, 2020.

Height (m)	Camera	Field		
		Debert	Folly Mountain	Portapique
0.57	Canon T6	0.429 BC	0.933 A	0.884 AB
0.57	LG G6	0.429 BC	0.944 A	0.830 ABC
0.98	Canon T6	0.833 ABC	0.833 ABC	0.756 ABC
0.98	LG G6	0.833 ABC	0.722 ABC	0.750 ABC
1.29	Canon T6	0.356 C	0.889 AB	0.649 ABC
1.29	LG G6	0.411 BC	0.833 ABC	0.722 ABC

3.4 Conclusions

The higher resolution, 1280x736, with the lowest threshold, 0.05, produced the best results for detecting sheep sorrel with the YOLOv3-Tiny CNN producing a peak F_1-score of 0.67 across all images captured in three fields in Nova Scotia. These parameters had little effect on the F_1-scores for hair fescue detection, which were consistently 0.80 or 0.81. These results were lower than the validation scores produced when the networks were trained, which may be the result of bias in the original image dataset. The small size of

sheep sorrel leaves indicates the higher resolution may have been necessary to adequately represent their visual features. Camera selection had minimal effect on hair fescue detection except in the Debert field at a height of 0.98 m. The Logitech c920 camera was not viable for sheep sorrel detection, as 19 of 27 parameter combinations resulted in zero detections. This may have been due to either lower quality images compared to the Canon T6 and LG G6 or because images from the Logitech c920 were not used to train the CNN. A lens height of 0.57 m produced the best results for sheep sorrel in two out of three fields. Preprocessing images to accentuate the green colours may cause the sheep sorrel to be more visible, potentially improving detection results. This would add another step to image processing, potentially reducing processing speed. Results from the LG G6 camera indicate that the quality of smartphone pictures is adequate for identifying hair fescue and sheep sorrel in field images. Future work should involve selecting a high-quality camera for use on a smart sprayer and collecting an unbiased image dataset for retraining the CNNs. Preprocessing techniques should also be examined for their effect on CNN accuracy and their impact on processing speed. Additionally, the CNNs should be tested for use in real-time on a smartphone, to allow wild blueberry growers to identify hair fescue and sheep sorrel in their fields. Using a CNN to target hair fescue, sheep sorrel, and other weeds in wild blueberry fields on a smart sprayer can reduce herbicide use and create cost-savings for growers.

CHAPTER 4: PROCESSING REQUIREMENTS FOR INTEGRATION OF CONVOLUTIONAL NEURAL NETWORKS ON A SMART SPRAYER

4.1 Introduction

Wild blueberries (*Vaccinium angustifolium* Ait.) are a perennial crop native to northeastern North America with a total production of over 108 million kg in 2019 (Robinson, 2020). In 2017, the crop contributed over $100 million to Nova Scotia's economy, including $65.9 million in exports (Wild Blueberry Producers Association of Nova Scotia, 2018). Production occurs in a two-year cycle with the flower buds beginning to grow from August to October in the first (sprout) year before lying dormant during the winter. The plants continue growing in the second (crop) year with fruit being harvested in August and September when approximately 90% of the berries are ripe (Farooque et al., 2014). The bare branches are pruned by flail mowing or burning after harvesting, restarting the growth cycle (Hall et al., 1979). Wild blueberries are desired for their health benefits, including anti-aging and anti-inflammatory properties (Beattie et al., 2005), and high antioxidant content (Kay & Holub, 2002) which helps reduce the risk of cardiovascular disease and cancer (Lobo et al., 2010).

Weeds, a major yield limiting factor in wild blueberry production (K. I. N. Jensen & Yarborough, 2004; Yarborough, 2006; Yarborough & Bhowmik, 1993), are traditionally managed with application of liquid herbicides (K. I. N. Jensen & Yarborough, 2004; McCully et al., 1991). Hair fescue (*Festuca filiformis* Pourr.) and sheep sorrel (*Rumex acetosella* L.) are the fourth and first most common weeds in Nova Scotia wild blueberry fields, respectively (White, 2019). Pronamide, which is currently used to manage hair fescue (White & Kumar, 2017), costs $413.65 ha^{-1} for a uniform application (Esau et

al., 2019). Hexazinone, used for sheep sorrel management (K. I. N. Jensen & Yarborough, 2004; Kennedy et al., 2010), costs \$168.03 ha^{-1} to uniformly apply (Esau et al., 2019). Hair fescue and sheep sorrel had field uniformities of 25% and 63% respectively in 2019 (White, 2019), which presents an opportunity for increased herbicide application efficiency using a variable rate smart sprayer. Smart sprayers use sensors to intelligently select which areas of a field to apply agrochemicals, reducing the volume of agrochemical needed for field management. Commercial options available for other cropping systems such as See & Spray (Blue River Technologies, 2018), GreenSeeker (Trimble Inc., 2020a), WeedSeeker (Trimble Inc., 2020b), and AiCPlus (Agrifac Machinery B.V., 2016) are not suitable for wild blueberry management due to the highly variable topography of wild blueberry fields.

Smart sprayers for wild blueberry management have relied on ultrasonic sensors to detect plant height (Zaman et al., 2011) or image data from cameras to detect foliage (Esau et al., 2014, 2016, 2018; Rehman et al., 2018, 2019). The ultrasonic sensors detected weeds taller than the wild blueberry plant canopy but could not detect weeds at the same height or shorter than the canopy (Zaman et al., 2011). A smart sprayer relying on green colour segmentation could successfully isolate green foliage from blueberry branches and bare ground (Esau et al., 2014, 2016, 2018). This resulted in herbicide savings of up to 78.5% (Esau et al., 2018), but the algorithm could not discriminate different weed species containing green. This sprayer used 27 spray nozzles and 9 cameras equally spaced along a 13.7 m boom and travelled at a maximum speed of 1.77 m s^{-1} (Esau et al., 2018). Another imaging system relying on colour co-occurrence matrices was created to detect goldenrod (*Solidago* spp.) in wild blueberry fields, but is not easily scalable to other weeds (Rehman et al., 2018, 2019). This sprayer included 8 nozzles and 4 cameras equally spaced along a

6.1 m boom (Rehman et al., 2019). The smart sprayers relying on imaging systems included graphical user interfaces (GUIs) for operators to control spray settings and view images from the cameras in real-time (Esau et al., 2014, 2018; Rehman et al., 2019). The GUIs were designed for use on the Windows operating system (Microsoft Corporation, Redmon, WA, USA), so commercial deployment of these sprayers would require paid Windows licences. A cross-platform GUI which also works on Linux-based operating systems such as Ubuntu (Canonical Ltd., London, UK) would remove this additional cost.

Deep learning convolutional neural networks (CNNs) are a novel processing technique used to automatically classify images or objects within images in real-time (LeCun et al., 2015). Datasets with thousands of images labelled according to their classification are used to train a CNN to classify new, unlabelled images through backpropagation (Rumelhart et al., 1986). Image processing using CNNs has been used in various aspects of agriculture since 2015 (Kamilaris & Prenafeta-Boldú, 2018), including classification of weeds in strawberry fields (Sharpe et al., 2019), various vegetables (Sharpe et al., 2020), turfgrasses (Yu, Sharpe, et al., 2019a, 2019b), and ryegrasses (Yu, Schumann, et al., 2019).

In chapter 2, three object-detection and three image-classification CNNs were trained to identify hair fescue and sheep sorrel in images of wild blueberry fields at four network resolutions from 1280x736 to 864x480 pixels. The CNNs accurately identified both target weeds, but the conclusion noted that the processing speed of the CNNs should be tested to determine if they were viable for use in a real-time smart sprayer. The three object-detection CNNs, YOLOv3, YOLOv3-Tiny (Redmon & Farhadi, 2018), and YOLOv3-Tiny-PRN (C.-Y. Wang et al., 2019), can process 416x416 resolution images at

46, 330, and 400 frames per second (FPS) respectively (C.-Y. Wang et al., 2020) on an Nvidia GTX 1080 Ti graphics processing unit (GPU, Nvidia Corporation, Santa Clara, CA, USA). The three image classification networks used were Darknet Reference (Redmon, 2016), EfficientNet-B0 (Tan & Le, 2019), and MobileNetV2 (Sandler et al., 2018). Darknet Reference operates at 345 FPS at a resolution of 224x224 on an Nvidia Titan X GPU (Redmon, 2016). Most consumer-grade Nvidia GPUs currently available contain between 6 GB and 11 GB (Nvidia Corporation, 2019, 2020b) of video random access memory (vRAM) needed to process CNNs.

This study examined the processing speed and vRAM requirements when processing multiple instances of the CNNs on a mobile GPU to determine if they would be viable for controlling spray applications in real-time. The three object-detection and three image-classification CNNs trained in chapter 2 were each tested at four resolutions ranging from 864x480 to 1280x736 pixels to process video frames from up to eight cameras simultaneously. A cross-platform GUI was developed in the Python programming language to control and view CNNs from multiple live camera feeds for use on a smart sprayer. Using CNNs to control spray applications will lead to greater herbicide application efficiency, resulting in major cost savings for wild blueberry producers.

4.2 Materials and Methods

4.2.1 Processing Hardware

An MSI workstation laptop (WS65 9TM-1410CA, Micro-Star International Co., Ltd, New Taipei, Taiwan) running Windows 10 Pro (Microsoft Corporation, Seattle, WA, USA) was used for testing the neural networks and developing the GUI. The laptop contained an Intel Core i9-9980H central processing unit (CPU, Intel Corporation, Santa

Clara, CA, USA) running at 2.30 GHz with 64 GB of system random access memory (RAM). The laptop's GPU, an Nvidia Quadro RTX 5000 Max-Q with a core clock speed of 600 MHz, boost clock speed of 1350 MHz, and 16 GB of vRAM was used to process the neural networks.

4.2.2 Speed and Memory Requirements for Processing Multiple Camera Streams with Convolutional Neural Networks

A version of the Darknet framework compatible with Windows (Redmon et al., 2020) was installed on the laptop and compiled with CUDA (Compute Unified Device Architecture, v10.0, Nvidia Corporation, Santa Clara, CA, USA), the CUDA Deep Neural Network Library (cuDNN, v7.6.4, Nvidia Corporation), and OpenCV (v3.4.9, The OpenCV Team). Eight Logitech c920 USB 2.0 cameras (Logitech International S.A., Lausanne, Switzerland) were connected to the laptop and the video captured by each camera was processed using a separate instance of the Darknet framework. Four of these cameras were connected through USB 3.1 ports on the laptop. The remaining four cameras were connected to the laptop via ethernet through a Raspberry Pi 3 Model B+ (The Raspberry Pi Foundation, Cambridge, UK) running motionEyeOS (Crisan, 2019) (**Figure 16**). A USB hub could not be used to connect all eight cameras due to bandwidth limitations of USB 2.0. Object-detection networks YOLOv3, YOLOv3-Tiny, and YOLOv3-Tiny-PRN were tested with 1, 2, 4, 6, and 8 cameras running simultaneously at 1920x1080 resolution. The networks were processed with internal resolutions of 1280x736, 1024x576, 960x544, and 864x480 with a batch size of 1. This test took place indoors instead of in-field conditions, so the CNNs were loaded using weights trained on the COCO dataset (Lin et al., 2014) by their respective authors rather than the weights trained to detect hair fescue and sheep sorrel in chapter 2. The average FPS for each camera reported by Darknet, and

the total amount of vRAM needed, as reported by Task Manager, in each test were recorded. This test was repeated using with identical parameters using the image-classification networks Darknet Reference, EfficientNet-B0, and MobileNetV2, loaded with weights trained on the ImageNet dataset (Deng et al., 2009) by their respective authors.

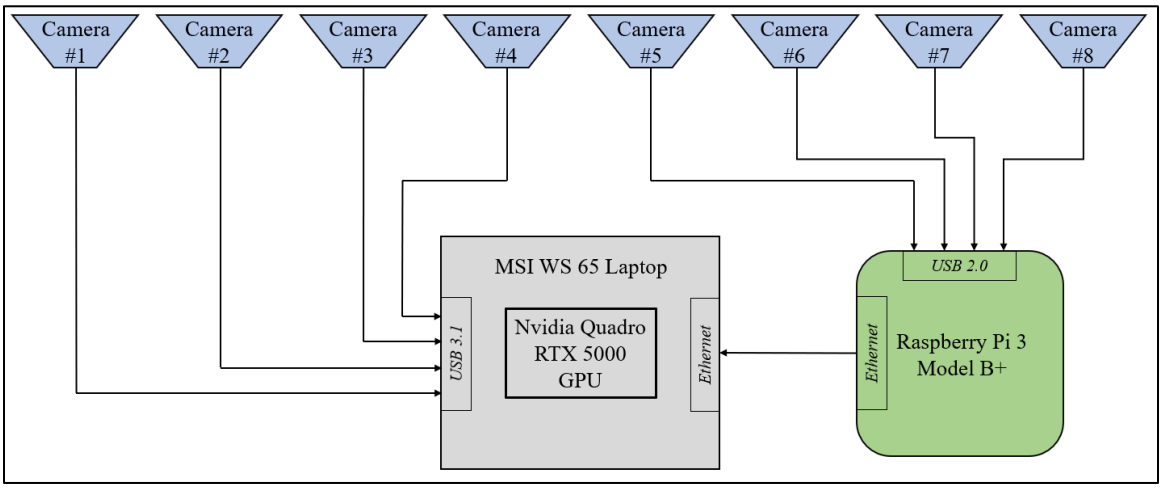

Figure 16: Block diagram of 8-camera setup used for testing. Four Logitech c920 cameras were connected to the laptop through USB 3.1 ports. A Raspberry Pi 3 Model B+ was connected to the laptop via ethernet to connect four additional Logitech c920 cameras.

4.2.3 Development of a Graphical User Interface for Controlling Convolutional Neural Networks on a Smart Sprayer

A cross-platform GUI was developed in the Python programming language (v3.8.2, Python Software Foundation, Wilmington, DE, USA) to control CNNs on a smart sprayer. The GUI was designed for an 8-camera, 8-nozzle sprayer with all cameras attached to a central computer for processing. Functions from the Tkinter library were used to create user input options and display CNN outputs. The GUI allows the user to select whether to use an object-detection or image-classification CNN for controlling spray applications. YOLOv3-Tiny and Darknet Reference were selected as the object-detection and image-

classification networks, respectively for their high level of accuracy and fast processing speeds. The user controls the internal network resolution, threshold for the object-detection network, and the target weed for spraying. The currently supported target weeds are hair fescue and sheep sorrel, which use the weights trained in chapter 2. The GUI was tested in Windows 10 Pro on the laptop and Ubuntu 16.04 LTS (Canonical Ltd., London, UK) on a custom-built PC with an Intel Core-i9 7900X CPU running at 3.3 GHz with 32 GB of RAM and Nvidia RTX 2080 Ti GPU clocked at 1650 MHz.

4.3 Results and Discussion

4.3.1 Memory Use and Processing Speed for Object-Detection Networks

The amount of vRAM needed to process multiple object-detection CNNs was found to scale linearly with the number of cameras being processed (**Figure 17**). The 16 GB of vRAM in the RTX 5000 GPU was enough for processing eight CNNs simultaneously at all resolutions except 1280x736 with YOLOv3. Extrapolating the memory measurements for YOLOv3 at 1280x736 linearly gives an estimated vRAM requirement of 18.4 GB for 8 cameras. YOLOv3-Tiny and YOLOv3-Tiny-PRN required 6.4 GB and 6.1 GB of vRAM, respectively, to process 8 video streams at 1280x736. Lowering the processing resolution resulted in a reduction of vRAM use for all three CNNs. YOLOv3 saw a 35.5% reduction in vRAM use when the processing resolution was changed from 1280x736 to 864x480. Adjusting the resolution by the same interval with YOLOv3-Tiny and YOLOv3-Tiny-PRN produced vRAM reductions of 18.2% and 21.3%, respectively. Understanding this relationship will be useful for selecting CNNs and GPUs for future research. The Nvidia GeForce RTX 3090, which contains 24 GB of vRAM (Nvidia Corporation, 2020c), is the only consumer-grade Nvidia GPU in production which contains enough vRAM to process

8 video streams with YOLOv3 at 1280x736 resolution. For a single camera at 1280x736 resolution, YOLOv3 required 2.3 GB of vRAM. YOLOv3-Tiny and YOLOv3-Tiny-PRN both required 0.8 GB of vRAM to process a single camera stream at 1280x736. Using YOLOv3-Tiny or YOLOv3-Tiny-PRN for processing video streams would be preferable because of their smaller memory requirement.

As video streams were added, the average framerate decayed non-linearly (**Figure 18**). The fastest CNN for processing 8 cameras at 1280x736 resolution was YOLOv3-Tiny-PRN, which achieved 9.6 FPS. YOLOv3-Tiny was slightly slower at 8.8 FPS. YOLOv3 processed images at 2.7 FPS at 1024x576 when 8 video streams were used simultaneously. YOLOv3-Tiny was the fastest CNN for processing a single video stream at 1280x736, achieving a framerate of 83.6 FPS. YOLOv3-Tiny-PRN was slightly slower at 82.2 FPS, and YOLOv3 was substantially slower at 16.2 FPS. Lowering the network resolution from 1280x736 to 864x480 improved processing speed by an average of 113.3% for YOLOv3, 80.3% for YOLOv3-Tiny, and 81.5% for YOLOv3-Tiny-PRN. In chapter 3, the minimum vertical FOV was found to be 0.43 m (**Table 10**). If the maximum ground speed used by Esau et al. (2018) is maintained (1.77 m s^{-1}), this results in a minimum framerate requirement of 4.12 FPS to ensure every area of the sprayer path is processed. YOLOv3-Tiny and YOLOv3-Tiny-PRN both meet this requirement. Considering the similar vRAM usage and processing speed for YOLOv3-Tiny compared to YOLOv3-Tiny-PRN, and its greater ability to detect sheep sorrel, YOLOv3-Tiny is the best option for use on a smart sprayer.

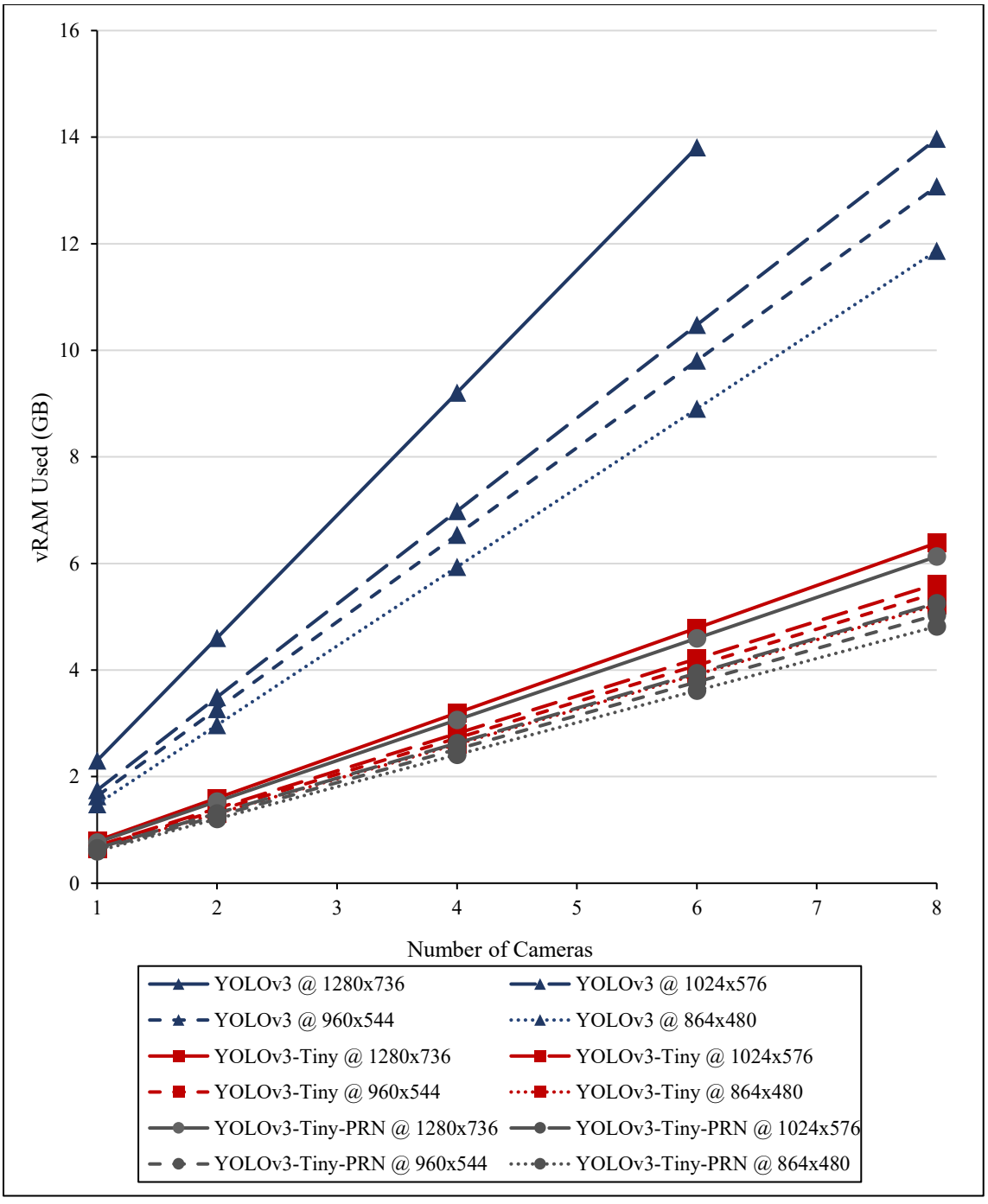

Figure 17: Graph of total vRAM required to process multiple cameras for object-detection networks. YOLOv3 at 1280x736 resolution was not tested with 8 cameras because it exceeded the 16 GB of vRAM available in the GPU.

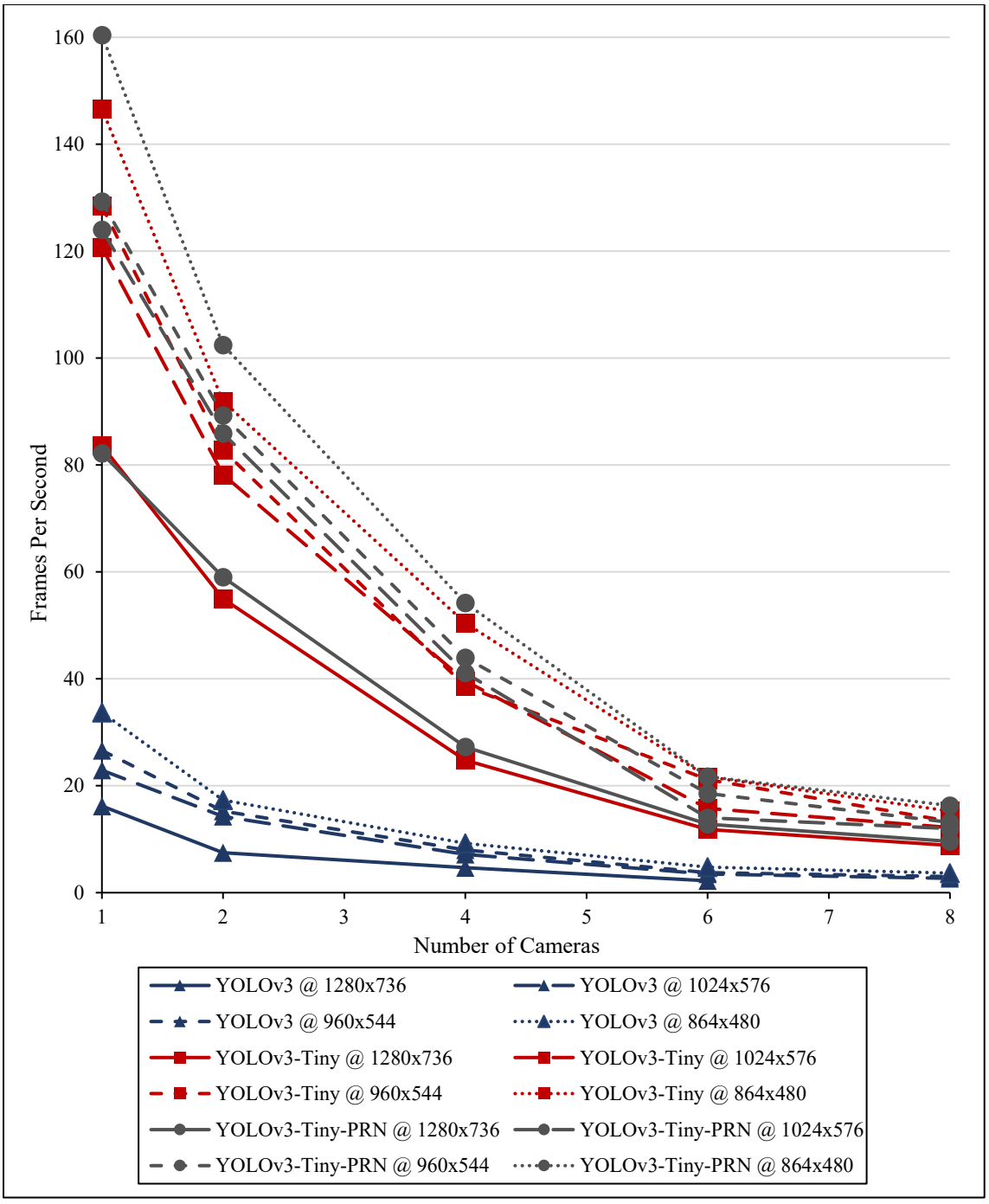

Figure 18: Graph of average processed frames per second from each camera against the number of cameras being processed for object-detection networks. YOLOv3 at 1280x736 resolution was not tested with 8 cameras because it exceeded the 16 GB of vRAM available in the GPU.

4.3.2 Memory Use and Processing Speed for Image-Classification Networks

Similar to object-detection CNNs, the amount of vRAM needed to process multiple image-classification CNNs was found to scale linearly with the number of video streams being processed (**Figure 19**). Processing video streams with EfficientNet-B0 presented the same problem as with YOLOv3, where the GPU did not have enough vRAM to process 8 CNNs at 1280x736 resolution. Extrapolating the memory measurements for EfficientNetB0 at 1280x736 linearly gives an estimated vRAM requirement of 17.6 GB for 8 cameras. MobileNetV2 required 10.3 GB of vRAM to process 8 video streams simultaneously at 1280x736, while Darknet Reference required 6.0 GB. As with object-detection CNNs, lowering the processing resolution resulted in a reduction of vRAM use for all three image-classification CNNs. EfficientNet-B0 saw the largest benefit from lowering the processing resolution, a reduction of vRAM use by 44.0% when the resolution was changed from 1280x736 to 864x480. Darknet Reference and MobileNetV2 saw reductions of 16.2% and 36.1%, respectively over the same interval.

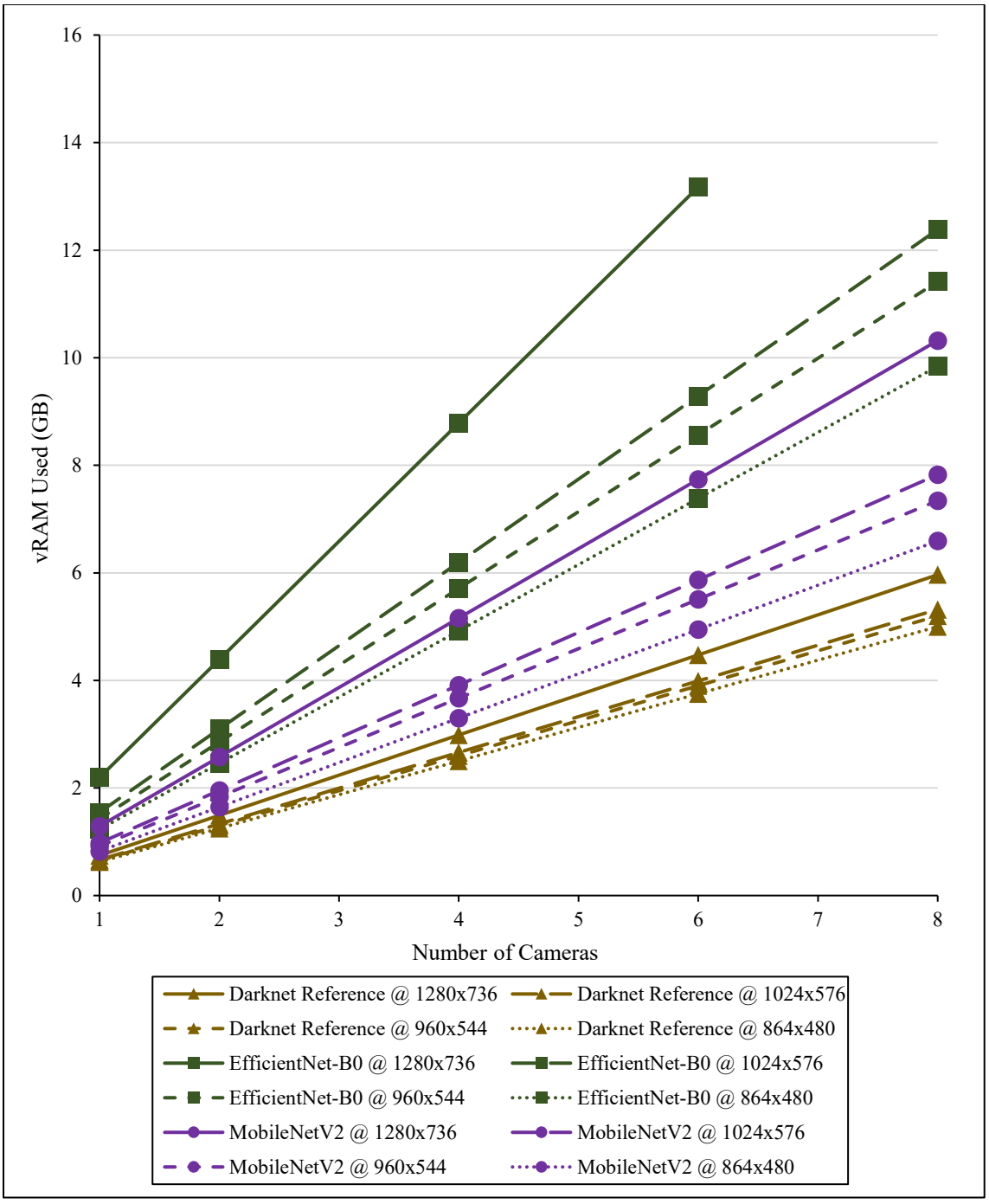

Figure 19: Graph of total vRAM required to process multiple cameras for image-classification networks. EfficientNet-B0 at 1280x736 resolution was not tested with 8 cameras because it exceeded the 16 GB of vRAM available in the GPU.

As with object-detection networks, the processing speed of image-classification networks decayed non-linearly as more cameras were added (**Figure 20**). The fastest CNN for processing 8 cameras at 1280x736 resolution was Darknet Reference, which achieved 19.2 FPS. MobileNetV2 was significantly slower than Darknet Reference, achieving 5.3 FPS under the same parameters. EfficientNet-B0 was slowest network for processing 8 cameras, with a framerate of 2.1 FPS at 1024x576. Lowering the network resolution from 1280x736 to 864x480 improved processing speed by an average of 135.8% for EfficientNet-B0, 108.6% for MobileNetV2, and 81.5% for Darknet Reference. Darknet Reference was the fastest CNN for processing a single video stream at 1280x736, achieving a framerate of 133.8 FPS. EfficientNet-B0 and MobileNetV2 were significantly slower at 12.0 FPS and 52.3 FPS, respectively. Darknet Reference and MobileNetV2 both exceed the minimum framerate requirement of 4.12 FPS. The faster processing speed of Darknet would allow a less powerful GPU to be used for processing, creating cost-savings. Considering the smaller vRAM usage and faster inference speed compared to EfficientNet-B0 and MobileNetV2, Darknet Reference is the best option for use on a smart sprayer.

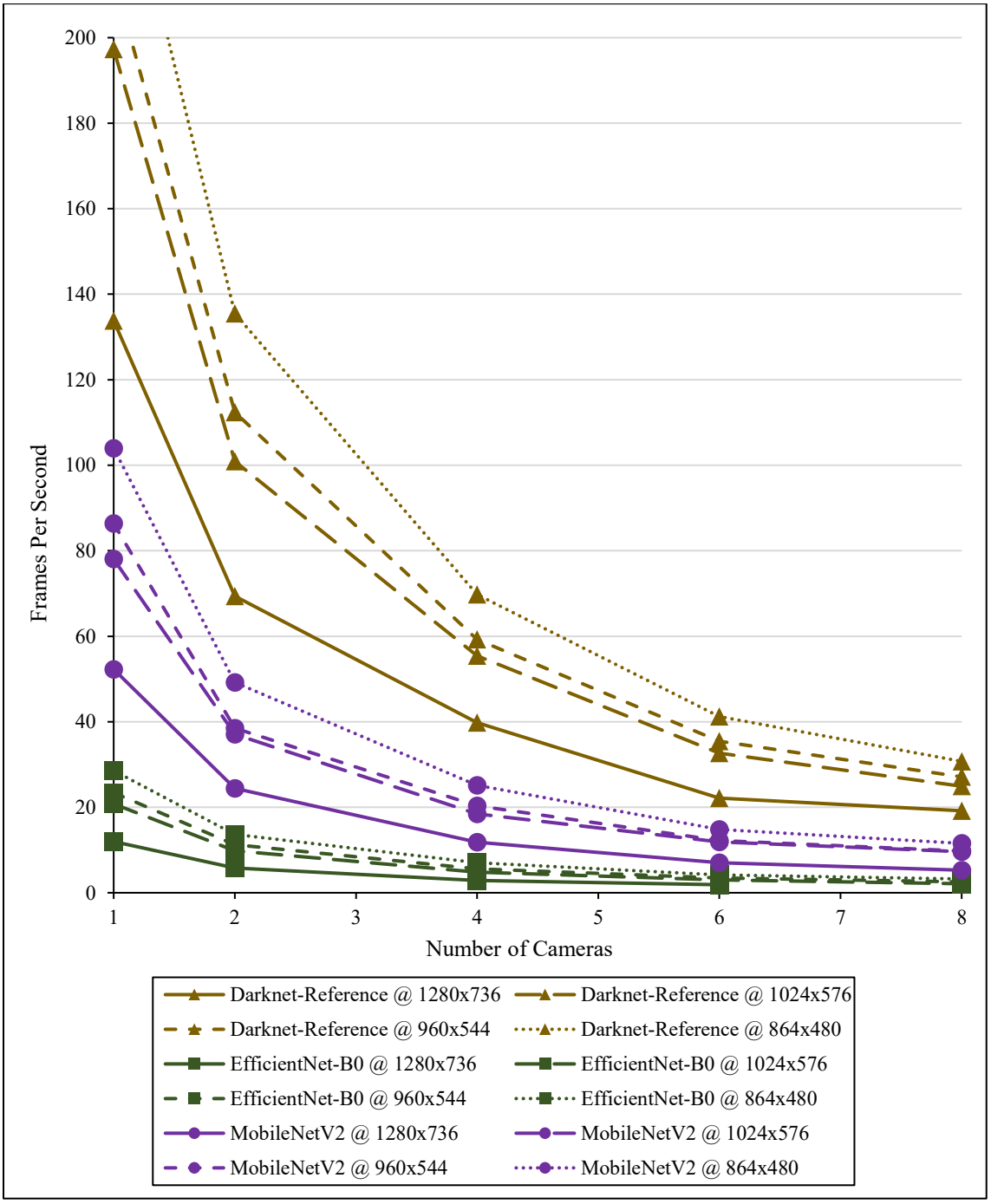

Figure 20: Graph of average processed frames per second from each camera against the number of cameras being processed for image-classification networks. EfficientNet-B0 at 1280x736 resolution was not tested with 8 cameras because it exceeded the 16 GB of vRAM available in the GPU.

YOLOv3-Tiny and Darknet Reference could both process 8 cameras at 1280x736 resolution in the GPU's 16 GB of vRAM, as they required 6.4 GB and 6.0 GB of vRAM, respectively. The vRAM required by these network limits the number of cameras which can be used from 7 (in a 6 GB GPU) to 13 (in an 11 GB GPU). Using a low-cost embedded GPU system such as an Nvidia Jetson (Nvidia Corporation, 2020a) to process images from each camera is an alternative to using a single central GPU to process all cameras. An array of embedded GPUs would allow for easier scaling of the entire system, as the GPUs could simply be added or removed with the cameras. Nvidia Jetson GPUs range in price from $59 USD to $899 USD and vRAM capacities from 2 GB to 32 GB (Nvidia Corporation, 2020a). Future work should involve testing embedded GPU systems to determine their viability for use in a real-time smart sprayer.

4.3.3 Graphical User Interface for Controlling Convolutional Neural Networks

A cross-platform GUI for controlling spray applications using CNNs was developed in Python and verified to run on Windows 10 and Ubuntu 16.04 (**Figure 21**). The GUI includes a frame of input settings in the upper-left corner which control the CNN parameters. Each camera includes a frame with options for accessing the camera showing the output frame produced by the CNN. The upper-right corner includes a "Start" button to initialize CNN processing for all selected cameras using the parameters defined by the user, and a "Stop" button to end CNN processing.

Figure 21: GUI written in Python 3.8.2 using the Tkinter library for controlling herbicide application on an 8-camera, 8-nozzle sprayer using CNNs. The top window shows the appearance on Windows 10 and the bottom window shows the appearance on Ubuntu 16.04.

The input options frame includes settings for detection mode, resolution, threshold, and target weed (**Figure 22**). YOLOv3-Tiny is used for target identification in "Detector" mode, while Darknet Reference is used for target identification in "Classifier" mode. The Darknet framework requires network resolutions to be a multiple of 32, so the user input options are automatically rounded up to the next multiple of 32 if the user inputs an incompatible value. The detection threshold can be modified when the GUI is used in "Detector" mode. Lower thresholds will yield more true and false positive detections, while

higher thresholds will yield less. The “Target” option selects whether to load the trained weights for hair fescue or sheep sorrel.

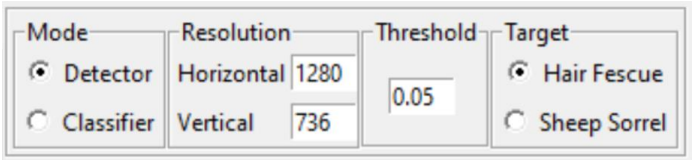

Figure 22: Options frame of the GUI. The user selects whether to use an object-detection or image-classification CNN, the internal network resolution, detection threshold (for object-detection only), and target weed.

The GUI includes a frame for each camera to be processed using the CNN (**Figure 23**). The frame is initialized with an image showing the camera number which is replaced with the output image produced by the CNN during processing. The options below the image display require the user to select the connection type and address for the CNN to connect to the camera.

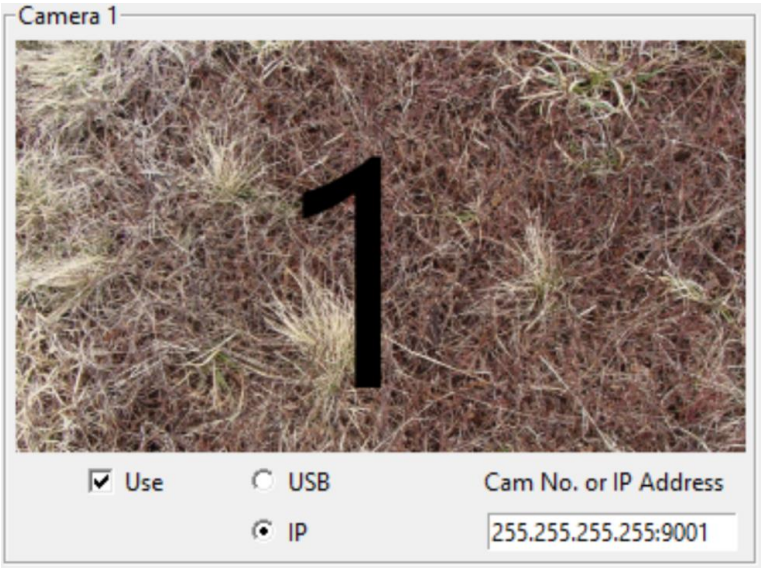

Figure 23: A camera frame from the GUI. A base image with a camera reference number is shown when the GUI is initialized. The user selects the camera connection method (USB or IP), and the address of the camera.

4.4 Conclusions

An important consideration for using CNNs to control herbicide application on a smart sprayer is the amount of processing power required. The amount of vRAM needed to process the CNNs scaled linearly with the number of cameras being processed simultaneously. The average framerate decayed non-linearly as the number of cameras was increased.

YOLOv3-Tiny and YOLOv3-Tiny-PRN required 6.4 GB and 6.1 GB of vRAM, respectively, to process images from 8 video streams at 1280x736 resolution. With 8 cameras, YOLOv3-Tiny and YOLOv3-Tiny-PRN achieved average processing speeds of 8.8 FPS and 9.6 FPS per camera, respectively. YOLOv3 would require approximately 18.4 GB of vRAM to process 8 video streams at 1280x736 pixels simultaneously, which did not fit in the 16 GB available in the Nvidia RTX 5000 GPU. When the resolution was decreased to 1024x576, YOLOv3 achieved an average processing speed of 2.7 FPS using 14.0 GB of vRAM to process 8 cameras simultaneously. Given the large amount of vRAM needed and slow processing speed, YOLOv3 is not viable for use on a real-time smart sprayer. Considering the similar vRAM usage and processing speed for YOLOv3-Tiny compared to YOLOv3-Tiny-PRN, and its greater ability to detect sheep sorrel, YOLOv3-Tiny is the best object-detection CNN for use on a smart sprayer.

Among image-classification CNNs, Darknet Reference achieved the fastest processing speed (19.2 FPS) using the smallest amount of vRAM (6.0 GB) for processing images from 8 video streams at 1280x736 pixels. MobileNetV2 achieved an average processing speed of 5.3 FPS using 10.3 GB of vRAM under the same parameters. EfficientNet-B0 required more vRAM than was available in the GPU to process 8 video

streams at 1280x736. Reducing the resolution to 1024x576 allowed EfficientNet-B0 to achieve an average framerate of 2.1 FPS using 12.4 GB of vRAM.

The results of this study identify YOLOv3-Tiny and Darknet Reference as the optimal object-detection and image-classification CNNs for use on a variable-rate smart sprayer. The Python-based GUI allows for control of the YOLOv3-Tiny and Darknet Reference CNNs for identification of hair fescue and sheep sorrel. The cross-platform nature of the GUI was verified to work on Windows 10 and Ubuntu 16.04, allowing flexibility during deployment. Using a CNN to control herbicide applications from a variable-rate smart sprayer will reduce the volume of herbicide needed for field management, resulting in cost-savings for wild blueberry producers.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The commercial wild blueberry industry is continually evolving and innovating to stay economically viable. The reduced price of wild blueberries and increasing labour costs have put a strain on the industry which requires creative, novel solutions to alleviate. Reducing agrochemical input costs through selectively applying herbicide only on areas of fields with weed cover is one such solution. This study trained and evaluated six convolutional neural networks for identifying hair fescue and sheep sorrel in wild blueberry fields. The trained networks were evaluated using three different cameras at varying target distances. Processing speed and video memory requirements were tested for each network, and a cross-platform GUI was developed for controlling the networks on a variable rate smart sprayer.

Images were collected from 58 Nova Scotia wild blueberry fields to train and test the networks. During the initial development, YOLOv3-Tiny was identified as the best object-detection network due to its high level of accuracy in detection of both weeds and fast processing speed. For detection of at least one target weed per image, all three object-detection networks produced F_1-scores of at least 0.95 for hair fescue at 1280x736 resolution and a threshold of 0.15. The results of changing the threshold were variable for sheep sorrel detection, with only small changes in the F_1-score. Object-detection results for sheep sorrel may be improved by labelling each individual sheep sorrel leaf or by selection regions of weed cover using semantic segmentation methods. Among image-classification networks, Darknet Reference produced the best overall results with F_1-scores of 0.92 or greater at all resolutions for classification of both weeds when trained with 7,200 images.

Darknet Reference was less consistent than the other five CNNs with smaller dataset sizes, with the F_1-score dropping to 0.77 when it was trained with 1,890 images. All other CNNs consistently produced F_1-scores above 0.90 when trained with dataset sizes from 472 to 3,780 images.

During independent field testing, the higher resolution, 1280x736, with the lowest threshold, 0.05, produced the best results. Sheep sorrel was detected by YOLOv3-Tiny with an F_1-score of 0.67 across all images in the dataset. F_1-scores for hair fescue detection, were consistently 0.80 or 0.81 regardless of resolution. These results were lower than the validation scores produced when the networks were trained, which may be the result of bias in the original image dataset. Camera selection did not have a significant effect on hair fescue detection except at a height of 0.98 m in the Debert field. A lens height of 0.57 m produced the best results for sheep sorrel in two out of three fields. The Logitech c920 camera failed to detect any sheep sorrel in 19 of 27 parameter combinations, indicating that it is not viable for this purpose. This may have been due to either lower quality images compared to the Canon T6 and LG G6 or because the CNNs were not trained with images from the Logitech c920.

The amount of vRAM needed to process the CNNs increased linearly with the number of cameras being processed simultaneously, while the average framerate decayed non-linearly. YOLOv3 would require approximately 18.4 GB of vRAM to process 8 video streams simultaneously at 1280x736 pixels, which did not fit in the 16 GB available in the Nvidia RTX 5000 GPU. YOLOv3-Tiny and YOLOv3-Tiny-PRN required 6.4 GB and 6.1 GB of vRAM, respectively, to process images from 8 video streams at 1280x736 resolution. With 8 cameras, YOLOv3-Tiny and YOLOv3-Tiny-PRN achieved average

processing speeds of 8.8 FPS and 9.6 FPS per camera, respectively. YOLOv3 achieved an average processing speed of 2.7 FPS using 14.0 GB of vRAM to process 8 cameras simultaneously at 1280x736. Given the large amount of vRAM needed and slow processing speed, YOLOv3 is not viable for use on a real-time smart sprayer. YOLOv3-Tiny and YOLOv3-Tiny-PRN had comparable processing requirements; however, YOLOv3-Tiny produced much better results for sheep sorrel detection than YOLOv3-Tiny-PRN. EfficientNet-B0 required more vRAM than was available in the GPU to process 8 video streams at 1280x736. Reducing the resolution to 1024x576 allowed EfficientNet-B0 to achieve an average framerate of 2.1 FPS using 12.4 GB of vRAM. Darknet Reference achieved the fastest processing speed (19.2 FPS) using the smallest amount of vRAM (6.0 GB) for processing images from 8 video streams at 1280x736 pixels. MobileNetV2 achieved an average processing speed of 5.3 FPS using 10.3 GB of vRAM under the same parameters. The GUI allows for control of YOLOv3-Tiny and Darknet Reference for identification of target weeds on Windows 10 and Ubuntu 16.04. Using a CNN to target hair fescue, sheep sorrel, and other weeds in wild blueberry fields on a smart sprayer can reduce herbicide use and create cost-savings for growers.

5.2 Recommendations

The results of this study indicate that deep learning CNNs can be trained to identify hair fescue and sheep sorrel in images of wild blueberry fields. YOLOv3-Tiny and Darknet Reference are both effective networks which can quickly perform deep learning inferences on images. Future work should involve testing of multiple high-quality cameras to determine the best option for use on a variable rate smart sprayer. Images should be captured with the selected camera and the CNNs should be retrained to potentially increase

identification accuracy. Additional CNNs can be tested to determine if there is a better option for weed identification. YOLOv4, which was published after most of the data analysis for this thesis was complete, is approximately 30% more accurate than YOLOv3 on the COCO dataset with similar processing speed (Bochkovskiy et al., 2020). Labelling images through semantic segmentation of regions of target weeds may improve detection accuracy over the bounding box method used by traditional object-detectors. Additionally, the CNNs should be trained to detect other common weeds in wild blueberry fields to maximize the potential use on a smart sprayer. Potential future targets include bunchberry (*Cornus canadensis* L.), goldenrod (*Solidago* spp.), haircap moss (*Polytrichum commune* Hedw.), and hawkweed (*Hieracium* spp.).

Other field management tasks in wild blueberry such as fungicide timing, pest scouting, and yield estimation can benefit from CNN technology. Currently, wild blueberry growers in Nova Scotia call a hotline to obtain information to help choose the appropriate timing for disease prevention. Diseases such as monilinia blight (*Monilinia vaccinii-corymbosi*) and botrytis blight (*Botrytis cinerea*) can begin growing at the F2 stage of bud development (Agriculture and Agri-Food Canada, 2016). A smartphone application with a CNN which identifies this bud stage could provide site-specific information to aid growers with timing fungicide application. A CNN which could recognize the diseases after their development would aid growers with future field management decisions.

A smartphone application for field scouting in wild blueberry would not be limited to disease management. Weed identification CNNs such as the ones developed in this thesis could be integrated into a smartphone application to provide growers with instantaneous information regarding the species of weeds in their fields and the current best practices for

management. Additionally, the ripeness detection CNN developed by Schumann et al. (2019) could be improved to provide a yield estimation prior to harvest, indicating whether a section of the field was worth the input cost to manage.

Finally, the CNNs should be deployed on a smart applicator to control herbicide use in wild blueberry fields. Although liquid herbicides are traditionally used for management, a machine vision system based on CNNs would also work to control application of granular herbicides, such as Dichlobenil, if the appropriate hardware for variable-rate application was developed. Overall, CNNs provide an opportunity to create cost-savings for the wild blueberry industry through limiting excess herbicide use and providing field-specific information for better management decisions. Widespread adoption of this novel technology in the wild blueberry industry has the potential to lower input costs and increase yield, making the industry more financially sustainable.

REFERENCES

Agriculture and Agri-Food Canada. (2016). *Blueberry diseases guide*. Agriculture and Agri-Food Canada. https://www.perennia.ca/wp-content/uploads/2018/04/blueberry-disease-guide.pdf

Agrifac Machinery B.V. (2016). *Agrifac - AiCPlus*. https://www.agrifac.com/condor/new-innovations/aicplus

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. *17th Conference on Database Systems for Business, Technology, and Web (BTW 2017)*, 79–88.

Beattie, J., Crozier, A., & Duthie, G. (2005). Potential health benefits of berries. *Current Nutrition & Food Science*, *1*(1), 71–86. https://doi.org/10.2174/1573401052953294

Bishop, C. M. (1995). Regularization and complexity control in feed-forward networks. *Proceedings International Conference on Artificial Neural Networks (ICANN'95)*, *1*, 141–148.

Blue River Technologies. (2018). *See & spray agricultural machines*. http://www.bluerivertechnology.com/

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv*. https://arxiv.org/pdf/2004.10934.pdf

Bottou, L. (1998). Online learning and stochastic approximations. *On-Line Learning in Neural Networks*, *17*(9).

Boureau, Y., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 111–118.

Cauchy, A.-L. (1847). Methode generale pour la resolution des systemes d'equations simultanees. *Compte Rendu Des Seances de L'Acad'emie Des Sciences*, *25*(2), 536–538.

Chellapilla, K., Puri, S., & Simard, P. Y. (2006). High performance convolutional neural networks for document processing. *International Conference on Frontiers in Handwriting Recognition (ICFHR)*.

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, *22*(12), 3207–3220. https://doi.org/10.1162/NECO_a_00052

Crisan, C. (2019). *motionEyeOS: A video surveillance OS for single-board computers*. GitHub Repository. https://github.com/ccrisan/motioneyeos/wiki

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN : Object detection via region-based fully convolutional networks. *30th Conference on Neural Information Processing Systems (NIPS 2016)*.

Dale, A., Hanson, E. J., Yarborough, D. E., McNicol, R. J., Stang, E. J., Brennan, R., Morris, J. R., & Hergert, G. B. (1994). Mechanical harvesting of berry crops. *Horticultural Reviews*, *16*, 255–382. https://doi.org/10.1002/9780470650561.ch8

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Devries, T., & Taylor, G. W. (2017). Dataset augmentation in feature space. *ArXiv*, 1–12. https://arxiv.org/pdf/1702.05538.pdf

Duchi, J. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Elfadel, I. M., & Wyatt Jr., J. L. (1994). The "softmax" nonlinearity: Derivation using statistical mechanics and useful properties as a multiterminal analog circuit element. *Advances in Neural Information Processing Systems 6*, 882–887.

Employment and Social Development Canada. (2018). *Historical minimum wage rates in Canada*. https://open.canada.ca/data/en/dataset/390ee890-59bb-4f34-a37c-9732781ef8a0

Esau, T. (2019). Fruit quality effects - A researcher's perspective. *WBPANS Annual General Meeting, March 2019*.

Esau, T., Zaman, Q., Groulx, D., Corscadden, K., Chang, Y. K., Schumann, A., & Havard, P. (2016). Economic analysis for smart sprayer application in wild blueberry fields. *Precision Agriculture*, *17*(6), 753–765. https://doi.org/10.1007/s11119-016-9447-8

Esau, T., Zaman, Q., Groulx, D., Farooque, A., Schumann, A., & Chang, Y. (2018). Machine vision smart sprayer for spot-application of agrochemical in wild blueberry fields. *Precision Agriculture*, *19*(4), 770–788. https://doi.org/10.1007/s11119-017-9557-y

Esau, T., Zaman, Q. U., Chang, Y. K., Groulx, D., Schumann, A. W., & Farooque, A. A. (2014). Prototype variable rate sprayer for spot-application of agrochemicals in wild blueberry. *Applied Engineering in Agriculture*, *30*(5), 717–725. https://doi.org/10.13031/aea.30.10613

Esau, T., Zaman, Q. U., MacEachern, C., Yiridoe, E. K., & Farooque, A. A. (2019). Economic and management tool for assessing wild blueberry production costs and financial feasibility. *Applied Engineering in Agriculture*, *35*(5), 687–696. https://doi.org/10.13031/aea.13374

Farooque, A. A. (2015). *Performance evaluation of a commercial wild blueberry harvester using precision agriculture technologies and mathematical modeling*. Doctoral Thesis, Dalhousie University.

Farooque, A. A., Zaman, Q. U., Groulx, D., Schumann, A. W., Yarborough, D. E., & Nguyen-Quang, T. (2014). Effect of ground speed and header revolutions on the picking efficiency of a commercial wild blueberry harvester. *Applied Engineering in Agriculture*, *30*(4), 535–546. https://doi.org/10.13031/aea.30.10415

Fennimore, S. A., Smith, R. F., Tourte, L., LeStrange, M., & Rachuy, J. S. (2014). Evaluation and economics of a rotating cultivator in bok choy, celery, lettuce, and radicchio. *Weed Technology*, *28*(1), 176–188. https://doi.org/10.1614/wt-d-13-00051.1

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, *17*(9). https://doi.org/10.3390/s17092022

Girshick, R. (2015). Fast R-CNN. *ArXiv*. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. https://doi.org/10.1109/CVPR.2014.81

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (1st ed.). The MIT Press. http://www.deeplearningbook.org

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., & Warde-Farley, D. (2014). Generative adversarial nets. *27th Conference on Neural Information Processing Systems (NIPS 2014)*.

Google. (2020). *[Map of wild blueberry sites used for image collection in spring 2019]*. https://www.google.com/maps/d/viewer?mid=1lU1AOkV7-V_TIF9_Vr-WtjKpOhq_XKxe&ll=45.356685825285595%2C-63.663151666666664&z=10

Government of New Brunswick. (2020). *2020 Wild blueberry weed control selection guide - Fact sheet C1.6.2*. https://www2.gnb.ca/content/dam/gnb/Departments/10/pdf/Agriculture/WildBlueberries-BleuetsSauvages/C162-E.pdf

Hall, I. V., Aalders, L. E., Nickerson, N. L., & Vander Kloet, S. P. (1979). The biological flora of Canada 1. *Vaccinium angustifolium* Ait., sweet lowbush blueberry. *The Canadian Field-Naturalist*, *93*, 415–430.

Hall, I. V., Craig, D. L., & Lawrence, R. A. (1983). A comparison of hand raking and mechanical harvesting of lowbush blueberries. *Canadian Journal of Plant Science*, *63*(4), 951–954. https://doi.org/10.4141/cjps83-119

Harris, M. (2008). Many-core GPU computing with NVIDIA CUDA. *Proceedings of the 22nd Annual International Conference on Supercomputing*, 1. https://doi.org/10.1145/1375527.1375528

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CPVR)*. https://doi.org/10.1109/CVPR.2016.90

Hinton, G. (2012). *Neural networks for machine learning - Lecture 6*. Coursera. https://www.cs.toronto.edu/~hinton/coursera_lectures.html

Hong, S., Minzan, L., & Zhang, Q. (2012). Detection system of smart sprayers: Status, challenges, and perspectives. *International Journal of Agricultural and Biological Engineering*, *5*(3), 10–23. https://doi.org/10.3965/j.ijabe.20120503.002

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Howard, J. (2018). *Practical deep learning for coders*. University of San Francisco. http://course18.fast.ai

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology*, *195*, 215–243. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=23914&isnumber=907

Hughes, A., White, S. N., Boyd, N. S., Hildebrand, P., & Cutler, C. G. (2016). Red sorrel management and potential effect of red sorrel pollen on *Botrytis cinerea* spore germination and infection of lowbush blueberry (*Vaccinium angustifolium* Ait.) flowers. *Canadian Journal of Plant Science*, *96*(4), 590–596. https://doi.org/10.1139/cjps-2015-0285

Ioffe, S., & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *ArXiv*.

Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, *1*, 295–307.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? *2009 IEEE 12th International Conference on Computer Vision*, 2146–2153. https://doi.org/10.1109/ICCV.2009.5459469

Jensen, J. R. (2005). *Introductory digital image processing: A remote sensing perspective* (K. C. Clarke (ed.); 3rd ed.). Prentice Hall Series in Geogrphic Information Science.

Jensen, K. I. N., & Yarborough, D. E. (2004). An overview of weed management in the wild lowbush blueberry - Past and present. *Small Fruits Review*, *3*(3–4), 229–255. https://doi.org/10.1300/J301v03n03_02

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

Kay, C. D., & Holub, B. J. (2002). The effect of wild blueberry (*Vaccinium angustifolium*) consumption on postprandial serum antioxidant status in human subjects. *British Journal of Nutrition*, *88*(4), 389–397. https://doi.org/10.1079/bjn2002665

Kennedy, K. J., Boyd, N. S., & Nams, V. O. (2010). Hexazinone and fertilizer impacts on sheep sorrel (*Rumex acetosella*) in wild blueberry. *Weed Science*, *58*(3), 317–322. https://doi.org/10.1614/WS-D-09-00081.1

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *ArXiv*, 1–15. https://arxiv.org/pdf/1412.6980.pdf

Kinsman, B. (1993). *The history of the lowbush blueberry industry in Nova Scotia 1950-1990*. The Blueberry Producers' Association of Nova Scotia.

Klonsky, K. (2012). Comparison of production costs and resource use for organic and conventional production systems. *American Journal of Agricultural Economics*, *94*(2), 314–321. https://doi.org/10.1093/ajae/aar102

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). *ImageNet classification with deep convolutional neural networks*.

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

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Biochemical and Biophysical Research Communications*, *86*(11), 2278–2324.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *European Conference on Computer Vision (ECCV 2014)*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Lobo, V., Patil, A., Phatak, A., & Chandra, N. (2010). Free radicals, antioxidants and functional foods: Impact on human health. *Pharmacognosy Reviews*, *4*(8), 118–126. https://doi.org/10.4103/0973-7847.70902

McCully, K. V., Sampson, M. G., & Watson, A. K. (1991). Weed survey of Nova Scotia lowbush blueberry (*Vaccinium angustifolium*) fields. *Weed Science*, *39*(2), 180–185. https://doi.org/10.1017/s0043174500071447

Nair, V., & Hinton, G. (2010). Rectificed linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf

Neal, R. M. (1996). *Lecture notes in statistics no. 188 - Bayesian learning for neural networks*. Springer. https://doi.org/https://doi.org/10.1007/978-1-4612-0745-0

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate O($1/k^2$). *Soviet Mathematics Doklady*, *27*(2), 372–376.

Nvidia Corporation. (2019). *GeForce RTX 2080 Ti graphics card*. https://www.nvidia.com/en-us/geforce/graphics-cards/rtx-2080-ti/

Nvidia Corporation. (2020a). *Embedded systems developer kits & modules from NVIDIA Jetson*. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

Nvidia Corporation. (2020b). *GeForce GTX 16 series graphics cards*. https://www.nvidia.com/en-us/geforce/graphics-cards/gtx-1660-ti/

Nvidia Corporation. (2020c). *GeForce RTX 3090 graphics card*. https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3090/

Partel, V., Kakarla, S. C., & 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*(December 2018), 339–350. https://doi.org/10.1016/j.compag.2018.12.048

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, *4*(5), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, *435*(7045), 1102–1107. https://doi.org/10.1038/nature03687

Rahman, M. A., & Wang, Y. (2016). Optimizing intersection-over-union in deep neural networks for image segmentation. *International Symposium on Visual Computing*, 234–244. https://doi.org/10.1007/978-3-319-50835-1_22

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. *Proceedings of the 26th International Conference on Machine Learning*.

Redmon, J. (2016). *Darknet: Open source neural networks in C*. http://pjreddie.com/darknet/

Redmon, J. (2018). *YOLO: Real-time object detection*. https://pjreddie.com/darknet/yolo/

Redmon, J., Bochkovskiy, A., & Sinigardi, S. (2020). *Darknet: YOLOv3 - Neural network for object detection*. GitHub Repository. https://github.com/AlexeyAB/darknet

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *ArXiv*. https://arxiv.org/pdf/1506.02640.pdf

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, faster, stronger. *ArXiv*. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv*. https://arxiv.org/pdf/1804.02767.pdf

Rehman, T. U., Zaman, Q. U., Chang, Y. K., Schumann, A. W., & Corscadden, K. W. (2019). Development and field evaluation of a machine vision based in-season weed detection system for wild blueberry. *Computers and Electronics in Agriculture*, *162*(July 2018), 1–13. https://doi.org/10.1016/j.compag.2019.03.023

Rehman, T. U., Zaman, Q. U., Chang, Y. K., Schumann, A. W., Corscadden, K. W., & Esau, T. (2018). Optimising the parameters influencing performance and weed (goldenrod) identification accuracy of colour co-occurrence matrices. *Biosystems Engineering*, *170*, 85–95. https://doi.org/10.1016/j.biosystemseng.2018.04.002

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(7), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Robinson, D. (2020). Acadia blueberry price bulletin. *Acadia Blue Publishing*.

Rumelhart, D., Hinton, G., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Santoni, M. M., Sensuse, D. I., Arymurthy, A. M., & Fanany, M. I. (2015). Cattle race classification using gray level co-occurrence matrix convolutional neural networks. *Procedia Computer Science*, *59*, 493–502. https://doi.org/10.1016/j.procs.2015.07.525

Schieffer, J., & Dillon, C. (2014). The economic and environmental impacts of precision agriculture and interactions with agro-environmental policy. *Precision Agriculture*, *16*(1), 46–61. https://doi.org/10.1007/s11119-014-9382-5

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schumann, A. W., Mood, N. S., Mungofa, P. D. K., MacEachern, C., Zaman, Q. U., & Esau, T. (2019). Detection of three fruit maturity stages in wild blueberry fields using deep learning artificial neural networks. *2019 ASABE Annual International Meeting*. https://doi.org/10.13031/aim.201900533

Sharpe, S. M., Schumann, A. W., & Boyd, N. S. (2019). Detection of Carolina geranium (*Geranium carolinianum*) growing in competition with strawberry using convolutional neural networks. *Weed Science*, *67*(2), 239–245. https://doi.org/10.1017/wsc.2018.66

Sharpe, S. M., Schumann, A. W., Yu, J., & Boyd, N. S. (2020). Vegetation detection and discrimination within vegetable plasticulture row-middles using a convolutional neural network. *Precision Agriculture*, *21*, 264–277. https://doi.org/10.1007/s11119-019-09666-6

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ArXiv*. https://arxiv.org/pdf/1409.1556.pdf

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Steinkraus, D., Buck, I., & Simard, P. Y. (2005). Using GPUs for machine learning algorithms. *Eight International Conference on Document Analysis and Recognition (ICDAR'05)*, 1115–1120. https://doi.org/10.1109/ICDAR.2005.251

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on Machine Learning*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CPVR)*. https://doi.org/10.1109/CVPR.2015.7298594

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the International Conference on Machine Learning (ICML)*.

Tetko, I. V, Livingstone, D. J., & Luik, A. I. (1995). Neural network studies. 1. *Joural of Chemistry Information in Computer Science*, *35*, 826–833.

Thomas, A. G. (1985). Weed survey system used in saskatchewan for cereal and oilseed crops. *Weed Science*, *33*, 34–43. https://doi.org/10.1017/S0043174500083892

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, *157*(January), 417–426. https://doi.org/10.1016/j.compag.2019.01.012

Trimble Inc. (2020a). *GreenSeeker crop sensing system*. https://agriculture.trimble.com/product/greenseeker-system/

Trimble Inc. (2020b). *WeedSeeker 2 spot spray system*. https://agriculture.trimble.com/product/weedseeker-2-spot-spray-system/

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., & Smeulders, A. W. M. (2012). Selective search for object recognition. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-013-0620-5

Venkataramanan, A., Honakeri, D. K. P., & Agarwal, P. (2019). Plant disease detection and classification using deep neural networks. *International Journal on Computer Science and Engineering*, *11*(9), 40–46.

Wang, C.-Y., Liao, H.-Y. M., Chen, P.-Y., & Hsieh, J.-W. (2019). Enriching variety of layer-wise learning information by gradient combination. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. http://arxiv.org/abs/1911.11929

Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. *ArXiv*. https://arxiv.org/pdf/1712.04621.pdf

White, S. N. (2018). Determination of Festuca filiformis seedbank characteristics, seedling emergence and herbicide susceptibility to aid management in lowbush blueberry (*Vaccinium angustifolium*). *Weed Research*, *58*(2), 112–120. https://doi.org/10.1111/wre.12286

White, S. N. (2019). Final weed survey update and research progress on priority weed species in wild blueberry. *Wild Blueberry Producers Association of Nova Scotia Annual General Meeting*.

White, S. N., & Kumar, S. K. (2017). Potential role of sequential glufosinate and foramsulfuron applications for management of fescues (*Festuca* spp.) in wild blueberry. *Weed Technology*, *31*(1), 100–110. https://doi.org/10.1614/wt-d-16-00086.1

Wild Blueberry Producers Association of Nova Scotia. (2018). *June 2018 dispatch*. http://www.nswildblueberries.com/members/newsletters/download?path=June2018.pdf

Wu, D., Wu, Q., Yin, X., Jiang, B., Wang, H., He, D., & Song, H. (2020). Lameness detection of dairy cows based on the YOLOv3 deep learning algorithm and a relative step size characteristic vector. *Biosystems Engineering*, *189*, 150–163. https://doi.org/10.1016/j.biosystemseng.2019.11.017

Xiong, H. Y., Barash, Y., & Frey, B. J. (2011). Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context. *Bioinformatics*, *27*(18), 2554–2562. https://doi.org/10.1093/bioinformatics/btr444

Yang, Q., Xiao, D., & Lin, S. (2018). Feeding behavior recognition for group-housed pigs with the Faster R-CNN. *Computers and Electronics in Agriculture*, *155*(October), 453–460. https://doi.org/10.1016/j.compag.2018.11.002

Yarborough, D. E. (2004). Factors contributing to the increase in productivity in the wild blueberry industry. *Small Fruits Review*, *3*(1–2), 33–43. https://doi.org/10.1300/J301v03n01_05

Yarborough, D. E. (2006). Innovations in weed management in wild blueberry fields in Maine. *Acta Horticulturae*, *715*, 197–202. https://doi.org/10.17660/ActaHortic.2006.715.28

Yarborough, D. E. (2012). Establishment and management of the cultivated lowbush blueberry (*Vaccinium angustifolium*). *International Journal of Fruit Science*, *12*(1–3), 14–22. https://doi.org/10.1080/15538362.2011.619130

Yarborough, D. E. (2017). Blueberry crop trends 1996 - 2017. *WBPANS Annual General Meeting, November 2017*.

Yarborough, D. E., & Bhowmik, P. C. (1993). Lowbush blueberry-bunchberry competition. *Journal of the American Society for Horticultural Science*, *118*(1), 54–62. https://doi.org/10.21273/jashs.118.1.54

Yu, J., Schumann, A. W., Cao, Z., Sharpe, S. M., & Boyd, N. S. (2019). Weed detection in perennial ryegrass with deep learning convolutional neural network. *Frontiers in Plant Science*, *10*(October), 1–9. https://doi.org/10.3389/fpls.2019.01422

Yu, J., Sharpe, S. M., Schumann, A. W., & Boyd, N. S. (2019a). Deep learning for image-based weed detection in turfgrass. *European Journal of Agronomy*, *104*(November), 78–84. https://doi.org/10.1016/j.eja.2019.01.004

Yu, J., Sharpe, S. M., Schumann, A. W., & Boyd, N. S. (2019b). Detection of broadleaf weeds growing in turfgrass with convolutional neural networks. *Pest Management Science*, *75*(8), 2211–2218. https://doi.org/10.1002/ps.5349

Zaman, Q. U., Esau, T. J., Schumann, A. W., Percival, D. C., Chang, Y. K., Read, S. M., & Farooque, A. A. (2011). Development of prototype automated variable rate sprayer for real-time spot-application of agrochemicals in wild blueberry fields. *Computers and Electronics in Agriculture*, *76*(2), 175–182. https://doi.org/10.1016/j.compag.2011.01.014

Zeiler, M. D., & Fergus, R. (2013). Visualizing and understanding convolutional networks. *ArXiv*. https://doi.org/10.1111/j.1475-4932.1954.tb03086.x

Zhou, Y. T., & Chellappa, R. (1988). Computation of optical flow using a neural network. *IEEE International Conference on Neural Networks*, *2*, 71–78. https://doi.org/10.1109