Near infrared imaging and image pre-processing to improve the automatic detection of Canada geese

by

Jacqueline Szeto

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical Engineering

Department of Electronics Engineering

Carleton University

Ottawa, Ontario

Abstract

Migratory shorebirds populations are adversely affected by climate change and loss of habitat thus careful monitoring of their populations is important for early detection of population loss. Current counting methods generally rely on intrusive and time-consuming manual identification. This work is part of a larger project to develop automated classification and counting methods using a remotely piloted aircraft system (RPAS). In addition to the use of RPAS, this work will also investigate if near-infrared (NIR) imaging captured by the RPAS yields detection improvements. Healthy vegetation reflects NIR wavelengths of light which can potentially create a greater contrast between an object and the surrounding vegetation. Pre-processing NIR raw images to enhance the contrast between vegetation and Canada geese (*Branta canadensis*) to improve object detection using the convolutional neural network (CNN) YOLOv4-Tiny have been investigated in this study. Training was done on a small dataset (423 and 1,269 images), a large dataset (2,000 images), artificially generated dataset (2,000 images) and hybrid datasets consisting of real and artificially generated images of Canada geese (4,000 and 5,000 images). A RPAS was used to obtain greyscale and NIR test images of geese decoys using the RPAS onboard camera and a NIR specific camera at varying altitudes. The NIR pre-processed ground test images showed detection improvements in both number of detections and confidence score percentages when validated against the YOLOv4-Tiny detector that was trained on an augmented dataset and the hybrid datasets. However, the greyscale aerial test images generally outperformed the NIR pre-processed images.

Keywords: Image Pre-processing, Convolutional Neural Networks, Greyscale, Augmentation, Remotely Piloted Aircraft Systems, Migratory Shorebirds

Acknowledgements

I would like to thank my supervisors Dr. Alan Steele and Dr. Jeremy Laliberté for providing guidance and support throughout my research. I would also like to give a big shout out to Alexsander Costa, Brendan Ooi, Skylar Bruggink, Gerhard Bruins, and the rest of the Molson team for helping and supporting me along the way.

I want to thank Ciena Corporation and my managers for sponsoring and supporting my post-secondary education whilst employed full-time.

Also, a big thank you to my family for their continuous support. Last but not least, thank you Justin for always being there for me, keeping my sanity in check and making sure there was always a warm cooked meal for me on my many long evenings.

Table of Contents

List of Figures

List of Tables

List of Appendices

List of Abbreviations

AI	Artificial intelligence
ANN	Artificial neural network
AOV	Angle of view
API	Application programming interface
CIE	Commission internationale de l'éclairage
CNN	Convolutional neural network
CUDA	Compute unified device architecture
DGN	Digital negative
FOV	Field of view
GPU	Graphical processing unit
GSD	Ground sampling distance
HVS	Human visual system
IOU	Intersection over union
JPEG	Joint Photographic Experts Group
mAP	Mean average precision
NDVI	Normalized difference vegetation index
NIR	Near infrared
PAN	Path aggregation network
RGB	Red, green, blue
RPAS	Remote piloted aircraft system
SAT	Self adversarial training

SPP	Spatial pyramid pooling
YOLO	You Only Look Once

Chapter 1: Introduction

Neural networks are powerful systems utilized in artificial intelligence (AI) which can be employed in object detection, image classification and text prediction applications for example (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010; Karasawa, et al., 2017; Abujar, et al., 2019). The goal of this research is to assist ornithologists and biologists in the automatic detection and counting of migratory shorebirds. In this study, a convolutional neural network (CNN) is employed in an object detection application to detect Canada geese (*Branta canadensis*). The Canada goose was chosen as the initial trial species because it presents melanin colour similarities with the various shorebird species that seasonally reside in James Bay which is home to a significant shorebird migration site (Friis, 2020). This bird species is also easily found in various public parks in the City of Ottawa for easy accessibility to collect image samples which were gathered using a near infrared (NIR) and red, green, blue (RGB) cameras. In addition, the RGB and NIR cameras were used to capture aerial images of Canada geese decoys as aerial image samples as we were unable to obtain aerial images containing real Canada geese. The pre-processing of RGB and NIR raw test images were then compared to verify if pre-processing NIR images yields detection improvements when tested against the You Only Look Once Tiny Version 4 (YOLOv4-Tiny).

1.1 Motivation

Migratory shorebirds have been shown to be sentinels of global environmental changes by monitoring their populations (Piersma & Lindström, 2004). This is because globally shorebird populations have been on the decline which is attributed to loss of habitat and breeding ground changes, and human activities (Piersma & Lindström, 2004; Andres, et al., 2012; Murray, et al., 2018). To track migratory shorebird populations, biologists and ornithologists have been

counting species manually which is labor intensive, potentially disruptive to the birds due to the possible use of helicopters to aerially count birds, and at risk for human errors. To reduce the burden of manual counting and potential human errors, this study investigates the use of CNNs to aid in the automatic detection and identification of Canada geese from image samples collected using a remote pilot aircraft system (RPAS) at varying altitudes, in addition to image samples collected at ground level.

1.2 Problem Statement

The focus of this research was aimed at enhancing images with respect to image contrasts to improve detection rates to aid in the automatic counting of shorebirds in vegetation. To heighten contrasts in the test images captured by the NIR camera, the NIR raw images were pre-processed to increase object contrast with respect to the image background containing vegetation due to the NIR reflectivity on healthy vegetation which potentially creates a contrast between the object (shorebird) and vegetation background. Pre-processing methods were investigated on the NIR raw false colour images using an open-source image pre-processing software *Rawtherapee* and MATLAB. The YOLOv4-Tiny detector was trained on RGB converted greyscale image datasets containing images of Canada geese that were openly sourced, provided by other institutions and artificially generated. In addition to the greyscale dataset, the greyscale images were augmented using three augmentation methods, namely contrasts manipulation and Gaussian smoothing, which were used to train the YOLOv4-Tiny detector. The use of artificially generated (i.e. synthetic or computer generated) images to train the YOLOv4-Tiny detector was also investigated. Custom artificial images were generated in an open-source 3D graphics software tool *Blender* which simulated aerial images of Canada geese. These images were

artificially generated due to the lack of aerial perspective images in the datasets containing real images of Canada geese.

Figure 1 details the workflow of this study as there are two main parts of the training and detection of the YOLOv4-Tiny detector: obtaining the training dataset and the test data images. The training dataset contains individual datasets containing RGB images of Canada geese which are converted to greyscale via MATLAB and the test data images involves the pre-processing of NIR raw images that are captured by the RPAS and ground level camera equipment using the open-source photo editor, *Rawtherapee*. In addition, the RGB test images were also converted to greyscale to compare detection results with the NIR pre-processed version. Due to the rarity of openly available aerial images of Canada geese, ground level images were sourced, and artificial aerial images were generated to train and test the YOLOv4-Tiny detector for the initial investigation.

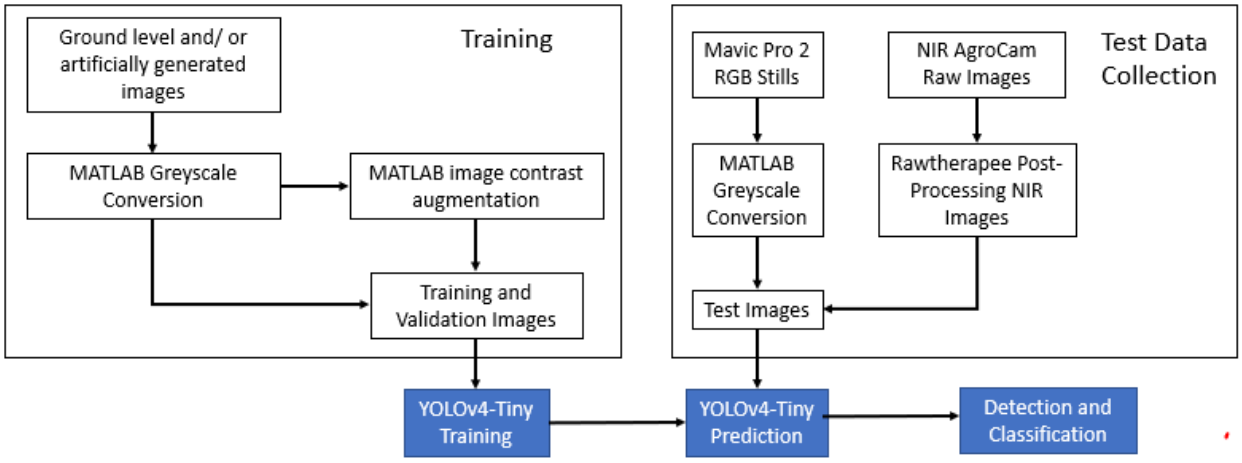


Figure 1 - A high-level flow chart detailing the process of the training and test methods for this study.

1.3 Contributions

In this preliminary study, the investigation of pre-processing NIR raw test images yielded an improvement in object detection by means of contrasts enhancements. In addition to pre-

processing NIR raw images, training the detector on an augmented dataset also yielded improvements in detection, most notably in the NIR test images in both the greyscale converted and pre-processed versions. It is also noted that the combination of real images and artificially generated images of Canada geese also generated detection improvements. The use of NIR equipment fitted to a RPAS also significantly reduces purchasing and operating costs compared to the traditional method to aerially count animals by counting from a helicopter.

The preliminary findings were presented as a full-length paper and an oral presentation to the 2021 Aerial Evolution Association of Canada Student Paper Competition (Szeto, Steele, & Laliberte, 2021) .

1.4 Thesis Outline

Chapter 2 will discuss the literature review and background pertaining to the topics discussed in this study. In Chapter 3, the methodologies used in this research will be discussed which includes the equipment used, YOLO version selection and training, the image pre-processing methods and sourcing of test images. Chapter 4 and Chapter 5 will present each detector study along with its detection results using different datasets. Lastly, the discussion and conclusions are explored in Chapter 6 and Chapter 7, respectively.

Chapter 2: Literature Review and Background

This chapter explains the literature review that served as the basis for this research. The main topics of this chapter are divided into the following sections:

2.1 Neural Networks

2.2 Animal Surveying Using Remotely Piloted Aircraft System

2.3 Multispectral Imaging

2.4 Image Pre-processing

Note that a fundamental background of neural networks is presented in Section 2.1.

2.1 Neural Networks

It was stated in *Artificial Intelligence: With an Introduction to Machine Learning* (Neapolitan & Jiang, 2018) that the roots of neural networks go as far back in the 1940s when AI was sought to be modelled after neurons of the human brain. The first theoretical approach to such model involved a binary variable that simply toggled between 1 and 0. Then in the mid-1950s, a program called the *Logical Theorist* was developed by Allen Newell, Herbert A. Simon and Cliff Shaw to simulate human like problem solving (Neapolitan & Jiang, 2018). This paved way for the development of complex AI systems which in the 1980s, knowledge-based AI systems were created. With the improvements in computing power in the 20th century, more complex AI algorithms were developed for deep learning purposes, which included CNNs.

The fundamental design of an artificial neural network (ANN) is modelled after the human brain (Figure *2* a)) in that a neural network is composed of artificial dendrites, synapses, somas and axons which are analogous to the components of the brain neuron.

Each component is described as follows:

- Dendrite – an input source to the neuron where each branch is an input path.
- Synapse – a connection between neurons where the transmission of electric nerve impulses takes place.
- Soma – the main body of the neuron where it processes retrieved information and executes logical functions.
- Axon – output line of the neuron which sends signals to the synapse.

There are two main operations in neural networks in the machine learning aspect: synaptic and somatic. During the synaptic operation, the strength of synaptic connections is formed (also known as "weights" in machine learning terms) which carries information from the previous neuron. The mathematical operations are executed in the somatic operation which will output a signal on the axon containing the aggregated information (weights) at a defined threshold (Figure 2 b)).

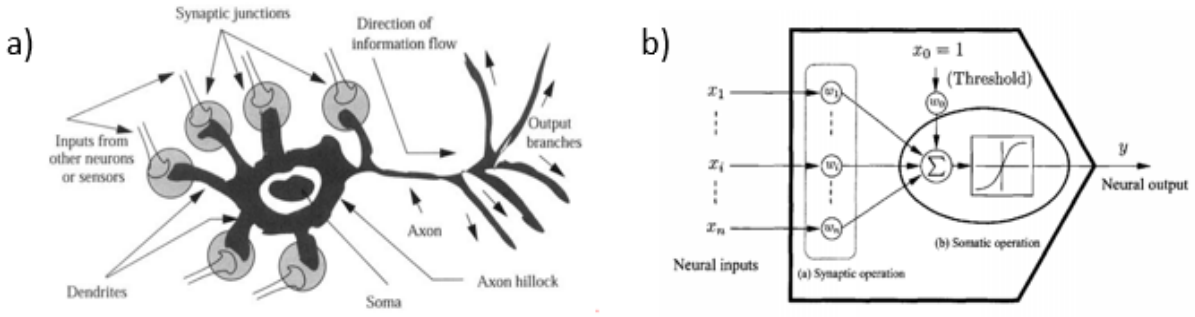


Figure 2 – a) An illustration of the human brain neuron and b) the neural network representation of a brain neuron (Gupta, Liang, & Noriyasu, 2003).

Thus, a neural network is composed of many individual neurons which are connected via synaptic connections where each neuron accounts of computational properties of the network. This is also called the fully connected layer and is where object detection probabilities are determined in CNNs (Figure 3). Each neuron connection has a value associated with it in the

hidden layer called weights and it is these weights that determine object prediction. These weights are self-learned during the training process and contain specific feature information of an object at each layer. The weights determine what features are present for example determining the hair style and face bone structure to establish if a female or male is identified. The weights are updated by calculating the difference between the measured value and the true value resulting from a loss function to revise the shared weights via backpropagation as a method of optimization. Once all the features are identified, the classification is done at the output layer using the information captured from the hidden layers.

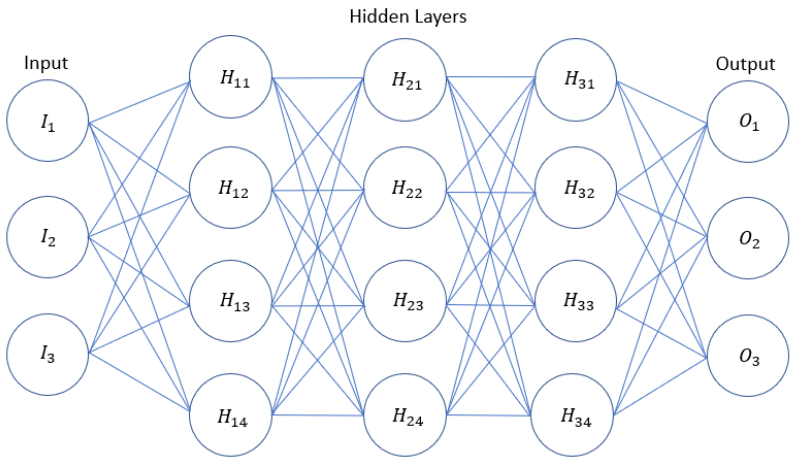


Figure 3 - A high level diagram of an ANN. The hidden layer which is denoted H_x contains the number of "neuron" layers.

To propagate to the next neuron in the hidden layer, an activation function is implemented in the ANN to activate the next neuron which is analogous to "neuron firing" that takes place when there is an electrical communication between neurons in our brains. The number of hidden layers is determined by design and could vary from one neural network model to another.

2.1.1. Convolutional Neural Networks

Convolutional neural networks (CNNs) are neural networks that involve the convolution mathematical operation of two matrices: the input matrix and the filter matrix, also known as kernels. The purpose of the convolution is to multiply the input and kernel matrices to extract specific object features from the input matrix, which results in a third matrix, the feature map. During training, the kernel matrix gets refined via backpropagation optimization which ultimately fine tunes the object information extracted from the input matrix which is then produced in the feature map. The input matrix contains pixel information from the input image which comprises of image intensity information in the form of a numeric representing on a scale from 0 – 255, where 0 contains no light and 255 contains maximum light. During the feature extraction process, the kernel moves in *strides* during the convolution process as seen in Figure 4 where is kernel is moving with a stride of one which produces a feature map when the convolution complete.

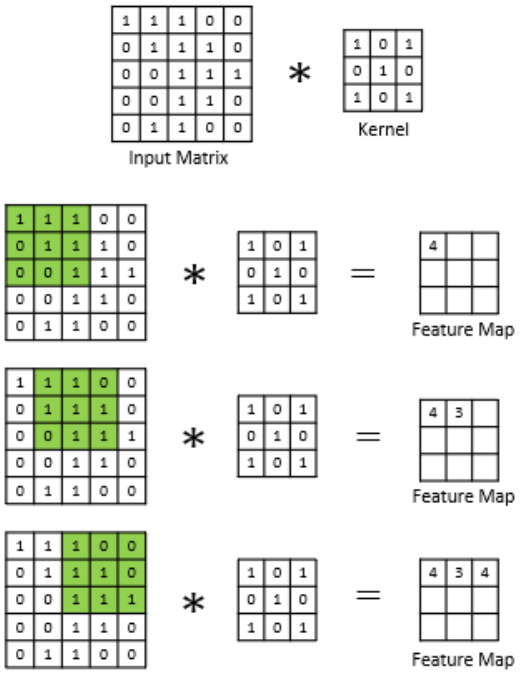


Figure 4 - Example of a convolution operation between the input matrix (5x5) and a kernel (3x3) of stride 1. The resulting feature map contains key information extracted from the input matrix.

The feature map contains key information about the input image such as colour or object edges. Once the feature map is produced, it then goes into the max pooling layer for further generalization and creates translational invariance. The convolution and max pooling layers can be repeated many times within a CNN model based on design (Figure 5).

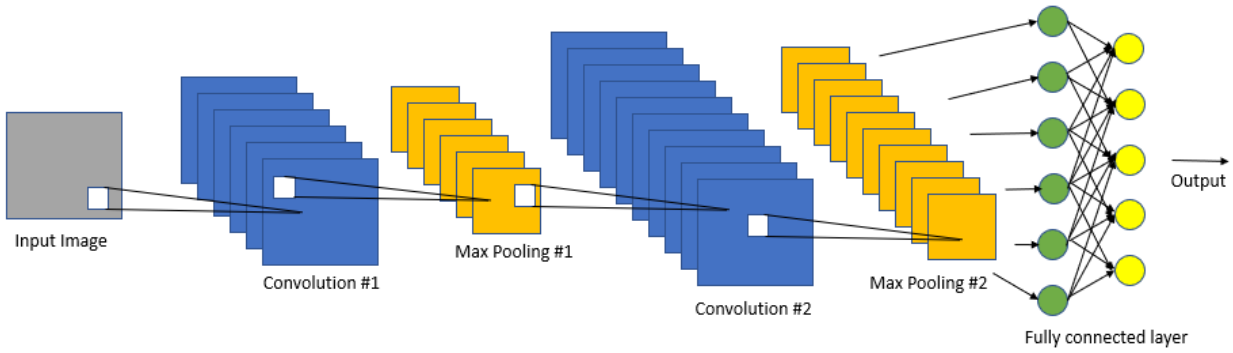


Figure 5 - Illustration of a convolutional neural network with two convolution and max pooling layers which are linked to a fully connected layer.

CNNs have gained popularity in recent years such that CNNs have become the base model for computer vision applications since the algorithms used in CNNs are an effective method to understand and extract object features in an image as learnable biases and weights (Bhatt, et al., 2021). The employment of CNNs have been widely utilized in the detection, recognition, and classification applications in various fields of study including autonomous vehicles, parasite detections and health imagining (Karasawa, et al., 2017; Abdurahman, 2021; Kayalibay, 2017). The first known CNN architecture was named LeNet which was developed in 1998 (Bhatt, et al., 2021). Since then, there have been many CNNs architectures developed where some of the popular architectures ResNet, GoogleNet, AlexNet, for example, utilize different optimization techniques within their algorithms. In addition to developing new CNN architectures, there have been studies to improve existing architectures such as enhancing

YOLOv3 and YOLOv4 models to detect small samples of blood smears for malaria parasite identification (Abdurahman, Fante, & Aliy, 2021).

2.1.2. You Only Look Once Convolutional Neural Network

YOLO is an open-source state-of-the-art supervised regression-based one stage CNN detection algorithm capable of object detection of multiple object classes in real time which consists of convolutional layers and a fully connected layer (Redmon, et al., 2016). Unlike traditional CNNs for object classification, which often use a sliding window approach to compute convolutions one stride at a time, the YOLO algorithm analyzes the image in its entirety (hence "you only look once" - YOLO) during the training and detection process. The YOLO algorithm splits the input image into grid cells where each cell predicts the bounding box around the detected object. YOLO uses an opensource neural network framework written in C and CUDA called Darknet which is also created by the same developer as YOLO (Redmon J. , 2013-2016). Darknet provides the fundamental backbone of YOLO and is required for YOLO to operate, as the backbone is responsible for feature extraction of the input image during training and testing of images or video frames.

YOLO has several convolution and max pooling layers in series (how many depends on the YOLO version) and two fully connected layers. When the extracted features have passed through the convolutional and max pooling layers, the data goes into the fully connected layer where object detection occurs which predicts both the bounding box coordinates and object class probabilities, also known as confidence scores. The fully connected layer uses a nonlinear activation function and non-max suppression to select the best prediction and the very last layer uses a linear activation function. A loss function is employed to refine the training and is defined in Redmon, et al., 2016. YOLO is trained using an input image dataset that is split into separate

training and validation sets, where the training set is used to train the detector object features and the validation set is strictly used to validate the detector on what it has learned from the training dataset.

The bounding box prediction that occurs in the last layer is responsible for determining the intersection over union (IOU) threshold to quantify the overlap between the ground truth (annotated) bounding box and the predicted bounding box and is defined in (1) and illustrated in Figure 6.

$$IOU = \frac{Area\ of\ overlap}{Area\ of\ union} \tag{1}$$

An IOU threshold is a pre-defined variable in the neural network and a positive detection is identified when the calculated IOU exceeds the pre-defined target IOU threshold.

Figure 6 – The ground truth box (red) is the annotated bounding box, and the predicted bounded box (yellow) is the estimate from the detector algorithm. The IOU is a result of the overlap between the ground-truth and predicted bounding box (Halley Szeto).

In addition to the IOU to measure detector performance, another metric, the mean average precision (mAP) is the main metric used in image classification when using neural

networks. The mAP determines how well the neural network detector can accurately identify all the true positives. The mAP uses the following metrics for calculation:

- Precision – how many of the total bounding box predictions are correct and is defined by

$$P = \frac{TP}{TP + FP} \tag{2}$$

  where TP is the number of true positives and FP is the number of false positives.

- Recall – how many of actual bounding targets were identified and is defined by

$$R = \frac{TP}{TP + FN} \tag{3}$$

  where TP is the number of true positives and FN is the number of false negatives (failed to predict that an object was present).

The precision and recall values are calculated for each image in the dataset which will form a precision-recall curve where precision is plotted against recall. The resulting area under the precision-recall curve is the mAP. It is ideal to have a high mAP that yields a high recall since the recall correlates to the precision of the detector.

2.1.2.1. Initial You Only Look Once Study of Canada Geese Detection

An initial study using the YOLO detector to detect Canada geese was initially completed by a former undergraduate co-op student of Dr. Alan Steele, Matthew Walsh. The study was completed using the embedded computing board NVIDIA Jetson Nano Developer Kit (NVIDIA Corporation, n.d.) made specifically for deep learning development and applications. YOLO was installed and configured on the Jetson Nano where a variety of default YOLO models were tested and analyzed using a custom training dataset of Canada geese. A real time detection

program, *Opendatacam* (Groß, et al., n.d.) was also installed and configured to test real time detection of videos containing Canada geese using the trained algorithm of the custom dataset.

To train a custom YOLO detector, Mr. Walsh sourced 423 RGB openly available images of Canada geese using the Fatkun Batch Download Image Google Chrome tool, which is publicly available in the Google Chrome Web store, and the images were manually annotated using an online annotation tool *makesense.ai* (Skalski, 2019) where the ground truth boxes were labelled as "Goose".

Different YOLO versions were examined, and it was found that YOLOv3-Tiny had better performance since it was able to detect most geese in a National Geographic test video containing various clips of Canada geese when *Opendatacam* was used. The study, which used training datasets of 160, 185 and 423 of Canada geese images, found that the dataset size correlated with the training duration determines the accuracy of the detection results (Walsh, Artificial Intelligence Based Monitoring System, 2020).

The study presented in this thesis will continue the investigation of the YOLO detector to detect Canada geese by increasing the dataset size, augmenting training, and validation images, and employing artificially generated images to train YOLO.

2.1.3. Neural Network Training with Greyscale Images

Greyscale images are single channel images and have been fundamental in feature extraction techniques such as Oriented FAST Rotated BRIEF (ORB), Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) (Lowe, 1999; Rublee, Rabaud, Konolige, & Bradski, 2011). Since greyscale omits the colour components (the red, green, and blue colour channels), the greyscale pixels only contain intensity information which defines image contrast (Nixon & Aguado, 2012). This is because each greyscale pixel intensity is

relative to its surrounding pixels, making it easier to distinguish object features. The pixel intensity information follows the 0-255 scale as described in Section 2.1.1 and greyscale can be described in many shades of grey which provides enough information for the detector analyze and extract features (Nixon & Aguado, 2012). This also results in object edge contrast since the pixels on the boundary of the object potentially differs in intensity to its surroundings. This was the case in Bui, et al., 2016, where their CNN was trained on RGB images but tested on RGB converted greyscale images and found that the greyscale test images yielded an improvement of +1.4% in detector accuracy over the same RGB test images (Bui, et al., 2016). In addition, because greyscale lacks colour information, it requires less computing memory to process convolutional calculations (Bui, et al., 2016; Santoso, Suprapto, & Yuniarno, 2020) and allows for faster training time (Ng, Tay, & Goi, 2013).

There have also been studies using greyscale images to train CNNs. In Santoso, et al., 2020, YOLOv3-Tiny was trained on greyscale images and used RGB as test images to detect copper inscriptions which yielded a high average detection accuracy of 97.93% (Santoso, Suprapto, & Yuniarno, 2020). An investigation completed by Ng, et al., 2013, compared training CNNs with greyscale, RGB and YUV images, and it was found that training on greyscale images produced the lowest detector error rate (Ng, Tay, & Goi, 2013).

2.1.4. Training Neural Networks with Artificially Generated Images

Due to challenges of gathering a large dataset containing specific scenarios such as collecting images of spilled loads on freeways (Zhou, et al., 2021) that could be hazardous to the photographer, there have been investigations on training neural networks with computer generated synthetic images to mitigate this issue. In addition, synthetic images can reduce manual labour and time to gather and annotate images for a large dataset (Dewi, et al., 2021).

Another advantage of using artificially generated images is that environmental factors which may be difficult to reproduce such as varying backgrounds, unusual weather conditions, seasonal effects and time of day can be manipulated (Barisic, et al., 2022).

The use of synthetic images proved promising results when the synthetic images were trained alongside with real images. In Barisic, et al., 2022, it was shown that a detector trained on synthetic images and then fine-tuned on real images for 20 training cycles yielded a mAP improvement of 5% compared to the mAP of the real image only trained detector (Barisic, Petric, & Bogdan, 2022). It was also shown in Dwei, et al., that training the YOLOv4 detector with a dataset containing a combination of real and synthetic images increased the detector accuracy by 20% compared to the accuracy of the real image only trained detector (Dewi, et al., 2021).

In the case for this study, there is a lack of publicly available aerial images of Canada geese that we can use to train our detector. Thus, we have explored generated synthetic images of aerial images at varying heights to train the YOLO detector alongside real images of Canada geese.

2.2 Augmented Training Dataset

CNNs are prone to overfitting which causes the trained detector to perform poorly when detecting objects in test images. This occurs when the detector is trained too closely to the images contained in the training dataset (Shorten & Khoshgoftaar, 2019) which creates problems when the detector is tested with images not part of the training dataset. To reduce overfitting, image augmentation has been shown as a valid method to artificially increase the training dataset and enhance model generalization (Shorten & Khoshgoftaar, 2019; Bloice, Stocker, & Holzinger, 2017). This is because datasets with augmented data showed better performance than

datasets without augmented images (Shijie, Wang, Peiyi, & Siping, 2017). Examples of image augmentation include image rotation, cropping, scaling, the addition of noise, and adjusting the contrast of the image.

Because of the benefits found with the implementation of augmented images in the dataset, images have been augmented in the training and validation datasets used in this study. The performance of the non-augmented trained detector was also analyzed and compared with the augmented trained detector.

2.3 Animal Surveying Using Remote Piloted Aircraft Systems

Previous methods of animal surveying involved using manned aircrafts however, due to high costs associated with aerial surveying, alternative methods of surveying were investigated. The employment of RPAS to monitor wildlife habitats and migration patterns are a proven technique to assist scientists in understanding animal behavior and the impacts of global warming on animal populations. This is because RPASs offer many advantages such as low operation costs, simpler logistics and excellent temporal and spatial resolutions (Linchant, Lisein, et al., 2015). There are three main animal groups where RPASs have been utilized to survey: aquatic animals, large terrestrial animals, and birds (Linchant, et al., 2015).

Bird surveying has long been a hobby for bird enthusiasts, in addition to surveying for scientific research, where the main method of surveying is the use of binoculars. In 2018, a surveying team arrived at a shorebird conservation site James Bay, Ontario, to monitor shorebird populations and habitats, however, their shorebird counts were strictly done by manual observation (Iron, 2018). This requires the surveyor to survey on foot which could potentially cause disturbance to the animal, is labor intensive and prone to human error. The use of RPAS to replace manual surveying provides an aerial view from a specific flying height which will

provide a larger view of the breeding grounds or habitats for easier counting. D. Chabot and D. Bird were early pioneers using RPASs to complete animal surveying starting in 2012 where they found that employing RPAS to survey flocks of birds have been shown to precisely detect birds that have contrast with its surroundings (Chabot & Bird, 2012). In addition, D. Chabot and D. Bird have investigated automatic object classification using image sampling points in 2013 (Chabot & Bird, 2013) which paved way for utilizing AI to automatically count the number of bird occurrences. In Kellenberger, et al., 2021, the use of CNNs to detect and count 21,000 seabirds on the coast of West Africa took 4.5 hours, as opposed to three weeks of manual labor to complete the same task (Kellenberger, Veen, Folmer, & Tuia, 2021).

2.4 Multispectral Imaging

The visible light spectrum is the spectrum that yields colours that the human eye can see and falls under the wavelength range starting with violet at 400 nm to red at 700 nm, where wavelengths above 700 nm are in the infrared spectrum. There are four main ranges within the infrared spectrum: near infrared, infrared, thermal infrared and far infrared which operates at wavelengths of 1 um, 10 um, 100um and 1 mm respectively (Johnsen, 2012). In this study, images in the NIR spectrum will be investigated as NIR optical equipment is utilized.

Healthy vegetation is known to reflect NIR wavelengths which could be used as an indicator of plant health (Woodhouse, et al., 1994), thus the difference between the NIR reflectivity and the NIR absorption on non-reflective surfaces can potentially provide enhanced contrast between the animals and the surrounding background vegetation to improve detection. However, NIR wavelengths are absorbed by water (Langford, et al., 2001) which results the water appearing dark in the NIR images thus NIR may not be suitable in image scenarios containing scenes of water. In addition to potentially enhancing contrasts in images, the NIR is

explored for this study because this spectrum preserves detail in the images rather than gathering emitted heat signatures that are obtained in the further infrared spectrum. In addition, the equipment required to gather images in NIR is simpler and less expensive than the equipment needed to gather images in wavelengths greater than the NIR wavelength.

2.4.1. Bird Reflectivity

It has been found that some species of birds are able to reflect NIR wavelengths which could be credited to an evolutionary trait for thermal regulation which is particularly true for the feathers of birds of paradise found in Australia (Medina, et al., 2018). In addition, there have been studies where bird feathers were shown to reflect NIR wavelengths given the condition of the feather barbule structure and density (Stavenga, et al., 2015). However, feather densities are correlated to the size of the bird with larger birds having thicker, larger, and sparser feathers (Kozák, 2011) and it was also found that birds containing less feather densities, mainly feather barbules, reflected less NIR wavelength (Stuart-Fox, et al., 2018). Given that the Canada goose is a larger size bird, there may be limited NIR reflection on the geese feathers which will potentially enhance contrast of the geese in the NIR images if its surroundings consist of healthy vegetation.

2.5 Image Pre-processing

Image pre-processing is a method to enhance images to yield improvements in detail of the photograph before it is fed into the neural network for training or detection. To pre-process images, the image must be in the raw format since the raw image allows the user to edit the image “as is” since raw images contain direct information captured by the camera’s sensor. Other camera image extensions, such as Joint Photographic Experts Group (JPEG), are pre-processed and contain a lossy image compression. Some of the common pre-processing techniques are editing the contrast, luminosity, saturation and adding noise. In this study, the

training image datasets will be pre-processed by converting the images into greyscale and strictly be focusing on editing the image contrasts and including Gaussian smoothing. As for the test images, the images will be pre-processed in both MATLAB and in an open-source photo editor to enhance image contrasts.

There are many colour models that can be utilized to manipulate an image ranging from application specific colour spaces, human visual system (HVS) spaces and Commission Internationale de l'éclairage (CIE) spaces (Tkalcic & Tasic, 2003). However, due the limited offerings of the open-source pre-processing software, *Rawtherapee*, which was used to pre-process raw test images in this research, we will only be briefly discussing the colour spaces that is offered by the software which are the RGB and CIE L*a*b colour spaces.

2.5.1. The RGB Colour Space

The RGB colour space is an additive colour space of light using green, red, and blue as primary colours based on the HVS colour space and is modelled after colour sensitive cones in the human eye, where there are three cones in the retina which are sensitive to red, blue, and green wavelengths. This colour space is represented as a spectral power distribution of the primary colours as a function of wavelength which is known as Grassmann's Law given in equations (4)-(6) where $I(\lambda)$ is the spectral power distribution and $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, and $\bar{b}(\lambda)$ represent the colour matching functions with respect to the selected primary colour.

$$R = \int_0^\infty I(\lambda)\bar{r}\,(\lambda)d\lambda \tag{4}$$

$$G = \int_0^\infty I(\lambda)\bar{g}\,(\lambda)d\lambda \tag{5}$$

$$B = \int_0^{\infty} I(\lambda)\bar{b}\,(\lambda)d\lambda \tag{6}$$

The RGB model is considered the most basic colour space model and can be further transformed to yield variants of additional colour spaces. However, the RGB model does provide difficulty in determining specific colours as there is a strong relationship between the primary colours. In addition, this colour space is mainly used in electronics and can vary on different monitors as light is the main driver to produce the saturation and luminance of the colour.

2.5.2. The CIE L*a*b Colour Space

The CIE L*a*b colour space was proposed and standardized by the CIE and was designed to have a uniform colour space to align with the human visual perception. There are three channels associated with the CIE L*a*b colour space where L represents the human perception of lightness and, a and b are axis representing shades of colour, green to red and blue to yellow respectively. A division of white points (white references) are used to normalize the colour values of this colour space. This colour space is a derivative of the CIE XYZ colour space where the CIE XYZ colour space is represented by a tristimulus system where red, green, blue which are characterized by a XYZ coordinate system. The equations represented by the CIE L*a*b colour space are transformed from the CIE XYZ colour space which are given in equations (7)-(9) where X, Y Z denote the tristimulus values and X_N, Y_N and Z_N represents the white colour stimuli reference.

$$L^* = 116f\left(\frac{Y}{Y_N}\right) - 16 \tag{7}$$

$$a^* = 500\left(f\left(\frac{X}{X_N}\right) - f\left(\frac{Y}{Y_N}\right)\right) \tag{8}$$

$$b^* = 200\left(f\left(\frac{Y}{Y_N}\right) - f\left(\frac{Z}{Z_N}\right)\right) \quad (9)$$

The advantage of this model is that it is device independent, meaning that the colour produced is not dependent on the electronic hardware and instead, provides a white reference which offers the amount of light to produce the colour. However, one of the drawbacks of this colour space is if the white references or the XYZ values are incorrect, it could potentially manipulate the target outcome colour into a varying shade.

Chapter 3: Methodology

In this chapter, the equipment and software used in this study will be discussed. The main sections in this chapter are:

3.1 Equipment

3.2 You Only Look Once Convolution Neural Network

3.3 Software Applications

3.4 Test Images

3.1 Equipment

This section will discuss the RPAS, cameras and computer system to run and process the YOLOv4-Tiny detector algorithm used in this research. The DJI Mavic Pro 2 RPAS was used as the RPAS to take aerial images of Canada geese decoys and the NVIDIA Jetson Nano embedded system was selected to train and test the YOLOv4-Tiny detector. A set of two cameras with RGB and NIR capabilities were purchased from AgroCam along with a camera tripod to complete an initial study comprising of ground test images of Canada geese. The NIR camera would then be later mounted to the RPAS to take NIR raw images from an aerial perspective.

3.1.1. DJI Mavic Pro 2 RPAS

The DJI Mavic Pro 2 RPAS (Figure 8 c)) was employed as the RPAS for this study as it was readily available. The Mavic Pro 2 is an advanced category RPAS as defined by Transport Canada which is approved for flights in a controlled airspace and near bystanders within 30 m (Transport Canada, 2021). The RPAS is compact, and weights 907 g and its unfolded dimensions are 322×242×84 mm. The onboard camera contains a 1” CMOS sensor and has a field of view (FOV) of 77°, and the video function can gather videos in 4K, 2.7K and 1K full high definition (DJI, 2022). The camera is located at the head of the RPAS which is attached to a gimbal and

can be dynamically controlled. For our videos used in this study to capture RGB stills, the 4K option was selected. The field work with the RPAS was done with Dr. Jeremy Laliberte who is a licensed RPAS operator.

3.1.2. Optical Equipment

Off-the-shelf digital cameras were initially used to collect trial images in the visible and NIR spectrum. The Panasonic Lumix (DMC-LS75) camera with its IR cut filter removed was used to capture NIR images, and a Canon PowerShot (SD13000 IS) was used to take images in the visible spectrum. Standard off-the-shelf cameras contain IR cut filters in their lenses to block infrared wavelengths that the camera's sensor can capture, which could distort visible RGB colours in the images. When the IR cut filter is removed from the lenses, this allows IR wavelengths to pass through the lenses which is captured by the camera's sensor. Figure 7 illustrates the effect of removing the IR cut filter from a digital camera.

Figure 7 – Sample NIR image of vegetation on a sunny day taken with the Panasonic Lumix camera with the IR cut filter removed (Alan Steele).

The off the shelf cameras used in the initial trial were not compatible with RPAS to use during flights thus additional cameras were identified and purchased. Two optical cameras which are compatible with the Mavic pro 2 were purchased from AgroCam (Figure 8 a)), a company

that specializes in equipment for normalized difference vegetation index (NDVI) applications using RPAS and offers camera capabilities in the NIR and RGB spectrum. The AgroCam cameras came with a 3D printed mount made specifically for the Mavic Pro 2 RPAS. The mount was manually attached onto the RPAS where the camera is mounted to the bottom of the RPAS (Figure 8 c)). A tripod and a dual camera mount (Figure 8 b)) were also purchased so the cameras could be mounted as a pair to gather ground level test images of Canada geese in public parks within the City of Ottawa to undertake the initial testing of the YOLOv4-Tiny detector. The RGB and NIR cameras were configured on the left and right respectively each time the tripod was used to take images. A set of geese decoys were also purchased to capture aerial images from the Mavic Pro 2 as taking images of real Canada geese proved challenging as the specifications of the cameras were not suited to take images at RPAS flight altitudes that would not disturb the birds.

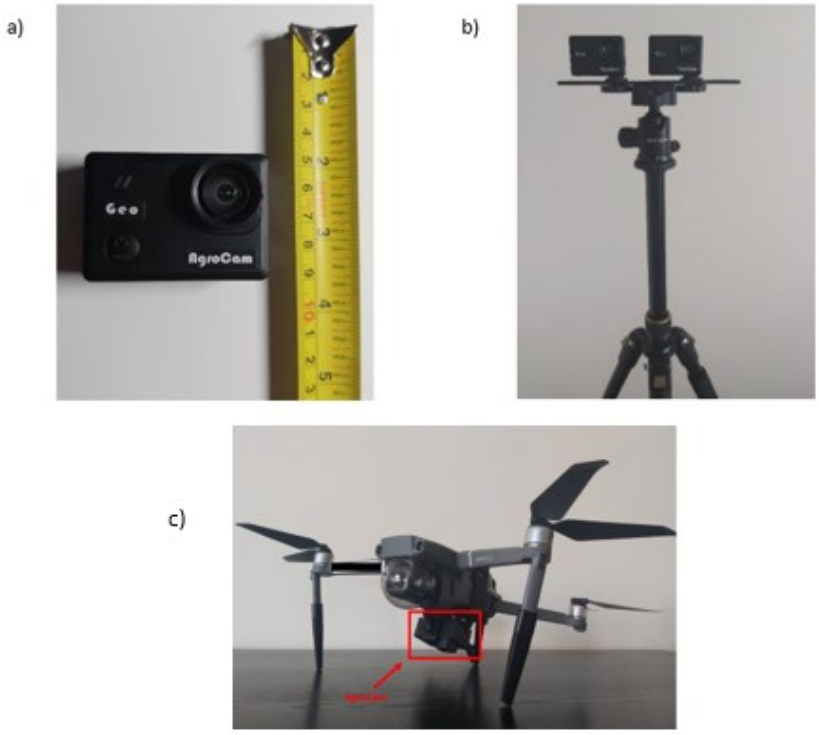


Figure 8 – Images of the a) AgroCam, b) dual camera mount with both the NIR and RGB cameras mounted, and c) the NIR camera mounted on the Mavic Pro 2 RPAS.

The AgroCam NIR camera also contains a blue filter lens that suppresses the RGB wavelengths and is only sensitive the NIR wavelengths and the violet wavelength (Norward Expert LCC, 2017) as seen in the spectral response of the AgroCam blue filter lens in Figure 9.

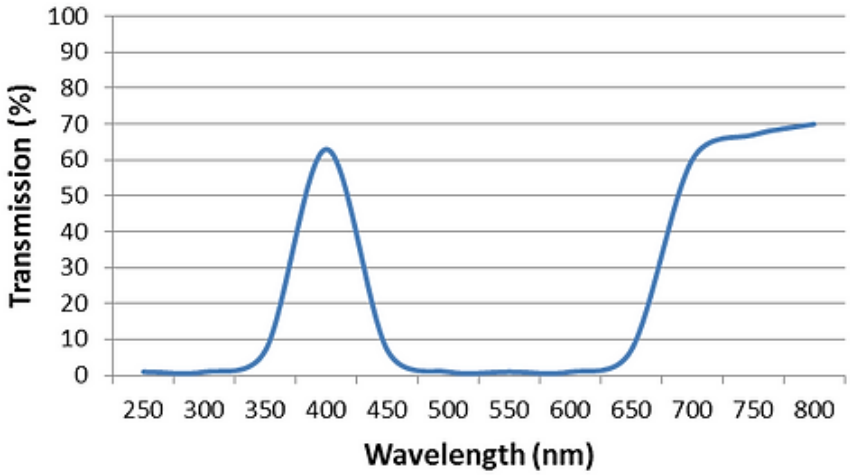


Figure 9 - The spectral response of the blue filter lens on the NIR AgroCam camera. The lens is sensitive to NIR wavelengths and wavelengths in the violet range *(Norward Expert LCC, 2017)*.

The AgroCam cameras has a sensor size of 1/2.3", a focal length of 4.35 mm and a FOV of 82° and is capable of capturing images in the raw format. Given that the sensor size of the AgroCam camera is smaller than the sensor of the Mavic Pro 2 and the focal length of the Mavic Pro 2 is larger than the AgroCam camera, it may seem apparent that the images taken with the Mavic Pro 2 seem closer than the images taken with the AgroCam when the images are taken by the same flying altitude. To prove this, we can calculate the angle of view (AOV) of the cameras which is dependent on the sensor size and the true focal length of the lenses. Using equation (10), the AOV for the Mavic Pro 2 and the AgroCam is 65° and 71.37° respectively, given the sensor dimensions as stated previously. An example showing the camera AOV differences is highlighted in Figure 10.

$$AOV = 2 * atan\left(\frac{Sensor\ Dimension\ (mm)}{2xFocal\ Length\ (mm)}\right) \quad (10)$$

This verifies that images taken by the onboard camera of the Mavic Pro 2 will seem closer compared to the images taken by the AgroCam due to its smaller AOV.

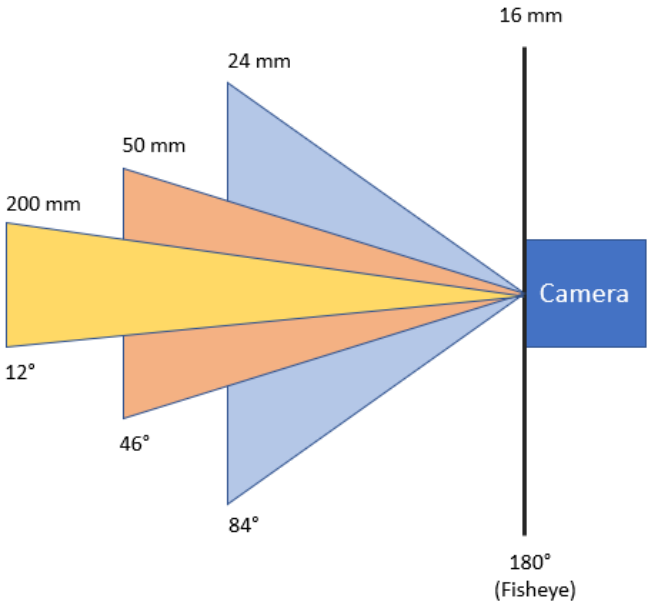


Figure 10 – A comparison of 12°, 46°, 84° and 180° AOVs. Notice that the smaller the AOV, the more telescopic the lens become.

3.1.3. NVIDIA Jetson Nano

The NVDIA Jetson Nano developer kit model P3450 (Figure 11) was selected as the computing hardware to train and test the YOLO detector. The Jetson Nano is a Linux operated, low power, embedded system-on-module computer that can develop and execute AI algorithms. The developer kit contains 4 GB of memory and utilizes the NVIDIA 128-core Maxwell architecture and Quad-core ARM Cortex-A57 MPCore for the graphical processing unit (GPU) and central processing unit (CPU) respectively. The Jetson Nano uses the NVIDIA developed compute unified device architecture (CUDA) for the GPU which allows for parallel computing and utilizes application programming interfaces (APIs) to operate the GPU for general purpose processing to accelerate its CPU performance. To start using YOLO on the Jetson Nano, YOLO

and its associated files and programs were downloaded from Github (Bochkovskiy A. , n.d.) and installed onto the computer.

Figure 11 – The fully encased NVIDIA Jetson Nano that is used in this study (Matthew Walsh).

3.2 You Only Look Once Detection

The YOLO model was pre-trained on the ImageNet 1000-class competition dataset (Russakovsky, et al., 2015) and can detect 20 class objects from the Pascal VOC 2012 dataset (Pascal2, n.d.) which includes the following class objects: person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa and tv/monitor. If YOLO detects an object in an image, it will produce a bounding box around the object. The bounding box will also have a confidence score which determines the accuracy of the bounding box and probability of the bounding box containing the detected object. To test the pre-trained model on non-RGB images, the Canon PowerShot was used to capture images containing cars and a cat in greyscale and sepia (Figure 12).

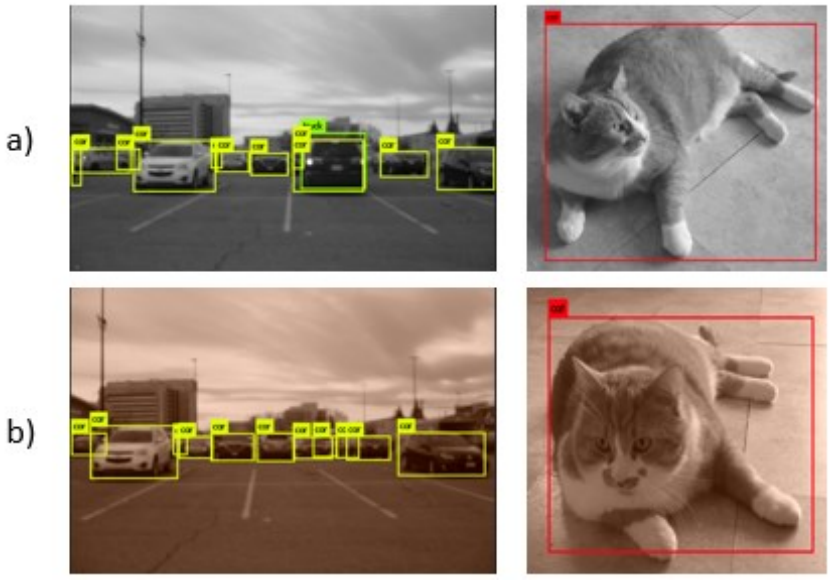


Figure 12 – "Cat" and "car" were able to be detected regardless of a) greyscale and b) sepia image hues of the images taken by the Canon PowerShot digital camera (Jacqueline Szeto).

In both the greyscale and sepia instances, the detector was able to detect the cat with a 100% confidence score. The confidence scores of the car detections in the parking lot images are between 27%-98% and 34%-99% for the greyscale and sepia images respectively. This test confirms that YOLO can detect using spatial information in addition to colour information since the ImageNet dataset contains RGB images.

3.2.1. You Only Look Once Version Selection

The recommended YOLO version for the computing hardware used in this research is YOLOv3-Tiny (Walsh, Jetson Nano setup and Opendatacam Installation, 2020) however, a newer version, YOLOv4, was released at the time of this work. YOLOv4 utilizes an enhanced cross stage partial Darknet53 (CSPDarknet53) backbone instead of the Darknet53 backbone employed in YOLOv3. The cross stage partial network (CSPNet) was developed to improve computation cost and enhance the learning capability of the detector (Wang, et al., 2020). YOLOv4 also comes with "bag of specials" and "bag of freebies" enhancements which contain the following (Bochkovskiy, Wang, & & Liao, 2020):

- Bag of specials
    - Spatial pyramid pooling (SPP) – removes the fixed size constraint of the network and enhances the receptive field sizes of the detection layer maps.
    - Path aggregation network (PAN) – Propagates instance segmentation information while maintaining spatial information.
- Bag of freebies
    - Image augmentation using mosaic, CutMix and self-adversarial training (SAT).

YOLOv4 also has a 'tiny' version where it offers a condensed neural network which contains fewer convolutions of 29 compared to its YOLOv4 counterpart which encompasses 53 convolutions. Though the performance of YOLOv4-Tiny is less than YOLOv4, YOLOv4-Tiny reduces hardware capability requirements during training and detection because it requires less computing power and memory (Bochkovskiy, et al., 2021). To compare the YOLOv3-Tiny and the YOLOv4-Tiny detectors, a series of detection tests were done on geese images obtained with the AgroCam RGB camera at various public parks within the City of Ottawa. The default pre-trained weights and configuration files from the YOLO GitHub were used to test the detectors as an initial test. As seen in Figure 13 and Figure 14, it can be clearly seen that YOLOv4-Tiny performs better than YOLOv3-Tiny. Though geese were detected as "bird" in the YOLOv3-Tiny test, the bounding boxes are incomplete (they do not bound the entire object) and there are misdetections as some geese are labeled "cow" in a) and d) of Figure 13. In addition to detecting more geese in image Figure 14 b) in the YOLOv4-Tiny results, YOLOv4-Tiny was able to detect all vehicles present in the samples however, there are still misdetections in all test samples.

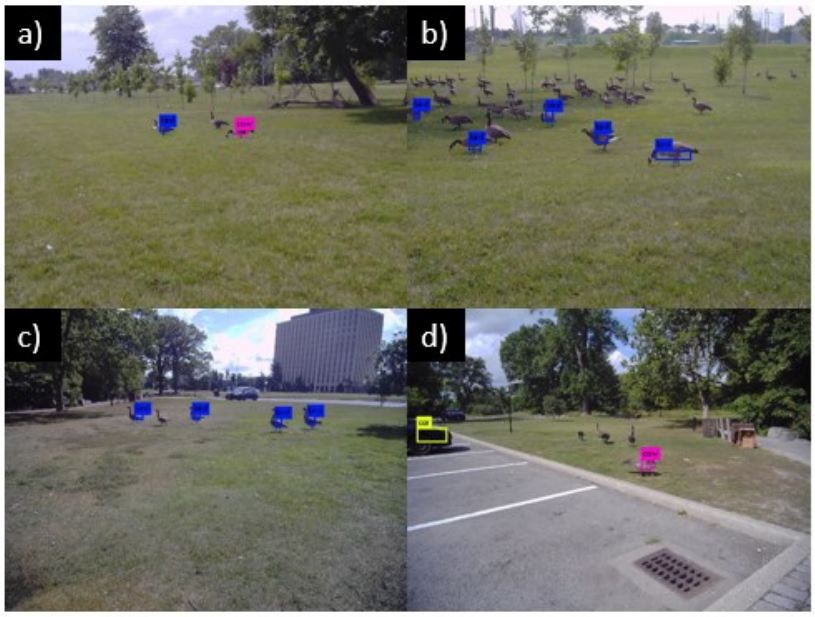


Figure 13 – Detection results using YOLOv3-Tiny.

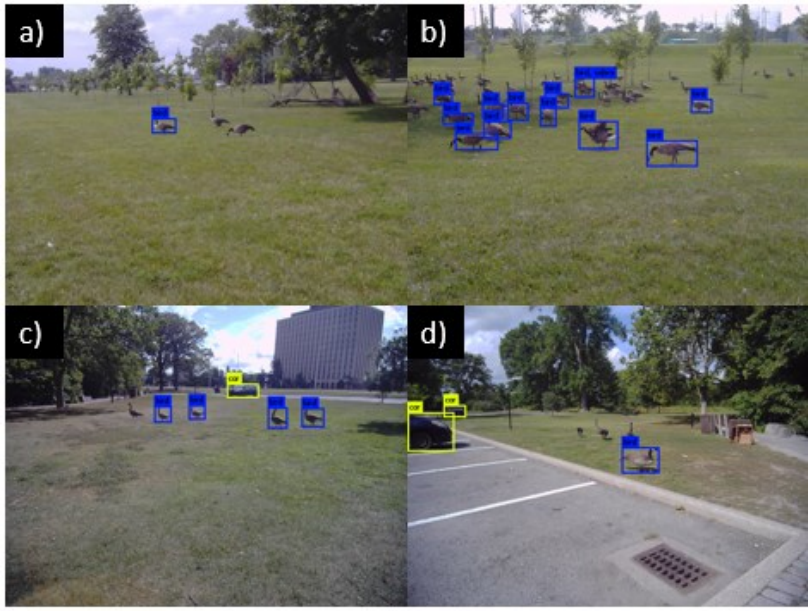


Figure 14 – Detection results using YOLOv4-Tiny.

YOLOv4-Tiny might not be able to detect all geese in the test samples above because the image dataset that was used to pre-train the detector may not contain images of birds in varying positions and resolutions. However, based on the results above, YOLOv4-Tiny was selected to be used as the algorithm in this study.

3.2.2. You Only Look Once Training

To start training the YOLOv4-Tiny detector on the custom Canada geese dataset, the following command is used in a command window in Linux:

./darknet detector train cfg/geese.data cfg/YOLOv4-Tiny-cfg YOLOv4-Tiny.conv.29

Where the following files are geese.data, YOLOv4-Tiny.cfg and YOLOv4-Tiny.conv.29. These files were downloaded from the Darknet GitHub (Bochkovskiy A. , n.d.) and include the following:

- geese.data – Contains the file directory paths to each image, along with its annotations (images and its corresponding annotation .txt file must have the same file name and be located in the same folder), in the training and validation sets which are uploaded into the Darknet folder path. The dataset is split into the training and validation datasets prior to the start of detector training.
- YOLOv4-Tiny.cfg – Configuration file of the YOLOv4-Tiny algorithm which comprises of CNN parameters, such as network resolution, subdivision and batch and stride, which can be edited in this file. A sample of this file can be viewed in Appendix C.
- YOLOv4-Tiny.conv.29 – Initial YOLOv4-Tiny weights file which gets updated during the training process.

The batch, subdivision, maximum batch, height and width, and class parameters in the yolo4-tiny.cfg file were the parameters that were edited for this research as the batch, subdivision, height, and width are dependent on the memory requirements of the CPU. Each parameter is defined as follows:

- Batch (default configuration value = 64) – Number of samples taken from the training dataset to train at a time.
- Subdivision (default configuration value = 1) – A divisor for the batch number to split the number of images sent to the GPU for parallel processing known as “mini batches”. The lower the divisor, more samples are sent for processing which requires more CPU memory. This parameter is strictly used to divide processing work.
- Maximum batch (default configuration value = 2000200) – Number of training iterations. It is recommended that the maximum batch is set as 2000*(number of class objects) as defined in the YOLO developer notes in GitHub.
- Height and width (default configuration value = 416) – Network resolution as YOLO will automatically resize input images to the defined height and width dimensions to feed it into the network algorithm for processing. The larger the dimensions, the more memory the detector will use and increase training time but will improve detector performance (Redmon & Farhadi, 2018; Bochkovskiy A. , n.d.). The value must be a multiple of 32.
- Class parameter (default configuration value = 80) – The number of object classes the detector will be trained or is trained to detect. YOLO can be trained to detect multiple object classes simultaneously or can be trained to only detect one class.

The pre-trained YOLOv4-Tiny is trained to detect 80 class objects, but in the case of this research, we are only concerned with the detection of one class object since we are training the detector with one annotation label which is “goose”. Thus, 80 was changed to one for the class parameter. In addition, it was required that the filters parameter before each YOLO layer in the .cfg file was changed to support one object class following the formula filters = (classes + 5) x 3. The batch and subdivisions parameters determine how the training dataset is split into “sets”

which is sent to the GPU to process and is limited by the capability of the computing hardware. While the default batch value is 64, there have been research investigating the optimal batch number for training. In Radiuk, 2017, a batch size between 16 and 1,024 was studied on the MNIST dataset containing 60,000 samples and it was found that using a higher batch size yielded higher detector accuracy but required longer training times and more CPU memory.

YOLO produces a weight file after each 1,000th iteration during training and is bounded by a maximum batches parameter which is defined as (number of classes) × 2000 (Bochkovskiy A. , n.d.). In addition to the weights file produced at each 1,000th iteration, YOLO will also produce a best weights file once training is complete which contains the optimal weights in comparison to other weights that were produced. The number of weight updates should be used cautiously as the model could be overfitted if the weights are updated too much since the number of weight updates is directly related to the training convergence. If the model is overtrained, then it is trained too closely to the dataset containing the training images and will not perform well against test images that do not bear resemblance to the images contained in the training dataset. YOLO calculates its mAP values against the images contained in the validation dataset.

Each weight file that was produced from the duration of training can be queried to generate a report containing the calculated IOU and mAP values associated with each weight file. The weight statistics also contain additional performance metrics such as the precision and recall values, and detection count. To test the mAP and IOU values for each weight, the following command was used:

./darknet detector map cfg/geese.data cfg/YOLOv4-Tiny-cfg backup/"weights file"

3.3 Software Applications

Software was required to pre-process the manually captured test images and as well to augment the images in the training and validation dataset. An open-source software program *Rawtherapee* was used to pre-process the NIR test images and MATLAB was used to augment and convert RGB to grayscale images in the training and validation training sets. In addition, the open-source software *Blender* was used to create and generate the artificial aerial images of Canada geese.

3.3.1. Rawtherapee

Since there is currently no standardization on raw extension images, camera manufacturers may have their own inhouse raw extension format, other than raw, that could only be viewed using software provided by the camera manufacturer. While AgroCam did not have inhouse pre-processing software, we were provided with an image converter by AgroCam (Gabor, 2018) that converted the raw images to digital negative (DGN) extension images. The DGN format maintains the raw data captured by the camera sensor but has greater usability because it is a generic raw format and not bounded by defined camera specifications. To pre-process images obtained from the AgroCam cameras, the raw images were pre-processed using an opensource photo editor *Rawtherapee*.

RawTherapee 5.8 (RawTherapee, n.d.) is an opensource image processing software capable of processing raw images and provides RGB and L*a*b colour spaces to pre-process images. The L*a*b colour space was chosen to pre-process the raw NIR images as the model represents the human perception of colour. The main objective of pre-processing the raw NIR images was to strictly gain object contrast improvements. This was done by changing several parameters within the software to enhance the contrast of the geese within its surrounding vegetation.

As seen in the original raw image in Figure 15 a), the NIR raw image will produce a false colour image because of the absorption of the red spectrum and reflection of the NIR spectrum (Butcher, Mottar, Parkinson, & Wollack, 2022). However, this pink hue is not concerning for this work as the pink hue can be eliminated as seen in the contrast enhanced pre-processed image in Figure 15 b).

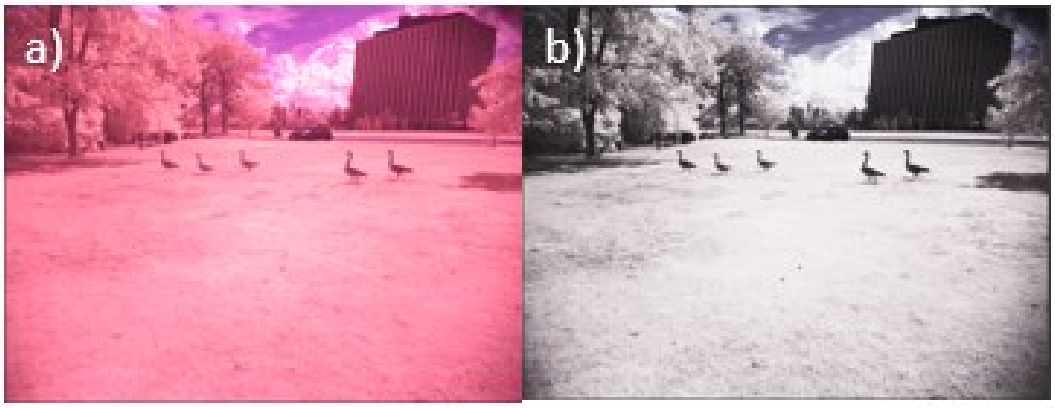


Figure 15 – The a) original NIR false colour raw image taken from the NIR AgroCam and b) the pre-processed raw image.

To achieve the state of the pre-processed image, the following parameters were changed in the *Rawtherapee* application:

	Parameter	Default Value	Pre-processed Value
Exposure: L*a* b Adjustments	Lightness	0	-13
	Contrast	0	20
	Chromaticity	0	-74
Colour: Channel Mixer	Red Channel*	0	57.8

* *Only the red parameter was changed*

Table 1 – *Rawtherapee* edited parameters used to pre-process the NIR raw test images.

In addition to the edited parameters above, the tone curve was also edited. The tone curve characterizes all tones in the image and is specifically used to vary the image brightness or darkness. The x-axis is the tone axis which goes from shadow (black/dark), mid-tones (grey) and

finally highlights (white). The y-axis represents the brightness of a specific x-point tone. In the case of this research, the brightness of the pixels in the mid-tone range were marginally decreased as seen in Figure 16 b) compared to the original curve as shown in Figure 16 a).

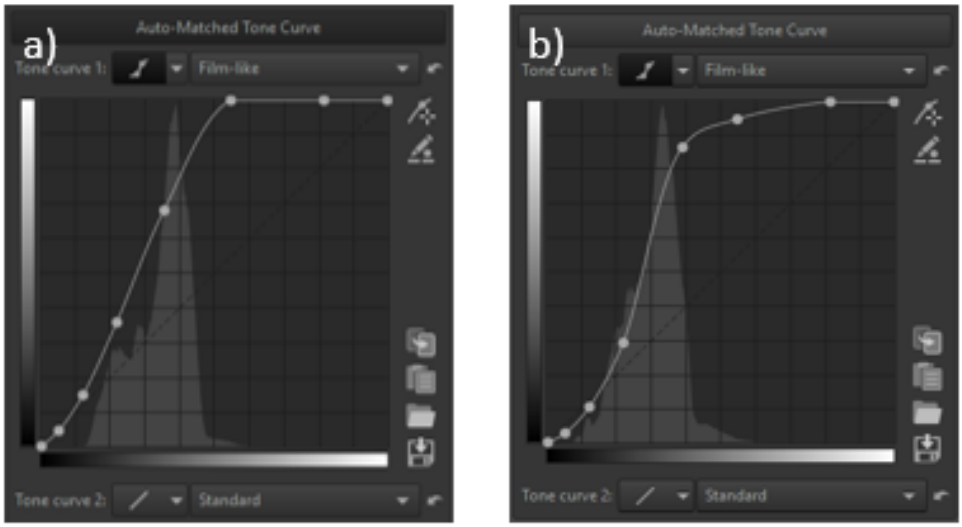


Figure 16 – The a) original and b) edited tone curves in *Rawtherapee*.

3.3.2. Image Annotation

All datasets used in this research were annotated using an online annotation tool *makesnese.ai*. This tool allows users to directly upload images into the application and manually apply multiple bounding boxes on an image with an associated bounding box tag. Bounding box coordinates are created for each object that is annotated within the image. For this research, the tag "goose" was used to annotate the images and the annotations were saved in the YOLO format as .txt files.

Bounding box annotations are a critical aspect of neural networks since it is with these bounding boxes that the CNN learns specific features of an object inside the box, hence, is it crucial that the objects are properly bounded. The bounding boxes must be tight as possible around the object such that there should not be any additional image background pixels between

the edges of the object and bounding box. This is because non-tight bounding boxes could degrade the detector training in properly recognizing pixels in an object class (Redekop & Chernyavskiy, 2021; Wang & Xia, 2021). In addition, all instances of the objects in the images must be annotated as a single box to inform the detector what objects in the image are *not* part of the object class (Su, Deng, & Fei-Fei, 2012). The only instances where geese were not annotated for this case of this study was if the geese were in a position such that it was hard for the human eyes to discern. This was prevalent in images where clusters of geese were taken or if geese are in the image background and the resolution is small (Figure 17 c)) and (Figure 17 d)). If the goose was obstructed by multiple geese or vegetation such that most of the body and/or neck was covered, the bounding box was ignored (Figure 17 a)). However, images where the body of the geese was turned such that the back was facing the camera, or the neck was down because the geese was grazing, or grooming were annotated (Figure 17b)). This was done so that the detector could learn and generalize varying positions of the goose.

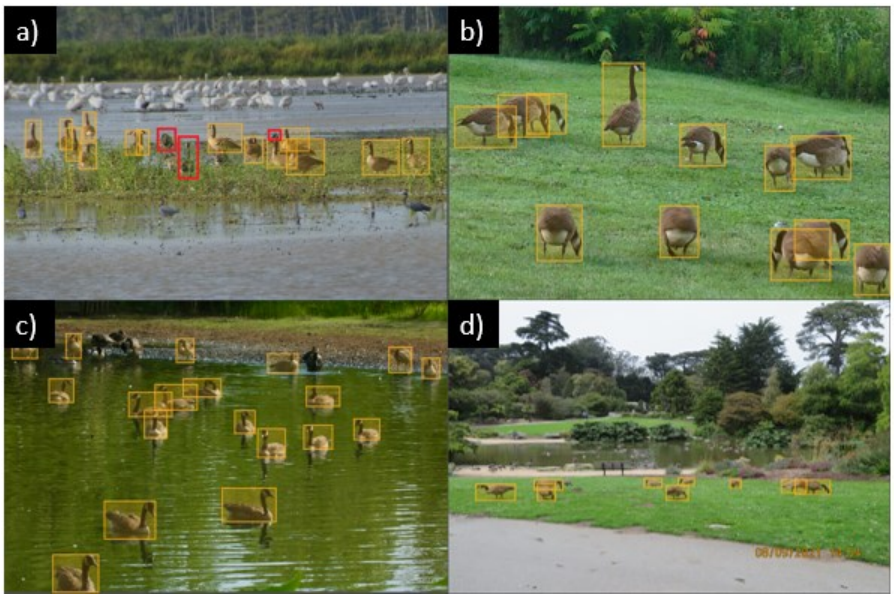


Figure 17 - Examples of the dataset annotation that was completed in *markesense.ai*. a) Geese obstructed by vegetation (highlighted in the red box) were ignored, b) varying positions of geese were annotated, and c) and d) shows examples where the annotation of geese was ignored due to object resolution.

3.3.3. MATLAB

Since this research is only working with one object class, the colour information was omitted so that the detector can focus on spatial information rather than colour. Therefore, the RGB images were converted to single-channel greyscale images. The *im2gray* function in MATLAB was used to convert the training dataset and test images to greyscale. This function converts the true colour RGB image into a greyscale intensity image by maintaining the image luminance and omitting the saturation and hue (Figure 18). The input argument of the *im2gray* function is a m-by-n-by-3 numerical array and the output is returned as a m-by-n numeric array. To complete the conversion, the *im2gray* function first creates a weighted sum of the R, B and G components and are then mapped to a luminance value based on each weighted sum respectively when converted to greyscale. The converted greyscale images were saved as 8-bit depth JPEG images. The MATLAB script written to convert the RGB images into greyscale can be found in Appendix B.

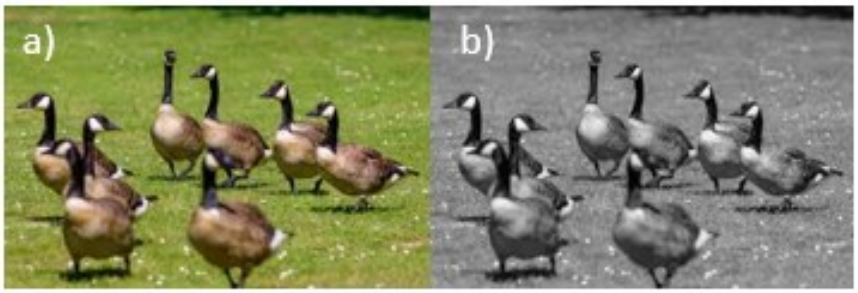


Figure 18 – The a) RGB image and the b) MATLAB converted greyscale image using the *im2gray* function.

3.3.3.1. Image Augmentation

Due to the time-consuming manual work of annotating images, images were augmented in a manner where the bounding box of the original image remained unaffected. Techniques such as image rotation and image resizing were avoided, and contrast adjustments, and Gaussian smoothing were selected as the augmentation method for our training dataset. The contrast

adjustments were done using the *imadjust* function in the MATLAB Image Processing Toolbox which was used to darken and lighten the original greyscale images. The *imadjust* function is an image intensity adjuster that maps the greyscale intensity values to new intensity values. The input argument of the *imadjust* function is a *m*-by-*n* numeric matrix and the output is the adjusted *m*-by-*n* numeric matrix. The low and high intensity values were adjusted to either lighten or darken the original greyscale image where the input parameters of the function are the following

```
imadjust(I,[low_in high_in])
```

where I is the input image, and low_in and high_in are the contrasts limits of between 0 and 1 of the input image. The low_in and high_in values were set to 0.0 and 0.6 for low contrast augmentation and 0.33 and 1 for high contrast augmented images. The *imgaussfilt* function in MATLAB was used to incorporate the Gaussian smoothing on the images which essentially blurred and degraded the quality of the image. This function uses a defined positive sigma value which is the standard deviation of the Gaussian distribution where the input parameters are

```
imgaussfilt(A,sigma)
```

where A is the input image, and the sigma is the standard deviation of the Gaussian distribution. For this study, the sigma was set to 6.

The values chosen for image contrast and Gaussian smoothing was selected based on whether it produced significant changes from the original images but remained discernable to the human eyes. An example of the contrast adjustments and Gaussian smoothing with respect to the baseline greyscale image can be seen in Figure 19. The MATLAB scripts written to augment the baseline greyscale images can be found in Appendix B.

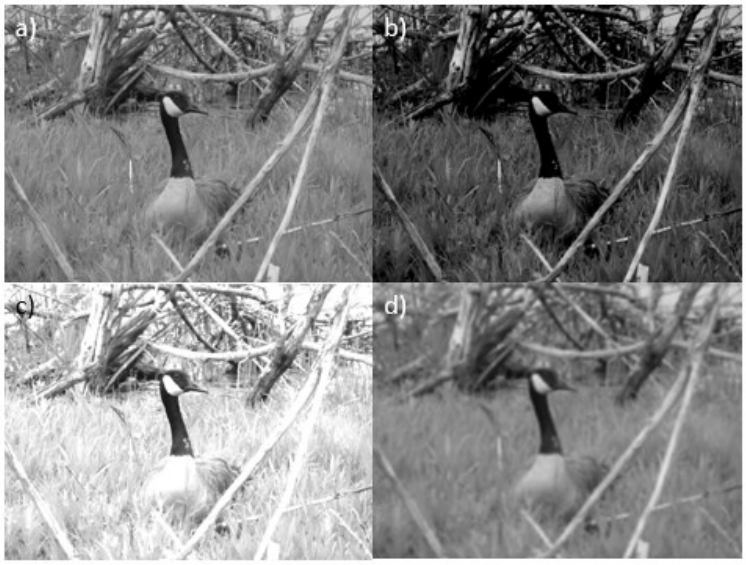


Figure 19 - Comparison of the a) greyscale, b) dark contrast, c) light contrast and d) Gaussian noise MATLAB pre-processed images.

3.3.4. *Blender*

Blender is an open-source software used specifically for rendering 3D graphics and images. In the case for our study, *Blender* was used to generate artificial images of aerial shots of Canada geese to simulate images captured by a RPAS to train the detector on an aerial perspective because of the lack of direct overhead images in the datasets containing real images. The *Blender* dataset used in this study was designed and provided by a Carleton undergraduate research assistant student Skyler Bruggink. There were two image datasets generated for this study the first containing 2,000 images with a simulated camera height of between 10.30 m and 23.46 m and the second dataset containing 1,000 images which simulated camera heights between 5 m and 30 m. While the first dataset contained direct overhead aerial shots, there were more oblique angle shots as if the RPAS is taking off from the ground. The second dataset was generated to fill in for the lack of direct overhead aerial shots of the first dataset.

Blender permits the camera to be placed anywhere in the image scene which allowed flexibility in the image perspective such that images could be generated at a ground level, aerial

or an in-between viewpoint. A python script was developed to generate the images and randomize the goose position and angle, as well as trees and rocks if present in the image. The randomization algorithm was also developed by Mr. Bruggink which divided the image scene into a grid and places a goose in an individual grid, therefore, no two geese are placed together in the same grid thus, there are no geese overlaps in these images. The script was also responsible for producing the simulated aerial heights in each generated image. Sample images of the *Blender* dataset can be seen in Figure 31 of Section 4.3.

The *Blender* dataset was generated using Corsair DDR4 3600MHz memory, Ryzen 5 5600X 6-Core 3.7 GHz CPU and Nvidia GeForce RTX 3060 GPU and was manually annotated using the *makesense.ai* online annotation tool.

3.4 Test Images

The test images of Canada geese were initially gathered at ground level at various public parks within the City of Ottawa. The ground test images consisted of RGB images that were converted to greyscale, NIR JPEG false coloured images converted to greyscale (enabling the raw setting on the AgroCam will produce a JPEG image in addition to the raw image), and the NIR raw image that was pre-processed in *Rawtherapee*. There was a total of four ground test images used in this study, however, only one out of the four images are shown in the results section of each study. The results of the three remaining images can be viewed in Appendix A.

The capture of real aerial images of large gatherings (>1000 individual animals) of Canada geese using the Mavic Pro 2 was attempted at Alfred Lagoon in Plantagenet, ON. Though it was reported previously that a flying altitude of 67 m above Canada geese did not cause any agitation on the birds (Bech-Hansen, et al., 2020), a flying altitude of 75 m was unsuccessful as the birds got agitated when the RPAS approached them. Subsequent flight

heights of 85 m and 100 m were attempted but it still made the large groups of geese swim in an opposite direction in the lagoon, away from the RPAS. Thus, it was decided to use a set of geese decoys to capture aerial images for testing purposes. The decoy set contained four individual geese which were placed strategically at the River Field at Carleton University which included placing two geese closely together and the other two far apart from each other. The AgroCam was configured to automatically take NIR images every five seconds however, this interval feature was not available on the onboard camera of the Mavic Pro 2 so the 4K video feature (30 fps) was used to capture a RGB video during the entirety of the flight. JPEG stills were then collected from the videos to compare with the test images captured with the NIR camera. Only the greyscale and NIR pre-processed images are used for testing when the detector was tested on aerial images.

Flight passes over the geese decoys were completed at altitudes of 5 m, 10 m and 25 m and the flight and image logs were used to determine the altitude of each image taken. The computer program *Snippet* was used to capture 8-bit depth JPEG images from the video. The NIR raw images captured from the AgroCam camera were pre-processed in *Rawthereapee* and 24-bit depth[1] images were obtained for testing.

Since the AgroCam camera was placed at an angle which respect to the Mavic Pro 2, the true altitude of the camera at which the images were taken during flight needed to be determined. Measuring from the table as a reference point as seen in Figure 20 to calculate the placement of the AgroCam with respect to the Mavic Pro 2, the angle was calculated as 16.5°. Thus, the true

[1] Although 8-bit conversion was selected in *Rawtherepee* to convert the raw image to a JPEG image, image properties of the converted images still reported a 24-bit depth image.

flight height of the images used as test images are 5.21 m at 5m, 10.43 m at 10 m and 26.07 m at 25 m.

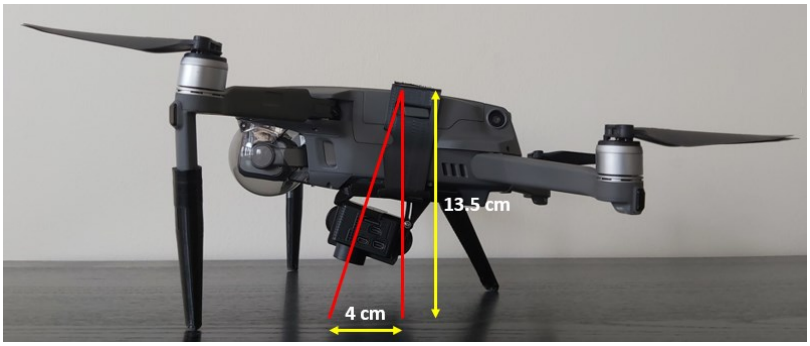


Figure 20 - Measurement of AgroCam placement on the Mavic Pro 2.

The supported gimbal tilt angle range on the Mavic Pro 2 is -90° to +30° with respect to the aircraft frame. The Mavic Pro 2 camera gimbal tilt angle was set to -71.2° when it was used to capture the aerial test images. Therefore, the true height of the Mavic Pro 2 captured images at 5.28 m, 10.56 m and 26.4 m at 5 m, 10 m, and 25 m respectively.

3.4.1. Image Spatial Resolution

The altitude at which the images were taken plays an important factor of how much details are contained in the image due to spatial resolution. Spatial resolution is the measure of the number of pixels per constructed image and is described by the ground sampling distance (GSD) which determines the size of a single pixel in the image. GSD is defined by

$$GSD = \frac{(flight\ height) * (sensor\ width)}{(focal\ length) * (image\ width\ in\ number\ of\ pixels)} \quad (11)$$

where flight height is the RPAS flying attitude, sensor width is the dimensions of the camera's sensor, focal length is the true camera focal length, and image width is the dimension of the image or video frame measured in pixels. Figure 21 shows illustration of the GSD with respect to RPAS.

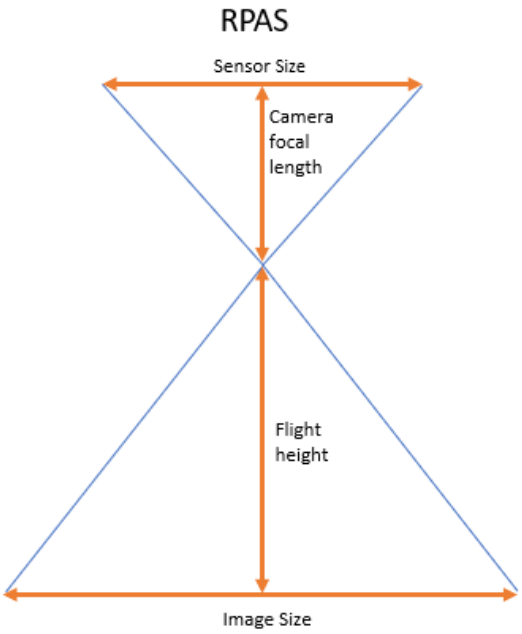


Figure 21 - Illustration of the ground sampling distance which is dependent of the camera sensor size, true focal length of the camera, RPAS flight height and image width measured in pixels.

To determine the GSD of an image, the sensor, image, and video dimensions of the camera data were used from the AgroCam and Mavic Pro 2 respectively. The GSD values of the AgroCam and the Mavic Pro 2 camera's true flight heights are depicted in Table 2.

Camera Type	True Flight Height	GSD (cm/pix)
AgroCam	5.21 m	0.187
	10.43 m	0.375
	26.07 m	0.936
Mavic Pro 2	5.28 m	0.170
	10.56 m	0.339
	26.4 m	0.850

Table 2 - GSD values of test images captured by the AgroCam and the Mavic Pro 2 cameras.

As it can be seen from the calculations, the image pixel size becomes larger with increasing flight altitude which degrades the spatial resolution of the image in comparison to the same image taken at a smaller flight altitude due to the increasing pixel size.

Chapter 4: You Only Look Once Training of Image Datasets

This chapter explores the training and testing of the YOLOv4-Tiny detector using three dataset studies which are separated into their own sections within this chapter:

- Small dataset consisting of 423 images of real Canada geese which is then augmented to increase the dataset to 1,269 images.
- Large dataset consisting of 2,000 images of real Canada geese. 500 images were selected and augmented to increase the dataset to 2,000 images.
- A dataset containing 2,000 synthetic computer-generated images of simulated aerial of Canada geese. 500 images were selected to augment to grow dataset into 2,000 images.

For each study, the dataset preparation, the training, results, and discussion are presented.

4.1 Small Dataset Study

The dataset used in the initial study was sourced using the Fatkun Batch Download Image Google Chrome tool and is openly available in the Google Chrome Web store. This tool allows user to bulk download images from their Google Chrome tabs. Since there are a limited number of publicly available images of Canada geese, 423 RGB ground level images of Canada geese in JPEG format were sourced for the initial study and consisted of various image scenarios such as images of only the geese head, large flocks captured from a distance, full bodies of geese and flocks flying at a distance. The image file size of this dataset ranged between 2 KB and 20 KB.

4.1.1 Dataset Preparation

This dataset was first converted to greyscale via MATLAB and followed the original 90/10 split used previously in the initial study of the YOLO detector where 384 and 39 images

were allocated into the training and validation datasets respectively. After testing the greyscale image trained YOLOv4-Tiny detector, the dataset was then augmented to artificially grow the overall dataset to 1,269 images since the original dataset did not meet the minimum required number of images for training as per the YOLO recommendation of 2,000 images per dataset (Bochkovskiy A. , n.d.). Only darkening and lighting contrast augmentation was completed for this study. The 1,296 images were divided into 1,015 training images and 254 test images following an 80/20 dataset split. The 80/20 ratio is known as Pareto's Principle, named after an Italian economist Vilfredo Pareto, and is used as a dataset split to train neural networks. There has been several research indicating that after attempting a variety of ratio splits, the 80/20 split was optimal for datasets consisting of approximately 2,000 images (Baranwal, Khandelwal, & Arora, 2019; Prashanth, Mehta, & Sharma, 2020). The image ordering of the greyscale, dark contrast and light contrast image sets in the augmented dataset are 1-423, 424-846 and 847-1269 respectively where dataset split between the training and validation images are shown in Table 3. The split was done in a specific manner such that the training and validation sets contains the augmented images, in addition to the greyscale images.

Validation	Training	Validation	Training	Validation	Training
1-86	87-509	510-593	594-1017	1018-1101	1102-1269

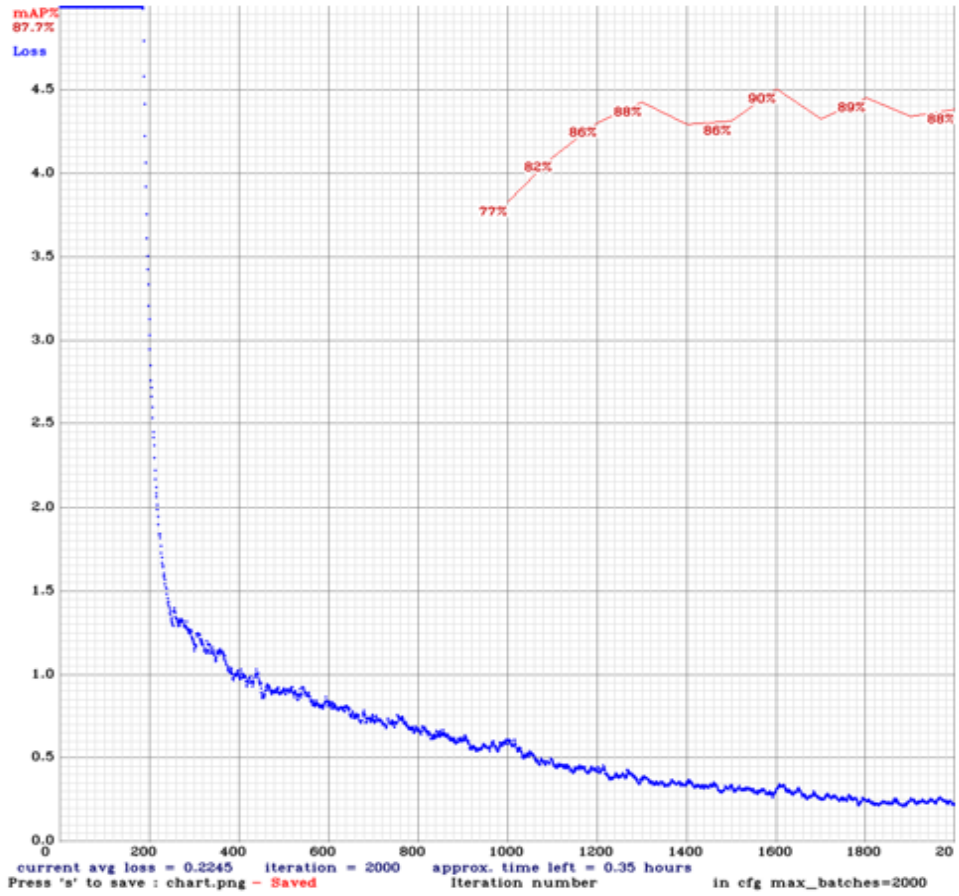
Table 3 – The greyscale, dark contrast and light contrast split between the training and validation datasets of the Small Dataset Study.

4.1.2 You Only Looking Once Training of Dataset

The 423 greyscale image dataset was first trained on the recommended maximum batches value of 2,000 using a maximum batches and subdivision values of 64 and eight respectively, and width and height of 416. While the best weights file yielded a mAP of 90%, one can clearly

observe that the training loss function does not converge in Figure 22 a), where the loss function is depicted as the blue graph, which indicates there are still significant gradient calculations (weight updates) to complete until the detector is optimized. When the maximum batches parameter is set to 20,000, the loss function converges, and the best weights mAP is 93.72% but the convergence comes with caution as the loss has potentially converged too much which may have caused overfitting as seen in Figure 22 b). Also shown in the graphs is the mAP graph which is represented by the red graph.

a)



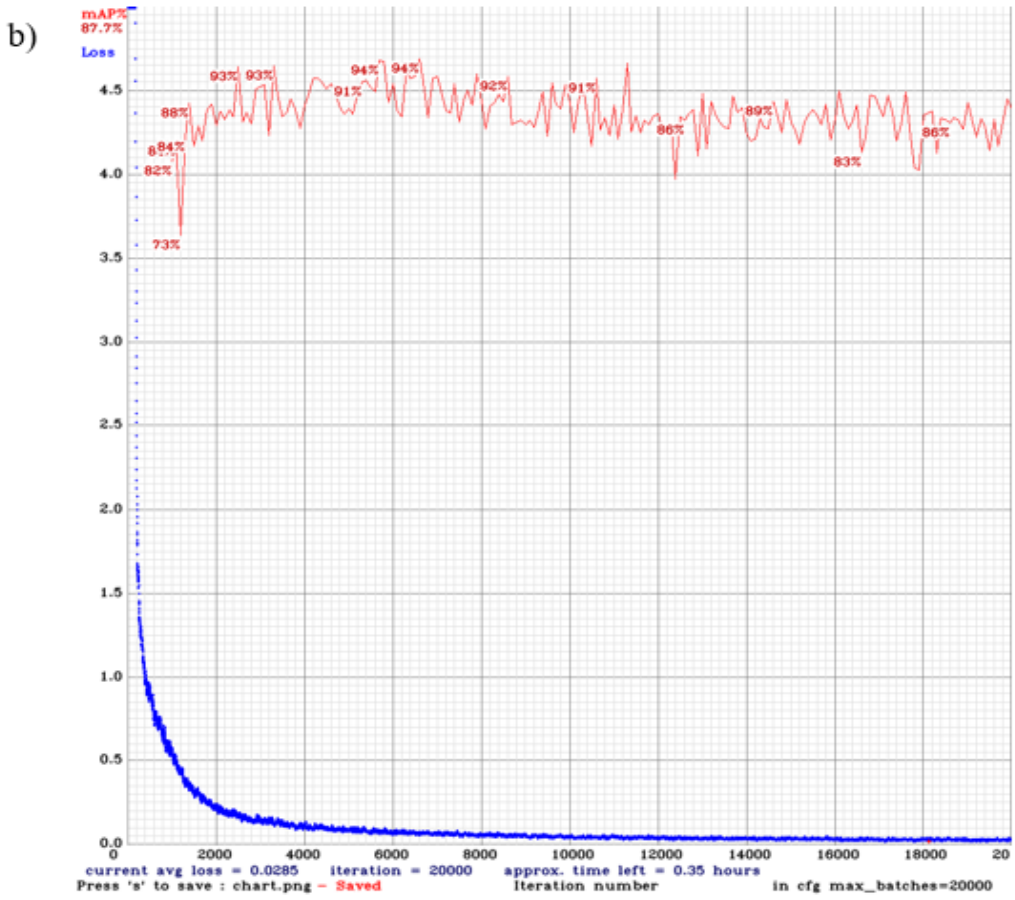


Figure 22 – The YOLOv4-Tiny mAP and loss graphs over a) 2,000 and b) 20,000 iterations. The blue and red graph depicts the loss and mAP repsectively.

Though a best weights file is produced at the end of training, it was found that there was a better or equally performing weights file in addition to the best weights file since each 1,000th weight performance was compared with the best weights file. For the greyscale and augmented training, the other optimal weights occurred at iteration 13,000. The mAP and IOU of the best weights and the weights at 13,000 of the greyscale and the augmented dataset when maximum batches is set to 20,000 is reported in Table 4.

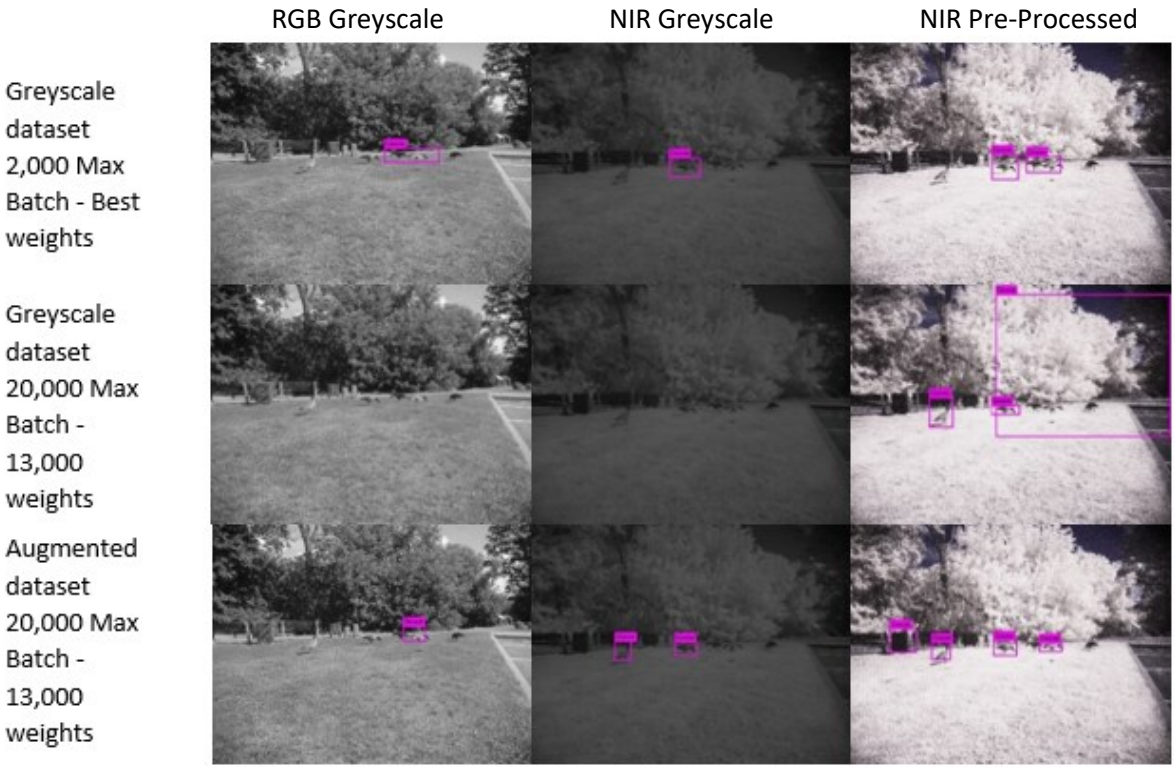
Dataset Type	Dataset Size	Weight File	Weight mAP	IOU
Greyscale	423	Best	93.7%	74%
		13,000	89.6%	73.8%
Augmented	1,269	Best	98.6%	87.7%
		13,000	98.6%	89.9%

Table 4 - YOLOv4-Tiny mAP and IOU performance metrics of the greyscale and augmented datasets of the best weights and the weights at iteration 13,000 when the maximum batches is set to 20,000.

4.1.3 Results

The initial training at maximum batches of 2,000 yielded optimal performance using the best weights file in comparison to the weights file produced at each 1,000th iteration, albeit detecting the two geese as one goose on the NIR post-processed image compared to the RGB and NIR greyscale image in Figure 23. The training with maximum batches at 20,000 improved the detection of geese against vegetation in the NIR pre-processed image. As it can be seen in Figure 23, the trained detector was unable to detect geese in the RGB and NIR greyscale images but was able to properly detect two geese in the NIR pre-processed image, albeit one missed detection. This result was obtained using the weights generated at iteration 13,000 and any weights above or below 13,000 deteriorated detector performance. The augmented dataset of 1,269 was also trained with a maximum batch of 20,000. This yielded marginal detection improvements in the NIR pre-processed test image as three geese were detected using the weights obtained at iteration 13,000. It is also noted that one goose in the NIR greyscale test image was detected which was not detected using the weights obtained from the detector that was trained on the greyscale dataset. Weights above or below 13,000 also deteriorated detector performance in this case. Note that the confidence scores in the table below the detection results are read left to right where each confidence score value correlates to each predicted bounding box in each test image. Using the

NIR Pre-Processed Greyscale dataset 2,000 Max Batch – Best weights results as an example, 93% and 26% correlates to the first and second predicted bounding box respectively.



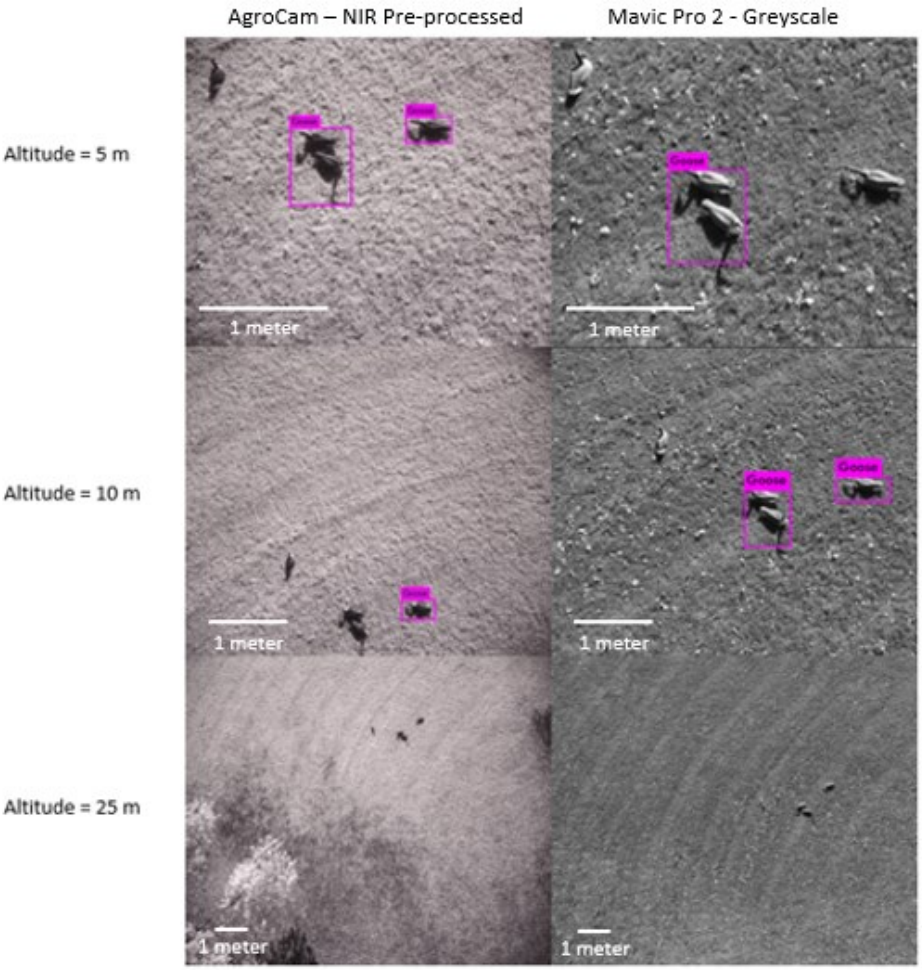
Detection confidence scores (left to right)

37%	84%	93%, 26%
0%	0%	51%, 88%, 45%
39%	43%, 74%	33%, 32%, 100%, 43%

Figure 23 – Ground image test results of the greyscale and augmented trained detector of the Small Dataset Study.

Aerial images of the geese decoys taken by the Mavic Pro 2 and the NIR AgroCam camera were then tested using the weights obtained at 13,000 for both the greyscale and augmented trained datasets. There were no significant improvements in using the NIR pre-processed images versus greyscale images when tested against the greyscale trained detector as seen in Figure 24. Both the greyscale and the NIR pre-processed image could not detect any geese in the image taken at 25 m and produced approximately the same number of detections on

the 5 m and 10 m images. The detector detected the cluster of two geese as one goose in both the NIR pre-processed and greyscale image.



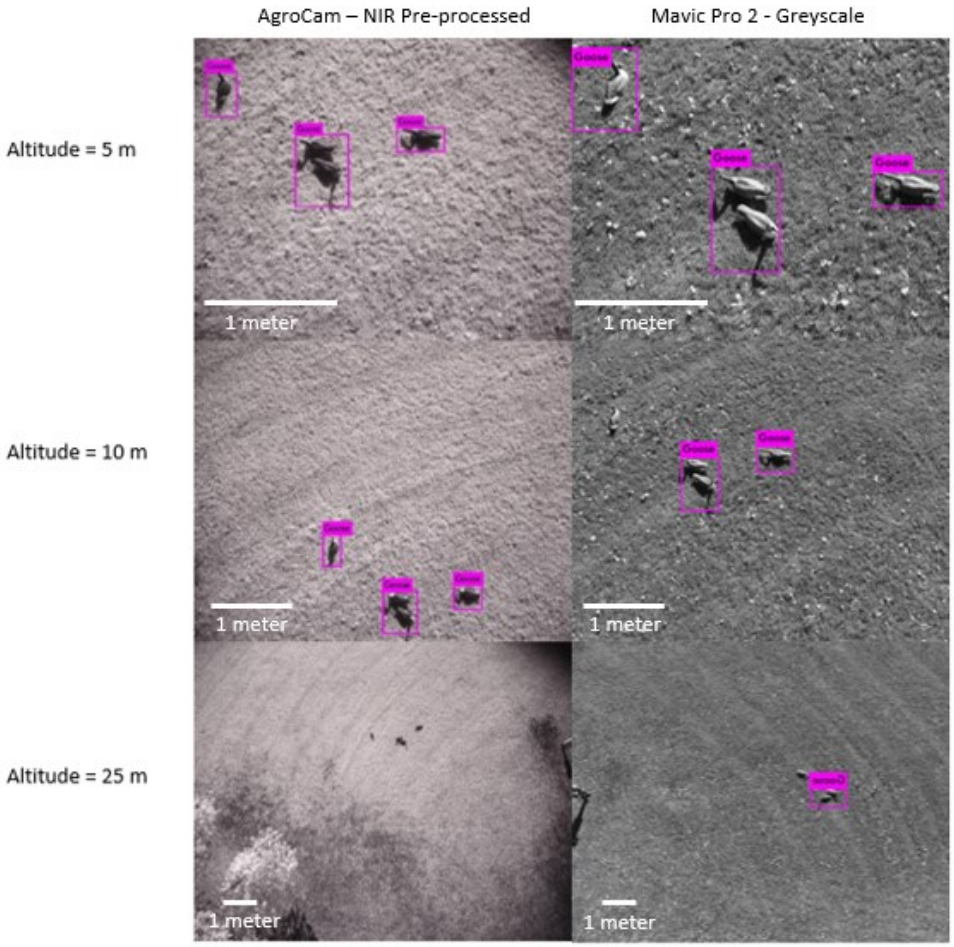
Detection confidence scores (left to right)

46%, 56%	41%
25%	92%, 84%
0%	0%

Figure 24 – Aerial results using the 13,000th weights of the greyscale trained detector of the Small Dataset Study.

However, when testing the images against the YOLOv4-Tiny detector that was trained on the augmented dataset, the NIR pre-processed image was able to detect all geese in the 5 m and 10 m images, although the detector detected two geese as one which is common between the NIR pre-processed and greyscale image as shown in Figure 25. Interestingly, the detector was

able to detect geese in the 25 m greyscale image. Overall, using the detector that was trained on the augmented dataset shows improvement in detections compared to the detections using the YOLOv4-Tiny detector that was trained on the greyscale dataset.



Detection confidence scores (left to right)

78%, 86%, 98%	27%, 78%, 98%
41%, 85%, 97%	98%, 97%
0%	53%

Figure 25 – Aerial results using 13,000th weights of the augmented trained detector of the Small Dataset Study.

4.1.4 Discussion

The NIR reflectivity of the vegetation provided an improvement in contrast between the geese and surrounding vegetation for the human eyes, but the detector still had trouble discerning

geese in some of the aerial test images. This may be attributed to the small dataset used and the type of images contained in the dataset since all images were sourced online which resulted in a mix of image scenarios and geese positions. Surprisingly, although the dataset was trained on non-aerial images, the detector was still able to detect geese in aerial images. It is possible that the detector learned very specific features such as colour shadings, shape, and plumage of the geese during training which is why the detector can detect the geese from an overhead perspective.

There are marginal improvements when using pre-processed NIR images to test the detector especially when the detector has been trained on the augmented dataset. All geese were detected in the 5 m and 10 m in the NIR pre-processed aerial test images of the augmented trained detector. In addition to the augmented trained detector, the bounding boxes around the NIR pre-processed aerial test images of the augmented trained detector are much tighter than the bounding boxes of the greyscale aerial test images. It is also noted that the confidence score percentages of the NIR pre-processed test image at 5 m for the augmented trained detector are higher than the confidence percentages of the greyscale results.

It is surprising that the detector was able to detect a goose in the 25 m greyscale aerial test image of the augmented trained detector, however, this could be a misdetection. Notice that the bounding box around the object at 25 m is not tight which indicates the training needs to be refined. The training dataset also did not contain any images that included very small resolutions of Canada geese which could provide feature learning on small resolutions of the bird.

As per YOLO guidelines outlined in GitHub, the maximum batches parameter in the configuration file should be set to number of (classes)×2000. In the case of this study, only one object class is delt with therefore by definition, the maximum batches should be set to 2,000,

however 2,000 is recommended for a dataset containing 2,000 images. Since the datasets used in this study is less than 2,000 images, a larger maximum batches value was used to allow the detector to adequately learn object features from enough training iterations. As clearly seen from the training graphs, training over 2,000 iterations did not cause the loss to converge which is highlighted in the detection results as the best weights of the detector that was trained over 2,000 iterations underperformed in comparison to the 13,000 weights files obtained from the detector that was trained for 20,000 iterations.

4.2 Large Dataset Study

A larger dataset was provided from the Cornell Macaulay Library which contains 4,000 images of Canada geese. The images in the dataset contained a variety of ground level and in-flight actions images of the birds, however, all 4,000 images could not be annotated simply because of subpar image resolution such that it was difficult for the human eye to discern the bird in some of the images. 2,000 images within the 4,000 images were manually analyzed and selected to be included in the dataset. Images that were selected contained strictly sedentary shots of geese to match the shape of the geese decoys used in the aerial test images which involves omitting images containing explicit head shots or flying positions. The image file size in this dataset ranged from 9 KB to 2.3 MB.

4.2.1 Dataset Preparation

The initial split of the 2,000 image Cornell dataset also followed the 80/20 split, and the dataset was arbitrarily divided into 1,600 training and 400 validation images where it was then converted to greyscale using MATLAB. The online annotation tool *makesense.ai* was used to annotate the images. The first 500 images of the original 2,000 dataset were selected for augmentation where image contrast was edited (light and dark), and Gaussian smoothing was implemented. The training and validation datasets were strategically selected such that there is

the light and dark contrast adjustments and Gaussian smoothing exposure in both the training and validation datasets. The training and validation datasets are split accordingly as per Table 5 where the greyscale, dark contrast, light contrast, and Gaussian smoothing image sets in the dataset are 1-500, 501-1,000, 1,001-1,500 and 1,501-2,000 respectively.

Training	Validation	Training	Validation	Training	Validation	Training	Validation
1-400	401-500	501-900	901-1,000	1,001-1,400	1,401-1,500	1,501-1,900	1,901-2,000

Table 5 – The augmentation split between the training and validation datasets of the Large Dataset Study.

4.2.2 You Only Looking Once Training of Dataset

To be consistent with the initial 423 image dataset training parameters, the same training parameter configuration was attempted on the 2,000 image dataset from Cornell to verify if there is an improvement with loss convergence now that a larger dataset is used. However, there were problems with maintaining the same parameters on the Cornell dataset. Due to the image file size of the Cornell dataset which averaged 3.35 MB (range from 9 KB to 12 MB) compared to the dataset obtained by the Fatkun Batch Download Image Google Chrome tool which averaged 4 KB (range from 1 KB to 20 KB), this presented memory issues on the Jetson Nano when training the Cornell dataset as training would suddenly stop with a memory error in the Linux bash terminal. The Cornell dataset was then characterized on the memory limitations of the Jetson Nano where different network resolutions, batch and subdivision values were attempted. The training parameters are shown in Table 6 with either pass or fail, where fail represents a failure in training completion due to memory issues. Each test was trained for 2,000 iterations and the mAP and IOU values were generated from the best weights file.

	Batch	Subdivision	Network Resolution	Pass or Fail	mAP	IOU
Test #1	32	32	416	F	-	-
Test #2	32	8	384	F	-	-
Test #3	64	16	384	F	-	-
Test #4	64	32	384	P	80.70%	63.23%

Table 6 – Training characterization of the greyscale Cornell dataset due to memory limitations on the Jetson Nano.

Although the large dataset contained the recommended number of images in the complete training dataset, it was still found that the loss has not fully converged as seen in the training graph in Figure 26.

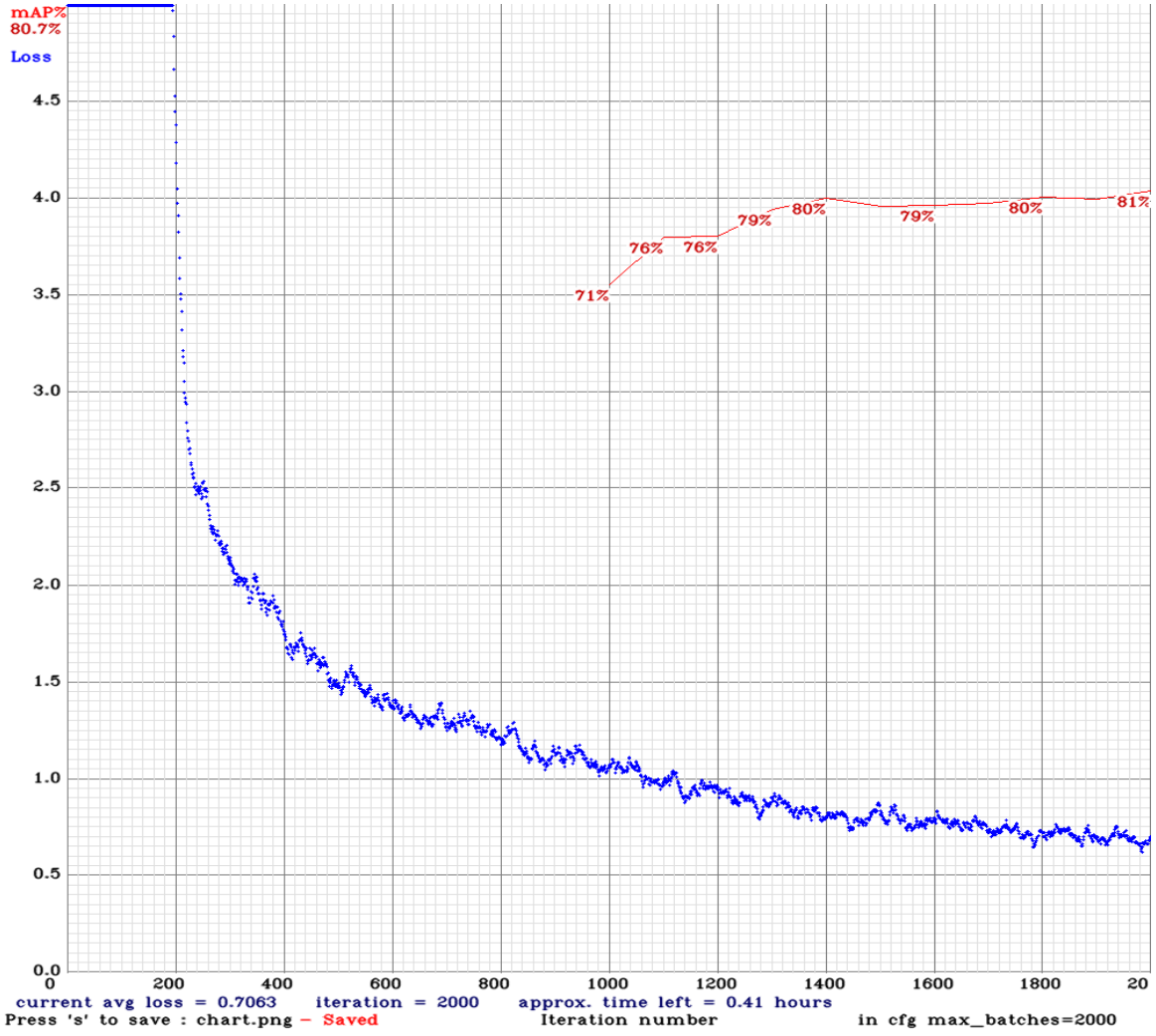


Figure 26 – The YOLOv4-Tiny training graph of Test #4 for 2,000 iterations. The red and blue graphs depict the mAP and loss respectively.

Test #4 was also attempted with 10,000 iterations but memory issues were encountered once again, and training was only able to produce up to the 7,000th weights file. The weights at iteration 4,000 had the best detection performance out of all the weight files analyzed for this training. The mAP and IOU for the 4,000th weights file is 80.86% and 66.96% respectively.

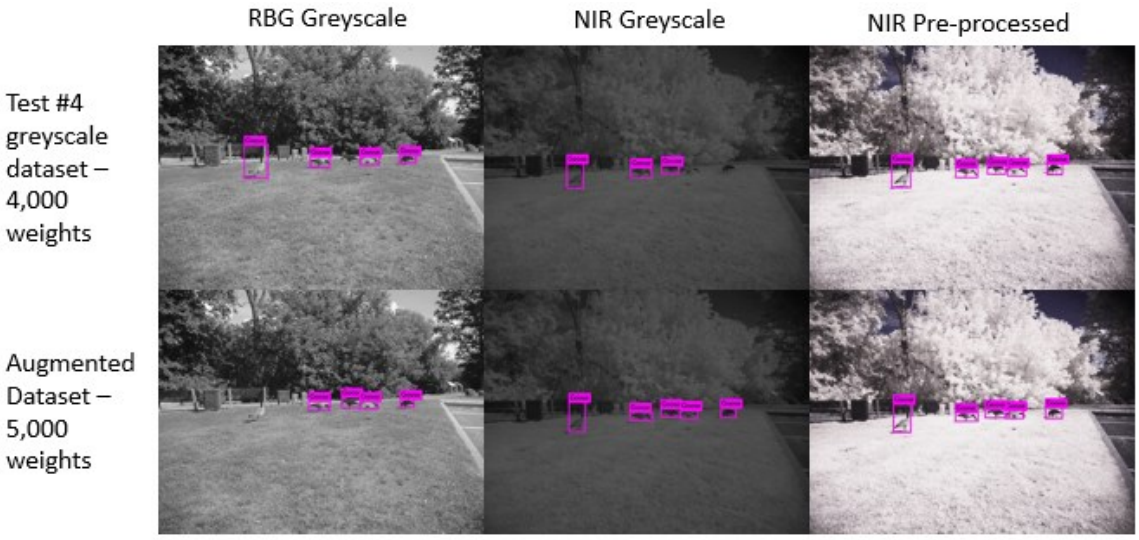
Since 10,000 iterations failed the Test #4 training, it was decided to select the smallest sized (lower resolution) 500 images of the greyscale Cornell dataset for augmentation. The 500 images set was grown to a 2,000 image dataset by creating augmented sub-sets by lightening contrast, darkening contrast, and applying the Gaussian smoothing, to the original greyscale images in addition to maintaining the original 500 greyscale images in the augmented dataset. Since lower resolution images were used, the network resolution was increased to 480x480 and a batch and subdivision of 64 and eight was used, however, this also caused memory problems on the Jetson Nano. The subdivision was then changed to 16 while maintaining the network resolution and batch values and the detector was able to do full pass of training over 10,000 iterations. It was found that the best performing weights file occurred at iteration 5,000 which yielded a mAP and IOU of 74.06% and 65.53% respectively.

The split of the non-augmented and augmented dataset following an 80% and 20% for its training and test datasets respectively. To split the augmented dataset, the first 400 images of the augmented sub-set were allocated to the training dataset and the last 100 images in the sub-set were allocated to the validation dataset.

4.2.3 Results

There are clearly detection improvements when training with a larger training dataset when the detector was tested on ground level images. Compared to the results of the Small Dataset Study, there are more geese detections across the RGB greyscale, NIR greyscale and

NIR pre-processed test images as seen in Figure 27 albeit not all geese were detected in both the greyscale converted and NIR pre-processed test images. It is interesting to note that the detection and confidence percentages improved when the augmented trained detector was used.

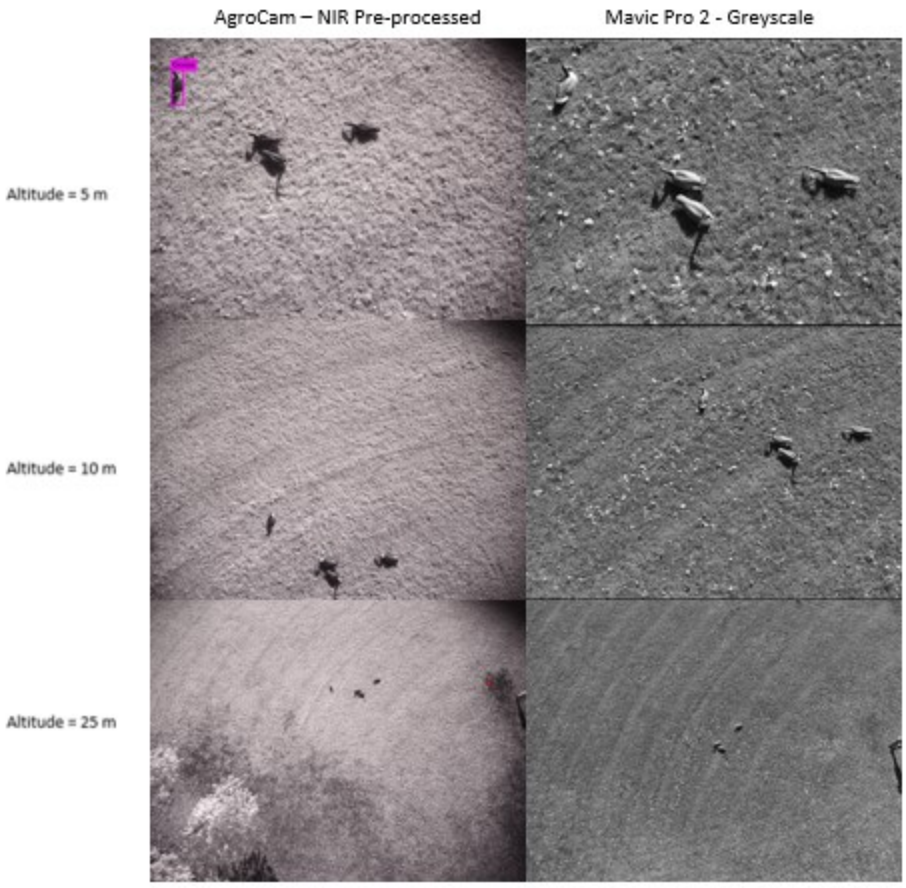


Detection confidence scores (left to right)

70%, 94%, 69%, 31%	77%, 86%, 36%	91%, 92%, 57%, 57%, 32%
88%, 50%, 78%, 36%	84%, 97%, 94%, 96%, 77%	94%, 91%, 93%, 83%, 27%

Figure 27 – Ground image test results of the greyscale and augmented trained detector of the Large Dataset Study.

Though there were improvements on the ground level test images, there was a degradation in detection performance when the detector was tested against the aerial test images in comparison of the aerial results in the Small Dataset Study when the greyscale trained detector was used. In Figure 28, there was only one goose detected in the NIR pre-processed images at 5 m while no other geese were detected at subsequent altitudes even in the greyscale aerial test images.

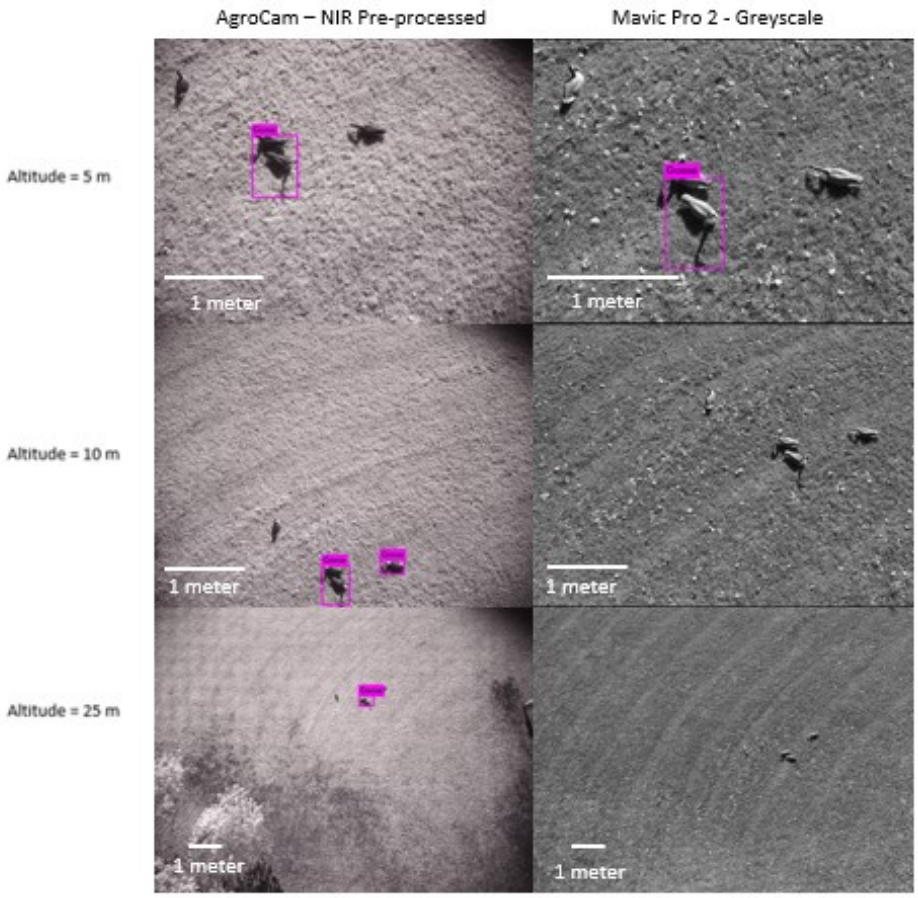


Detection confidence scores (left to right)

26%	0%
0%	0%
0%	0%

Figure 28 – Aerial results using the 4,000 weights of the greyscale trained detector of the Large Dataset Study.

When the augmented trained detector was tested, there were more detections compared to the results of the greyscale trained detector (Figure 29). In addition, the number of detections in the NIR pre-processed aerial test images were greater than the greyscale aerial test images. However, the augmented aerial results of the Large Data Study had less detections than the augmented aerial results of the Small Dataset Study.



Detection confidence scores (left to right)

82%	40%
69%, 94%	0%
59%	0%

Figure 29 – Aerial results using the 5,000 weights of the augmented trained detector of the Large Dataset Study.

4.2.4 Discussion

Training with a larger dataset improved geese detections at ground level images especially when the dataset has been augmented. It is worth noting that in both the large and small dataset studies, the augmented trained detector was able to detect a goose in the 25 m aerial test images. However, the results in this section also showed that using a larger complete dataset did not yield an improvement in the aerial results. This could be because of the increased ground level images in the larger dataset which caused the detector to generalize more on ground level geese.

The augmentation showed improvement in detection and confidence percentages in the NIR greyscale and NIR pre-processed images in the ground test images. This could be attributed to the enhanced contrast the NIR images provided compared to the RGB greyscale converted images. In the ground images test using the 5,000 weights of the augmented trained detector, there was one goose that was not detected in the RGB image but was detected in the NIR greyscale and NIR pre-processed image. Figure 30 shows the same ground level test image of the a) greyscale, b) NIR false colour (default NIR JPEG capture on the AgroCam), c) NIR greyscale and d) NIR pre-processed test images but with no detection bounding box to obstruct details in the images. As it can be seen in the greyscale image, the goose that was not detected in the far left seems to be camouflaged in the grass with a quick glance. However, in the NIR false colour image, it can be clearly seen that there is indeed a goose in the far right as there is an enhancement in contrast. This detail is then transferred to the NIR greyscale image in since it is a direct greyscale conversion. This enhancement in contrast is also seen in the NIR pre-processed image that was pre-processed in *Rawtherapee*.

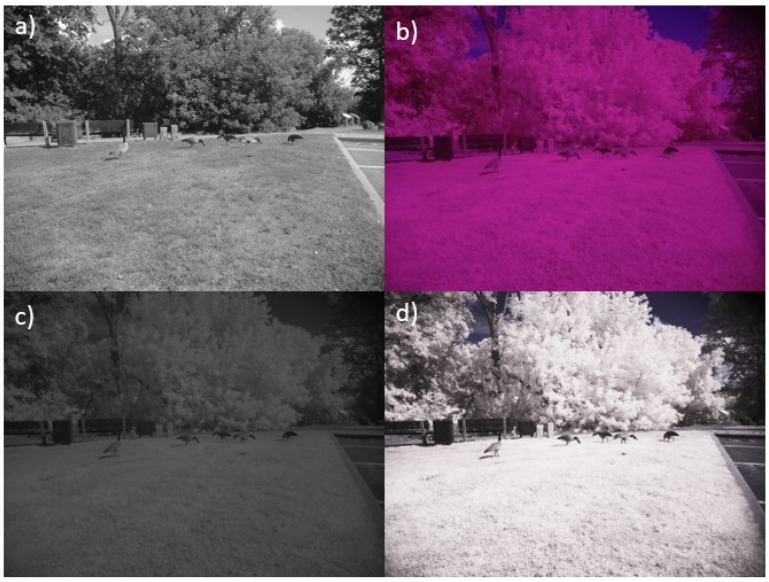


Figure 30 – Comparing image contrasts of the a) RGB greyscale, b) NIR false colour, c) NIR greyscale and d) NIR pre-processed test images.

There were more detections present in the NIR pre-processed aerial test images compared to the greyscale aerial test images when the augmented trained detector was used since there was only one detection in the greyscale aerial test images compared to the four detections in the NIR pre-processed aerial images. The enhanced contrast of the pre-processed NIR image improves the discerning of the specific features learnt by the detector which is potentially why the detector was able to have more detections in the NIR pre-processed images rather than the greyscale images.

4.3 *Blender* Dataset Study

A third dataset that was used in this study was provided was by a Carleton University undergraduate research assistant student, Skyler Bruggink, where his work involved generating simulated aerial images taken from a RPAS using a 3D modelling software *Blender*. This dataset contained 2,000 JPEG images and ranged from 72 to 323 KB in file size. The images contained six image scenarios using pre-defined background drops which were further manipulated to make it more realistic. The backdrop includes scenes of rocky shores, grass fields, dried grass fields and water (Figure 31).

The simulated altitudes of the images ranged from 9.28 to 23.46 m. The dataset consisted of a mixture of images taken at oblique angles (as if the RPAS is taking off) and direct aerial shots (as if the RPAS is flying directly over the geese), however, there seems to be more images taken at an oblique perspective rather than a direct aerial view in this dataset.

Figure 31 – Samples of the artificially generated images via *Blender*. Images a) and c) are samples of a direct overhead shot and b) shows a simulated shot taken at an oblique angle.

4.3.1 Dataset Preparation

The first test of this dataset consisted of converting the 2,000 RGB images into greyscale via MATLAB to train the detector with. However, four images were omitted from the dataset as the images could not be annotated due to the geese being obscured by objects within the image that was hard for the human eye to discern. Therefore, the true size of this dataset is 1,996 images and was annotated using the online annotation tool *makesense.ai*. The training and test split for this dataset was 80% and 20% respectively.

The second test comprised of the augmenting a selection of 500 greyscale images from the 2,000 image dataset. The images were carefully selected such that each image backdrop was accounted for. The augmentation methods of contrast lightening, dark contrast lightning and Gaussian smoothing, in addition to the original greyscale images, created four sub-sets that totaled 2,000 images. The first 100 images and last 400 images of each sub-set were allocated to the validation and training datasets respectively and followed the 80% and 20% split for training and validation allocations.

4.3.2 You Only Looking Once Training of Dataset

Both the greyscale and augmented datasets were trained using a network resolution of 480x480, batch of 64 and subdivision of 16, and was trained over 10,000 iterations. The best

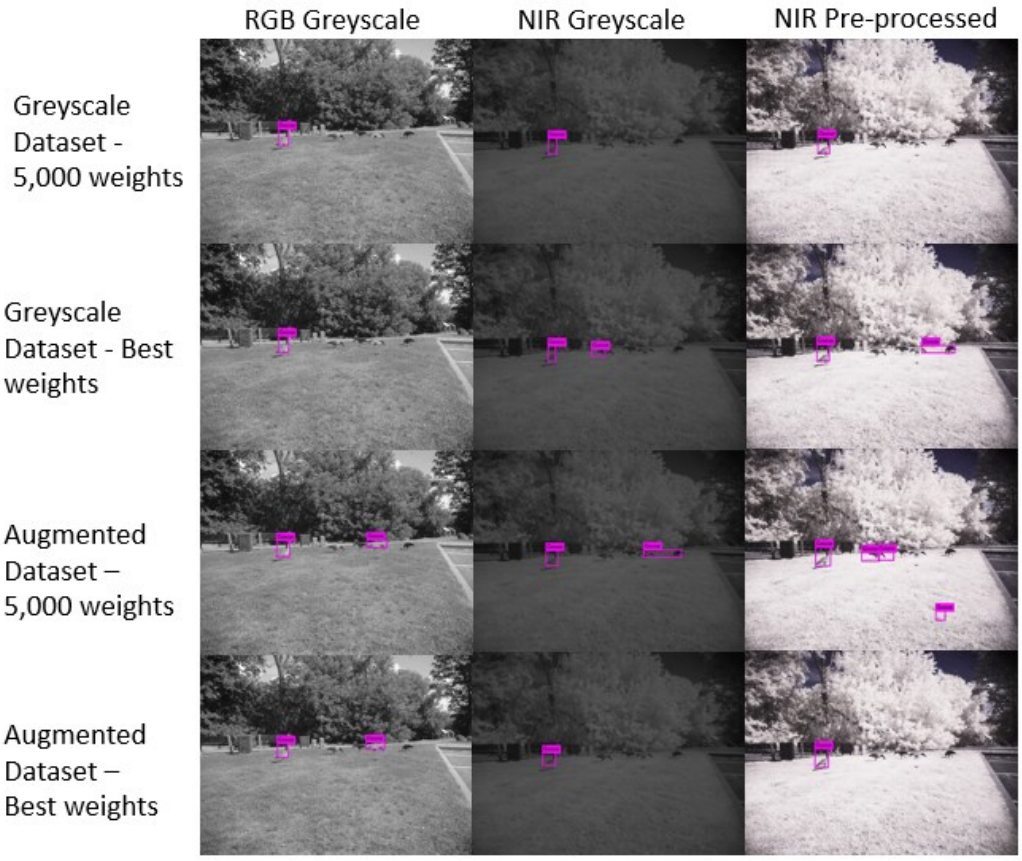
weights from the greyscale dataset training generated a mAP of 95.52% and IOU 74.52%, and the best weights of the augmented dataset training generated a mAP of 97.35% and IOU of 73.65%. Each weight file produced at every 1,000th training iteration was compared with the best weights file and the 5,000 weights file was deduced as the better performing weights file within the set of the 1,000th weight files when tested against the set of test images. The mAP and IOU of the 5,000 weights file for the greyscale dataset and augmented dataset is 93.86% and 78.39%, and 97.14% and 69.18% respectively.

4.3.3 Results

There were no discernable differences between the best weights and ideal weights obtained at iteration 5,000th for both the greyscale and augmented trained detectors. As it can be seen in Figure 32, there were several misdetections across all test images except for the greyscale trained dataset using the 5,000 weights. Not only were they misdetections, but the predicted bounding boxes were completely incorrect in that it either generated on half the goose or it identified a whole goose in addition to a goose head as one single goose. The 5,000th weight file also performed subpar for the augmented trained detector as there are more misdetections in the 5,000 weights file compared to the best weights file.

The 5,000 weights of the greyscale trained detector seemed the most stable in that it had less detection errors compared to the other ground image tests within this study. It is also noted that the most left goose that was consistently detected using the best weights of the greyscale trained detector had better confidence percentages compared to the other tests that detected the same goose. As for the augmented test results, there were more misdetections when the 5,000 weights were tested and little to no detection improvements when the best weights were used. Since there was no conclusive result on the ground test images to determine what weight file

should be used to test the aerial images, the aerial test images were tested against all possible weight files of the greyscale and augmented trained detectors. In the case for the aerial tests, the 5,000th weights of the greyscale trained detector and the 5,000th weights of the augmented trained detector yielded the best results which are used going forward in this study.



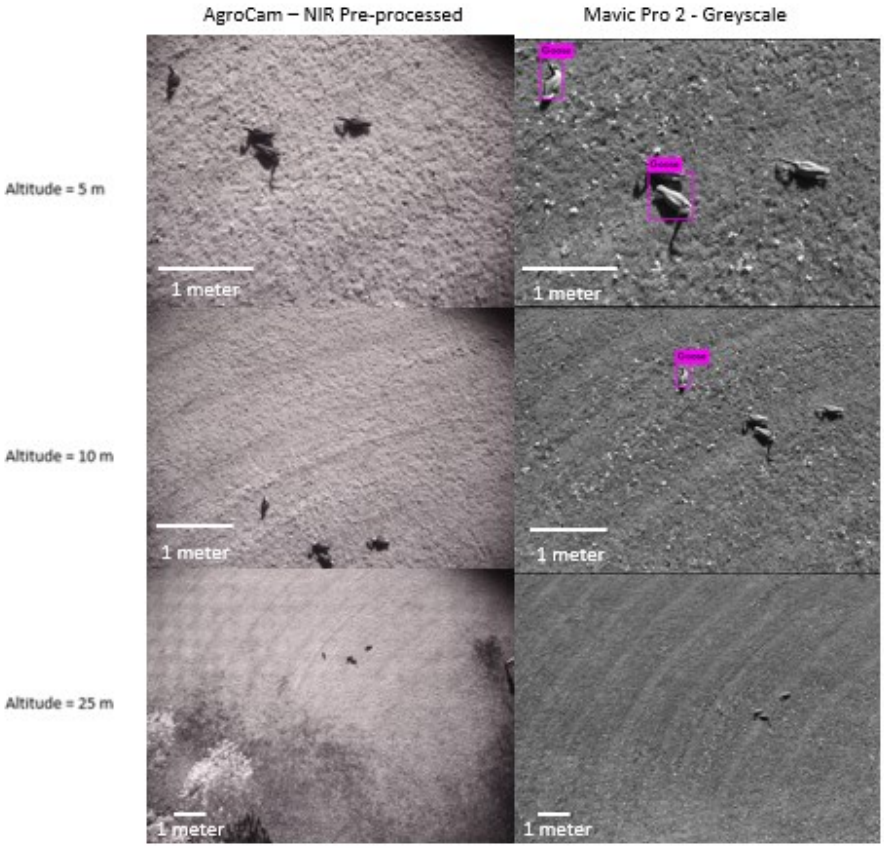
Detection confidence scores (left to right)

78%	67%	85%
97%	80%, 20%	93%, 61%
73%, 35%	66%, 30%	91%, 36%, 39%, 30%
59%, 44%	54%	80%

Figure 32 – Ground test results of the detector trained on the *Blender* Dataset Study.

The 5,000 weights file of the greyscale trained detector was not able to detect any geese in the NIR pre-processed aerial test images but was able to detect geese in the 5 m and 10 m greyscale aerial test images. However, there are less geese detected in this test compared to the

Small Dataset Study when the greyscale trained detector was tested. One observation to note in the 5 m greyscale aerial test result is that the bounding box around the goose that is placed closely next to another goose is tighter than the bounding box generated in the Small and Large Dataset Study when the same goose is detected (Figure 33).



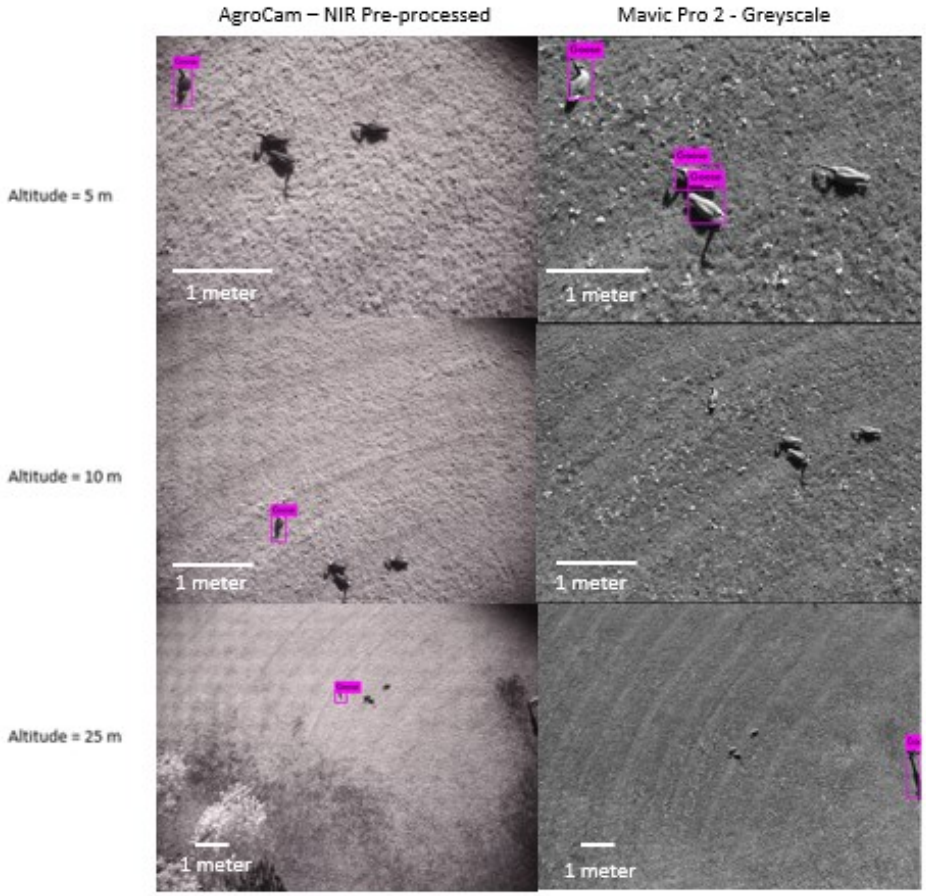
Detection confidence scores (left to right)

0%	74%, 28%
0%	26%
0%	0%

Figure 33 – Aerial test results of the greyscale trained detector using 5,000 weights of the *Blender* Dataset Study.

Although there were more detections using the 5,000 weights of the augmented trained detector on the ground test images compared to the best weights, several bounding boxes were not accurate as seen in Figure 32. However, when the 5,000 weights were tested against the

aerial test images, the detector was able to detect two geese decoys that were placed closely together in the 5 m greyscale test image as individual detections as seen in Figure 34.



Detection confidence scores (left to right)

85%	91%, 35%, 89%
68%	0%
62%	69%

Figure 34 - Aerial test results of the augmented trained detector using 5,000 weights of the *Blender* Dataset Study.

4.3.4 Discussion

It can be clearly seen from the results of the artificial image trained detector performed worse than the detectors trained on real images of Canada geese. This could be because the artificially generated dataset lacked a variety of geese positions and orientations that are present in the real image datasets. This is noted in the ground level images where the detector had

trouble detecting geese on all test images. It is most likely that this dataset had more aerial-like images which do not align with ground-based images such as the images that were used to test the detector with. However, although there were virtually no improvements in the number of geese detections in the aerial images, this detector was the only detector that was able to discern two closely placed geese in the 5 m greyscale aerial test image when the augmented trained detector was tested, whereas in the Small and Large Dataset Study, the two geese were misdetected as one goose.

There were no obvious detection improvements in the NIR pre-processed aerial test images compared to the aerial results in the Small and Large Dataset Study. One hypothesis regarding this behavior is the contrast between the Canada geese and the image backdrops in the artificially generated dataset. Notably, the images with the water and grass backgrounds are darker which offer little contrast between the Canada geese and their background when the images are converted to greyscale. Figure 35 illustrates how a darker RGB background results in minimal contrast between the Canada goose and its background when the image has been augmented.

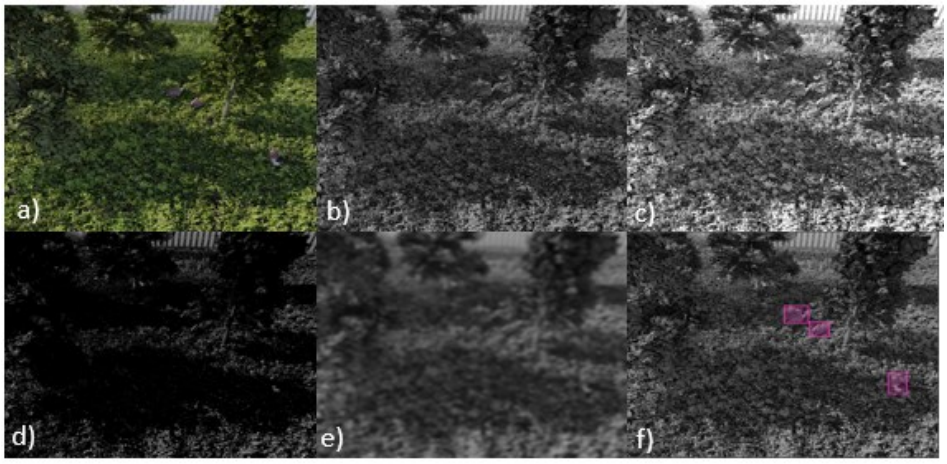


Figure 35 – The effect of the augmentation of the artificially generated images used in this study. Images in order are a) RGB image, b) greyscale image, c) light contrast image, d) dark contrast image, e) Gaussian noise and f) the annotated greyscale image.

This can potentially present a challenge when training the detector as the edges of the Canada goose are difficult to discern against the image backdrop.

One thing to point out in the ground test images is that the predicted bounding box did not cover the entirety of the geese as compared to the predicted bounding boxes of the ground test images in the Small and Large Dataset Study using real images of Canada geese. This is most likely because the artificially generated images modelled the bird in a perched position which lacked details of goose feet and only the goose body was annotated. This shows that image details are important when training the detector so that the predicted bounding boxes encompasses the entire detected object in the test images. Thus, it is very important to include all necessary details in the generated dataset for proper detector learning.

Chapter 5: Investigation into Combining Real and Artificial Images Datasets

In addition to the training and testing of the three datasets mentioned in Chapter 4, two independent studies of combining real and the artificially generated images of Canada geese were investigated since it was found that the combination of real and artificially generated images improved detection rates (Section 2.1.4). This chapter is broken into the following sections:

5.1 Study #1: Hybrid #1 - Combination of Cornell and *Blender* Images

5.2 Study #2: Hybrid #2 – Hybrid#1 Dataset with Additional Artificially Generated Aerial Specific Images

5.1 Study #1: Hybrid #1 - Combination of Cornell and *Blender* Images

5.1.1 Dataset Preparation

Images from the Cornell dataset and the artificially generated dataset provided by Skylar Bruggink (Section 3.3.4) were combined to form a complete dataset consisting of real and artificial geese images. 2,000 greyscale images from both the Cornell and artificially generated datasets were used to create a 4,000 image dataset. The dataset training and validation dataset followed an 80% and 20% split.

5.1.2 You Only Look Once Training of Dataset

The original greyscale images of the Cornell dataset were initially selected to be part of the dataset but due to memory limitations of the Jetson Nano, the training of this dataset would unexpectedly kill the training progress during the first 1,000th iteration of training. Since only the 1,000th weights file was produced, there was insufficient weights data to test with as previous

training studies found that the ideal weights file occurred around the 5,000th iteration. The root cause of the memory issue came from the Cornell dataset as the average image size of the dataset was 3.35 MB and using a YOLOv4-Tiny network resolution of 480x480. Although the Cornell greyscale dataset was successfully used to train YOLOv4-Tiny in the Large Dataset Study, the network resolution was 384x384. It is ideal that the network resolution, batch and subdivision values are consistent with the previous studies in Chapter 3 for proper comparison of results. Thus, it was decided to omit the greyscale dataset and incorporate the augmented Cornell dataset since the dataset was able to handle a network resolution of 480x480, batch of 64 and subdivision of 16.

This dataset was trained with a network resolution of 480x480 and with a subdivision and batch of 64 and 16 respectively and was trained on 10,000 iterations. Though this dataset was successful at completing a full round of training, there were low memory warnings present during training. The best weights file had the greatest performance compared to all the 1,000th weights file that were analyzed. The mAP and the IOU of the best weights file is 90.34% and 75.76% respectively.

5.1.3 Results

Though there is a misdetection seen in the greyscale ground test image, all the geese were detected in the NIR greyscale and NIR pre-processed ground test images (Figure 36). The confidence percentages were noticeably better in the NIR type test images versus the percentages captured by the greyscale test image.

Detection confidence scores (left to right)

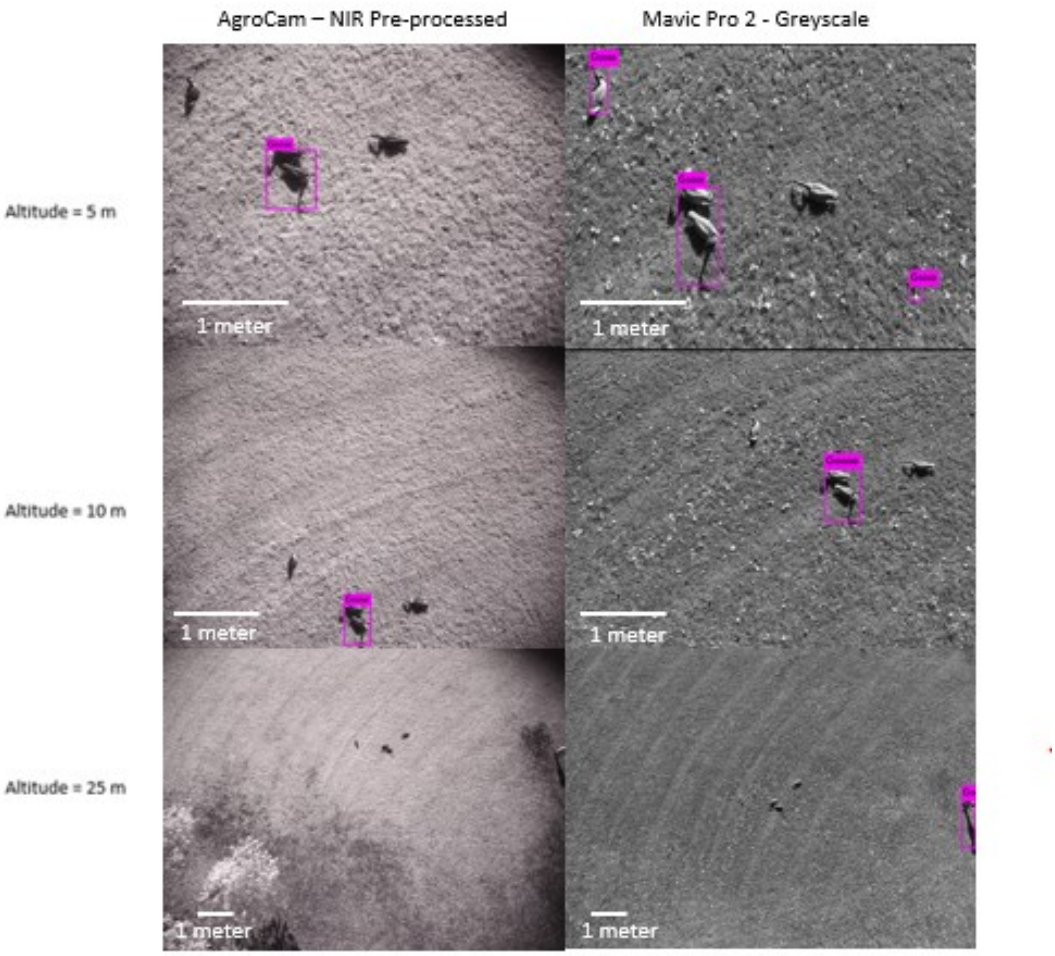
92%, 48%, 96%, 60%, 40%	96%, 87%, 85%, 92%, 68%	98%, 92%, 94%, 89%, 75%

Figure 36 – Ground test results using the best weights of the Hybrid #1 Study.

The detector was able to detect two individual geese in an image consisting of two geese overlapping each other in another ground test image and was the first trained detector to do so (Figure 37). This image was tested in the previous tests where it was consistently being detected as one goose if it was detected.

Figure 37 – Comparison of the a) original image, b) detection result of three geese (bounding boxes are overlapped) and c) detection result of the Cornell dataset trained detector where the detector only detected two geese.

Although there were minor improvements in that the detector was able to discern overlapping geese as two geese instead of one, the detector struggled to properly detect geese in the aerial test images. The most notable error in detection is that two geese is detected as one goose which occurred in both the greyscale and NIR pre-processed 5m and 10 m aerial test images (Figure 38). It can also be observed that a leaf in the grass of the 5 m greyscale test image was detected as a goose.



Detection confidence scores (left to right)

58%	94%, 68%, 58%
70%	79%
0%	61%

Figure 38 – Aerial results using the best weights of the Hybrid #1 Study.

5.1.4 Discussion

This detector which was trained on 2,000 real images and 2,000 artificially generated images performed well on ground test images but underperformed in both the NIR pre-processed and greyscale aerial test images. It can be deduced that this may be the fact since the artificial dataset contained minimal direct overhead shots and more aerial oblique shots, which most likely increased the detector's perspective on ground level images. This seems to align with the results as the detector was able to deduce two overlapping geese as two individual geese in the greyscale

ground test image Figure 37. However, it is suspected that the lack of unique geese models in the artificial dataset caused a leaf in the aerial test image to be detected as a goose. Upon close-up inspection of the leaf reveals that the leaf bears a resemblance of the geese models in the artificially generated images (Figure 39). This could be a result of the artificial image implementation since it used the same goose model in every generated image thus lacking variety in different geese positions that would aid with the information generalization of the detector.

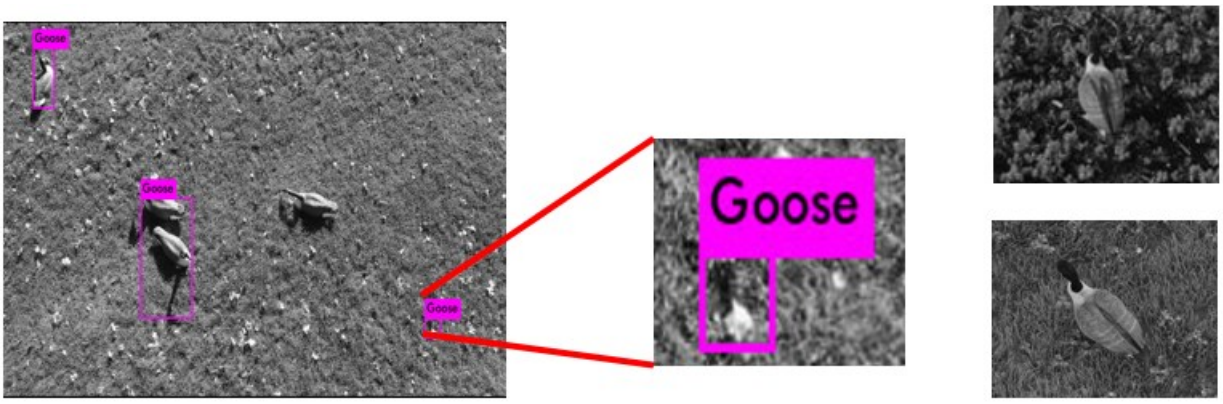


Figure 39 – A closer look at the leaf that was detected as a goose in the 5 m greyscale test image in comparison to the geese models in the training dataset (left).

Both the NIR greyscale and NIR pre-processed images were able to detect all geese in the ground test images but there was minimal detection in the NIR pre-processed aerial test images. Again, this could be the result of having more ground like images in the training dataset. In addition, since the augmented Cornell dataset was used instead of the pure greyscale dataset due to memory limitations of the Jetson Nano, the detector may have generalized better on test images that have a heightened contrast, which is the case of the NIR greyscale and NIR pre-processed images, since there were contrast manipulation and Gaussian smoothing in the augmented dataset. It is also important to note that the detection confidence scores of the NIR type images in the ground test images outperformed the scores of the greyscale images and were

able to detect all geese in the test images properly. Between the confidence scores of the two NIR type images, the NIR pre-processed results had the best scores.

5.2 Study #2: Hybrid #2 – Hybrid #1 Dataset with Additional Artificially Generated Aerial Specific Images

5.2.1 Dataset Preparation

In this dataset, an additional 1,000 artificially generated images of simulated pure aerial perspectives were added to the Hybrid #1 dataset, increasing the number of images in the training dataset to 5,000 images to create the Hybrid #2 dataset. This additional dataset was also provided by Skyler Bruggink and contained the same image backdrops as the dataset used in Section 4.3. The aerial heights in this dataset were simulated to match the altitudes of the aerial test images used in this research which includes replicated altitudes of 5 m, 10 m, 15 m, 20 m, 25 m, and 30 m. This dataset also strictly contains direct overhead shots rather than oblique shots which were presented in the initial *Blender* dataset.

The additional dataset was annotated using the online *makesense.ai* annotation online tool and was converted to greyscale using MATLAB. The complete dataset consisting of 5,000 images followed 80% and 20% split for the training and test datasets respectively. The average size of the additional dataset was 257.62 kB and all images in this dataset have been compressed using the JPEG extension. The images from the Cornell dataset remained augmented while the artificially generated images were strictly in greyscale.

5.2.2 You Only Look Once Training of Dataset

The dataset was trained with using a batch and subdivision value of 64 and 16 and a network resolution of 480x480 over 10,000 iterations. During the training of this dataset, the Jetson Nano encountered memory problems and crashed the training. However, the training was

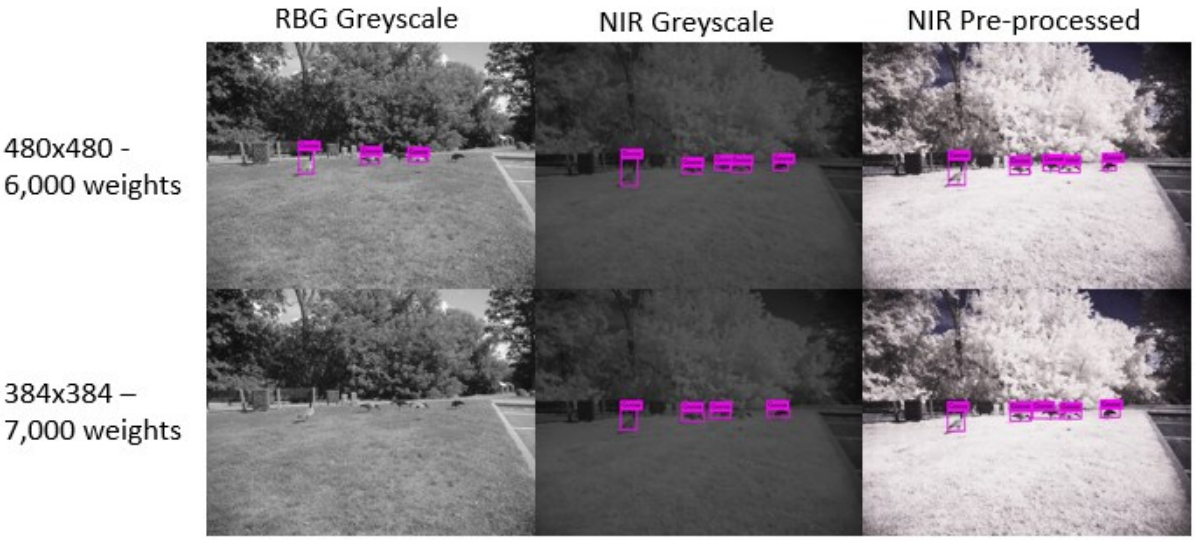
able to reach 7,000 iterations before it crashed and based in the previous studies conducted in this research, it was concluded that there were enough produced weight files to run an analysis. A best weights file was produced and both the best weights file and the weights at each 1,000th iteration generated weights file was analyzed. It was found that the weights at the 6,000th iteration performed the greatest. The mAP and IOU of the 6,000th weights file is 90.91% and 71.32 % respectively.

Since the initial attempt of training caused memory issues, the network resolution was brought down from 480x480 to 384x384 while leaving the batch and subdivision values the same, but this also caused memory issues as the training crashed mid-training. It was decided to send two images instead of four for parallel processing thus the subdivision value was changed to 32 while keeping the network resolution as 384x384 and the batch value as 64. While the detector was able to do a full pass of training using a smaller network resolution and sending less images for parallel processing, there were still low memory warnings during training. It was found that the weights at the 7,000th iteration had the best performance detection in this case. The mAP and the IOU of the 7,000th weights file is 88.60% and 71.49% respectively.

5.2.3 Results

The weights at iteration 6,000 of the 480x480 network resolution detector produced significant improvements in the detection of geese in the NIR type images of the ground test images (Figure 40). It was also notable that the confidence percentages of the NIR pre-processed test image are higher than the confidence scores of the NIR greyscale test image. There was significant detection degradation when the 7,000th weights from the 384x384 network resolution trained detector were used on the same ground test images as it did not detect any geese in the greyscale test image but is it important to note that all geese were detected in the NIR pre-

processed test image. Based on these initial findings, the 6,000th weights file of the 480x480 network resolution trained detector was used to test the aerial test images going forward for this test.

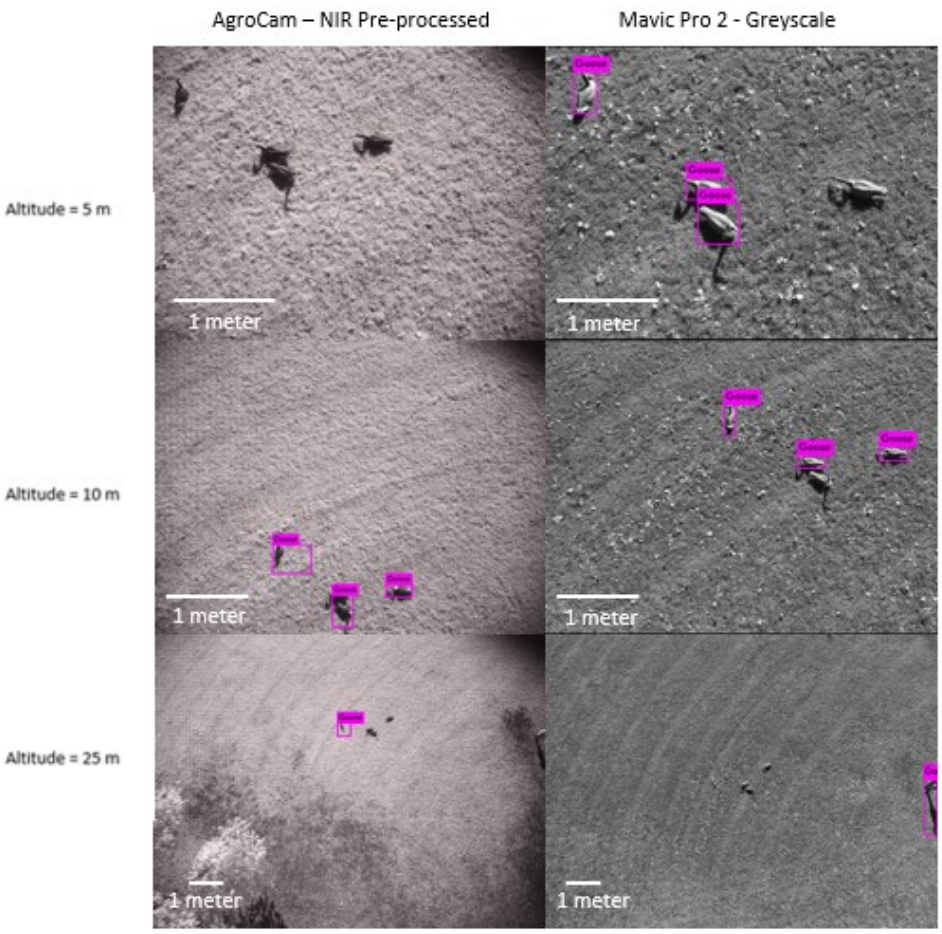


Detection confidence scores (left to right)

64%, 66%, 90%	96%, 91%, 70%, 89%, 69%	96%, 97%, 95%, 94%, 85%
0%	90%, 94%, 38%, 46%	91%, 82%, 41%, 68%, 39%

Figure 40 – Ground test results of the Hybrid #2 Study.

The 6,000th weights file had significant improvements in detections in the greyscale aerial test images in comparison to its NIR aerial test image counterpart. As it can be seen in Figure 41, the detector was able to discern the closely placed geese in the 5 m greyscale test image as two individual geese, as well as detecting three of the four geese in the 10 m greyscale test image. In addition to detecting the closely placed geese as two individual geese, the predicted bounding box around the geese in they greyscale aerial test images are much more refined compared to the NIR pre-processed test images where some of the predicted bounding boxes are not tight around the detected object.



Detection confidence scores (left to right)

0%	90%, 87%, 60%
27%, 29%, 68%	61%, 35%, 55%
31%	53%

Figure 41 – Aerial detection results using 6,000 weights of the Hybrid #2 Study.

5.2.4 Discussion

It can be clearly seen that the addition of pure aerial images in the overall training dataset improved the detection in the aerial test images. The aerial results seen in this test were also the best results out of all the training and detection studies completed for this research in terms of accuracy. Not only did the detector improve on aerial detections, but the detector was also still able to detect all geese present in the NIR type ground test images even with an increase in aerial type images in the overall training dataset. However, it was found that training with a larger

dataset comes at a cost of computer memory as more details are extracted during training which requires more memory. As it was seen in the results, changing the training parameters such as the network resolution from 480x480 to 384x384 so that the Jetson Nano was able to complete a full training pass significantly degraded detection performance. This is expected as less information from the input image are extracted since changing the network resolution to a smaller resolution resizes the input images to the defined network resolution. This causes the input images to lose detailed information that is seen and learnt when the network resolution is larger.

From the aerial test results, it can be clearly seen that the greyscale images outperformed their NIR pre-processed test image counterpart which is a stark contrast compared to the results of the ground test images. This could be because the aerial images used in the overall training dataset for this study were not augmented and were left as greyscale images. The only augmented images that were real images were ground images from the Cornell dataset which were used strictly because of the memory problems on the Jetson Nano during the Hybrid #1 study, which could also explain why the detector performed well against the NIR type ground test images. In Figure 42, it can be clearly seen that the greyscale background of the artificially generated image is similar to the background of the aerial test images. It is also important to note that of the 5,000 images used to train this detector, 3,000 images were artificially generated aerial images that were converted to greyscale which could influence how to detector behaves when it sees an aerial test image in greyscale form. Though the detector was able to detect geese in the NIR pre-processed aerial test images, some of the predicted bounding boxes are not a definitive box around the detected geese (i.e. not tight around the detected object) which indicates that training still needs to be refined to test against the NIR aerial images.

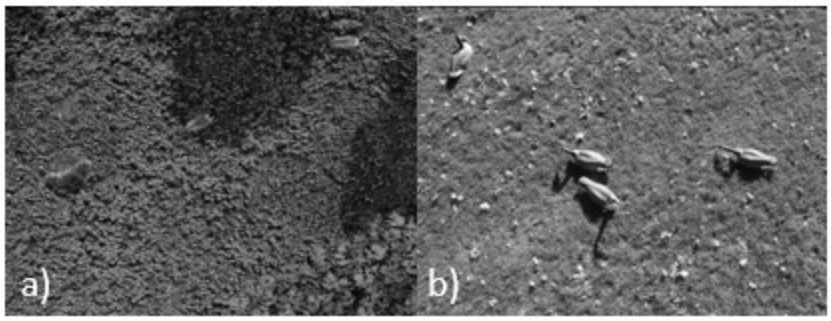


Figure 42 – Similarities of the greyscale background of the a) artificially generated image with a grassy landscape and b) real test image also with a grassy landscape.

Chapter 6: Discussion

From the results presented in Chapter 4 and Chapter 5, this work produces several key findings. The collection of images in the NIR spectrum and RGB (then converted to greyscale) were tested and compared to verify if NIR spectrum images yielded detection improvements results. The training and detection results of the YOLOv4-Tiny training parameters and the different types of datasets that were presented in this study were investigated. The flying altitude of the RPAS during aerial test image collection also played a major role in the object resolution since the camera focal length and RPAS distance to the object determines the true image resolution. To mitigate the lack of real aerial images contained in the overall dataset, a preliminary study of using artificially generated aerial images for this study was explored. Each key finding is discussed in detail in the following sections:

6.1 NIR Imaging

6.2 YOLOv4-Tiny Training

6.3 YOLOv4-Tiny Detection

6.4 RPAS Flying Altitudes

6.5 Artificially Generated Images

6.1 NIR Imaging

The use of NIR imaging in our test images showed promising results when the detector was trained and tested on real images. In the initial testing of the NIR images in Section 4.1, the Small Dataset Study, there were marginal detection improvements in both the NIR greyscale, and the NIR pre-processed ground test images compared to their greyscale counterpart. This is especially true when the images were tested using both the greyscale and augmented trained detectors. There were further detection improvements in the NIR greyscale and NIR pre-

processed ground test images once the overall dataset grew to 2,000 images as seen in the Large Dataset Study in Section 4.2. In this case, additional geese were detected in the NIR type ground test images when the greyscale trained detector was tested. Additional improvements were seen when the augmented detector was tested as all geese in the NIR type ground test images were detected. It also can be noted that the confidence scores were higher in the NIR pre-processed ground test images.

From the results of all the studies conducted for this research, it was seen that in general, the confidence scores of the NIR type ground test images exceeded the scores of the greyscale ground test images. In the Large Dataset Study where the dataset meets the minimum recommended number of images, the augmented test confidence scores of the NIR type ground test images surpassed the scores of the greyscale confidence scores. It was also noticed that the scores of the NIR greyscale and NIR pre-processed results are comparable with one another. In the *Blender* study, the NIR pre-processed confidence score of the far-left goose in the ground test image that was detected in each detection test had the best confidence score (Figure 32). To further complement this finding, the confidence scores of the NIR pre-processed ground test images of the Hybrid studies performed better than the NIR greyscale confidence scores. It is also noted that the confidence scores of the greyscale aerial test images were generally better than the NIR pre-processed aerial test images. Pre-processing the raw images of the decoys turned the geese dark and it is suspected that it altered the detector's perspective since the images in the overall dataset did not contain geese as dark as the geese in the NIR pre-processed images. It is assumed that in the NIR pre-processed aerial test images, the colour of the geese decoys did not match the detector's expectation of goose based on the training dataset, which could influence the confidence score.

The improvement in detection of the NIR greyscale and NIR pre-processed images were also significant when the augmented detector was used to test the ground test images. This showed that adding light and dark contrast adjustments, as well as adding noise to the images (Gaussian smoothing), assisted the detector in generalizing the features of geese and was able to properly detect both varieties of the NIR test images as the images provided an enhanced contrast in object edge which potentially leads to a higher probability in detection. This is clearly seen in Figure 43 as the white plumage between the tail and leg is slightly camouflaged with the image backdrop in the a) greyscale image whereas in the b) NIR greyscale and c) NIR pre-processed image, the white plumage is easily discernable. Colour, pattern and shape may be some of the features the detector had learned during training and if the detector is not able to discern the white plumage in the greyscale image, it will potentially alter the detectors' perspective on what features to look for when fed with a test image that lacks specific details. Comparing the NIR type images, it can be seen that the contrast of the NIR pre-processed image is much greater than the NIR greyscale image, which could be the reason why the confidence scores were higher for the NIR pre-processed test images in comparison to the NIR greyscale test images.

Figure 43 - Different contrasts is seen in the a) greyscale versus the b) NIR greyscale and c) NIR pre-processed images.

There were also detection improvements seen in the NIR pre-processed aerial test images with regards to the number of detections versus the same greyscale aerial test images when the

detector was trained on a dataset containing real images of Canada geese. However, in the tests using the artificially generated dataset, the greyscale aerial test images generally performed better than its NIR pre-processed image counterpart. The reason for this is explained in Section 4.3.4. Thus, it is suspected that if the detector was trained on a dataset that contained real aerial images or if the artificial images had varying background gradients similar to those of real images, a detection improvement in the NIR pre-processed aerial test images would most likely be seen.

6.2 YOLOv4-Tiny Training

This study investigated a variety of training parameters that determined the accuracy and training time of the YOLOv4-Tiny detector by analyzing the weights files produced at every 1,000th iteration and comparing it with the best weights file. The effect of the overall dataset size, which ranged from 423 to 5,000 images, and the type of images (geese orientation in real images and aerial specific images) were analyzed, and findings were noted.

There was a disparity between the best weights and the optimal weights since the best weights file did not necessarily produce the best detection results. This is because YOLOv4-Tiny bases its mAP and IOU metrics on the validation image set and is dependent on the type of images within the validation image set and the size of the validation image set, which is reliant on the overall dataset size. The dataset size did influence the training because the optimal weights were produced other than the best weights file. An example of this was seen when the training of the Small Dataset Study is compared to the other studies that utilized 2,000 or more images to train the detector with. In the Small Dataset Study, the augmented dataset consisted of 1,269 images and the detector was trained for 20,000 iterations where the optimal weights were found at iteration 13,000. In the case for the other datasets that had 2,000 or more images that

were used in this study, it was found that the optimal weights were typically found at about iteration 5,000 when the detector was trained for 10,000 iterations.

The type of images in the overall training set also plays an important role in the mAP and IOU values as it is the validation image set that determines the mAP and IOU values. A caveat to this is that the test images used may not be similar to the images within the validation image set, which is most likely why the mAP value is high but is not able to detect all geese in the test images. Since the optimal weights were obtained at a specific point within the training iterations, it was found that detector performance deteriorated when subsequent weight files after the optimal weights were used, thus it can be concluded that overfitting occurred at these weights. Although overfitting did occur in all the trainings done for this research as a large maximum batch value was used, it is better to overfit the detector rather than not providing enough iterations to properly learn and generalize object features, which requires more time allocation to properly train the detector.

Another point to make regarding the images of the complete training dataset is the kind of images used to train the detector. Surprisingly, the detector was able to detect geese from an aerial perspective when it was trained on ground level images. There was also an improvement in detections in both the greyscale and NIR pre-processed aerial images when tested in the Small Dataset Study using the augmented trained detector. A reason for this could be because the detector was trained to look for a “tear drop” like feature for the body with a black stick (the neck) protruding from the body. Figure 44 a) shows a sample image in the Small Dataset Study that was used in the training dataset. The “tear drop” shape with a protruding black rod is seen in the Figure 44 b) NIR pre-processed image which shows the similarities between the images

although one was taken from a ground perspective and the other was taken in an aerial perspective.

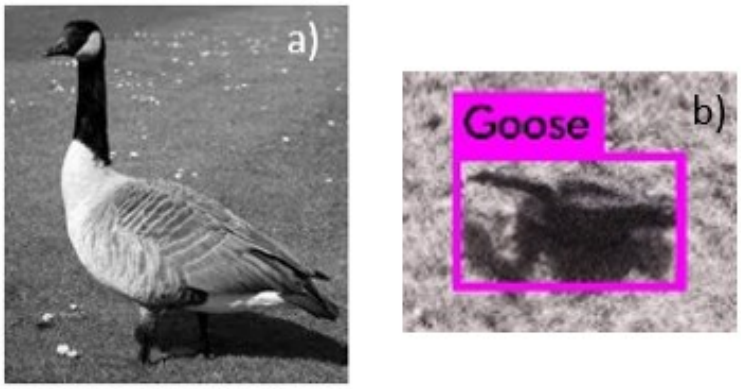


Figure 44 – Similarities between a a) ground perspective goose and b) an aerial perspective goose.

A significant limitation of the trained detectors used in this study was that the detectors were not robust in detecting clusters and overlapping geese. This is seen most notably in the detection of the closely place geese in the aerial test images and overlapping geese in the ground test images seen in Appendix A where the detector would detect only one instance of geese instead of multiple instances. One possible reason for this is the annotated bounding boxes overlap one another if all geese are annotated if the image contains a flock of geese. Taking Figure 45 as an illustration, when the annotated bounding box is applied to the overlapping geese, it can be seen that the annotated bounding box includes a partial annotation of the geese in the foreground.

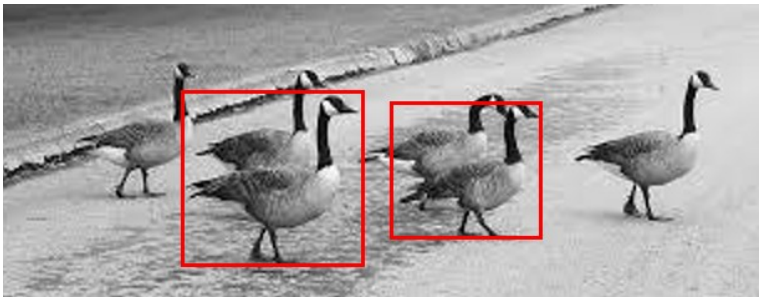


Figure 45 – Annotated bounding boxes showing a partially annotated goose along with the main goose that is annotated.

Because the annotated bounding box technically captures the information of two geese instead of one targetted goose, the detector may have learnt features appearing in pairs of geese which is why the detector is able to detect two geese since it was techinally trained to look for two seperate geese features instead of one. It is possible that we may have been providing the detector with inaccurate information when including training images that have clusters of geese where the annotated bounding boxes contain a partial second goose.

6.3 YOLOv4-Tiny Detection

While the Small Dataset Study was able to detect the geese decoys in both the NIR pre-processed and greyscale test images when the augmented trained detector was used, there was a clear degradation in detection performance when the aerial images were tested against the Cornell augmented dataset in the Large Dataset Study. This could be because the Cornell dataset contained 2,000 images of ground images versus 1,269 ground images contained in the Small Dataset Study. By training the detector with more ground images, it could be that with a larger dataset comes with more reinforcement of features during training. Thus, in the Large Dataset Study results, it is suspected that the detector is biased towards geese at ground level perspective. This is confirmed in that all geese were successfully detected in the NIR type images in the Large Dataset Study but not in the Small Dataset Study. This can also be said for the *Blender* Dataset Study where the overall dataset contained aerial and oblique images of Canada geese. In this case, the detector performed poorly on the ground test images but was the first detector in this study to detect the closely placed geese as two individual geese in the 5 m greyscale aerial test image (Figure 34). This could be attributed to the fact that the detector generalized more on geese information from an aerial perspective simply because there were more aerial type images contained within the overall dataset.

It can be noticed that in the aerial results of the Small Dataset Study and Large Dataset Study, some geese were detected in the 10 m test image when they were not detected in the 5 m test images in both the NIR pre-processed and greyscale aerial test images. This could be a result of using a detector that was trained on ground-based images to detect geese from an aerial perspective since the detector did not learn specific features from an aerial perspective and lacks generalization when tested against aerial images. This occurrence is only seen when the detector has been trained on real image since the aerial detections seem more stabilized when aerial images are introduced in the complete training dataset.

Another aspect to note is the detection of the geese decoys shadows in the aerial test images in both the NIR pre-processed and greyscale aerial test images in the Small Dataset Study and the Large Dataset Study. As mentioned in Section 6.2, the reason for this could be because the detector is looking for a protruding rod feature since the detector was trained on real geese images. When strictly aerial type images were used to train the detector in the *Blender* Dataset Study, the detection of the decoy shadows were omitted when geese were detected in the aerial test images. This is most likely due to the fact that the geese models in the *Blender* dataset matches the geese decoys in that the geese necks in the artificially generated images are much shorter than the necks presented in in the real images (Figure 46).

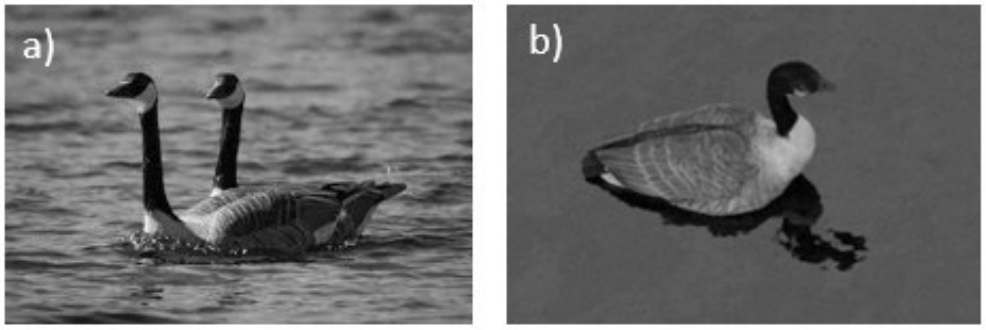


Figure 46 – A sample image from a) the real geese dataset and b) the artificial image dataset showing the differences in neck length.

In addition, the type of training images determines the outcome on whether the decoy is detected or not. In the Hybrid #1 Study, the decoy shadows were detected in the aerial test images where detections were successful. However, in the Hybrid #2 Study, we see that the shadows are omitted in the successful aerial detections. It is possible that the addition of 1,000 pure aerial images in the Hybrid #2 dataset influenced the detector to detect geese from an aerial perspective which omits detecting the long neck since the artificial images do not contain images of geese with long necks and strictly looking at geese from an aerial perspective.

6.4 RPAS Flying Altitudes

The RPAS flying altitude had an effect of the detection probabilities of the geese. As shown in the results section, there were minimal geese detected in the 25 m altitude test image, but more detections were seen when the RPAS flew closer to the decoys. This is attributed to Johnson's criteria where the resolution of the object in the image is correlated to the number of pixels of the object (Ooi, 2019). The more pixel cycles there are within the object, the greater the chances of the object being recognized and identified. This can also be confirmed in the GSD calculations that was done for the Mavic Pro 2 and AgroCam cameras in Section 3.4.1 which found that as the RPAS flew closer to the ground, more pixels were present giving the image more details. Thus, the further away the image is taken from the object, the less pixels the object will contain which will potentially cause missed detections due to poorer image resolutions. The aerial test images used in this research were taken at flight altitudes at 5 m, 10 m, and 25 m however, 5 m is unrealistic to fly over real animals as it is too close and causes disturbance to the animals. This was confirmed when the research team flew the Mavic Pro 2 close to a flock of geese at Alfred Lagoon in Plantagenent, ON. The Mavic Pro 2 was flown at an altitude of 75 m

but when the RPAS approached the flock of geese, it made the birds agitated and they either flew away or swam to the opposite side of the lagoon, away from the RPAS.

Although the detector was able to detect geese in 25 m aerial test images as seen in some of the results presented in this research, it is a questionable detection since the geese resemble grains of rice in the default resolution of the image. However, upon closer inspection when the image is zoomed in at 1083% using the Windows Photo application (Figure 47), the outline of the geese is discernable. As mentioned in the YOLOv4-Tiny Training discussion in Section 6.2, it is suspected that the detector is trained to look for a tear drop like shape with a rod protruding from the shape, in addition to other patterns shapes such as the tapering of the end body and the differentiation between the wing and the body. This could be the reason why the detector was able to detect the geese at a flying altitude of 25 m.

Figure 47 - The 25 m test image zoomed in at 1083% using the Windows Photo application which still highlights specific geese features such as the contrast between the wing and body and the length of the neck.

6.5 Artificially Generated Images

Due to the lack of aerial images presented in the real images obtained for the Small Dataset Study and Large Dataset Study, the aerial images used in this study were strictly artificially generated using the 3D rendering software *Blender* as described in Section 3.3.4. The

artificially generated aerial images proved to be a viable solution to obtain training images that are challenging to obtain and lacking from the overall dataset. Artificially generating the training images also allows many degrees of freedom to what to simulate such as scenery, geese randomization, weather and image perspectives that would help the detector generalize object information during training. In the artificially generated aerial images used in this study, a variety of backdrop sceneries were used as well as different geese positioning due to a randomization algorithm.

It has been shown that the detector can be fully trained on a training dataset that has been completely artificially generated. In the *Blender* Dataset Study in Section 4.3 where the detector was trained with 100% artificially generated images that simulated aerial perspectives at oblique angles, the detector was able to discern two closely placed geese as two individual geese which was not achievable in the Small Dataset Study and Large Dataset Study when the best performing trained detector was used. In addition, the augmented trained detector was consistent in detecting the top left goose in the NIR pre-processed aerial 5 m, 10 m, and 25 m test images (Figure 34). This shows that there is potential in training the YOLOv4-Tiny detector with strictly artificially generated images. Although the ground level detections of the *Blender* Dataset Study were subpar, this is most likely because there were no ground level images in the overall training dataset.

Artificially generated images can also complement a training dataset with real images as seen in the Hybrid #1 and Hybrid #2 studies. Since the real images consisted of ground level images and the artificial images comprised of aerial images, the mixing of artificial aerial images and real ground images in the overall training dataset allowed the detector to learn features from a ground and aerial perspective since the detector was able to detect geese in the ground and

aerial test images. Since the aerial test in the Hybrid #1 Study performed well below the results of the Hybrid #2 Study, this shows that the addition of 1,000 artificially generated aerial images in the Hybrid #2 dataset can improve aerial detections. This demonstrates that artificially generated images can be included in the overall dataset to account for specific positions to detect such as in the case for this research, detecting geese from an aerial perspective.

6.6 Cost Considerations

The NIR cameras used in this study are lightweight, low cost and can be mounted on the Mavic Pro 2, by using the 3D printed AgroCam mounts made specifically for the Mavic Pro 2, to gather aerial images and data. This method of gathering aerial images and data is significantly cheaper than employing a helicopter used for aerial counting. A helicopter rental costs approximately $1,300 CAD per hour (Excell'Jets, 2022) and requires man hours to manually count the animals and capture images. The use of a helicopter also comes with risks in the event of a helicopter crash. In addition, the use of the NIR AgroCam camera used in this study is considerably cheaper than thermal cameras such as the FLIR Outdoor Thermal camera at $3,3985 CAD (ITM Instruments Inc., 2022) for example. Using the RPAS to gather aerial images and employing AI to automatically count to animals is much safer, cheaper and omits the need for manual counting, which is prone to human errors. Table 7 outlines the approximate cost of the total aerial equipment used in this study. Though the Mavic Pro 2 RPAS has been discontinued at the time this thesis was written, it’s successor, the Mavic 3, will be used in the cost analysis since the AgroCam mounts can be designed and 3D printed to custom fit any RPAS assuming it meets RPAS payload requirements. Note that the prices mentioned in this section and in the table are as of July 23, 2022 and has been converted from USD to CAD using a

conversion rate of 1.29 (July 23 2022). The prices also exclude shipping and customs fees, and other additional fees.

Mavic 3 (DJI, 2022)	$ 2,453.03
AgroCam Cameras (RGB and NIR cameras were bought as a set) (Norward Expert LLC, 2017)	$ 930.06
YOLOv4-Tiny	Free (open source)
Total Cost	**$3,383.09**

Table 7 - Total cost of equipment used for aerial imaging used in this study.

Chapter 7: Conclusions and Recommendations

This has been a study of NIR images, both greyscale converted (JPEG) and pre-processed (raw), to enhance image contrast to investigate if NIR type images will yield detection improvements compared to RGB converted to greyscale test images. In addition, several training variables such as dataset sizes, the type of images contained in the overall dataset and the YOLOv4-Tiny training parameters were also investigated to optimize training and detection results. A preliminary study on utilizing artificially generated aerial images was also investigated in this research.

In total, this research has completed five studies consisting of varying datasets and verified if there are any improvements in detection if the test images are in the NIR spectrum. The consensus of this study is NIR does improve object detection if the overall training dataset contains augmented images. One of the contributing factors of why the detector was able to perform well on NIR type test images was most likely due to the low and high contrast editing, and the addition of noise via Gaussian smoothing of the original greyscale images in the overall training dataset. This is simply because the NIR type images had contrast enhancements applied when the NIR image was either converted to greyscale or pre-processed in *Rawtherapee*. Although Gaussian smoothing does not manipulate the contrast, it does provide a new perspective of the images during training which increases the generalization of object information, which could also aid in the improvement of detections in NIR images presented in this study. Augmenting the datasets also proved as a viable solution to artificially grow a training dataset into a larger dataset for training purposes.

Another key finding is with regards to the use of artificially generated images employed in the *Blender* and Hybrid studies. Artificially generated images can serve as an alternative to

real image datasets where real images are challenging to procure, such as real aerial images of Canada geese and capturing real images over a specific backdrop. The combination of real and artificial images in overall training dataset provided promising detection results, although this study was not able to complete training using ideal training parameters due to memory limitations on the Jetson Nano.

The combination of ground and aerial images employed in the overall training set proves that the detector can be trained on many degrees of freedom warranted there are enough images to solidify feature recognition during training as seen the Hybrid #2 study. In the Hybrid #2 study where a combination of aerial and ground level images were used in the complete training dataset, the detector was able to detect both ground and aerial types test images. This can pave way for a "one-size fits all" approach when the detector is needed to detect birds from various perspective and angles.

7.1 Recommendations

Based on the conclusions of this study, several recommendations are provided to improve future training and detection of Canada geese using the YOLO detector to gain a deeper understanding to employ the detector in shorebird detection applications in the future. The sections recommended for future studies are broken in the following sections:

7.1.1 YOLO Training

It is advised to investigate the YOLOv4-Tiny training parameters for this specific application as training the YOLOv4-Tiny detector will need to be optimized. As such, the

training parameters, such as the batch and network resolutions used in this study can serve as a baseline for future studies. This study ran the training at 10,000 iterations however, the best performing weights seemed to be at the 5,000th weight mark. By reducing the number of iterations, training time will be saved. Another factor that needs to be considered is determining the ideal network resolution and batch values, though this is limited on the type of computer being used to train the detector. The training parameters should be characterized based on the type of computer used in the study and the training limits should be tested to ensure computer memory boundaries are respected while yielding optimal results. Since multiple instances of memory issues were encountered during the training for this study, it is also recommended to source a computer with larger RAM for training. In addition to studying the training of the YOLOv4-Tiny algorithm, it is recommended to study the algorithm during training to shed light on how geese features are learnt during the training process which will open a new perspective on how the detector generalizes information during training. This will provide an idea of what exactly the detector is looking for which could influence the type of test images we want to ideally source for optimum detections.

Another suggestion for training is to remove images that contain overlapping flocks of geese from the complete training dataset. In the aerial detection results of the Small and Large Dataset Study, it was seen that the geese decoys that were closely placed together were detected as one goose. This could be because of annotating images where overlapping geese are present as described in Section 6.2. Thus, it is recommended to omit these images from the overall training dataset as it was shown that the artificially generated aerial images, which contains no geese overlaps, can detect the closely placed geese decoys as two individual geese as seen in Section 4.3.3. In addition, further studies could be done to investigate methods and techniques to

improve detection in object clusters if images containing overlapping geese in the overall training dataset are to remain in the dataset.

In terms of future training investigations, it is advised to fully convert the hybrid datasets into an augmented dataset since only the Cornell images were augmented while the rest of the images within the hybrid datasets were greyscale. Training the detector with the augmented dataset showed improvement over its greyscale trained counterpart thus, it is recommended to investigate if there will be an improvement in detection if all images in the hybrid datasets have been augmented using the same augmentation techniques used in this study. This test would be applicable to the testing of greyscale aerial images as the augmentation of the artificially generated images in used in this research did not perform well on NIR pre-processed aerial test images.

7.1.2 Artificially Generated Images

Since the artificial aerial images used in this study was generated such that no two geese are placed together, images of this nature can be generated to study if it improves the detection of images containing clusters of geese or overlapping geese. Another aspect to improve in the artificially generated images is to add different backdrop gradients as if the image was taken in the morning, afternoon, or evening. As seen in Section 4.3.4, the current backdrops of the images seem dark even in its RGB form which makes the image appear darker when the image is converted to greyscale and when the dark contrast augmentation is applied.

It is also important to include varying geese positions if the detector is to detect live imaging of birds in the future. Positions of live birds are unpredictable and could result in various positions. It is ideal to generate images that include diverse geese positions in preparation to detecting live geese so that potential positions are easily discernable by the

detector. The *Blender* Dataset Study is a good example of strictly training with aerial like type images which did not perform well on ground test images.

7.1.3 RPAS Equipment

It was obviously not possible to fly the RPAS at a height where it did not cause a disturbance to the flocks of geese while gathering high resolution images to test the detector with. Thus, it is recommended to source a camera with a narrow FOV lens, also known as telephoto lenses, that could be utilized to simulate images taken at 5 m altitudes when flying at higher, less invasive altitudes. A study should also be done to investigate if the decibels from the Mavic Pro 2 causes a disturbance to Canada geese at various flying and take off ranges.

7.2 Conclusions

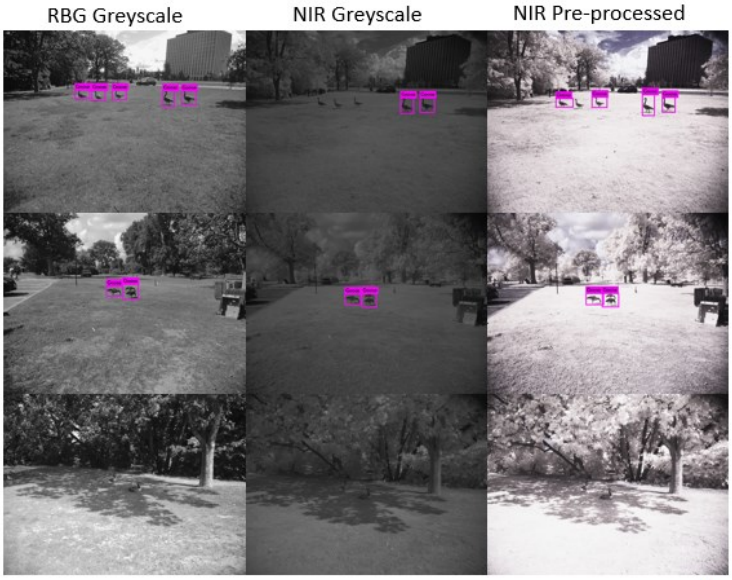
In this study, we have studied the utilization of NIR reflectivity on healthy vegetation in NIR raw images to enhance object contrast via image pre-processing to improve detection of Canada geese. We have shown that the YOLOv4-Tiny detector shows an improvement in detection performance of Canada geese when NIR type images were used as test images compared to RGB images that were converted to greyscale when the detector has been trained on greyscale image datasets. It was found that the NIR test images resulted in an increase in Canada geese detections and confidence scores of the predicted detections. Training the detector with an augmented dataset containing contrast manipulations and Gaussian smoothed images also improved performances of the NIR test images. In addition, the varying overall dataset size was investigated, and it was clearly seen in this research that as more images were used in training, detection became more refined. Due to the lack of aerial perspective images in the datasets containing real Canada geese images, custom artificial images were generated which the YOLOv4-Tiny was successfully trained on. The use of artificially generated images showed potential as the hybrid datasets, which combined real and artificial images, were able to detect

geese in both ground and aerial test images. As well, the use of NIR imaging equipment and a RPAS has been shown to be a safe and low-cost method to capture aerial images of animals.

Appendix A – Additional Ground Results

The additional ground test results of this study are shown in this section. Test images from left to right are greyscale, NIR greyscale and NIR pre-processed.

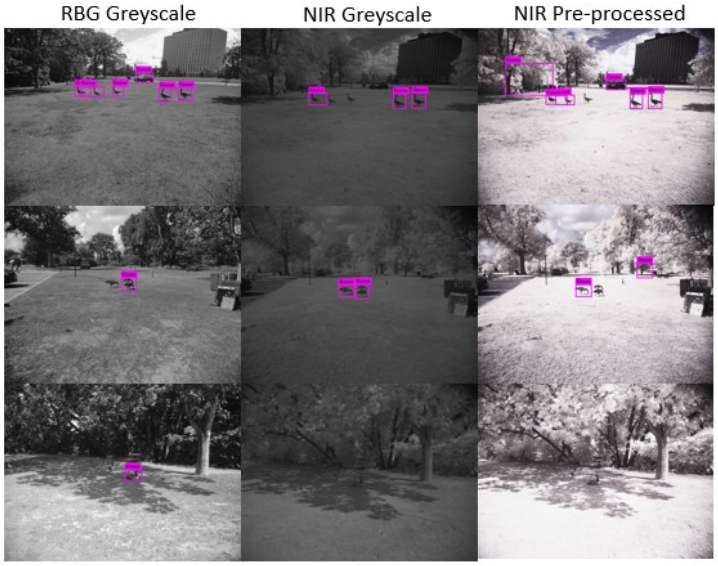
Small Dataset Study – Greyscale trained detector using 13,000 weights



Detection confidence scores (left to right)

82%, 89%, 85%, 97%, 96%	96%, 84%	51%, 88%, 45%
59%, 95%	93%, 81%	94%, 92%
0%	0%	0%

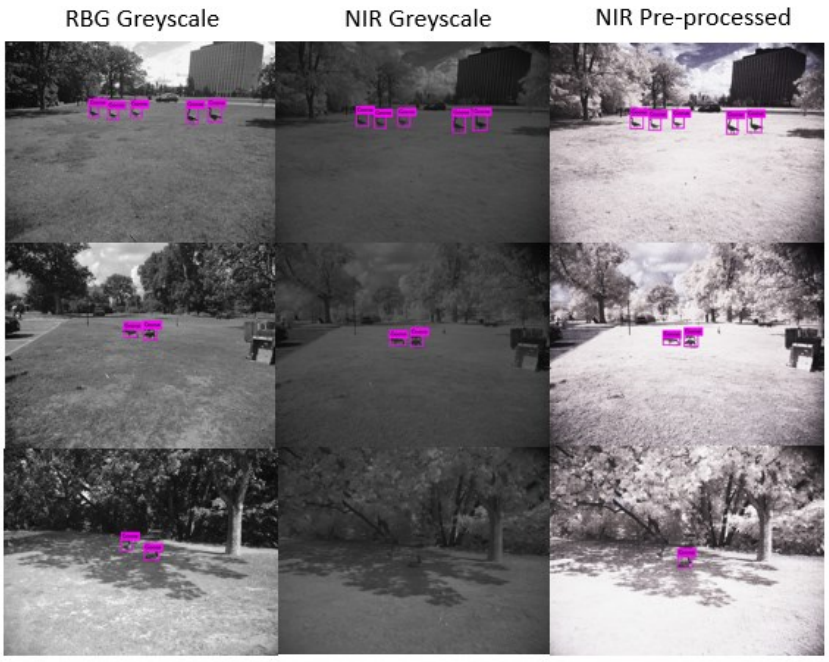
Small Dataset Study – Augmented trained detector using 13,000 weights



Detection confidence scores (left to right)

32%, 36%, 28%, 62%, 96%, 93%	57%, 97%, 97%	28%, 81%, 43%, 74%
72%	95%, 86%	52%, 87%
28%	0%	0%

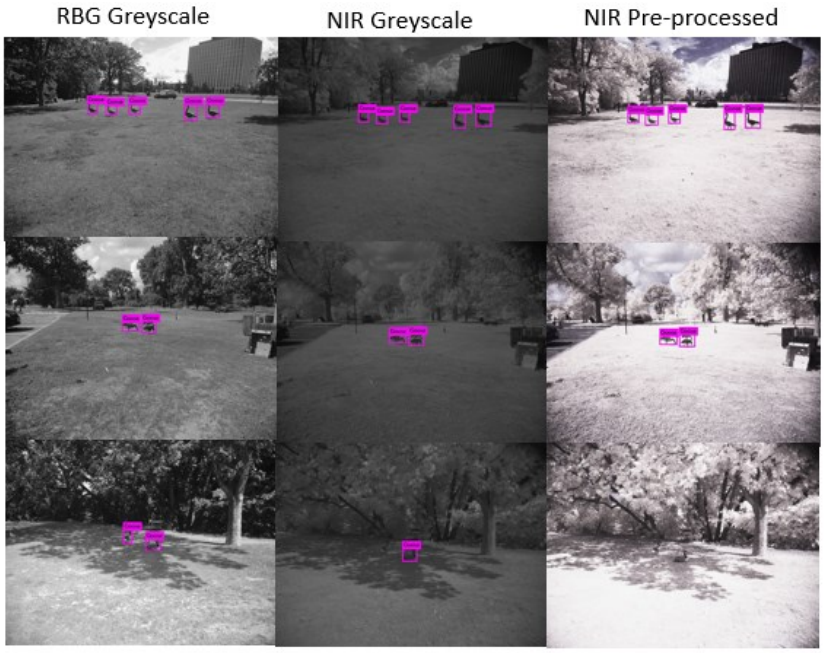
Large Dataset Study – Greyscale trained detector using 4,000 weights



Detection confidence scores (left to right)

59%, 73%, 78%, 93%, 96%	93%, 87%, 87%, 71%, 91%	96%, 86%, 83%, 81%, 97%
45%, 82%	49%, 45%	76%, 83%
51%, 81%	0%	55%

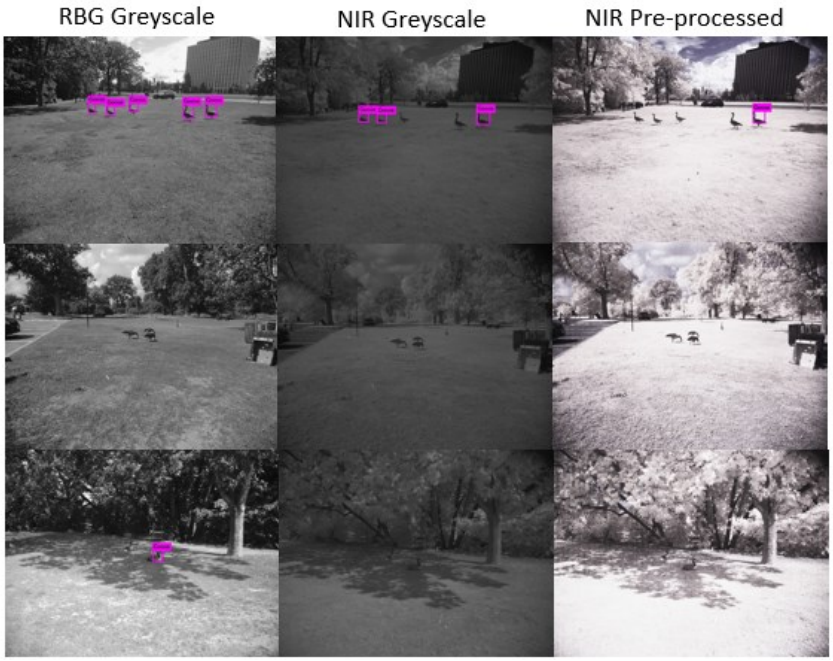
Large Dataset Study – Augmented trained detector using 5,000 weights



Detection confidence scores (left to right)

98%, 98%, 99%, 98%, 99%	98%, 96%, 97%, 99%, 98%	99%, 96%, 99%, 99%, 97%
99%, 99%	100%, 98%	100%, 98%
91%, 99%	95%	0%

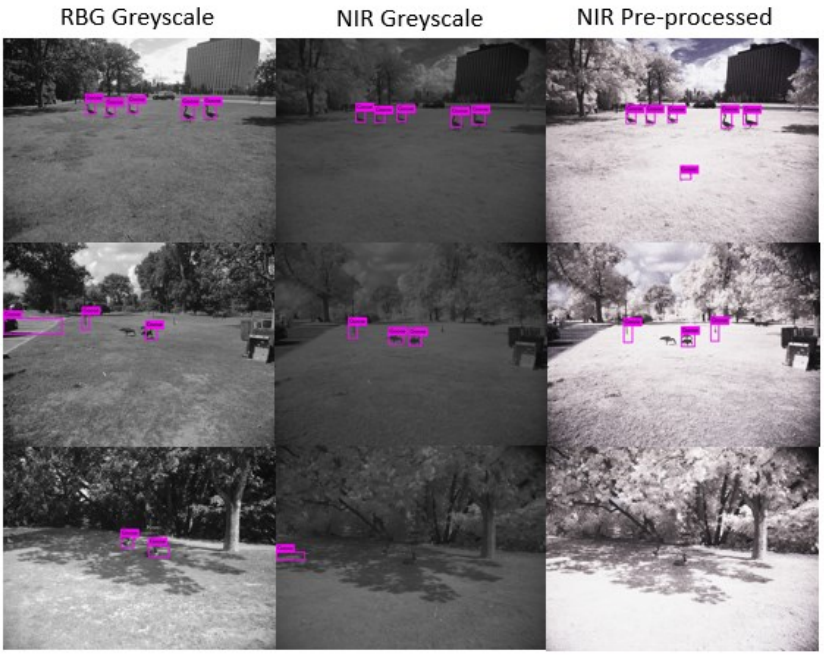
Blender Study – Greyscale trained detector using 5,000 weights



Detection confidence scoress (left to right)

42%, 88%, 96%, 71%, 69%	27%, 84%, 94%	25%
0%	0%	0%
43%	0%	0%

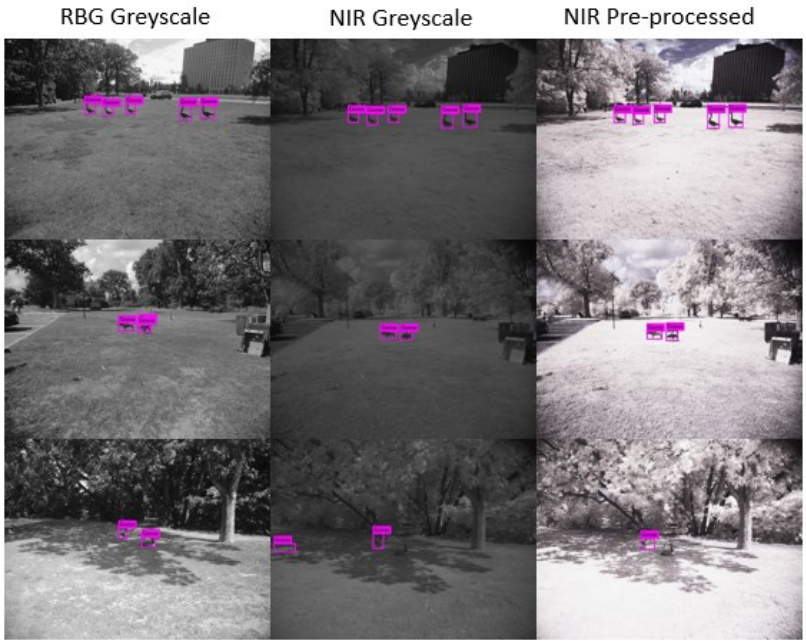
Blender Study – Augmented trained detector using 5,000 weights



Detection confidence scores (left to right)

94%, 51%, 87%, 85%, 97%	50%, 85%, 56%, 97%, 97%	89%, 96%, 88%, 34%, 91%, 92%
62%, 26%, 53%	32%, 30%, 71%	91%, 36%, 39%, 30%
37%, 80%	61%	0%

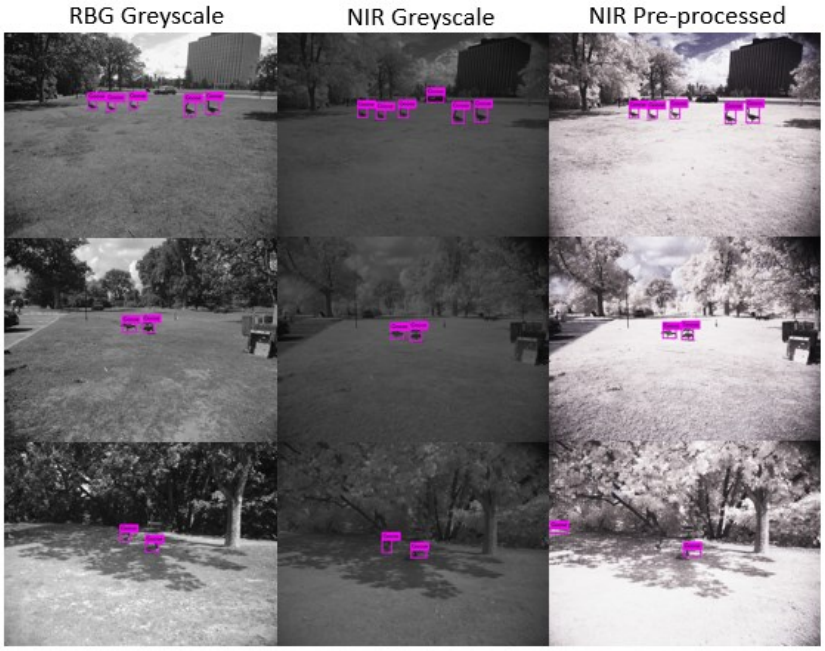
Hybrid#1 Study – Best weights



Detection confidence scores (left to right)

99%, 100%, 99%, 97%, 100%	98%, 96%, 98%, 96%, 99%	99%, 93%, 100%, 98%, 99%
99%, 96%, 56%	100%, 99%	100%, 100%
80%, 96%	33%, 27%	27%

Hybrid#2 Study – 6,000 weights



Detection confidence scores (left to right)

97%, 99%, 98%, 97%, 97%	97%, 83%, 86%, 55%, 94%, 96%	98%, 89%, 97%, 96%, 97%
95%, 77%	99%, 98%	98%, 98%
51%, 92%	53%, 26%	43%, 37%

Appendix B – MATLAB Augmentation Code

The following scripts were written in MATLAB and were used to augment the image datasets.

Greyscale Conversion

```
% This script converts an RGB image into greyscale
% Written by Jacqueline Szeto 2022

clc;
clear;
folder = dir('C:\Users\Jacqueline\OneDrive - Carleton
University\Documents\Thesis\Skylar Dataset\Additional Aerial\Full
Dataset\*.jpg');
number_of_images = numel(folder);
[~,order] = sort_nat({folder.name});
files_sorted = folder(order);
destination = 'C:\Users\Jacqueline\OneDrive - Carleton
University\Documents\Thesis\Skylar Dataset\Additional Aerial\Complete
Greyscale Dataset';
RGB_folder = 'C:\Users\Jacqueline\OneDrive - Carleton
University\Documents\Thesis\Skylar Dataset\Additional Aerial\Full Dataset';

% starting at 3 instead of 1 to offset the directory
% paths in fields 1 and 2 in the 2002x1 structure

for n = 1:number_of_images
    image = files_sorted(n).name;
    imagePath = [RGB_folder,'\', image];
    Bit_Depth = imfinfo(imagePath).BitDepth;
    RGB = imread(imagePath);
    greyscale = im2gray(RGB);
    %getting image number to match file name
    ImgNumArr = regexp(image,'\d*','match');
    ImgNumStr = char(ImgNumArr);
    imgName = [destination,'\Image',ImgNumStr,'.jpg'];
    if Bit_Depth == 8 | Bit_Depth == 24
        imwrite(greyscale,imgName);
    else
        imwrite(greyscale,imgName,'BitDepth',12);
    end
end
```

Light Contrast

```
% This script changes the input images to a lighter contrast
% Written by Jacqueline Szeto 2022

clc;
clear;
folder = dir('*.jpg');
number_of_images = numel(folder);
[~,order] = sort_nat({folder.name});
files_sorted = folder(order);
number = 1001;
```

```
for n = 1:number_of_images
    image = files_sorted(n).name;
    image_read = imread(image);
    new_image = imadjust(image_read,[0.0 0.6]);
    file_name = sprintf('%d.jpg', number);
    imwrite(new_image,file_name);
    number = number + 1;
end
```

Dark Contrast

```
% This script changes the input images to a darker contrast
% Written by Jacqueline Szeto 2022

clc;
clear;
folder = dir('*.jpg');
number_of_images = numel(folder);
[~,order] = sort_nat({folder.name});
files_sorted = folder(order);
number = 501;

for n = 1:number_of_images
    image = files_sorted(n).name;
    image_read = imread(image);
    new_image = imadjust(image_read,[0.33 1]);
    file_name = sprintf('%d.jpg', number);
    imwrite(new_image,file_name);
    number = number+1;
end
```

Gaussian Smoothing

```
% This script applies a Gaussian smoothing to the input images
% Written by Jacqueline Szeto 2022

clc;
clear;
folder = dir('*.jpg');
number_of_images = numel(folder);
[~,order] = sort_nat({folder.name});
files_sorted = folder(order);
number = 1501;

for n = 1:number_of_images
    image = files_sorted(n).name;
    new_image = imread(image);
    blurred_image = imgaussfilt(new_image,6);
    file_name = sprintf('%d.jpg', number);
    imwrite(blurred_image,file_name);
    number = number + 1;
end
```

Appendix C – YOLOv4-Tiny .cfg File

The YOLOv4-Tiny .cfg file presented in this appendix is one of the examples used for training in this study. The only parameters that varied between training were iteration, batch and subdivision. All other parameters remained the same throughout the research. Note that the parameters filters before each YOLO layer and classes were changed to allow training for one object class.

```
[net]
# Testing
#batch=1
#subdivisions=1
# Training
batch=64
subdivisions=16
width=480
height=480
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.00261
burn_in=1000

max_batches = 10000
policy=steps
steps=1600000,1800000
scales=.1,.1

#weights_reject_freq=1001
#ema_alpha=0.9998
#equidistant_point=1000
#num_sigmas_reject_badlabels=3
#badlabels_rejection_percentage=0.2

[convolutional]
batch_normalize=1
filters=32
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

[route]
layers=-1
groups=2
group_id=1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

[route]
layers = -1,-2

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky

[route]
layers = -6,-1

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[route]
layers=-1
```

```
groups=2
group_id=1

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

[route]
layers = -1,-2

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[route]
layers = -6,-1

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[route]
layers=-1
groups=2
group_id=1

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky
```

```
[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[route]
layers = -1,-2

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[route]
layers = -6,-1

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear
```

```
[yolo]
mask = 3,4,5
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=1
num=6
jitter=.3
scale_x_y = 1.05
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
ignore_thresh = .7
truth_thresh = 1
random=0
resize=1.5
nms_kind=greedynms
beta_nms=0.6
#new_coords=1
#scale_x_y = 2.0

[route]
layers = -4

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 23

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear

[yolo]
mask = 1,2,3
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=1
num=6
```

```
jitter=.3
scale_x_y = 1.05
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
ignore_thresh = .7
truth_thresh = 1
random=0
resize=1.5
nms_kind=greedynms
beta_nms=0.6
#new_coords=1
#scale_x_y = 2.0
```

References

Abdurahman, F., Fante, K. A., & Aliy, M. (2021). Malaria parasite detection in thick blood smear microscopic images using modified YOLOV3 and YOLOV4 models. *BMC bioinformatics, 22*(1), 112. doi:http://doi.org/10.1186/s12859-021-04036-4

Abujar, S., Masum, A. K., Chowdhury, S. M., Hasan, M., & Hossain, S. A. (2019). Bengali Text generation Using Bi-directional RNN. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1-5. doi:10.1109/ICCCNT45670.2019.8944784

Andres, B., Smith, P., Morrison, R., Gratto-Trevor, C., Brown, S., & Friis, C. (2012). Population estimates of North American shorebirds. *Wader Study Group Bull, 119(3)*, 178–194.

Baranwal, S., Khandelwal, S., & Arora, A. (2019). Deep Learning Convolutional Neural Network for Apple Leaves Disease Detection. *SSRN Electronic Journal*, 260-267. doi:https://doi.org/10.2139/ssrn.3351641

Barisic, A., Petric, F., & Bogdan, S. (2022). Sim2Air - Synthetic Aerial Dataset for UAV Monitoring. *IEEE Robotics and Automation Letters, 7*(2), 3757-3764. doi:10.1109/LRA.2022.3147337

Bech-Hansen, M., Kallehauge, R. M., Lauritzen, J. M., Sørensen, M. H., Jensen, L. F., Pertoldi, C., & Bruhn, D. (2020). Evaluation of disturbance effect on geese caused by an approaching unmanned aerial vehicle. *Bird Conservation International, 30*(2), 169-175. doi:10.1017/S0959270919000364

Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. *In: Montavon G., Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, 7700*, 437-478. doi:https://doi.org/10.1007/978-3-642-35289-8_26

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., . . . Ghayvat, H. (2021). CNN Variants for Computer Vision: History, Architecture. *Electronics, 10*(20), 2470. doi:https://doi.org/10.3390/electronics10202470

Bloice, M. D., Stocker, C., & Holzinger, A. (2017). Augmentor: An Image Augmentation Library for Machine Learning. *The Journal of Open Source Software, 2*(19), 432. doi:https://doi.org/10.21105/joss.00432

Bochkovskiy, A. (n.d.). *Yolo v4, v3 and v2 for Windows and Linux*. Retrieved from GitHub: https://github.com/AlexeyAB/darknet

Bochkovskiy, A., Wang, C., & & Liao, H. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv, abs/2004.10934.*

Bui, H. M., Lech, M., Cheng, E., Neville, K., & Burnett, I. S. (2016). Using grayscale images for object recognition with convolutional-recursive neural network. *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, 321-325. doi:https://doi.org/10.1109/CCE.2016.7562656

Butcher, G., Mottar, J., Parkinson, C., & Wollack, E. (2022, July 24). *Tour of the Electromagentic Specturm.* Retrieved from National Aeronautics and Space Administration: https://smd-prod.s3.amazonaws.com/science-pink/s3fs-public/atoms/files/Tour-of-the-EMS-TAGGED-v7_0.pdf

Chabot, D., & Bird, D. M. (2012). Evaluation of an off-the-shelf Unmanned Aircraft System for Surveying Flocks of Geese. *Waterbirds, 35(1)*, 170-174. doi:https://doi.org/10.1675/063.035.0119

Chabot, D., & Bird, D. M. (2013). Small unmanned aircraft: precise and convenient new tools for surveying wetlands. *Journal of Unmanned Vehicle Systems, 1(01)*, 15-24. doi:https://doi.org/10.1139/juvs-2013-0014

Dewi, C., Chen, R.-C., Yang-Ting, L., Xiaoyi, J., & Hartomo, K. D. (2021). Yolo V4 for Advanced Traffic Sign Recognition With Synthetic Training Data Generated by Various GAN. *IEEE Access*, 97228-97242. doi:10.1109/ACCESS.2021.3094201

DJI. (2022, July 23). *DJI Mavic 3*. Retrieved from DJI Store: https://store.dji.com/ca/product/dji-mavic-3?set_country=CA&vid=109821

DJI. (2022, July 25). *MAVIC 2 Specs*. Retrieved from MAVIC 2: https://www.dji.com/ca/mavic-2/info

Excell'Jets. (2022, July 23). *HOW MUCH DOES AN HOUR OF HELICOPTER FLIGHT COST?* Retrieved from Excell'Jets: https://www.excelljets.net/en/private-flights/helicopter-rental/helicopter-prices/?cn-reloaded=1

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1627-1645. doi:10.1109/TPAMI.2009.167

Friis, C. (2020). *James Bay Shorebird 2019 Report.* Environment and Climate Change Canada.

Gabor, I. (2018, May 9). *Raw2dng*. Retrieved from https://app.box.com/s/gfahlmla4kznzbzc6zafgl0s9p6mwuil

Groß, B., Kreutzer, M., Durand, T., Reimann, R., Porada, F., & Rittmeister, N. (n.d.). *Opendatacam*. Retrieved from An open source tool to quantify the world: https://opendata.cam/

Gupta, M. M., Liang, J., & Noriyasu, H. (2003). *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Iron, J. (2018). *James Bay Shorebird Project 2018 - Reports*. Retrieved from Jean Iron Photos: http://jeaniron.ca/2018/JB18/p10.htm

ITM Instruments Inc. (2022, July 23). *FLIR Scion OTM Outdoor Thermal Monocular, 24 x 18°, 640 x 512*. Retrieved from https://www.itm.com/product/flir-scion-otm260-outdoor-thermal-monocular-7tm-01-f130?ksearch_click=FLIR%2BScion

Johnsen, S. (2012). *The Optics of Life: a Biologist's Guide to Light in Nature.* Princeton University Press.

Karasawa, T., Watanabe, K., Ha, Q., Tejero-De-Pablos, A., Ushiku, Y., & Harada, T. (2017). Multispectral Object Detection for Autonomous Vehicles Takumi. *Thematic Workshops '17: Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 35-43. doi:https://doi.org/10.1145/3126686.3126727

Kayalibay, B., Jensen, G., & van der Smagt, P. (2017). CNN-based Segmentation of Medical Imaging Data. *ArXiv, abs/1701.03056*.

Kellenberger, B., Veen, T., Folmer, E., & Tuia, D. (2021). 21 000 birds in 4.5 h: efficient large-scale seabird detection with machine learning. *Remote Sensing in Ecology and Conservation, 7*(3), 445-460. doi:10.1002/rse2.200

Kozák, J. (2011). An overview of feathers formation, moults and down production in geese. *Asian-Australasian Journal of Animal Sciences, 24*(6), 881-887. doi:http://doi.org/10.5713/ajas.2011.10325

Linchant, J., Lisein, J., Semeki, J., Lejeune, P., & Vermeulen, C. (2015). A review of UASs in wildlife monitoring. *Mammal Review, 45*, 239-252. doi:https://doi.org/10.1111/mam.12046

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision, 2*, 1150-1157. doi:https://doi.org/10.1109/ICCV.1999.790410

Medina, I., Newton, E., Kearney, M. R., Mulder, R. A., Porter, W. P., & Stuart-Fox, D. (2018). Reflection of near-infrared light confers thermal protection in birds. *Nature Communications, 9*(3610). doi:https://doi.org/10.1038/s41467-018-05898-8

Murray, N., Marra, P., Fuller, R., Clemens, R., Dhanjal-Adams, K., Gosbell, K., . . . Studds, C. (2018). The large-scale drivers of population declines in a long-distance migratory shorebird. *Ecography, 41*, 867-876. doi:https://doi.org/10.1111/ecog.02957

Neapolitan, R. E., & Jiang, X. (2018). *Artificial Intelligence With an Introduction to Machine Learning* (2nd Edition ed.). New York: Chapman and Hall/CRC. doi:https://doi-org.proxy.library.carleton.ca/10.1201/b22400

Ng, C. B., Tay, Y. H., & Goi, B. M. (2013). Comparing image representations for training a convolutional neural network to classify gender. *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation*, 29-33. doi:10.1109/AIMS.2013.13

Nixon, M. S., & Aguado, A. S. (2012). *Feature Extraction & Image Processing for Computer Vision* (3rd ed ed.). Amsterdam: Elsevier/Academic Press.

Norward Expert LCC. (2017). *DETAILED SPECIFICATION - AgroCam NIR+blue lens*. Retrieved from AgroCam: https://www.agrocam.eu/specification-nir-blue-lens

Norward Expert LCC. (2017). *DETAILED SPECIFICATION - AgroCam NIR+blue lens*. Retrieved from AgroCam: https://www.agrocam.eu/specification-nir-blue-lens

Norward Expert LLC. (2017). *AgroCam Geo NIR + AgroCam Geo RGB (dual camera NDVI)*. Retrieved from AgroCam: https://www.agrocam.eu/product-page/agrocam-geo-nir-agrocam-geo-rgb-dual-camera-ndvi

NVIDIA Corporation. (n.d.). *Jetson Nano Developer Kit*. Retrieved from NVIDIA Developer: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

Ooi, B. X.-Z. (2019, December). *Development of a Computer Vision Framework for Improved Remotely Piloted Aircraft Operations*. [Master's Thesis, Carleton University]. doi:https://doi.org/10.22215/etd/2020-13960

Pascal2. (n.d.). *The PASCAL VOC project*. Retrieved from The PASCAL Visual Object Classes Homepage: http://host.robots.ox.ac.uk:8080/pascal/VOC/

Piersma, T., & Lindström, Å. (2004). Migrating shorebirds as integrative sentinels of global environmental change. *Ibis, 146*(s1), 61-69. doi:https://doi.org/10.1111/j.1474-919X.2004.00329.x

Prashanth, D. S., Mehta, R. V., & Sharma, N. (2020). Classification of Handwritten Devanagari Number – An analysis of Pattern Recognition Tool using Neural Network and CNN. *Procedia Computer Science, 167*, 2445-2457. doi:https://doi.org/10.1016/j.procs.2020.03.297

Radiuk, P. M. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science, 20*(1), 20-24. doi:https://doi.org/10.1515/itms-2017-0003

RawTherapee. (n.d.). Retrieved from https://www.rawtherapee.com/

Redekop, E., & Chernyavskiy, A. (2021). Medical Image Segmentation with Imperfect 3D Bounding Boxes. *In: Engelhardt S. et al. (eds) Deep Generative Models, and Data Augmentation, Labelling, and Imperfections. DGM4MICCAI 2021, DALI 2021. Lecture Notes in Computer Science, 13003*, 193-200. doi:https://doi-org.proxy.library.carleton.ca/10.1007/978-3-030-88210-5_18

Redmon, J. (2013-2016). *Darknet: Open Source Neural Networks in C*. Retrieved from http://pjreddie.com/darknet

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement.* ArXiv, abs/1804.02767.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564-2571. doi:10.1109/ICCV.2011.6126544

Russakovsky, O., Deng, J., Hao, S., Krause, J., Satheesh, S., Ma, S., . . . Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis, 115*, 211–252 (. doi:https://doi.org/10.1007/s11263-015-0816-y

Santoso, R., Suprapto, Y. K., & Yuniarno, E. M. (2020). Kawi Character Recognition on Copper Inscription Using YOLO Object Detection. *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, 343-348. doi:10.1109/CENIM51130.2020.9297873

Shijie, J., Wang, P., Peiyi, J., & Siping, H. (2017). Research on data augmentation for image classification based on convolution neural networks. *2017 Chinese Automation Congress (CAC)*, 4165-4170. doi:https://doi.org/10.1109/CAC.2017.8243510.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data 6, 60*. doi:https://doi.org/10.1186/s40537-019-0197-0

Skalski, P. (2019). *Make Sense*. Retrieved from https://www.makesense.ai/

Stuart-Fox, D., Newton, E., Mulder, R. A., D'Alba, L., Shawkey, M. D., & Igic, B. (2018). The microstructure of white feathers predicts their visible and near-infrared reflectance properties. *PLoS ONE, 13*(7), 1-14. doi:http://doi.org/10.1371/journal.pone.0199129

Su, H., Deng, J., & Fei-Fei, L. (2012). Crowdsourcing Annotations for Visual Object Detection. *HCOMP@AAAI, WS-12-08*, 40-46.

Szeto, J., Steele, A., & Laliberte, J. (2021). Near infrared imaging and image pre-processing to improve the automatic detection of Canada Geese. *2021 Aerial Evolution Association of Canada's Student Paper Competition.*

Tkalcic, M., & Tasic, J. F. (2003). Colour spaces: perceptual, historical and applicational background. *The IEEE Region 8 EUROCON 2003. Computer as a Tool., 1*, 304-308. doi:10.1109/EURCON.2003.1248032

Transport Canada. (2021, February 2). *Find your category of drone operation*. Retrieved November 2021, from Government of Canada: https://tc.canada.ca/en/aviation/drone-safety/learn-rules-you-fly-your-drone/find-your-category-drone-operation

Walsh, M. (2020). *Artificial Intelligence Based Monitoring System.* [Internal Document, Carleton Univeristy].

Walsh, M. (2020). *Jetson Nano setup and Opendatacam Installation.* [Internal Document, Carleton Univeristy].

Wang, C. Y., Mark Liao, H. Y., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 390-391.

Wang, J., & Xia, B. (2021). Bounding Box Tightness Prior for Weakly Supervised Image Segmentation. *In: de Bruijne M. et al. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. MICCAI 2021. Lecture Notes in Computer Science, 12902*, 526-536. doi:https://doi-org.proxy.library.carleton.ca/10.1007/978-3-030-87196-3_49

Zhou, S., Bi, Y., Wei, X., Liu, J., Feng, L., & Yuchuan, D. (2021). Automated detection and classification of spilled loads on freeways based on improved YOLO network. *Machine Vision and Applications*, 32-44. doi:https://doi.org/10.1007/s00138-021-01171-z