January 2018

Using Citizen Scientists To Inform Machine Learning Algorithms To Automate The Detection Of Species In Ecological Imagery

Marshall Mattingly

Using Citizen Scientists to Inform Machine Learning Algorithms to
Automate the Detection of Species in Ecological Imagery

by

Marshall Paul Mattingly III
Bachelor of Science, University of North Dakota, 2015

A thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

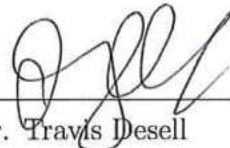in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota
May
2018
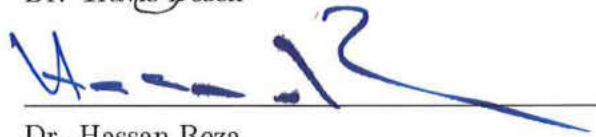
This thesis, submitted by Marshall Paul Mattingly III in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work as been done and is hereby approved.

_____
Dr. Travis Desell

_____
Dr. Hassan Reza

_____
Dr. Susan Ellis-Felege

_____

This thesis is being submitted by the appointed advisory committee as having met all of the requirments of the School of Graduate Studies at the University of North Dakota and is hereby approved.

_____
Grant McGimpsey
Dean of the School of Graduate Studies

May 1, 2018
_____
Date

PERMISSION

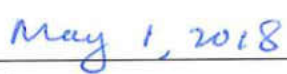Title   Using Citizen Scientists to Inform Machine Learning Algorithms to Automate the Detection of Species in Ecological Imagery

Department   Department of Computer Science

Degree   Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in his absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

<div align="right">

Marshall Paul
Mattingly III
May 2018

</div>

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Dedicated to my wife, Shaina, who encourages me to chase my dreams.
Her unconditional support drives me every day.

ABSTRACT

Modern data collection techniques used by ecologists has created a deluge of data that is becoming increasingly difficult to store, filter, and analyze in an efficient and timely manner. In just two summers, over 65,000 unmanned aerial system (UAS) images were collected, comprising the several terabytes (TB) of data that was reviewed by citizen scientists to generate inputs for machine learning algorithms. Uncontrolled conditions and the small size of target species relative to the background further increase the difficulty of manually cataloging the images. To assist with locating and identifying snow geese in the UAS images, a citizen science web portal was created as part of Wildlife@Home. It is demonstrated that aggregate citizen scientist observations are similar in quality to observations made by trained experts and can be used to train convolutional neural networks (CNN) to automate the detection of species in the imagery. Using a dataset comprising of the aggregate observations produces consistently better results than datasets consisting of observations from a single altitude, indicating that more numerous but slightly variable observations is preferable to more consistent but less numerous observations. The framework developed requires system administrators to manually run scripts to populate the database with new images; however, this framework can be extended to allow researchers to create their own projects, upload new images, and download data for CNN training.

CHAPTER 1

INTRODUCTION

Ecologists today are able to collect significant amounts of data, especially through the use of unmanned aerial systems (UAS). From 2007-2010, there were a handful of UAS projects related to evironmental sciences, ecology, and wildlife; now more than 5% of all publications in top journals related to those fields utlize UAS [1]. Storing, filtering, and analyzing this deluge of data in an efficient and timely manner is becoming increasingly difficult. In just two summers of UAS data collection, over 65,000 UAS images comprising over 3 TB (terabytes) of imagery has been generated. Manually examining this vast quantity of data for ecological species and events is a laborious task which must be overcome before any meaningful scientific conclusions can be drawn.

To aid wildlife ecologists in the cataloging of ecological video and images, an online hub called Wildlife@Home [1] was created as part of the Citizen Science Grid [2] at the University of North Dakota (UND). Collectively, the Wildlife@Home projects comprise more than 100,000 hours of video footage from nest cameras, 1,800,000 images from trail cameras, and 65,000 images from UAS flights taken at three different altitude: 75m, 100m, and 120m. The volunteers who review the images and video on Wildelife@Home are known as citizen scientist.

Placing a project on Wildlife@Home allows exponentially more observations to be made than in traditional small lab setting. It must first be shown that citizen scientists can make reliable ecological observations that can be used to draw sound scientific conclusions, even though only 1 out of 49 users in a voluntary survey on Wildlife@Home identified as having a wildlife biology or

---

[1]https://csgrid.org/csg/wildlife/

ecology background. Prior work has shown that citizen scientists on Wildlife@Home produce actionable results for both video footage [3, 4] and imagery [5, 6]. The end goal of the manual object detection is to use the observations, primarily from citizen scientists, as the inputs to image and video classification algorithms that can detect a majority of the targets in future imagery and video, respectively. This paper is focused on the detection of lesser snow geese (*Anser caerulescens caerulescens*), hereafter called snow geese (singular: snow goose), in UAS imagery from the Hudson Bay area of Canada, and further discussion of object detection will be on that specific topic unless otherwise specified. The snow geese come in two phases: white and blue, with a respective population proportion ratio of 2:1.

Automating the detection of snow geese from the same general geographic location has significant challenges even with a robust, representative set of manually detected observations. The first significant challenge is the relatively small area (18×18 pixels) of the snow geese when compared against the size of a typical UAS image (844×755 pixels to 2000×3000 pixels). This relatively small target object size is further exacerbated by the relative infrequency of snow geese within the imagery, with less than 1% of the imagery, by area, comprising of snow geese. The second major challenge is cryptic coloration, whereby the snow geese blend into or appear similar to other objects in the background. White phase snow geese appear remarkably similar to rocks within the landscape, and blue phase snow geese naturally look similar to the landscape.

The image classification technique used on in this research is convolutional neural networks (CNNs), which use manually labeled input data (generally of the same dimensions) to train a set of weights that can be used to detect objects of each target label in other images. CNNs have proven to be an effective, accurate method for image classification in other projects, such as the Modified National Institute of Standards and Technology (MNIST) handwritten digit

dataset [7–11]. CNNs were choses as the image classification technique for use on this project due to familiarity of the technique and the demonstrated accuracy of CNNs on other image classification projects.

A primary goal of this project is to identify, document, and work toward creating a general interface and workflow that can be utilized to create a citizen science web portal. The web portal should be able to be setup by a wildlife ecologist (or other researcher) on their own with a little help from their technical support team. This generalizable interface with support for templates, multiple database backends, and simple spawning of CNN tasks has not been fully developed. The requirements have been explored, Unified Modeling Language (UML) diagrams have been created to show some aspects of the system, and many portions of the system have been coded, including the scripts and web pages to ingest new images and produce training data. The current image review interface code is tightly coupled to Wildlife@Home and lacks an all-encompassing project management interface.

CNNs are only capable of automating object detection within images if the input data is truly representative of the target objects and have an abundance of examples from which to train. Considering the vast amount of time it takes a single person to go through millions of images, a web interface was designed to allow multiple people to go through the same images, labeling all objects within the image [5]. However, there is no guarantee that the objects detected by the citizen scientists are representative of the target objects. To verify that the citizen scientists inputs are representative enough to be used in CNN training, the observations of citizen scientist were compared against observations made by a trained expert. It was shown that aggregate citizen scientist observations, known as matched observations, are extremely similar to trained expert observations [5]. This research verifies that the prior results hold true with a much larger dataset.

Prior work on the UAS image dataset aggregated the different image altitudes – 75m, 100m, and 120m – into a single dataset. CNNs were trained and tested using images from all altitudes, instead of looking at each altitude individually [6]. To expand upon the prior work, individual CNNs were trained and tested for each separate altitude, with the datasets only coming from the respective altitudes. Comparing the more precise data from each individual altitude against the more abundant aggregate dataset can lead to a discussion of which is more important for CNNs used in this research.

This paper is organized into the following sections: Chapter 2 introduces the background knowledge required for understanding this paper and presents related works that focus on citizen science and automated object detection within wildlife ecology. Chapter 3 details how the UAS data were collected, how the Wildlife@Home interface allows citizen scientists to label objects within the images, and the specific collections of data used for training the CNNs. Chapter 4 details the methodology describing the components of the image review interface and algorithms used to extract objects from the observations made by citizen scientists. Chapter 5 compares the expert and citizen scientists observations to examine the viability of citizen scientist to generate data that is good enough for input into the CNNs; compares a technique of using individual CNNs for each altitude (75m, 100m, and 120m) instead of an aggregate CNN that merges the altitudes into a single dataset, such as in prior work; and examines the scalability of the CNN code used in this and prior work. Finally, Chapter 6 refines the results of this paper and discusses future work from this research.

CHAPTER 2

RELATED WORK

Citizen Science Projects

Citizen science is a modern technique whereby the general population can help scientists process and catalog data. Since 2011, there has been a surge in the number of citizen science projects, such as PlanetHunters [12], which allows citizen scientists to identify planet candidates from the NASA Kepler public release data. However, there are few projects using citizen science within avian ecology. NestWatch [13] was a Cornell project that opportunistically captures a variety of cavity-nesting species from video cameras installed in bird houses. CamClickr [14] was used to catalog nesting behavior in over 600,000 images, and was incorporated into a university biology curriculum to teach students how to accurately identify objects in images while being aware of potential observer biases [14]. eBird [15] provided spatio-temporal information about bird distribution and abundance by allowing users to upload user-taken images of bird observations through handheld devices. Data from eBird was compared against formal surveys, and showed that the opportunistic data gathered by citizen scientists differed by only 0.4% / year [16], allowing for citizen scientist inputs to inform avian ecological finding.

In fact, the surge of citizen science projects is large enough that a platform, called Zooniverse [17], was developed to allow for the rapid creation of citizen science projects. There were over 20 citizen science projects hosted by Zooniverse at the start of 2014, which has grown to 50 active citizen science projects in 2018. The best performing projects on Zooniverse are well-established projects relating to astronomy [18]. A couple of projects hosted on Zooniverse include

GalaxyZoo [19, 20], which allows citizen scientists to classify galaxies in images from the Sloan Digital Sky Survey [21], and Snapshot Serengeti [22], which allows citizen scientists to identify objects from camera traps in the Serengeti National Park. This research is differentiated from Zooniverse by the specificity of input; Zooniverse allows for basic questions about an image to be answered, such as the total number of a target species in the image, while this research allows boxes to be drawn over the target species in the image. This specificity is what allows the CNNs to be trained, while the ability to rapidly get a project into the hands of citizen scientists on Zooniverse is useful for getting aggregate answers quickly.

A major problem with citizen science is incentivizing the citizen scientists to provide accurate observations to ensure field scientists can draw correct conclusions. Gamification [23] is a technique that can be used to encourage citizen scientists to provide accurate observations. Gamification gives a small amount of points to citizen scientists for making observations, but a large amount of points when those observations can be matched to an expert observation (or several other citizen scientists). After the implementation of gamification on the UAS images project on Wildlife@Home, there were four (4) citizen scientists who completed over four (4) full mosaics, when compared against zero (0) citizen scientists who had completed even a single full mosaic.

Object Detection Techniques

CNNs are widely used for image classification. CNNs have a set of weights and biases which are used, along with the CNN architecture, to generate outputs, such as classifications of an image. Finding the best CNN architecture can be challenging, as some CNN architectures are more suited to a task than others. After an architecture has been decided, individual weights get trained using input classifications.

The architecture of CNNs typically consist of many "hidden" layers stacked between a single input layer and a single output layer, such that the output values of one hidden layer are the input values of the next layer in the stack until the final output layer is reached. Each hidden layer is comprised of feature maps and filters which process the data. There are several types of hidden layers, some of which contain weights associated with each feature map, while others contain no weights.

It has been shown that CNNs can be used to classify objects within the aggregate UAS imagery dataset [24]. The exact same CNN code and architecture is being used to train and run the CNNs on the different altitude datasets within this research to compare against the aggregate altitude dataset. The CNN architecture for this research is shown in Fig. 1 and Table 1.
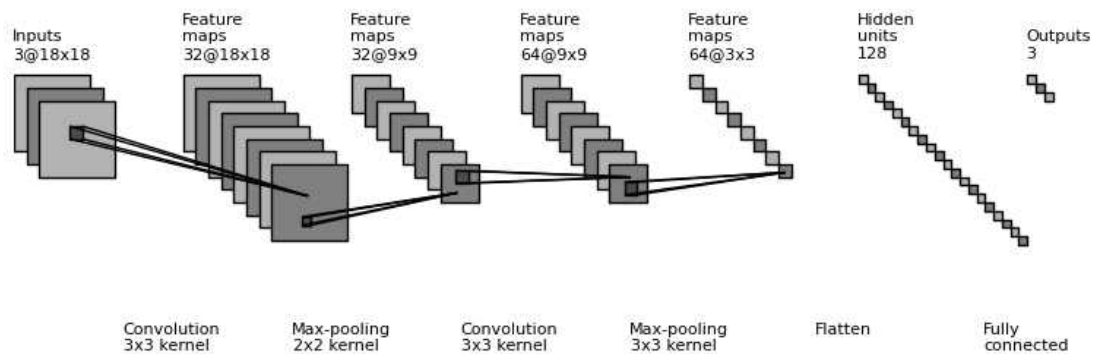
Figure 1: Visual representation of the architecture used for the CNNs in this paper for comparison with CNNs ran over the aggregate dataset combining altitudes.

Object detection in both video and still imagery is a rapidly advancing topic in computer science, with many different challenges and datasets used to validate and compare techniques amongst each other. The ImageNet Large Scale Visual Recognition Competition is a collection of image and object classification challenges, with two of the challenges in the past few years focusing on object detection in images with 200 fully labeled classes and object detection within videos of 30 fully labeled classes [25]. There are a multitude of techniques to perform the object detection, even when using CNNs as the baseline, with two

Table 1: Description of the architecture used for the CNNs in this research for comparison with CNNs run over the aggregate dataset combining altitudes.

Layer Type	Layer Dims	Filter / Pool Size	Stride	Filters	Padding
Input	18 x 18 x 3				
Convolutional	18 x 18 x 32	3	1	32	1
Max Pooling	9 x 9 x 32	2	2		
Convolutional	9 x 9 x 64	3	1	64	1
Max Pooling	3 x 3 x 64	3	3		
Fully Connected	1 x 1 x 128			128	
Fully Connected	1 x 1 x 3			3	

prominent techniques being region-based CNNs and whole-image CNNs.

Region-based CNNs, such as R-CNN [26], Fast R-CNN [27], and Faster R-CNN [28], attempt to identify areas of interest, known as regions, that are then run through the CNN. A region of interest (RoI) pooling layer maps a set of features from the variable sized RoI onto a fixed size feature map, with different RoI's sharing computations and memory where applicable to minimize runtime and storage requirements. Faster R-CNN greatly decreases the computational requirements of generating the RoI's by implementing a Region Proposal network.

Whole-image CNNs, such as You Only Look Once (YOLO) [29] and its refinement YOLOv2 [30], simply run the entire image through CNNs without any preprocessed RoI detection. The input images are split into many same-size regions and run through the CNN, with a bounding box and probability of detection produced for each region. This allows the CNN to be completed on a single pass, whereas region-based CNNs may run many sub-images from the same whole-image through the CNN with significant overlap.

Object Detection in Wildlife Ecology

A similar project by Xu and Zhu [31] sought to automate the classification of six (6) species of seabirds on complex and uncontrolled backgrounds. The image is segmented using Grabcut [32], which attempts to roughly identify all objects similar to the user-specified target object within the image. The segmented objects are then extracted and run through three (3) different models: k-Nearest Neighbor [33], Logistic Boost [34, 35], and Random Forest [36]. The outputs of these models are then used to vote on the final classification of the objects within the image. This non-CNN technique resulted in a recognition accuracy of 88.1% when run over 900 samples.

Villa et al. [37] trained CNNs over the data gathered from the Snapshot Serengeti project using four (4) datasets: *(i)* a raw unbalanced dataset, *(ii)* a raw balanced dataset, *(iii)* a balanced dataset that only includes animals that are present in the foreground of an image, and *(iv)* a dataset that includes segmented images that contained part (or parts) of an animal within the segments. Multiple CNN architectures were trained and compared on each of the four datasets. The CNNs trained on the unbalanced dataset were the worst and the best results were on the segmented dataset, with an accuracy of 58% and 88.9%, respectively.

Non-CNN object detection methods were used on mosaic UAS imagery of white-tailed deer with both visible (RGB) and thermal infrared (TIR) spectra [38]. Supervised and un-supervised pixel-based detection, which are extremely basic object detection methods, were both unsuccessful on the imagery. Object-based image analysis (OBIA), however, produced no false-positives while matching the 50% detection rate of manned aerial surveys on the TIR imagery [38]. OBIA on the RGB images had an extreme number of false positives, with 1,946 deer detected in an image with only 4 actual deer present, as an example [38]. This demonstrates that using a different spectrum,

such as TIR instead of RGB, on the same images using the same object detection methods can potentially improve results.

Feature-based analysis is an object detection method that is faster than CNNs and could potentially be used to pre-process a dataset to highlight interesting images for manual inspection. This would dramatically reduce the effort of citizen scientists, as many images that are obviously background-only with no target species can be skipped. Feature-based analysis using two features, color and shape, was used to detect birds from video recorded during UAS flights [39]. The omission and commission rates were less than 20% each [39], which gives credence to using feature-based analysis to preprocess UAS images before showing them to citizen scientists.

CHAPTER 3

WILDLIFE@HOME IMAGE DATASET

Data Gathering

The UAS imagery used in this project was collected using a Trimble UX5 [1] fixed
wing UAS (Fig. 2(a)). The images were collected in Wapusk National Park in
Manitoba, Canada in 2015 and 2016, and produced over 65,000 images (over 3
TB of data) in total. Flights occurred during the snow goose nesting season and
were flown at altitudes of 75m, 100m, and 120m above ground level. A 16
megapixel Sony camera placed in the nadir position recorded the images with an
80% overlap between consecutive images. Figure 2(b) is example of the flight
path over an area.

(a) UAS takeoff
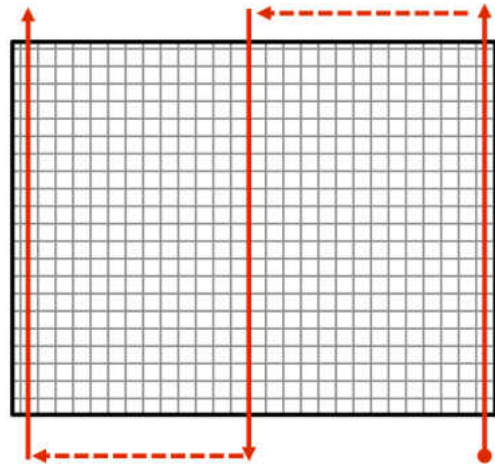
(b) UAS flight path

Figure 2: An example of the UAS takeoff procedure used by the Trimple UX5
and an example flight path that is flown using transects. Photo credit Susan
Ellis-Felege.

The images taken were then used to create mosaics for each flight. The

---

[1]http://uas.trimble.com/ux5

Trimble Business Center [2] (version 3.51) was used for the 2015 data and Pix4D [3] (version 3.2.23) was used for the 2016 data. In total, 36 distinct mosaics were created, each of which were several GB large and totaled over 50GB of mosaic data. Each mosaic was then split down into mosaic split images (MSIs) that could be shown to experts and citizen scientists through a web portal. From the 36 mosaics, 8,759 MSIs were created.

Wildlife@Home

Wildlife@Home is a collection of citizen science projects hosted on the UND Citizen Science Grid [2]. Users are able to volunteer time and computing resources by manually classifying objects within images or events within video, and by allowing a Berkeley Open Infrastructure for Network Computing (BOINC) [40] client to run on their idle computer to help with classification. BOINC has been used by Wildlife@Home to run background subtraction algorithms over their entire video dataset [41] and to train the Evolutionary eXploration of Augmenting Convolutional Topologies (EXACT) algorithm [10]. In the future, BOINC could be used to train and test the CNNs for object detection.

The data source from Wildlife@Home used in this research is the UAS imagery, as described in I. Citizen scientists are able to place bounding boxes around objects, indicating the exact location and label of all objects in the image. Target species objects in the UAS imagery make up less than 1% of the area in the UAS images, making manual detection particularly challenging. This is especially true for blue phase snow geese which naturally blend with the majority of the background.

---

[2]http://www.trimble.com/Survey/trimble-business-center.aspx
[3]https://pix4d.com/

Ecological Importance

All UAS images in this research were taken in Wapusk National Park in Manitoba, Canada, where there is an overabundance of snow geese causing widespread destruction to the habitat, with the reduction in snow goose population being considered the best method for recovery [42]. Currently, ground counts are used to estimate the snow goose population, whereby ecologists count the number of snow goose nests within a 50-meter radius each year. This methodology provides an estimate of the total snow goose population, but due to limited spacial coverage may not be representative of the actual density of the expanding population of snow geese. To cover a larger area in a more reasonable timeframe, UAS are flown to collect imagery which can then be either manually labeled or used with image classification algorithms.

It is also possible to do manned flights whereby the pilot and/or other surveyors in the plane attempt to count the geese directly. This allows for a greater area coverage, but is more costly and comes with more safety risks than UAS flights given the relatively low altitude and speed with which the plane needs to fly to allow accurate spotting of snow geese [43, 44].

User Interface

The Wildlife@Home review image interface (Fig. 3) allows users, which are made up of both citizen scientists and pre-determined experts, to go through a collection of image projects to make observations. Users are shown an image from a selected project and instructed to draw a bounding box around all target species for that project while minimizing the amount of excess background. After a bounding box is drawn, a corresponding observation entry is generated on the left-hand side of the interface where the user can label the observation with the help of example images included in the interface documentation. If there are no

target species in the image, the user can select "There's Nothing Here". The observations are then recorded to a database when the user presses "Submit".
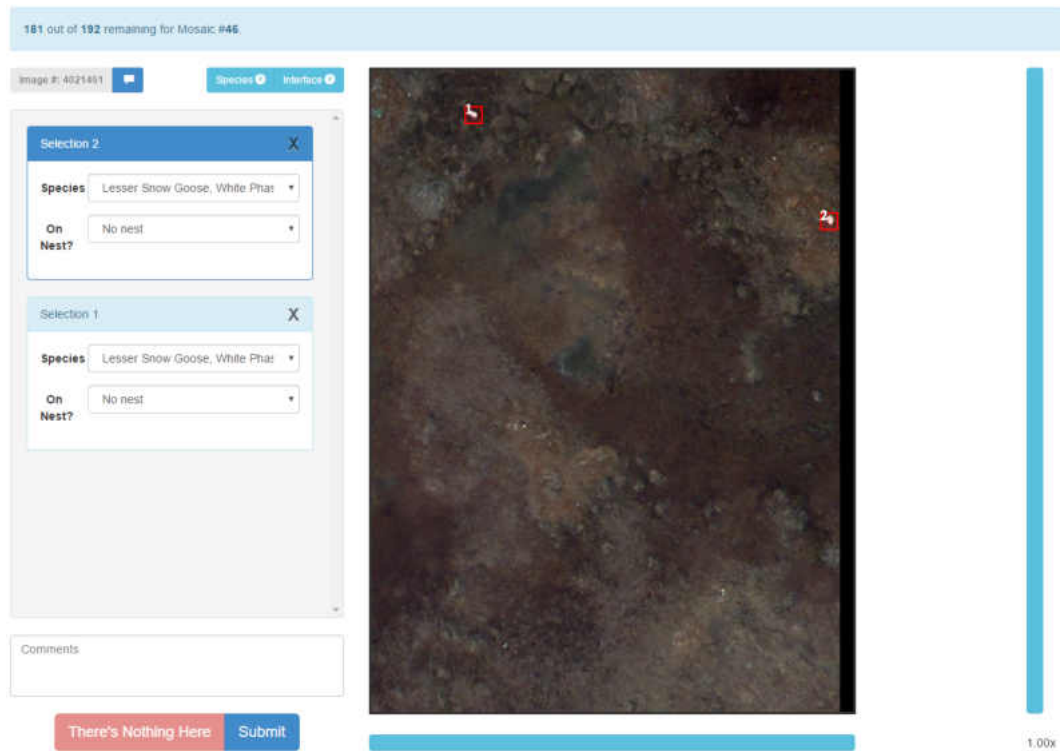
Figure 3: The graphical user interface (GUI) of the web portal for identifying objects in ecological imagery for the Wildlife@Home projects. This screenshot shows a UAS image with two white snow geese identified by the user.

Observations made using the web portal are either designated expert or unmatched. Expert observations are assumed to be true, good data (though there is still potential for human error) that can be used for the inputs to CNNs, while unmatched observations are the individual citizen scientists observations that come with a degree of uncertainty. To increase confidence in the unmatched data, matching algorithms were developed and compared, finally selecting on the 10-pixel corner point method from prior work [5]. This matches two unmatched observations into a single matched observation, with the intersecting area being used as the extents of the matched observation [5].

Storing the Data

All observations, including bounding boxes and labels, are stored in a MySQL database. The user observations are given one of three (3) designations:

1. Expert - if the recording user is a trained expert. Observations made by experts are presumed true and used to evaluate citizen scientists observations.

2. Unmatched - if the recording user is a citizen scientist. Individual observations made by citizen scientists come with no assumptions of quality, and has been shown to come with wide variability and limited quality [5].

3. Matched - if two citizen scientist observations are matched, the intersection of their bounding boxes is considered a matched observation. This has been shown to produce results with quality rivaling expert observations [5].

Considering the already demonstrated variable quality of the unmatched observations [5], only expert and matched observations are used for this research.

Technical Issues and Corrections

There was a mechanical error with the RGB camera used to capture images during the 2015 data collection that resulted in the images having a significant blue-shift. Instead of considering the 2015 and 2016 data as separate, thus decreasing the overall size of the dataset, the 2015 images were normalized against the 2016 images by finding the relative ratio of the average color of a white snow goose in each dataset. The red, green, and blue channels of all images in the 2015 dataset where multiplied by 233.0/150.0, 255.0/189.0, and 236.0/190.0, respectively. Each channel was rounded down to the nearest integer

15

and capped at a maximum value of 255. Manual inspection of the normalized 2015 images appears to be correct (Fig. 4), and the CNNs were able to train well on the combined and shifted dataset.

(a) Example original blue-shifted image from the 2015 dataset

(b) The same image from the 2015 dataset after normalization to the 2016 dataset

(c) Example image from the 2016 dataset

Figure 4: An example of the blue-shift error in an image from the 2015 dataset compared against its normalized image and an image from the 2016 dataset.

CHAPTER 4

METHODOLOGY

Prior to development of the web portal, basic software engineering practices were utilized to ensure that the product being developed matched the needs of the client, in this case a wildlife biologist. A software requirements document was written to discover the functional and non-functional requirements of the web portal. A software design document was generated from the software requirements, which included use case, activity, and class diagrams to inform the development of the web portal. Included within this methodology are descriptions and/or UML diagrams for: (i) the web portal for image detections; (ii) the database to store the information collected; (iii) the generation of training and testing data sets; and (iv) the the high performance neural network training.

Database Schema

The basic idea is to have the mosaic images related to a single project for filtering purposes (Fig. 5). The mosaic images are then split into smaller sub-images, as individual mosaics are tens of thousands of pixels in each dimension and up to several gigabytes large — much too large to download, let alone view, in a reasonable time. The split images are referenced to a generic image type that is shared with other projects on Wildlife@Home and stores the information about the file location of the image on network storage.

Image observations made by users are stored in a table relating to the generic image to which it relates, and includes whether the observation was made by an expert — expert information is defined in another database for security purposes. A single image observation is tied to 0 (in the case of **nothing_here**)

or more image observation boxes, which store the x, y, width, height, and species information of the observations.

Using this schema, joins can be used to gather all the information required to extract observations from specific projects or mosaics for both experts and citizen scientists. These observations can then be used to run the matching algorithms to generate all the data files required for training and to place copies of the corresponding images in the datasets into a folder for download.

Image Review Interface

The image review interface is the web-based application on which users make observations on the images. Since the mosaic images are split into hundreds or thousands of sub-images, the logic to determine the next image to present to the user is relatively complex. If the user has not started reviewing any mosaics for the project, a new mosaic in the project is randomly selected, and the user is shown the first sub-image from the mosaic. If the user has started a mosaic in the project, the next sub-image for the mosaic is shown. If there are no more sub-images for the started mosaic, the user has completed the mosaic and is shown a completion screen that displays how many images were reviewed and how many observations of each specific type were made.

Generating Training Data

Accessing the observation data directly from the server during training, testing, and validation is infeasible because the data in the database can change between CNN training iterations. To overcome this obstacle, the required data for training is stored in files that can be used off-line for training, testing, and validating. This is especially important to test how changes to the CNN parameters and algorithm can alter the results of a given dataset.

The IDX (non-acronym) file format, which is a simple file format for vectors and multidimensional matrices of various numerical types, is used to store the individual observations, including the label of the observation. This file format was chosen because it is used in the MNIST dataset [9], which is also used to test the CNNs developed for this research. This meant that no specialized file format needed to be developed to process observations and maximizes the ability of other researchers to compare results using the datasets produced in this research.

The byte-by-byte representation of an IDX file is shown in Table 2, which is encoded with the most-significant byte (MSB) first. The first two bytes are always 0. The third byte is a magic byte that informs the data type of every element in the IDX file (byte, short, int, float, and double are the options, with predefined byte widths). The fourth byte holds the number of dimensions ($1 \leq N \leq 255$). Then there are $4 \times N$ bytes that hold the width of each dimension $n \; \epsilon \; N$ as a 4-byte integer. Following the widths is the actual data, with $\prod_{n=1}^{N} W(n)$ total elements, where $W(n)$ is the width of the nth dimension.

Table 2: IDX file format description, showing the byte-by-byte values at each location, which are always encoded with most-significant byte (MSB) first.

Byte #	Width	Description
0	1	0
1	1	0
2	1	Data type (w)
3	1	Number of dimensions (N)
$4 + (n \times 4)$	4	Width of n ($W(n)$)
$4 \times (N + 1)$	w	Start of data

A custom binary format (BIN), known henceforth as a location file, was created to store more information about individual observations, as described in Table 3. The first value in the location file is an integer with the total number of MSI entries in the file. Each MSI entry is comprised of an integer for the MSI number and the total number of observations within the MSI. Each observation

entry is comprised of an integer for the species ID, top-left x pixel, top-left y pixel, width, height, and a hash to identify the user who made the observation, in that order.

Table 3: Custom binary format (BIN) file format description.

Byte #	Width #	Description
0	4	Number of MSI entries (M)
4	4	Number of observations within MSI (N)
8	4	Species ID
12	4	Top-left pixel x
16	4	Top-left pixel y
20	4	Width
24	4	Height
28	4	User hash
-	-	Repeat Species ID User hash for each $n\epsilon N$
-	-	Repeat observations for each $m\epsilon M$

Comma separated value (CSV) files are used to store the counts of each label within the image, as described in Table 4. Each line in the CSV file is "MSI number, Number of white-phase snow geese, Number of blue-phase snow geese". This simple spreadsheet file format is used to validate the output observations of the CNNs with the known expert observations for the same images. If multiple experts had observations for the same image, the average of their observation counts were taken and rounded down to the nearest whole number and stored in the CSV file.

Table 4: Comma separated value (CSV) file format used for validating the counts output by the CNNs against expert counts from the same images.

MSI #	# of white-phase snow geese in MSI	# of blue-phase snow geese in MSI
Repeat for each unique MSI		

High Performance CNN Training

The initial training and testing of the CNNs for this and prior research was done using the CPU-only on a Mac Pro using a 3.5 GHz 6-Core Intel Xeon E5 processor using the CNN code available at:
https://github.com/Connor-Bowley/neuralNetwork [6]. To test the scalability of the framework, the program was limited to 1-, 2-, 3-, 4-, 5-, and 6-cores. The code is an excellent candidate for conversion to BOINC, to allow an even greater distribution and parallelization of processing. The code conversion was attempted for this work, but was not completed.

Matching Observations

Two algorithms were developed for matching individual observations within a single image to observations made by a different user. These two methods are the corner-point method (Eq 1) and the percent area overlap method (Eq 2) [5]. The corner-point method determines the maximum distance between each of the four corners of the observation while the area overlap method determines the percentage of total area overlap between the two observations.

$$
\begin{aligned}
c_0 &= \sqrt{(x_{10} - x_{20})^2 - (y_{10} - y_{20})^2} \\
c_1 &= \sqrt{(x_{11} - x_{21})^2 - (y_{10} - y_{20})^2} \\
c_2 &= \sqrt{(x_{11} - x_{21})^2 - (y_{11} - y_{21})^2} \\
c_3 &= \sqrt{(x_{10} - x_{20})^2 - (y_{11} - y_{21})^2} \\
C &= \max(c_0, c_1, c_2, c_3)
\end{aligned}
\tag{1}
$$

$$L_i = \min(x_{11}, x_{21}) - \max(x_{10} - x_{20})$$

$$H_i = \min(y_{11}, y_{21}) - \max(y_{10} - y_{20})$$

$$A_i = \max(0, L_i \times H_i) \tag{2}$$

$$A_u = A_1 + A_2 - A_i$$

$$O = \frac{A_i}{A_u}$$

Extracting Matched Observations

After two observations have been determined to be matching observations using one of the matching algorithms, a new matched observation can be extracted. Since the individual observations are not in identical locations, two algorithms have been developed to extract an "agreed upon" matched observation from the individual observations. These two algorithms are the extract average method (Eq 3) and the extract intersection method (Eq 4) [5]. The extract average method determines the Cartesian coordinates for the top-left and bottom-right corners by averaging the left-, right-, top-, and bottom-most extents of the observations, while the extract intersection methods extracts the overlapping area between the observations.

$$(x_0, y_0) = \left(\frac{\sum_{i=1}^{n} x_{i0}}{n}, \frac{\sum_{i=1}^{n} y_{i0}}{n} \right)$$
$$(x_1, y_1) = \left(\frac{\sum_{i=1}^{n} x_{i1}}{n}, \frac{\sum_{i=1}^{n} y_{i1}}{n} \right) \tag{3}$$

$$(x_0, y_0) = \left(\max_{i=1}^{n}(x_{i0}), \max_{i=1}^{n}(y_{i0}) \right)$$
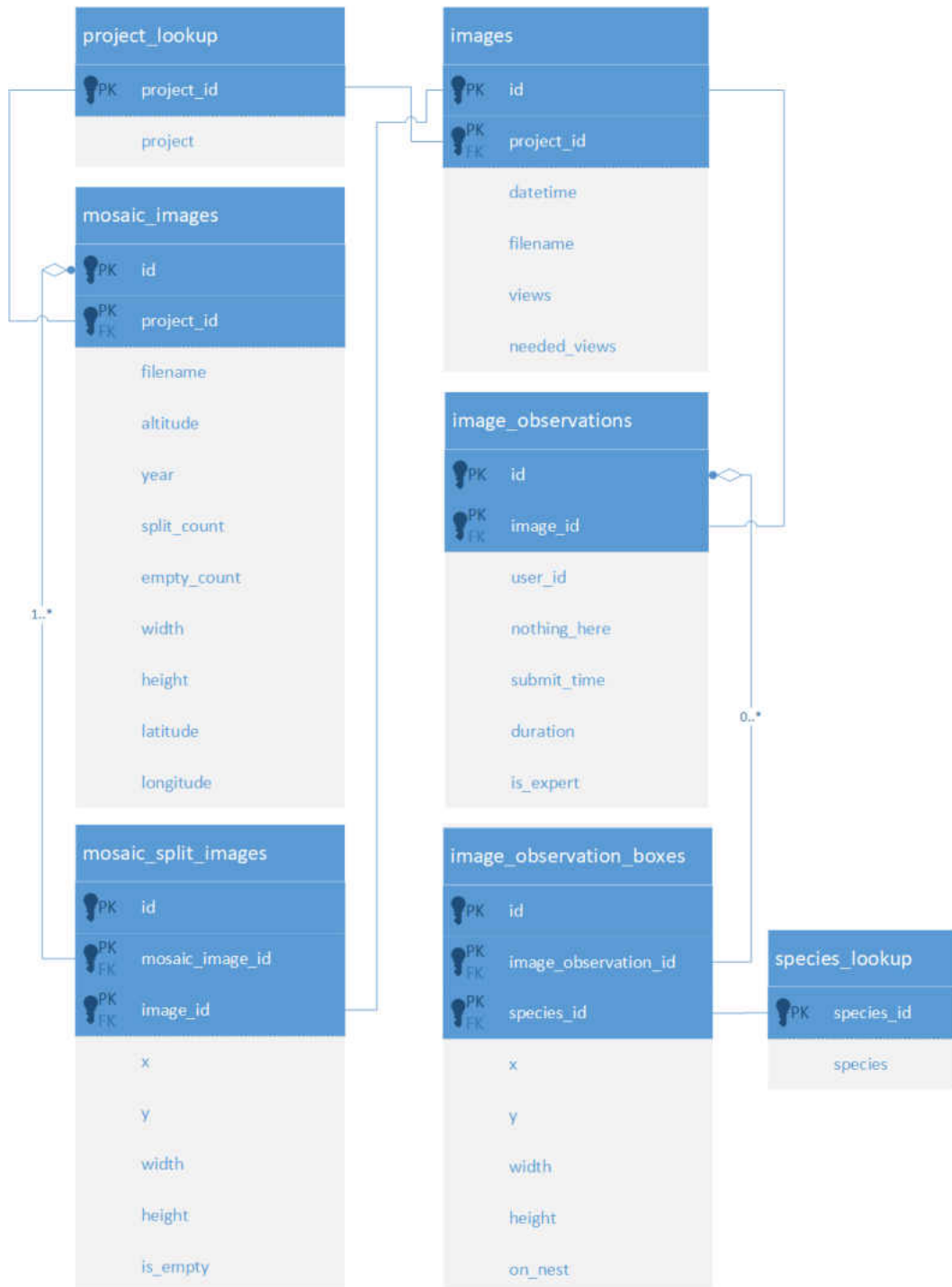$$(x_1, y_1) = \left(\max_{i=1}^{n}(x_{i1}), \max_{i=1}^{n}(y_{i1}) \right) \tag{4}$$

Figure 5: UML database diagram, based on the UML class diagram, showing the important entities within the database for storage of the mosaics and user observations relevant to this research. **user_id** is stored in another database for security purposes and not shown in this schema.

CHAPTER 5

RESULTS

Comparing Citizen Scientist vs Experts

Given the demonstrated applicability of the corner-point method for matching observations with a 10-pixel maximum corner distance, and the quality of the overlap algorithm in extracting aggregate user observations from the matched observations [5], a new technique was used to show that the methods hold over a larger sample for comparing against expert observations. Matched citizen scientist observations were extracted from mosaics which had 250 or more expert observations. For each of these mosaics, expert observations were attempted to be matched with at least one matched citizen scientist observation using the corner-point method with a 10-pixel maximum. If the expert could be matched, it was added to the binary matches for the mosaic.

The lowest percent of expert matches against aggregate citizen scientists is 73%, while the highest percent of expert matches is 92% (Table 5). This suggests that the matched citizen scientist observations extracted using the overlap algorithm still corresponded well with expert observations, even after the overlap extraction. This, along with the original results over a smaller dataset, gives confidence that the matched citizen scientist observations can be used to train the CNNs with accuracy comparable to expert observations.

Comparing individual altitudes vs one large dataset

The same application and code, with the same settings of 30 epochs and 5 training iterations with a feedback loop, was used to train and test the CNNs of

Table 5: The number of expert observations within mosaics that are able to be matched, using the corner-point matching algorithm, with at least one matched user observation with the aggregate observation of two citizen scientists, extracted using the overlap algorithm.

Mosaic #	Expert Observations	Matched Observations	Binary Matches	Percent Matched
44	1226	989	1093	89%
45	1522	1634	1403	92%
46	874	1340	720	82%
49	862	689	756	88%
50	282	190	214	76%
53	1465	1161	1076	73%
54	1033	941	856	83%
56	3295	1831	2696	82%
62	734	1010	1316	76%

individual altitudes as was used in prior work to train and test CNNs on the aggregate altitude dataset [6, 45]. A background to target object ratio of 3:1 was chosen for all the CNN training, as it was shown to be a ratio which maximized the training of the CNN without introducing too many over-fitting errors [45]. The CNNs were trained and tested at 75m, 100m, and 120m for both expert and matched citizen scientists observations for a total of 6 distinct CNNs.

There is a large discrepancy in the number of observations in the eight (8) datasets. The aggregate datasets had 2056 expert and 6560 matched observations, respectively. The 75m altitude dataset had 480 expert and 1540 matched observations. The 100m altitude dataset had 555 expert and 2035 matched observations. The 120m altitude dataset had 680 expert and 2350 matched observations.

Iteration 0 (Table 6) is the baseline iteration with no feedback loop implemented, providing a baseline of the CNN. The improvement in Iteration 1 shows how a single instance of retraining using the feedback loop can

Table 6: Results of the CNN feedback loop for the aggregate data and each altitude with a 3:1 background to target object ratio.

Training set	Iteration	Predict	Actual	|%Error|	Observations
Agg. Expert	0	850.00	331	156.80	2056
Agg. Expert	1	279.00	331	15.71	2056
Agg. Expert	best (3.00)	288.67	331	12.79	2056
Agg. Matched	0	1054.33	331	218.53	6560
Agg. Matched	1	330.00	331	0.30	6560
Agg. Matched	best (2.67)	318	331	3.93	6560
75m Expert	0	114	96	18.75	480
75m Expert	1	111	96	15.63	480
75m Expert	best (4.00)	107	96	11.46	480
75m Matched	0	155	96	61.46	1540
75m Matched	1	217	96	126.04	1540
75m Matched	best (0.00)	155	96	61.46	1540
100m Expert	0	750	111	575.68	555
100m Expert	1	579	111	421.62	555
100m Expert	best (4.00)	370	111	233.33	555
100m Matched	0	212	111	90.99	2035
100m Matched	1	330	111	197.30	2035
100m Matched	best (0.00)	212	111	90.99	2035
120m Expert	0	308	136	126.47	680
120m Expert	1	296	136	117.65	680
120m Expert	best (4.00)	275	136	102.21	680
120m Matched	0	217	136	59.56	2350
120m Matched	1	226	136	66.18	2350
120m Matched	best (3.00)	191	136	40.44	2350

dramatically reduce the error of the CNN in the case of the total aggregate data from prior research. The improvement in each subsequent iteration for the individual datasets, however, is not as prominent. Each training set had a different iteration which produced the least error, with the average best iteration shown for each dataset.

75m Individual Datasets

The 75m expert dataset produced the overall best results with both the lowest average error with 11.46% and the lowest standard deviation in individual results

with 24.15% (Fig. 6(a)). The 75m matched dataset was 61.47% with a standard deviation of 59.35% (Fig. 6(b)), which is the best among the individual matched datasets. The standard deviations on both 75m datasets is extremely high, indicating that either more data is required or further CNN configuration tuning needs to occur before this altitude can match the aggregate dataset CNN.

100m Individual Datasets

The 100m expert dataset average percent error improved notably with retraining, reducing the average percent error from 575.68% to 233.33% (Fig. 7(a)). The standard deviation, however, saw minimal reduction. The 100m matched dataset average percent error and standard deviation remained similar during retraining, with the baseline iteration being marginally better than the other iterations (Fig. 7(b)). This indicates similar issues to the CNN retraining of the 75m individual datasets.

120m Individual Datasets

The 120m expert (Fig. 8(a)) and matched (Fig. 8(b)) datasets both have marginally positive results on the average percent error from retraining. The standard deviation of the error varies between retrain iterations, both positively and negatively, for both datasets. This is further evidence that either the retraining algorithm requires more observations to be effective or that the CNN parameters need tweaking for smaller datasets.

General Notes

The expert and matched both produced the best individual CNN results with the 75m altitude dataset. This is likely explained by the size of the target objects being relatively larger from 75m altitude than from 100m or 120m

altitude. This highlights a major obstacle in generalizing the CNN for usage in multiple projects: the CNN configurations need be able to account for the target object size. Using the same CNN configuration with a target object size of 14px×14px works well for the 75m altitude, but may need to be adjusted for the 100m and 120m altitudes.

The variability of success using the retraining iterations is most likely the result of the randomized nature of the retrain loop. During retraining, known false positive outputs are verified against expert observations and included in the next retraining dataset. To limit the number of additional observations, some initial observations are randomly dropped from the retrain dataset. This was shown to produce consistently good results in the larger aggregate datasets, but may have a greater potential of removing important observations from the retrain dataset in the relatively smaller individual altitude datasets.

High Performance Scalability

The scalability of the CNN training program was tested by altering it to use a variable number of CPU-cores simultaneously. Performance was tested using a single training and testing iteration on the 75m expert dataset with no feedback loop. The Mac Pro 3.5 GHz 6-core Xeon E5 machine was limited to between 1-core and 6-cores.

The single-core runtime was almost 22 hours, while the 6-core run time was just under 3.5 hours, 96% of a pure linear improvement (Fig. 9). This shows that the CNN training program is an excellent candidate for multithreading and distributed computing. Converting the code to run using BOINC would allow for even larger datasets to be processed in a reasonable amount of time.
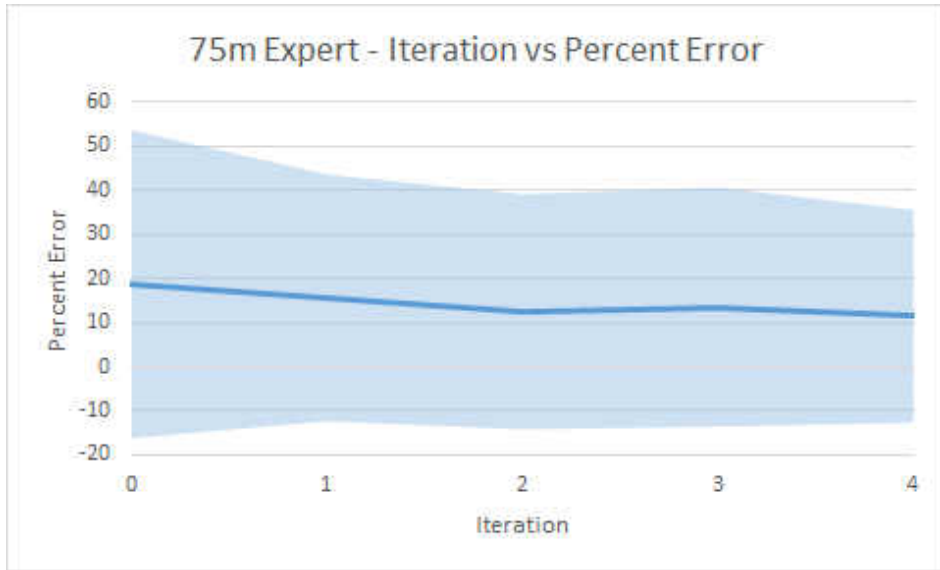
(a) 75m expert dataset average percent error with standard deviation shown. Re-training has a minimal improvement on both average percent error and standard deviation.
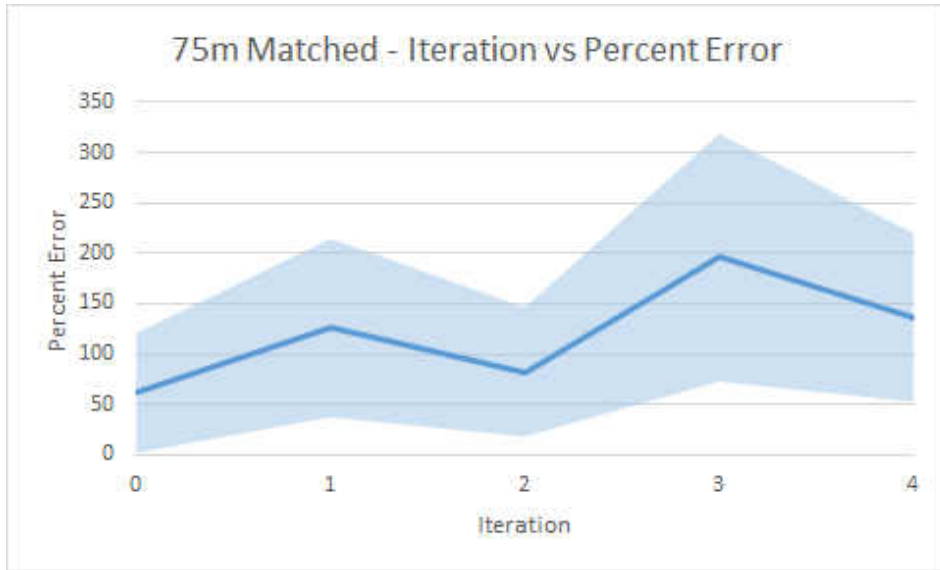
(b) 75m expert dataset average percent error with standard deviation shown. Re-training produces variably poorer results in both average percent error and standard deviation.

Figure 6: The average percent error in the 75m altitude individual datasets for expert (a) and matched observations (b). The average percent error is shown with the solid blue-line while the shaded area highlights the range of the standard deviation of percent error from individual mosaic images.

(a) 100m expert dataset average percent error with standard deviation shown. Retraining has a meaningful impact on the average percent error, decreasing from just under 600% to just over 200%, but a minimal impact on the standard deviation.

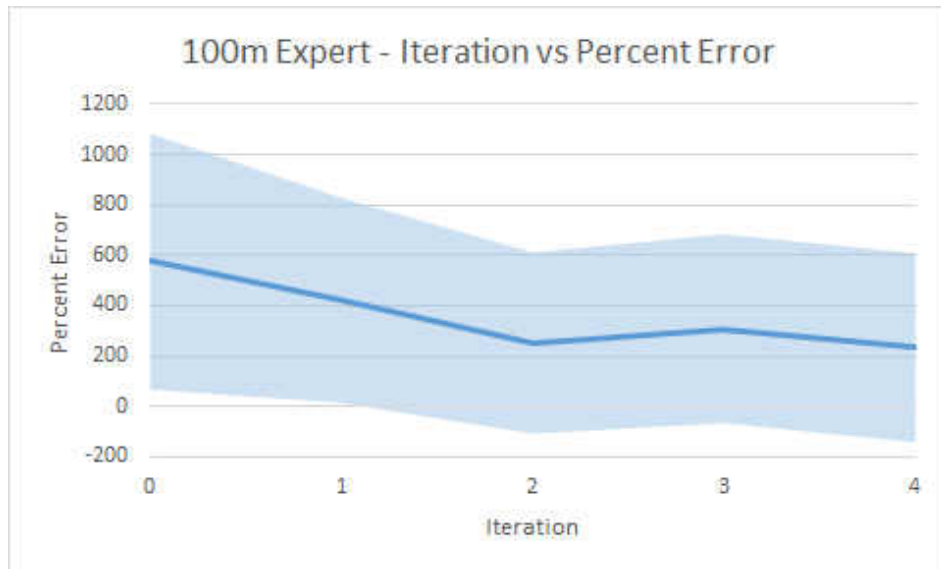(b) 100m matched dataset average percent error with standard deviation shown. Retraining has a minimal impact on both average percent error and standard deviation with results getting variably better and worse each iteration.
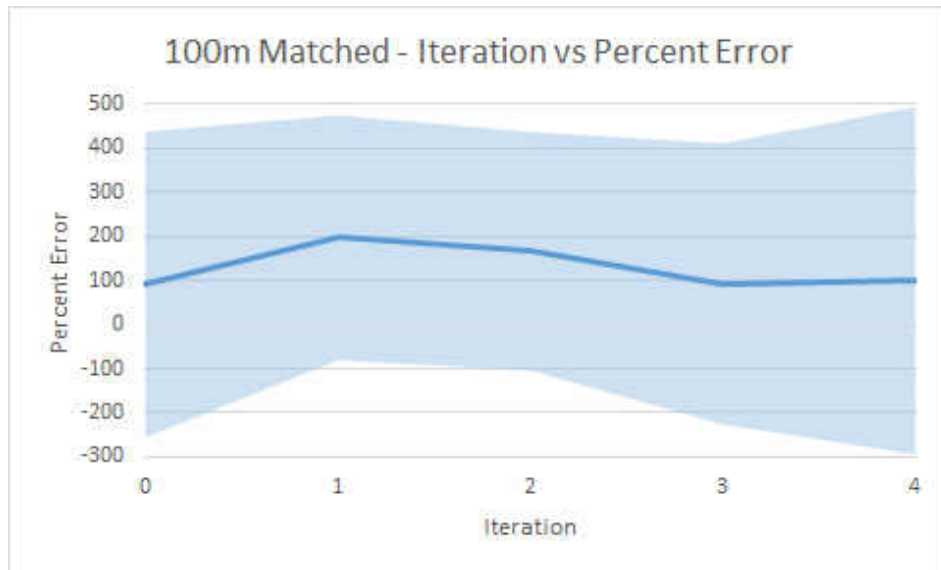
Figure 7: The average percent error in the 100m altitude individual datasets for expert (a) and matched observations (b). The average percent error is shown with the solid blue-line while the shaded area highlights the range of the standard deviation of percent error from individual mosaic images.
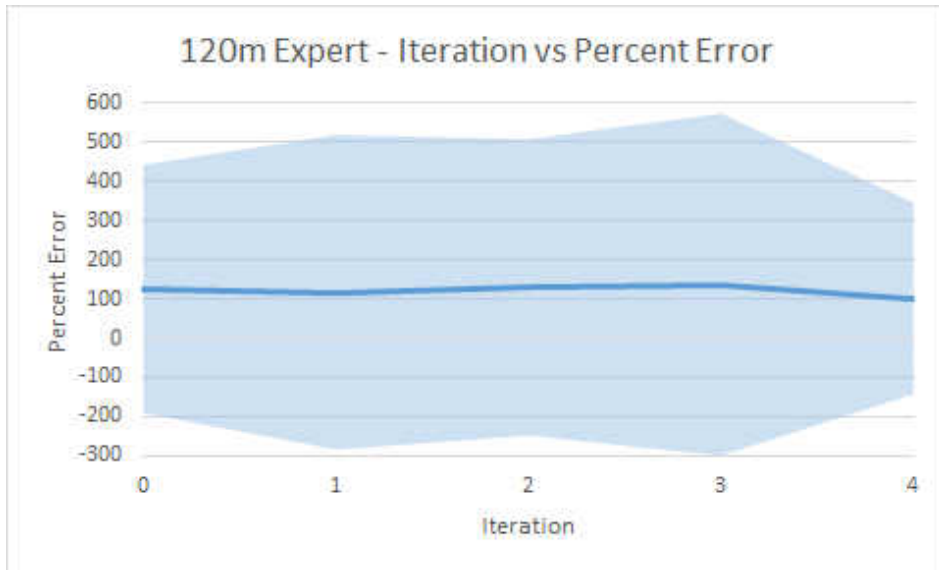
(a) 120m expert dataset average percent error with standard deviation shown. Retraining has a marginally positive impact on the average percent error while the standard deviation is meaningfully improved.
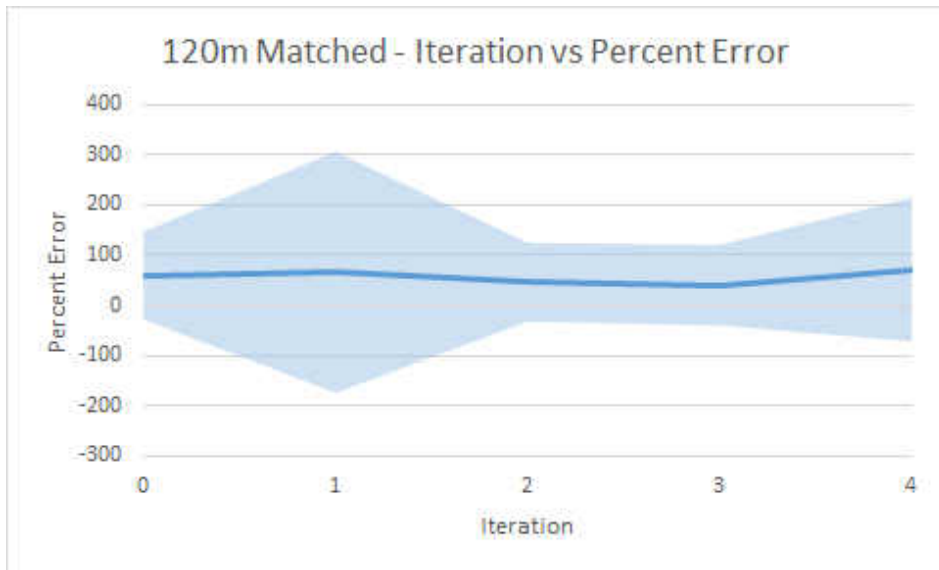
(b) 120m matched dataset average percent error with standard deviation shown. Retraining has a minimal impact on the average percent error, but has wildly variable negative and positive impact on the standard deviation.

Figure 8: The average percent error in the 120m altitude individual datasets for expert (a) and matched observations (b). The average percent error is shown with the solid blue-line while the shaded area highlights the range of the standard deviation of percent error from individual mosaic images.

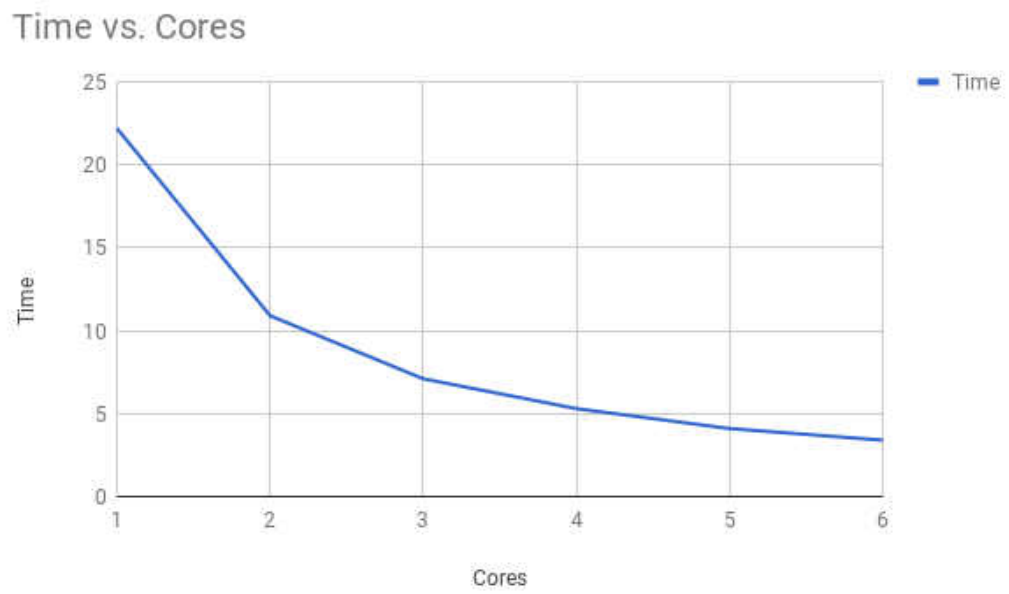Figure 9: The running time of a single iteration of the CNN training and testing program vs the number of cores utilized. The speedup is not perfectly linear due to the overhead of thread synchronization, but there is near-linear speedup.

CHAPTER 6

CONCLUSION

Citizen science can be a useful technique to distribute the classification of species within UAS imagery. Matching citizen scientist observations decreases the variability of the observations and provides matched observations that are comparable to observations made by trained experts. The techniques used in this research can be further used in other citizen science projects to rapidly categorize targets within images for use in CNNs, or to just gather aggregate data, which can then inform ecological conclusions.

The matched citizen scientist observations are comparable to expert observations and can be used to train CNNs to help automate the detection of species within the UAS imagery. The specific UAS imagery used in this research trained better on the CNNs using aggregate observations from all altitudes, instead of using individual datasets for each individual altitude — 75m, 100m, and 120m. This has interesting implications toward requiring a significant amount of training data, especially when the target objects make up less than 1% of the total area of the imagery, versus having more "correct" data.

The individual altitudes each contain significantly less observations than the aggregate dataset, but all of the observations are nearly identical in size. The dramatic improvement using retraining on the aggregate dataset, and the variable results of retraining on the individual datasets, seem to indicate that having more data is beneficial, even if the sizes are marginally different between target objects. This is also apparent in the large standard deviation of percent error in the individual altitude datasets, even after retraining. The aggregate dataset sees some improvement in minimum and maximum deviation during training, but has a significantly lower deviation to start with [45].

The CNN training and retraining program is currently limited to running on a single machine. It was demonstrated that there is a near linear improvement in processing speed by increasing the core count from 1-core to 6-cores using the prior developed program [6, 45]. This indicates converting the current algorithms to use BOINC can significantly decrease the amount of time required to test the CNNs, especially as the datasets get larger.

Future Work

This research demonstrated that a citizen scientists web portal can be used to generate inputs for CNNs which can be downloaded using a simple interface for single-computer training and testing. Any number of projects requiring users to mark up images can be created on Wildlife@Home, but inserting new data requires a developer to run the necessary scripts to insert that data into the databases to be exposed on the website. A natural extension of the project would be to create a generalizable framework that can allow researchers to setup their own project on Wildlife@Home, which would require creating an additional authentication system and further project management interfaces. The framework could be further generalized to work as a standalone product, but would require significant redesign.

The project management interfaces should include the ability for researchers to filter the observations and download a single zip file containing the requested data for CNN training. This would include all the images, IDX, BIN, and CSV files required for training. This download could then be used with the prior developed interface on a Mac or Linux machine to train and test the CNNs [6, 45].

The final extension of the project would be to convert the CNN algorithms to use BOINC to distribute the computing. This would allow citizen scientists who

work on the project to also volunteer their idle computing resources to the training and testing effort. However, the current code requires significant reworking to decouple OS-specific and single-machine code to support distributed computing. This could eventually be incorporated into the project management interface to start a BOINC job instead of downloading a zip file when the researcher wants to download training and testing data.

LITERATURE CITED

[1] DominiqueChabot, "Trends in drone research and applications as the journal of unmanned vehicle systems turns five," *Journal of Unmanned Vehicle Systems*, vol. 6, no. 1, pp. vi–xv, 2018.

[2] T. Desell, "Citizen science grid," 2017.

[3] T. Desell, R. Bergman, K. Goehner, R. Marsh, R. VanderClute, and S. Ellis-Felege, "Wildlife@Home: Combining crowd sourcing and volunteer computing to analyze avian nesting video," in *eScience (eScience), 2013 IEEE 9th International Conference on.* IEEE, 2013, pp. 107–115.

[4] "On the effectiveness of crowd sourcing avian nesting video analysis at Wildlife@Home, author = Travis Desell and Kyle Goehner and Alicia Andes and Rebecca Eckroad and Susan Ellis-Felege, booktitle = The 15th International Conference on Computational Science, address = Reykjavík, Iceland, month = June, day = 1-3, year = 2015."

[5] M. Mattingly, A. Barnas, S. Ellis-Felege, R. Newman, D. Iles, and T. Desell, "Developing a citizen science web portal for manual and automated ecological image detection," in *e-Science (e-Science), 2016 IEEE 12th International Conference on.* IEEE, 2016, pp. 223–232.

[6] C. Bowley, M. Mattingly, S. Ellis-Felege, and T. Desell, "Toward using citizen scientists to drive automated ecological object detection in aerial imagery," in *e-Science (e-Science), 2017 IEEE 12th International Conference on.* IEEE, 2017.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning

applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] P. Y. Simard, D. Steinkraus, J. C. Platt *et al.*, "Best practices for convolutional neural networks applied to visual document analysis." in *ICDAR*, vol. 3, 2003, pp. 958–962.

[9] Y. LeCun and C. Cortes, "MNIST handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[10] T. Desell, "Developing a volunteer computing project to evolve convolutional neural networks and their hyperparameters," in *e-Science (e-Science), 2017 IEEE 12th International Conference on.* IEEE, 2017.

[11] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1237.

[12] D. A. Fischer, M. E. Schwamb, K. Schawinski, C. Lintott, J. Brewer, M. Giguere, S. Lynn, M. Parrish, T. Sartori, R. Simpson, A. Smith, J. Spronck, N. Batalha, J. Rowe, J. Jenkins, S. Bryson, A. Prsa, P. Tenenbaum, J. Crepp, T. Morton, A. Howard, M. Beleu, Z. Kaplan, N. vanNispen, C. Sharzer, J. DeFouw, A. Hajduk, J. P. Neal, A. Nemec, N. Schuepbach, and V. Zimmermann, "Planet Hunters: the first two planet candidates identified by the public using the Kepler public archive data," *Monthly Notices of the Royal Astronomical Society*, vol. 419, no. 4, pp. 2900–2911, 2012.

[13] R. Bonney, C. B. Cooper, J. Dickinson, S. Kelling, T. Phillips, K. V. Rosenberg, and J. Shirk, "A developing tool for expanding science knowledge and scientific literacy," *BioScience*, vol. 59, pp. 977–984, 2009.

[14] M. A. Voss and C. B. Cooper, "Using a free online citizen-science project to teach observation & quantification of animal behavior," *The american biology Teacher*, vol. 72, no. 7, pp. 437–443, 2010.

[15] C. Wood, B. Sullivan, M. Iliff, D. Fink, and S. Kelling, "ebird: engaging birders in science and conservation," *PLoS biology*, vol. 9, no. 12, p. e1001220, 2011.

[16] J. J. Horns, F. R. Adler, and Çağan H.Şekercioğlu, "Using opportunistic citizen science data to estimate avian population trends," *Biological Conservation*, vol. 221, pp. 151 – 159, 2018.

[17] R. Simpson, K. R. Page, and D. De Roure, "Zooniverse: observing the world's largest citizen science platform," in *Proceedings of the 23rd international conference on world wide web*. ACM, 2014, pp. 1049–1054.

[18] J. Cox, E. Y. Oh, B. Simmons, C. Lintott, K. Masters, A. Greenhill, G. Graham, and K. Holmes, "Defining and measuring success in online citizen science: A case study of Zooniverse projects," *Computing in Science & Engineering*, vol. 17, no. 4, pp. 28–41, 2015.

[19] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg, "Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey," *Monthly Notices of the Royal Astronomical Society*, vol. 389, no. 3, pp. 1179–1189, 2008.

[20] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, M. J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg, "Galaxy Zoo 1: data release of morphological classifications for nearly 900,000 galaxies," *Monthly Notices of the Royal Astronomical Society*, vol. 410, no. 1, pp. 166–178, 2011.

[21] J. e. a. Adelman-McCarthy, "The 6th Sloan Digital Sky Survey Data Release, http://www.sdss.org/dr6/," July 2007, apJS, in press, arXiv/0707.3413.

[22] Lion Research Center, University of Minnesota.

[23] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments.* ACM, 2011, pp. 9–15.

[24] C. Bowley, A. Andes, S. Ellis-Felege, and T. Desell, "Detecting wildlife in uncontrolled outdoor video using convolutional neural networks," in *e-Science (e-Science), 2016 IEEE 12th International Conference on.* IEEE, 2016, pp. 251–259.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[26] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[27] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[30] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[31] S. Xu and Q. Zhu, "Seabird image identification in natural scenes using GrabCut and combined features," *Ecological Informatics*, vol. 33, pp. 24–31, 2016.

[32] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3.   ACM, 2004, pp. 309–314.

[33] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[34] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *Icml*, vol. 96, 1996, pp. 148–156.

[35] J. H. Friedman, "Additive logistic regression: a statistical view of boosting," *Ann. Statist.*, vol. 28, pp. 337–407, 2000.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] A. Gomez, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: identification of animal species in camera-trap images using very deep convolutional neural networks," *arXiv preprint arXiv:1603.06169*, 2016.

[38] L.-P. Chrétien, J. Théau, and P. Ménard, "Visible and thermal infrared

remote sensing for the detection of white-tailed deer using an unmanned aerial system," *Wildlife Society Bulletin*, vol. 40, no. 1, pp. 181–191, 2016.

[39] A. Abd-Elrahman, L. Pearlstine, and F. Percival, "Development of pattern recognition algorithm for automatic bird detection from unmanned aerial vehicle imagery," *Surveying and Land Information Science*, vol. 65, no. 1, p. 37, 2005.

[40] D. P. Anderson, "BOINC: A system for public-resource computing and storage," in *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on.* IEEE, 2004, pp. 4–10.

[41] K. A. Goehner, "Using computer vision and volunteer computing to analyze avian nesting patterns and reduce scientist workload," Master's thesis, University of North Dakota, 2015.

[42] S. L. Peterson, R. F. Rockwell, C. R. Witte, and D. N. Koons, "The legacy of destructive snow goose foraging on supratidal marsh habitat in the hudson bay lowlands," *Arctic, Antarctic, and Alpine Research*, vol. 45, no. 4, pp. 575–583, 2013.

[43] G. P. JONES IV, L. G. Pearlstine, and H. F. Percival, "An assessment of small unmanned aerial vehicles for wildlife research," *Wildlife Society Bulletin*, vol. 34, no. 3, pp. 750–758, 2006.

[44] K. S. Christie, S. L. Gilbert, C. L. Brown, M. Hatfield, and L. Hanson, "Unmanned aircraft systems in wildlife research: current and future applications of a transformative technology," *Frontiers in Ecology and the Environment*, vol. 14, no. 5, pp. 241–251, 2016.

[45] C. Bowley, "Training convolutional neural networks using an automated feedback loop to estimate the population of avian species," Master's thesis, University of North Dakota, Grand Forks, ND, 2017.