

11-10-2023

UAVs and Deep Neural Networks: An Alternative Approach to Monitoring Waterfowl at the Site Level

Zachary J. Loken

Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://repository.lsu.edu/gradschool_theses



Part of the [Artificial Intelligence and Robotics Commons](#), [Databases and Information Systems Commons](#), [Data Science Commons](#), and the [Natural Resources and Conservation Commons](#)

Recommended Citation

Loken, Zachary J., "UAVs and Deep Neural Networks: An Alternative Approach to Monitoring Waterfowl at the Site Level" (2023). *LSU Master's Theses*. 5850.
https://repository.lsu.edu/gradschool_theses/5850

This Thesis is brought to you for free and open access by the Graduate School at LSU Scholarly Repository. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Scholarly Repository. For more information, please contact gradetd@lsu.edu.

UAVS AND DEEP NEURAL NETWORKS: AN ALTERNATIVE APPROACH TO MONITORING WATERFOWL AT THE SITE LEVEL

A Thesis
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The School of Renewable Natural Resources

by
Zachary Jonathan Loken
B.S., University of Wisconsin, Stevens Point, 2019
December 2023

ACKNOWLEDGEMENTS

First, I would like to thank Dr. Kevin Ringelman, my advisor, for allowing me the opportunity to complete this Master's project. From keeping me out of trouble with HR to helping me tame my project-related thoughts, my gratitude for your support throughout this project is immense. I also must thank my committee members, A. Mini and D. Fowler, for their honest input and guidance throughout this project. Moreover, this project would not have been possible without the financial and professional support from multiple organizations and individuals. Ducks Unlimited, Inc. provided financial support for this project through the National Fish and Wildlife Foundation. The U.S. Fish and Wildlife Service, Louisiana Department of Wildlife and Fisheries, and Five Oaks Ag Research and Education Center provided fieldwork housing. I give countless thanks to the duck hunting clubs and landowners who allowed me access to their properties during the hunting season to conduct my research; this project was impossible without you. Additionally, a special thanks to M. Furtman, who generously provided over 1,000 professional photos of wintering waterfowl for this project at no additional cost. I would also like to thank my fellow lab mates and cohort of graduate students in the School of Renewable Natural Resources—D. Bakner, L. Bonczek, A. Dopkin, A. Lipford, K. Miranda, G. Rhodes, G. Rhyne, P. Rodrigues, E. Stein, and G. Zhiwei. My time at LSU was much more enjoyable thanks to each of you. To my technicians and undergraduate student workers—G. Rhodes, G. Rosseau, M. Baldino, and M. Wray—thank you. I could not have completed this project's fieldwork and data annotation aspects without you. Finally, my dog Bernie kept me company during many field excursions and late nights in the office—a companionship that'll never be forgotten.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	ii
ABSTRACT.....	iv
CHAPTER 1. SURVEY PROTOCOL FOR MONITORING NON-BREEDING WATERFOWL WITH UNCREWED AERIAL VEHICLES	1
1.1. INTRODUCTION.....	1
1.2. METHODS.....	5
1.3. RESULTS	14
1.4. DISCUSSION	15
CHAPTER 2. APPLICATION OF DEEP CONVOLUTIONAL NEURAL NETWORKS TO LOCATING AND IDENTIFYING SPECIES OF NON-BREEDING WATERFOWL IN UAV IMAGERY	20
2.1. INTRODUCTION.....	20
2.2. METHODS.....	22
2.3. RESULTS	38
2.4. DISCUSSION	42
APPENDIX A	50
LITERATURE CITED.....	51
VITA	60

ABSTRACT

Understanding how waterfowl respond to habitat restoration and management activities is crucial for evaluating and refining conservation delivery programs. However, site-specific waterfowl monitoring is challenging, especially in heavily forested systems such as the Mississippi Alluvial Valley (MAV)—a primary wintering region for ducks in North America. I hypothesized that using uncrewed aerial vehicles (UAVs) coupled with deep learning-based methods for object detection would provide an efficient and effective means for surveying non-breeding waterfowl on difficult-to-access restored wetland sites. Accordingly, during the winters of 2021 and 2022, I surveyed wetland restoration easements in the MAV using a UAV equipped with a dual thermal-visible high-resolution sensor to collect 2,360 optical images of non-breeding waterfowl. I then developed, optimized, and trained a SingleShot MultiBox Detection (SSD) object detection model with a deep convolutional neural network (VGG16) backbone to locate and identify eight different duck species in the UAV imagery. The final model achieved a total mean average precision and recall of 99.1% and 82.9%, respectively, after only 45 training epochs. The individual species class precision ranged from 65.3 to 86.1%, while the species class recall ranged from 69.7 to 88.6%. This study demonstrates the promise of UAV-based surveys for effectively surveying non-breeding waterfowl in structurally complex and difficult-to-access habitats and, additionally, provides a functional deep learning-based object detection framework for automated detection of non-breeding waterfowl in UAV imagery. This framework can be used to provide managers with an efficient and cost-effective means to count waterfowl on project sites—thereby improving their capacity to evaluate waterfowl response to wetland restoration efforts.

CHAPTER 1. SURVEY PROTOCOL FOR MONITORING NON-BREEDING WATERFOWL WITH UNCREWED AERIAL VEHICLES

1.1. INTRODUCTION

1.1.1. Waterfowl Conservation: Habitat and Population Objectives

The management of North American waterfowl is a conservation success story (Williams et al., 1999; Rosenberg et al., 2019), and much of this can be credited to the strategic vision of the North American Waterfowl Management Plan (NAWMP). The NAWMP was signed in 1986 by the U.S. Secretary of the Interior and Canadian Minister of the Environment to restore waterfowl (family Anatidae) populations to levels of the 1970s through habitat conservation. The NAWMP not only established waterfowl population objectives but also recognized the need for habitat restoration to reach population goals (NAWMP, 1986). In the absence of dedicated funding at the plan's conception, the population and habitat objectives outlined in the NAWMP were to be met through public-private partnerships across regionally based joint ventures (JVs) using pooled resources (NAWMP, 1986).

Continental waterfowl population objectives within the NAWMP were last updated in the 2018 update of the 2012 NAWMP revision, and the procedures for assigning regional objectives have also been updated at both international and regional scales (Fleming et al., 2017). Regional population abundance objectives are foundational components for establishing waterfowl habitat objectives by JVs (Fleming et al., 2017), and translating regional population objectives to habitat goals usually takes place through bioenergetic modeling. Bioenergetic modeling assumes food resources are the limiting factor for waterfowl during autumn/winter. Accordingly, bioenergetic models estimate the energetic needs of waterfowl populations for the duration of the non-breeding

season and translate those needs into habitat objectives by estimating the dietary energy provided by different habitat types (Petrie et al., 2011).

The most common method for establishing population objectives for non-breeding waterfowl has involved the state-level Midwinter Waterfowl Survey (MWS) and county-level harvest data (Fleming et al., 2017). These data are combined across the U.S. and then used to "step-down" continental waterfowl population objectives for each region based on MWS data and harvest distribution (Lancaster et al., 2021). Migration chronology data are then typically used to extrapolate the midwinter objective across the non-breeding planning period to estimate duck-use-days (DUDs) and associated energy requirements (Fleming et al., 2017), although other methods exist for translating a midwinter population objective into a habitat objective (Petrie et al., 2011). Regardless, it is clear that robust monitoring of waterfowl and habitats at multiple spatial and temporal scales must occur in order to achieve non-breeding objectives under the NAWMP.

1.1.2. Current Monitoring Techniques

Throughout the Mississippi Flyway, particularly the Mississippi Alluvial Valley (MAV), non-breeding waterfowl populations are typically surveyed via aerial transects flown by fixed-wing aircraft at low elevations (Andersson et al., 2015). However, early iterations of these surveys faced criticism due to limitations in sampling design (Eggeman and Johnson, 1989), a lack of consistent methodologies over space and time (Reinecke et al., 1992; Heusmann, 1999), and visibility biases (Gilbert et al., 2021). Currently, several statistically robust long-term population datasets at local (Ringelman et al., 2017), regional (Whitaker et al., 2019), and state-level scales (Herbert et al., 2021) exist. When combined with harvest and migration chronology data, these datasets provide a means for monitoring regional populations of non-breeding waterfowl. Nonetheless, estimates derived from these surveys are still subject to various limitations, including

visibility biases and high coefficients of variation (Reinecke et al., 1992; Soulliere et al., 2013; Mastro, 2019; Gilbert et al., 2020; 2021).

1.1.3. Shortcomings for Site-Level Metrics

Aerial transect surveys for non-breeding waterfowl have been shown to provide useful estimates of population indices for open landscapes at large spatial scales (Conroy et al., 1988) and to facilitate comparisons of relative abundance and distribution among years. However, existing aerial survey designs have proven inadequate for accurately estimating waterfowl use at individual sites—particularly those affected by visual obstruction (Ringelman et al., 2017). Even site-level counts conducted by observers on the ground are subject to potentially significant visibility biases in enclosed areas—which is especially problematic in a region such as the lower MAV (LMAV), dominated by a complex network of privately owned forested wetlands with minimal access. Thus, a serious gap in monitoring the non-breeding waterfowl of this region exists at the site level, and this deficiency is particularly notable because the site level is precisely where restoration and management actions occur.

For example, the Wetland Reserve Easement (WRE) program is one conservation tool that provides critical wintering habitat to non-breeding waterfowl in the LMAV. A primary goal of the WRE program is to optimize wildlife habitat (Evans-Peters, 2010), especially for migratory waterfowl (NRCS, 2023), which corresponds with the goals and objectives of the NAWMP (USFWS, 2010). The WRE program aims to restore and enhance wetland ecosystems through conservation easements on privately owned lands, which is especially important in the LMAV—a critical wintering region for non-breeding waterfowl in North America (Herbert et al., 2021). In the LMAV alone, at least 2,772 privately owned WREs spanning over 375,782 hectares of wetland are enrolled in the WRE program (Table 1.1; National Conservation Easement Database, 2022).

Considering such an expanse of privately-owned habitat for waterfowl wintering in the LMAV, precise and site-specific survey protocols are imperative to effectively evaluate waterfowl use of these areas to determine the efficacy of the WRE program and others like it.

Table 1.1. The total number of privately-owned WREs in the LMAV, grouped by state. The total area (ha) of easements in the LMAV enrolled in the WRE program is listed for each state. The mean, minimum, and maximum area are also included to portray variability in easement size.

State	WREs	Area (ha)			
		Sum	Mean	Min	Max
Arkansas	677	100,278	148	2	1,731
Illinois	30	1,496	50	5	257
Kentucky	60	4,983	83	8	550
Louisiana	923	121,187	131	4	3,026
Mississippi	702	107,883	154	3	2,175
Missouri	177	17,708	100	1	2,246
Tennessee	203	22,247	110	2	2,246
Total	2,772	375,782			

1.1.4. UAVs for Site-level Monitoring

Uncrewed aerial vehicles (UAVs) may be a promising solution to reform site-level methods for monitoring non-breeding waterfowl. UAVs offer very high-resolution spatial data, which has led to research on the utility of UAV-acquired imagery to monitor specific biological metrics like avian nesting status and success at the site level (Weissensteiner et al., 2015; Sardá Palomera et al., 2017). The increasing use of UAVs for ecological and environmental monitoring has seen a commensurate increase in their use for counting individuals and nests in bird colonies across a range of species and colony sizes (Chabot and Bird, 2015; Chabot and Francis, 2016). Over the last decade, this exponential increase in UAV-based bird surveys has revolutionized wildlife research (Linchant et al., 2015).

For example, UAVs have been especially useful in estimating large congregations of birds such as flocks of Snow Geese (*Anser caerulescens*) and colonial-nesting waterbirds (Chabot and Bird, 2012; Sardá Palomera et al., 2012). UAVs have also proven effective at locating cryptic birds

in dense habitats (Pöysä et al., 2018), especially when equipped with thermal imaging cameras (Scholten et al., 2019). In recent years, using UAVs and thermal imaging cameras has helped researchers find waterbirds that otherwise would have gone undetected, with less disturbance and much greater efficiency than traditional methods (Bushaw et al., 2019; 2020). Accordingly, I developed methods to survey small, enclosed sites for non-breeding waterfowl using a UAV equipped with a dual thermal-visible high-resolution sensor.

1.2. METHODS

1.2.1. Project Overview

I developed and implemented a novel monitoring protocol for measuring non-breeding waterfowl abundance at the site level. Specifically, I developed methods for using a UAV with high-resolution optical and thermal sensors to collect dual thermal-visible images and videos of waterfowl on privately owned WREs. As such, the data collection objectives for this project were to:

- (1) collect thermal images and videos for detecting and quantifying waterfowl,
- (2) capture optical images and videos for identifying waterfowl to species level and determining general habitat use behaviors (e.g., foraging, loafing), and
- (3) complete each UAV survey without disturbing or disrupting the natural behavior of waterfowl.

A comparison of deep learning-based methods for object detection of waterfowl in UAV footage furthered this project (Chapter 2). As such, this project provides managers with an operational framework for evaluating waterfowl habitat use by determining the best site-level monitoring methods for non-breeding waterfowl in the LMAV.

1.2.2. Permits, Regulations, Training, and Logistics

The Federal Aviation Administration (FAA) outlines several requirements under Part 107 (14 CFR § 107) that must be met to fly a commercial UAV in the United States legally. Moreover, all UAVs weighing between 0.25 and 25 kilograms must be registered with the FAA using an online registration system and operated under Part 107 rules in the United States. Part 107 requirements include rules for operation, remote pilot certification, operations over people, and waivers. It is important to note that these requirements are subject to change, and it is essential to stay updated with the latest regulations issued by the FAA under Part 107 to ensure compliance when flying a commercial UAV in the United States.

Accordingly, I ensured that the UAV was registered with the FAA for this project and that all pilots obtained a Part 107 Remote Pilot certification. Concerns regarding operations over people were minimal as the study sites were private easements located in remote areas. However, in cases where people, such as hunters, were detected during surveys, the pilot maintained a safe operating distance to avoid potential risks. Additionally, I obtained a certificate of waiver authorizing UAV operation beyond the visual line of sight (as stated in 14 CFR § 107.31) during waterfowl surveys, as the survey locations were not easily accessible by vehicle. I also obtained a special use permit to conduct wildlife surveys on National Wildlife Refuge (NWR) lands.

1.2.3. Study Area

The LMAV (Figure 1.1) is North America's largest floodplain, covering nearly 10 million hectares. Its landscape comprises a complex network of sloughs, oxbows, side channels, and backwater areas, fostering a diverse forested wetland ecosystem. The region's subtle topography shapes hydrology and significantly influences plant and wildlife diversity and structure (Killgore et al., 2014). Numerous fish and wildlife species thrive within these intricate aquatic and terrestrial

habitats, although many populations are currently facing stress or decline (Killgore et al., 2014). Notably, over 80 percent of the region's forests are privately owned, introducing complexity to restoration efforts and management goals for wintering waterfowl (Oswalt, 2013). Within the broader wildlife context, the LMAV is considered the heart of the Mississippi Flyway—where more than 40 percent of North America's waterfowl and 60 percent of all U.S. bird species migrate or winter (National Fish and Wildlife Foundation, 2019).

1.2.4. Site Selection

WREs within the LMAV region of Arkansas, Louisiana, and Mississippi were identified as potential sampling sites for this study. Potential sites were restricted to a 100-km radius around each field house to optimize efficiency by minimizing travel time between locations. For the winter 2021 field season, field housing locations included Tensas NWR in Louisiana and Cache River NWR in Arkansas. Field housing locations for the winter 2022 field season were Russell Sage State Wildlife Management Area in Louisiana and Five Oaks Research and Education Center in Arkansas. The final selection of sampling sites (Figure 1.1) was based on proximity to available field housing, diversity in habitat cover types, landowner permission to access, and the density of adjacent WREs. Specifically, areas with clusters of five or more adjacent WREs were targeted to maximize the number of sites sampled in a single day. The sampled habitat cover types encompassed flooded agriculture, seasonal emergent marsh, flooded bottomland hardwood, slough, perennial river, and oxbow lake.

1.2.5. Field Operation Details

UAV surveys were conducted during the winter season of 2021 (December 2021 – January 2022) and 2022 (November 2022 – January 2023). The surveys took place from 30 minutes before sunrise until around 11 a.m. Central Standard Time (CST) and from 4 p.m. CST until 30 minutes

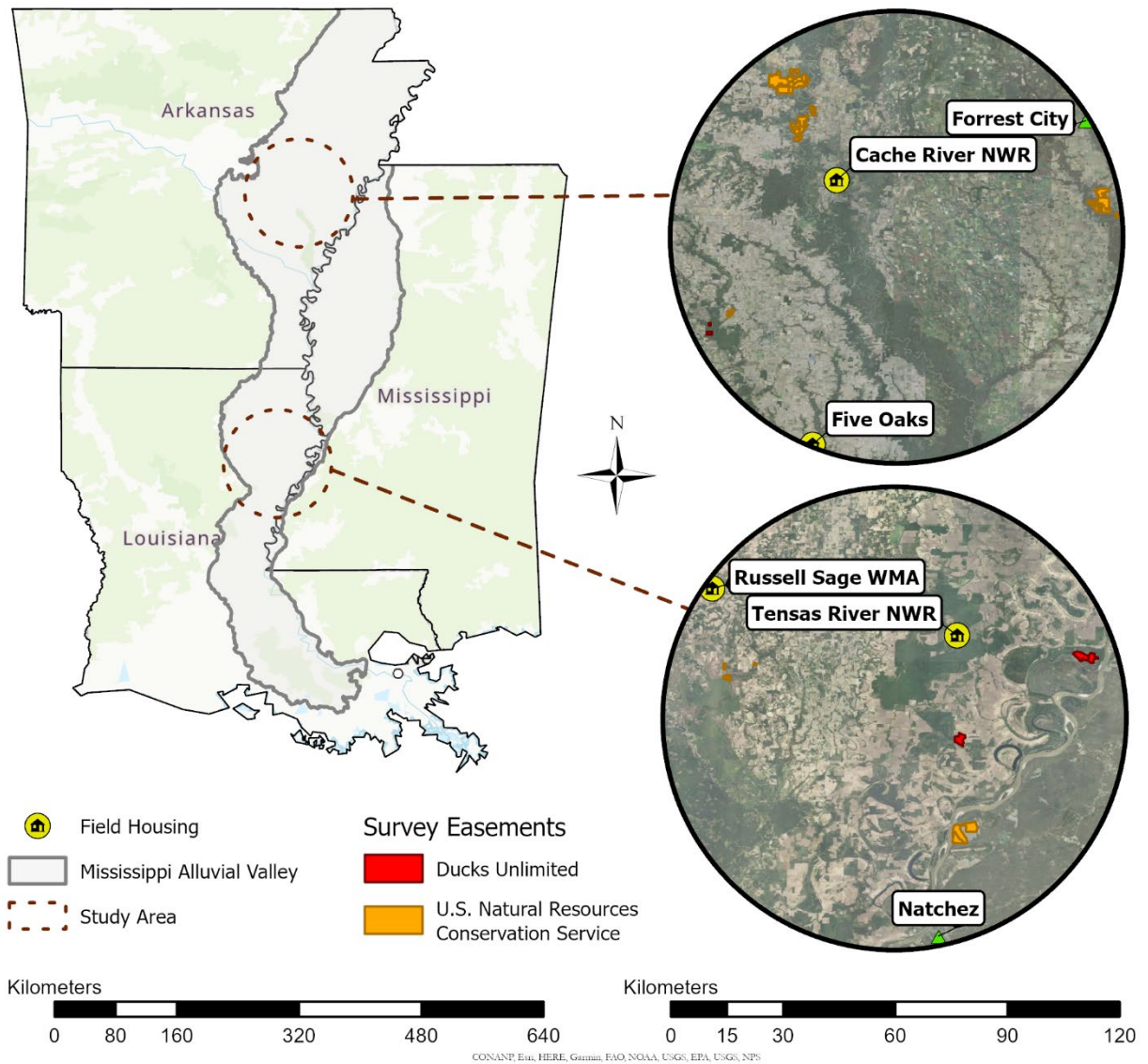


Figure 1.1. Study area boundaries and survey easement locations for fieldwork. Maps show the Mississippi Alluvial Valley boundary, northern and southern study areas, survey easement boundaries, and major U.S. Cities. Survey easement boundaries are color-coded according to agency. Natchez, MS and Forrest City, AR are shown for reference.

after sunset to minimize the occurrence of thermal false positives (e.g., the leaf of a hydrophyte is recorded with the same pixel values as waterfowl). Only days with clear or mildly overcast weather, zero precipitation, and wind speeds below 32 kilometers per hour were sampled to ensure optimal survey conditions. The temperature ranged from -2.2° Celsius (C) to 22.8°C during the surveys. Each survey lasted approximately one hour and was flown at an altitude of 68.5 m above

ground level (AGL), at a flight speed below 5 km/h, and with the camera tilted vertically downward between 25° and 40°. Due to feedback interferences between the UAV's payload sensor and remote control, surveys were confined to a radius of approximately 3.2 km from the launch point. Lastly, it was often necessary to visually sub-sample the thermal footage with the zoom camera during surveys to estimate relative species composition (Zhou et al., 2021).

1.2.6. UAV System and Operation Details

UAV flights were carried out using a DJI Matrice 300 RTK rotary-wing quadcopter, measuring 810 x 670 x 430 mm and weighing 6.3 kg. The quadcopter was powered by two 22.8 V lithium pro ion batteries, which provided a flight time of approximately 45 minutes when considering payloads. The DJI BS60 Intelligent Battery Station and eight additional batteries were utilized to ensure continuous operation in the field. The survey methodology employed an opportunistic approach to observe wintering waterfowl and other wetland-dependent bird species on WREs in the LMAV. After launching the UAV without specialized equipment from nearby upland areas, the pilot manually navigated the UAV to habitats pre-identified on aerial imagery as potential waterfowl habitat—which was often confirmed by discussions with landowners before each flight.

In addition to the pilot, one observer was always present to scan the airspace for potential collision hazards and maintain awareness of the UAV's position through direct visual observation. Although not necessary during either of the field seasons, the visual observer was also present to take control of the UAV in case the remote pilot in command lost the ability to do so. The UAV did not require any special provisions, such as internet access, to operate during remote fieldwork; however, the UAV controller was connected to Wi-Fi during surveys via a mobile hotspot wherever

cellular network signals were capable of rendering up-to-date satellite imagery as basemaps on the UAV remote control's integrated map.

1.2.7. Payload, Sensor, and Data Collection

The UAV was equipped with a DJI Zenmuse H20T Quad-Sensor—which contains a 20-megapixel (MP) zoom camera, a 12-MP wide-lens camera, a 1,200 m laser rangefinder, and a 640-x 512-pixel radiometric thermal camera. The thermal camera was used to collect thermal images (Figure 1.2) and videos for detecting and quantifying waterfowl, while the zoom camera was used to capture high-resolution optical images (Figure 1.3) and videos for identifying waterfowl to



Figure 1.2. Raw thermal image captured using the DJI Zenmuse H20T Quad-sensor's radiometric thermal camera. This thermal image of mallards was taken at an altitude of 68.5 m AGL with the camera tilted vertically downward 28.6° (i.e., gimbal pitch degree equal to -28.6).

species level where possible. Once waterfowl were detected on the UAV's thermal camera, the pilot would manually begin recording dual thermal-visible videos while panning and zooming to survey the entire congregation. Throughout each survey, the pilot would periodically trigger the zoom camera to capture high-resolution images of individuals and groups of waterfowl. Image and video data were stored as JPEGs on MicroSD cards during data acquisition, transferred to an external solid-state drive (SSD) post-survey, and backed up to the cloud after each field season.



Figure 1.3. High-resolution RGB image of the same scene as the thermal image in figure 2 captured using the DJI Zenmuse H20T Quad-sensor's zoom camera. This image of mallards was taken at 68.5 m AGL, using a 119.9 mm focal length, and a gimbal pitch degree equal to -28.6.

Ground sampling distance (GSD), the distance between pixel centers measured on the ground, of the sample thermal image (Figure 1.2) taken at 68.5 m AGL was approximately 6.09 cm/pixel. For the Zenmuse H20T thermal camera, the sensor width was 7.68 mm, the focal length was 13.5 mm, the image width was 640 pixels, and the image height was 512 pixels. The GSD of the sample zoom camera image (Figure 1.3), taken at 68.5 m AGL with a focal length of 119.94 mm, was approximately 0.08 cm/pixel. The Zenmuse H20T zoom camera had a sensor width of

7.6 mm, a modifiable focal length range of 6.83 to 119.94 mm, an image width of 5,184 pixels, and an image height of 3,888 pixels. Accordingly, the formula (Equation 1.1) for GSD is:

$$(Equation 1.1) \quad GSD = \frac{(S_W \times H \times 100)}{(F \times I_W)}$$

where GSD denotes ground sampling distance in cm/pixel, S_W denotes sensor width in mm, H denotes flight height in m, F denotes focal length in mm, and I_W denotes image width in pixels. Using the GSD of 6.09 cm/pixel for the sample thermal image (Figure 1.2), the image ground width and height were 39.0 and 31.2 m, respectively. Similarly, the image ground width and height of the sample zoom camera image (Figure 1.3) were 4.15 and 3.11 m, respectively. The calculation (Equation 1.2) to determine the ground width of an image is:

$$(Equation 1.2) \quad G_W = \frac{GSD \times I_W}{100}$$

where G_W denotes ground width in m, GSD denotes ground sampling distance in cm/pixel, and I_W denotes image width in pixels. To calculate the on-the-ground height (G_H) of an image using Equation 1.2, replace image width with image height (I_H ; pixels).

1.2.8. Thermal Imaging

Thermal imaging can be defined as the sum of radiance emitted from an object, radiance emitted from the surrounding environment, and atmospheric transmission (Equation 1.3).

$$(Equation 1.3) \quad R_{sen} = \tau \epsilon R_{obj} + \tau(1 - \epsilon)R_{env} + (1 - \epsilon)R_{atm}$$

R_{sen} is the radiance sensed by the thermal camera, R_{obj} denotes the radiance emitted by the object of interest, R_{env} is the radiance from the surrounding environment, R_{atm} is the radiance from the atmosphere, τ is the transmittance, and ϵ is emissivity. Most of the challenges associated with thermal imaging concern translating data into relative temperatures because radiometric thermal cameras store data as raw absolute values—not relative temperatures—in binary thermal images.

Furthermore, thermal imaging may be affected by the radiometric properties of objects and their surrounding environment—namely, transmittance (τ), emissivity (ϵ), and reflectance (ρ) (Maldague, 2001).

Atmospheric transmittance represents a significant challenge in thermal imaging analysis because it can change the radiometric temperature measurement, thus interfering with the active heat read by the sensor. Furthermore, humidity and temperature influence the transmission of radiance from targets; low humidity reduces transmitted atmospheric radiance (Nakaguchi and Ahamed, 2022), while high temperature increases transmitted radiance (Stefan-Boltzmann Law).

Emissivity is an object's effectiveness in emitting thermal energy compared with a blackbody, an idealized physical body that absorbs all incident electromagnetic radiation at the same temperature (Björn and Nilsson, 2018). Emissivity values are measured on a scale from 0 to 1, where 0 represents a perfect reflector, and 1 represents a blackbody. An object's emissivity depends on characteristics such as temperature, surface texture, material, spectral wavelength, oxidation, and sensor angle (Höpe, 2014). The equation for calculating emissivity (Equation 1.4) can be defined as the absorption of a blackbody minus the reflectance of the object (R_{obj}).

$$\text{(Equation 1.4)} \quad \epsilon_0 = 1 - R_{obj}(\lambda)$$

Reflectance can directly affect the interpretation of thermal values and is directly related to the angle of view from the camera to the target. Shorter distances to the target increase the reflectance captured by the camera and vice-versa. Radiometric correction, or calibration of image reflectance using illumination characteristics of the thermal sensor, is necessary to derive accurate temperature measurements from a raw radiometric image (Raeva et al., 2018).

1.3. RESULTS

1.3.1. Field Surveys and Data Collection

During the winters of 2021 and 2022, I conducted 57 flights across 27 survey days on 11 different WREs for a total of 9,953 hectares surveyed. During the surveys, I collected 2,360 thermal images, 2,360 optical images, 116 thermal videos, and 116 optical videos of non-breeding waterfowl. Table 1.2 summarizes my data collection efforts by property, field season, and state.

Table 1.2. Summary results from UAV fieldwork. 'Surveys' represents the number of times a site was visited, 'Flights' represents the number of times I launched the UAV at a site, 'Pictures' represents the total number of images (thermal and optical) collected at a site, 'Videos' represents the total number of videos (thermal and optical), and 'Hectares' represents the total hectares of each property surveyed and cumulative hectares surveyed across all flights.

Property	Season	State	Surveys	Flights	Pictures		Videos		Hectares	
					Thermal	Optical	Thermal	Optical	Property	Surveyed
BG	1	AR	1	2	10	10	3	3	55	55
FP	1	AR	1	1	0	0	1	1	1,196	1,196
JP	1	AR	1	2	50	50	4	4	921	921
JC	1	AR	4	9	0	0	16	16	97	390
CV	1	LA	1	1	0	0	1	1	517	517
MW	1	LA	1	1	25	25	2	2	154	154
RP	1	LA	1	4	31	31	7	7	499	499
RD	1	LA	1	2	25	25	3	3	39	39
MP	1	MS	1	3	4	4	5	5	1,844	1,844
Season 1 Total:			12	25	145	145	42	42	5,323	5,615
BB	2	AR	4	10	1,049	1,049	25	25	528	2,110
FO	2	AR	7	12	670	670	26	26	106	739
FP	2	AR	1	3	187	187	8	8	1,196	1,196
JC	2	AR	3	7	309	309	15	15	97	292
Season 2 Total:			15	32	2,215	2,215	74	74	1,927	4,337
Project Total:			27	57	2,360	2,360	116	116	7,250	9,952

Although the UAV-based survey for non-breeding waterfowl provided an effective and low-disturbance means of assessing waterfowl habitat use, the acquired images and videos must be manually processed to generate counts for each detected species. For further information on UAV data annotation and processing, including a breakdown of total abundance (i.e., manual counts) by guild and species, see Chapter 2.

1.4. DISCUSSION

1.4.1. Summary of Main Findings

This study successfully used UAVs equipped with a dual thermal-visible sensor to detect and monitor non-breeding waterfowl habitat use on enclosed WREs in the LMAV, aligning with findings from other literature investigating the efficacy of UAVs for studying wetland-dependent bird species. The two types of imagery proved highly complementary for facilitating detection (i.e., thermal sensor) and confirming identification (i.e., optical sensor) of cryptic subjects (McKellar et al., 2020) and additionally found several notable advantages to traditional ground-based methods. For example, this UAV-based survey method allowed for access to areas not easily accessible by foot (Ivošević et al., 2015) and likely provided higher detection rates than ground-based methods (Dundas et al., 2021) without disturbing waterfowl on project sites (McKellar et al., 2020; Ellis-Felege et al., 2021). The results demonstrate the promise of UAV-based surveys for effectively surveying non-breeding waterfowl and likely other wildlife in structurally complex and difficult-to-access habitats.

1.4.2. UAV-based Non-breeding Waterfowl Survey

Throughout the winter 2021 field season, I field-tested various combinations of flight altitudes, gimbal pitch degrees, flight speeds, survey times, and zoom levels on both the optical and thermal sensors until I found a combination of parameters that provided the best quality imagery without disturbing waterfowl. After testing various survey methods throughout the first field season, I developed a survey protocol based on the best combination of survey parameters implemented during the first field season. Table 1.3 summarizes the final survey parameters implemented in the final monitoring protocol for this study.

Table 1.3. The final survey parameters were determined after the first field season and implemented during the second field season.

Parameter	Value
Data Acquisition	Low-zoom optical image of the group, subsample at higher zoom
Flight Altitude	67 to 70 m AGL
Flight Speed	0 to 5 km/h over waterfowl
Flight Type	Manual (non-programmed)
Gimbal Pitch	-25 (side profile) to -40° (top profile)
Payload	Dual thermal-visible sensor
Sensor Resolution	Optical: 5K resolution; Thermal: 640- x 512-pixel resolution
Sensor Zoom Level	Optical: 10x to 40x zoom; Thermal: 1x to 2x zoom
Survey Time	Sunrise until 11 a.m. and 4 p.m. until sunset
Temperature Range	Limited to UAV and payload specifications
Weather Conditions	Clear or mildly overcast, zero precipitation, no fog, wind below 32 km/h

Expanding on Table 1.3., the proposed survey protocol for monitoring waterfowl abundance in enclosed areas is as follows. Surveys should be conducted using manual (non-programmed) flights systematically until all portions of the site that could support waterfowl have been surveyed (Bushaw et al., 2020; 2021). Waterfowl are mobile, habitats are dynamic, and it is often necessary to adjust the angle of the UAV and sensor relative to the target for successful species identification; pre-programmed flights are unlikely to capture the requisite imagery for species identification reliably and successfully and thus are not recommended. Manual flight patterns varied by habitat type, and as such, three different flight techniques were employed while searching for non-breeding waterfowl with the thermal camera:

- (1) For channelized features such as bayous and rivers, the UAV was flown down the center of the channel while panning the camera left and right to scan for waterfowl.
- (2) For open water habitats such as flooded agriculture and oxbow lakes, the UAV was flown to the center of the water body. I then divided the site into four imaginary quadrants and flew to the approximate centroid of each quadrant while panning the camera to scan for waterfowl.

(3) For flooded timber habitat, I targeted 'duck holes' (i.e., canopy openings) and sloughs where water was visible and panned the surrounding timber for waterfowl while navigating from one opening to the next.

Once waterfowl were detected on the thermal camera, the pilot switched to the optical camera and subsampled each group by zooming in on individuals as necessary to determine species composition. The species identification process was dynamic and variable between groups of waterfowl, so manual UAV operation was necessary; pre-programmed flights capturing imagery at fixed intervals are not likely sufficient for species identification. A zoom level of 20 on the optical camera was ideal, but any zoom level between 10 and 40 provided sufficient quality imagery for identifying species of drakes and hens. Waterfowl in imagery collected with a zoom level of less than ten were too small to identify to species level, and I concluded it was best to subsample groups at higher zoom levels for later computer processing of imagery back in the lab. In contrast, waterfowl in imagery collected with a zoom level greater than 40 were usually too pixelated and blurry.

A resolution of no less than 640 x 512 pixels for the thermal camera and a resolution of at least 20 MP for the optical camera is recommended. Survey flights between 67 and 70 m AGL provided sufficient quality data to achieve the objectives and ensured that waterfowl were not disturbed during the survey. Not disturbing waterfowl was especially important during this study, as I primarily operated on waterfowl hunting clubs during the hunting season. A gimbal pitch (i.e., tilt up or down) between -25° and -40° allowed for the best viewing angles of waterfowl, with less pitch and more pitch corresponding to side-profile and top-profile dominant views, respectively.

Surveys should take place from sunrise until 11 a.m. and from 4 p.m. until sunset to minimize the potential for thermal false positives. Similarly, despite the temperature range

ultimately being limited by UAV and payload specifications, anecdotal findings suggest that the thermal contrast of waterfowl and background features such as water or trees is more significant in lower temperatures than high. This increased thermal contrast is likely because there is less atmospheric radiance being transmitted at lower temperatures, creating a more significant difference between the radiance of the object and the atmospheric radiance. Similarly, the radiance of the environment would be smaller in low temperatures because there is less atmospheric radiance for the environment to absorb and later emit. Accordingly, following Equation 1.3, what this translates to in thermal imaging analysis is a decreased rate of false positives because R_{env} and R_{atm} are minimized causing R_{obj} (i.e., waterfowl) to be the primary source of radiance detected by R_{sen} .

1.4.3. Limitations of Proposed Survey

Although the proposed survey methods are viable and effective at surveying non-breeding waterfowl, limitations exist. For example, two UAV batteries last 45 minutes maximum when running multiple payload sensors, which poses several challenges when planning for remote fieldwork. Heavily forested landscapes, such as those found in the LMAV, can interfere with communication between the payload sensor and UAV remote control at distances greater than 3 km. As such, it was occasionally necessary to pause a survey mid-way and relocate to another portion of the property before finishing the survey. Moreover, the quality of data collected via UAV relies heavily on atmospheric conditions. Weather conditions such as rain, fog, high UV index, and heavy winds reduce the quality of data collected by UAVs, limiting the number of days that can be surveyed. Lastly, each UAV survey results in substantial amounts of image and video data that must be processed to derive valid abundance estimates.

1.4.4. Data Analysis Bottleneck

Even after UAVs have been successfully used to monitor large bird congregations that are not easily accessible on the ground, the vast amount of complex imagery collected introduces a data analysis bottleneck (Hayes et al., 2021). Post-processing is usually the most time-consuming part of UAV surveys (Barnas et al., 2019), but emerging deep-learning methodologies pose a potential solution to this data analysis bottleneck phenomenon (Hong et al., 2019; Francis et al., 2020). For example, deep neural networks (DNN)—networks with more than two hidden layers—are evolving as a prominent means of object detection and can be applied to UAV imagery for population monitoring (Osco et al., 2021). In general, DNNs have been proven to outperform conventional machine learning methods (He et al., 2016) and have led to a series of breakthroughs for object detection tasks (Aburasain et al., 2021). The application of DNNs to the UAV images of non-breeding waterfowl I collected during this project is the subject of Chapter 2.

CHAPTER 2. APPLICATION OF DEEP CONVOLUTIONAL NEURAL NETWORKS TO LOCATING AND IDENTIFYING SPECIES OF NON-BREEDING WATERFOWL IN UAV IMAGERY

2.1. INTRODUCTION

2.1.1. UAV-based Wildlife Surveys

In recent years, the use of UAVs has transformed wildlife surveys and ecological research (Linchant et al., 2015). UAVs offer a range of advantages, including the ability to access remote areas and challenging terrains, minimal disruption to wildlife, and the capture of high-resolution imagery. Collecting extensive UAV imagery has streamlined the acquisition of large amounts of data, potentially offering valuable insights into wildlife populations and habitat use. However, the proliferation of UAVs has introduced a new challenge: the efficient analysis of the vast and complex imagery generated during fieldwork.

The analysis of these image data has become a significant bottleneck in the utility of UAV surveys for wildlife research (Hayes et al., 2021). Manual processing is usually required to derive useful information from these UAV-captured images, consuming substantial time and computational resources while also requiring extensive storage and cataloging architecture (Barnas et al., 2019). Consequently, there is a pressing need for advanced analytical methods to automate the detection and classification of wildlife species in UAV imagery. This automation would alleviate the burden of manual image processing and fully harness the potential of UAV technology for wildlife conservation.

2.1.2. SingleShot MultiBox Detection

One promising solution to address this data analysis bottleneck phenomenon is the application of deep learning-based methods for object detection (Hong et al., 2019; Francis et al.,

2020). Deep learning algorithms have demonstrated remarkable capabilities in recognizing and locating objects within images finding applications across domains such as computer vision, natural language processing, and ecological research. Training deep learning algorithms for automated identification and classification of objects in UAV imagery holds the promise of streamlining data analysis and providing valuable insights into wildlife populations and their dynamics.

Among the array of deep learning-based algorithms for object detection, the SingleShot MultiBox Detector (SSD) is an excellent choice for multi-class wildlife detection in UAV imagery. SSD offers real-time object detection with high precision, rendering it well-suited for efficiently processing large datasets where more than one species class may appear together in an image. However, the conventional SSD model relies on a basic convolutional neural network architecture, which, while effective in many applications, grapples with a significant challenge: the vanishing gradient problem. The gradient flows through convolutional layers with the chain derivation rule in the backpropagation algorithm during deep learning training; if the initial gradient input value is less than 1 in a deep network, the gradient is prone to approach 0 during the calculation process, which contributes to training failure on some layers (Liu et al., 2021). This is the vanishing gradient problem. The network's ability to effectively capture complex patterns in the image data can be hindered and training times prolonged, resulting in a poorly trained model.

2.1.3. Deep Neural Networks for Classification

I employed an enhanced deep neural network backbone architecture while training the SSD algorithm to detect and identify non-breeding waterfowl in UAV imagery. Specifically, this deep neural network backbone addresses the vanishing gradient problem by introducing a rectified linear unit (Agarap, 2019; ReLU) activation layer after each convolutional layer in the neural

network. The ReLU activation function outputs a gradient of 1 when the input is greater than 0, thereby minimizing the chances of the gradients vanishing during training (Liu et al., 2021). Combining the SSD algorithm with an upgraded VGG backbone creates a robust model capable of detecting, locating, and classifying non-breeding waterfowl species in UAV imagery with extreme precision. By harnessing the power of deep learning and conquering the challenges posed by complex image analysis, this study aims to provide wildlife managers with a functional deep learning-based object detection framework for the automated identification of non-breeding waterfowl in UAV imagery.

2.2. METHODS

2.2.1. Analytical Objectives

I developed, tested, and implemented a deep learning-based object detection program for detecting non-breeding waterfowl from UAV imagery. Specifically, I modified the traditional SSD algorithm to use a deeper convolutional neural network (i.e., VGG16) backbone and then trained, optimized, and evaluated the object detection model using the UAV imagery of non-breeding waterfowl collected during the field portion of this project. I then combined the UAV imagery with an on-the-ground image dataset, re-trained using the amended dataset, and reevaluated model performance on unseen UAV imagery to assess whether adding very-high-resolution imagery (captured from the ground) during training improves model performance for UAV image detection. As such, the analytical objectives of this project were to:

- (1) train an optimized deep learning-based object detection model to accurately detect, localize, and identify species of non-breeding waterfowl in UAV imagery and

(2) amend the UAV imagery dataset with on-the-ground imagery of non-breeding waterfowl in the LMAV to compare model performance with the non-amended dataset. This section outlines the steps I implemented to train, optimize, and evaluate SSD300 with a VGG16 backbone in a PyTorch machine-learning framework. The associated code for the following sections can be found in Appendix A.

2.2.2. Processing UAV Imagery

Each instance of non-breeding waterfowl in the zoom imagery collected during fieldwork was annotated with a polygon mask using V7 Labs' DARWIN data annotation platform (Figure 2.1; DARWIN, 2023). Annotating each instance with a polygon mask provided the appropriate data for both object detection and instance segmentation frameworks without taking much longer than drawing a rectangular bounding box does. Object detection is the ability to classify multiple objects in an image and determine their locations by putting a bounding box around the objects, whereas instance segmentation adds a pixel mask for every detection that gives the shape of an object in addition to its class and location. This chapter focuses exclusively on object detection.

Non-breeding drakes were classified to species level wherever possible during data annotation. Non-breeding hens were grouped into one class due to the ambiguity of distinguishing between hen species in the UAV imagery. All other wildlife were classified to species level. Any images containing objects that were either too occluded or blurry to confidently determine the class were discarded from the annotation dataset during the review phase, in which every annotation was reviewed for classification accuracy. After all images containing detections were annotated and reviewed, the annotation data was exported in DARWIN 2.0 (JSON) format for reading into a PyTorch machine-learning framework (Paszke et al., 2019).



Figure 2.1. Sample image and ground-truth annotations after reading the dataset into Python. X- and Y-axis units are pixels.

Each zoom camera image had a separate JSON file with annotation data stored in nested array format. Each JSON file was read into Python version 3.9.18 as a nested array, flattened into a dictionary (i.e., normalized) so that each detection was stored as a unique observation, and all observations were finally appended to a Pandas data frame. The data frame was then filtered to include only the image name, class label, bounding box coordinates in PASCAL VOC format [x minimum, y minimum, x maximum, y maximum], and polygon mask path columns. The code used to read annotations and plot sample ground-truth polygons with their corresponding images can be found in Appendix A.1.

All UAV zoom imagery collected during the field portion of this project was consolidated into a single file directory. However, some images either did not contain any detections or contained non-breeding waterfowl that were too difficult to annotate confidently. Accordingly, it was necessary to filter the images folder that contained all zoom imagery to include only the images with corresponding annotations. A list of unique image file paths was created from the image name column in the annotation data frame and then used as an index to copy images with

detections from the original file directory to a new folder containing only images with corresponding annotation data.

Furthermore, every row in the annotation dataset containing a species label with less than 200 occurrences was removed to ensure the model had sufficient data for each class to learn from during the training and validation phases (Shahinfar et al., 2020). As such, each row in the data frame that contained a label value with less than 200 total instances in the dataset was removed. It was again necessary to filter the images directory to remove images that contained only species with less than 200 total instances in the dataset. Finally, the data frame containing annotations for species with greater than 200 occurrences in the dataset was saved as a CSV file in the same file directory as the folder containing all the corresponding images. The code used to filter the image collection can be found in Appendix A.2.

2.2.3. Data Pre-processing

PyTorch requires that class labels be numeric and not categorical. Therefore, Scikit-learn's *OrdinalEncoder* (Pedregosa et al., 2018) encoded categorical labels (e.g., MALL) in the data frame as numeric labels. It is important to note that class 0 is designated as the background class in PyTorch, so a value of one was added to the encodings to ensure that class 0 was reserved. Data augmentations are standard practice in computer vision and introduce controlled 'noise' into the training process, resulting in more robust models that generalize better (Shorten & Khoshgoftaar, 2019). Following the original SSD paper (Liu et al., 2016), I developed a data transformation pipeline for augmenting and transforming images and corresponding annotation data during loading into a custom PyTorch *Dataset* class.

The data augmentations applied during the training phase included *RandomZoomOut* and *RandomHorizontalFlip*. The *RandomZoomOut* augmentation randomly zoomed out of an image

using a minimum and maximum scale factor (i.e., side range) value of [1.0, 2.0], an occurrence probability of 0.3, and a fill value of [255, 20, 147] on the RGB scale—which translates to bright pink. A bright pink fill value was chosen to ensure that empty pixels created during a *RandomZoomOut* operation were filled with a color value that could not be easily confused with non-breeding waterfowl. The *RandomHorizontalFlip* augmentation randomly flipped a given image and its bounding boxes using an occurrence probability of 0.5. No data augmentations were applied to the dataset during the validation and testing phases.

In addition to data augmentations, transformations were needed to convert data into types suitable for processing by PyTorch models (Pedregosa et al., 2011). The transformations applied explicitly during the training phase included a *RandomIOUCrop* and *Resize* transformation. Each image was randomly cropped using a Jaccard index (Costa, 2021) across 40 trials, and the trial with the minimum Intersection-over-Union (IoU) overlap between ground truth objects was applied as the crop. Images were then resized to a uniform size of 300 by 300 pixels without maintaining the original aspect ratio. The only transformation applied specifically during the validation and testing phases was a *Resize* operation, which resized images to a uniform size of 300 by 300 pixels. A sample of each training transform and augmentation is shown in Figure 2.2.

Moreover, transformations applied during all phases—training, validation, and testing—included *ToImageTensor*, *ConvertImageDType*, *SanitizeBoundingBox*, and *Normalize*. PyTorch required that all images be image tensors containing float values; therefore, the *ToImageTensor* transform was used to transform all images into image tensors, and *ConvertImageDType* was used to convert all image tensor data into floating point values. The *SanitizeBoundingBox* operation removed bounding boxes and associated labels with any coordinate outside the image. PyTorch recommended that the *SanitizeBoundingBox* transform be applied at least once during the

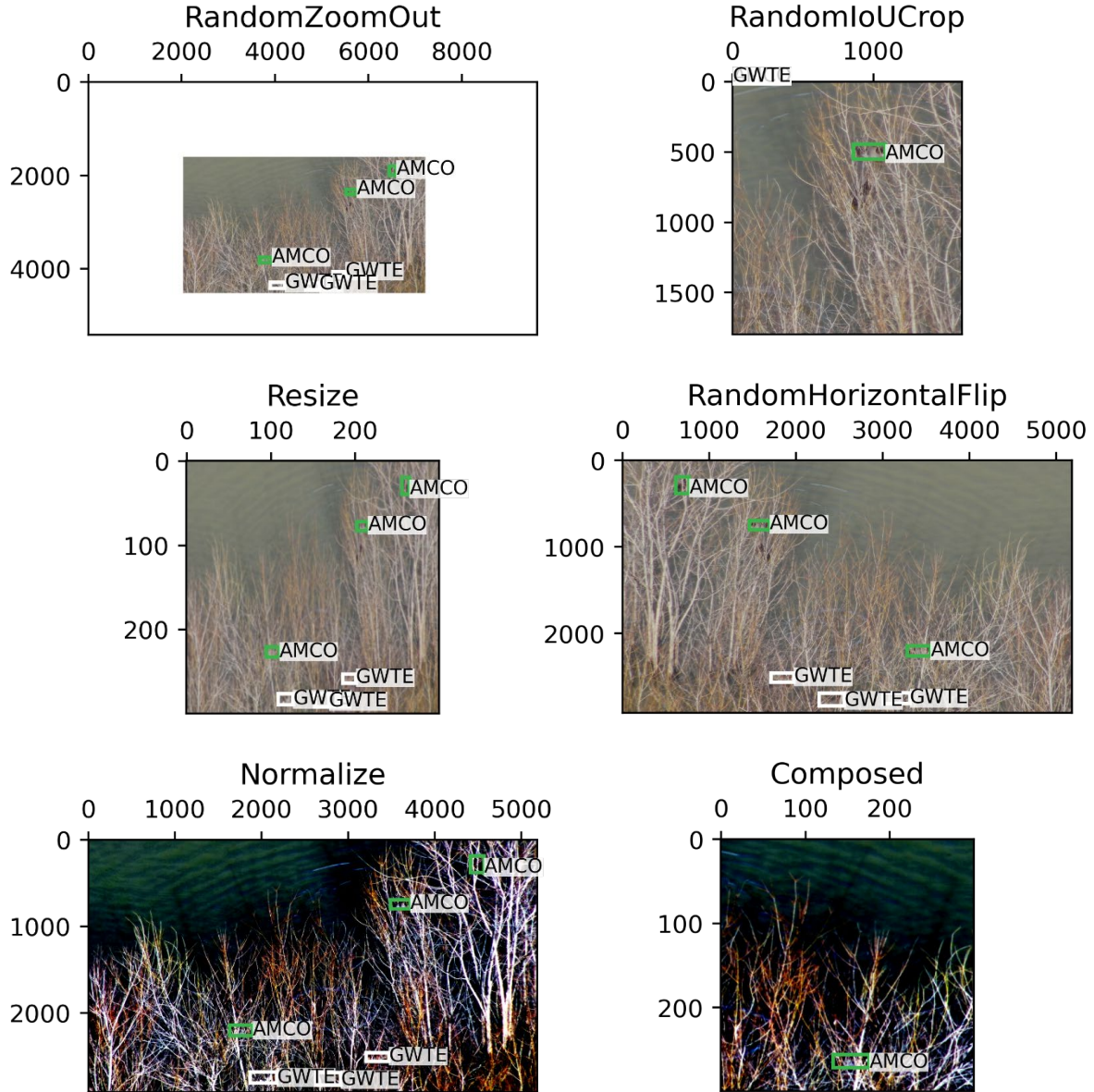


Figure 2.2. Each training transform applied to a sample image and its corresponding annotations. The last subplot titled ‘Composed’ applied all training transforms together to illustrate what a training sample looks like.

transformation pipeline to clean up out-of-image bounding box remnants resulting from image cropping operations. Finally, the floating point image tensor values were normalized using the mean and standard deviation values from the ImageNet (Russakovsky et al., 2015) dataset to normalize values to a range of $[0, 1]$. The ImageNet mean and standard deviation values were used to normalize the custom dataset because I implemented transfer learning using a VGG16 backbone

pre-trained on the ImageNet dataset. The code used to pre-process the object detection dataset can be found in Appendix A.2.

2.2.4. Loading Object Detection Dataset with PyTorch

A custom PyTorch *Dataset* class was written to load the object detection dataset and apply the appropriate transformation pipeline during each training, validation, and test phase. Specifically, the custom PyTorch *Dataset* class took three arguments: a path to the CSV file containing the annotation dataset, a path to the directory containing all corresponding images, and the data transformation function which accepted a Boolean value (i.e., true, false) of train or not. The main objective of the custom PyTorch dataset was to store each image and its corresponding annotation data as a list of dictionaries, where each list corresponded to one image, and each dictionary within the list contained annotation data for a single object in that image.

The dictionary keys required by PyTorch for SSD included *boxes*, *labels*, *image_id*, *area*, and *iscrowd*. The *boxes* dictionary key contained the bounding box coordinates in PASCAL VOC format (i.e., "XYXY"). The *labels* key contained the encoded numeric class labels. The *image_id* key contained the image index value in the custom PyTorch dataset. The *area* key contained the bounding box area, calculated by multiplying the bounding box width (i.e., maximum x minus minimum x) with the bounding box height (i.e., maximum y minus minimum y). Lastly, object instances that were part of a crowd were defined by the *iscrowd* key. For this project, all instances were considered not crowd.

The final step in the custom PyTorch dataset was applying the transformation pipeline to the images and target features in the dataset. If 'train' was set equal to true in the transformation function, all training augmentations and transformations were applied to the custom PyTorch dataset during loading. If false, all validation and test transformations were applied. I used the

same data transformation and augmentations as in the original SSD paper (Liu et al., 2016). The code used to load the custom object detection dataset into PyTorch can be found in Appendix A.2.

2.2.5. Defining SSD300 with VGG16 Backbone

The SingleShot MultiBox Detector (SSD) used a CNN backbone, passed an input image through it, and accepted the convolutional outputs known as feature maps from different levels of the network. These feature maps were then passed through the regression and classification heads responsible for predicting the box's location and class. Since the feature maps of each image contained outputs from different layers of the network, their size varied, and thus, they could detect objects of different shapes and sizes. Several default boxes were tiled on each feature map, which were considered rough prior guesses. For each default box, the SSD predicted whether there was an object, its class, and its offset.

The SSD network architecture was as follows. The *DefaultBoxGenerator* class was responsible for generating the default boxes of SSD. It produced a set of predefined boxes of specific width and height, tiled across the image as the first rough prior guesses of where objects might be located. The *DefaultBoxGenerator* class was parameterized by a set of hyperparameters (i.e., aspect ratios and scales) that controlled the shape and tiling of default boxes. The default hyperparameter values for the PyTorch *DefaultBoxGenerator* class were used as this implementation automatically provided good guesses by using a self-scaling function to estimate the best scales and ratios based on the input data (Paszke et al., 2019).

The *SSDMatcher* class then matched the default boxes to the ground truth boxes. After estimating the IoU of all possible combinations between the default and ground truth boxes, the matcher was used to find the best candidate ground truth for each default box with overlap higher than the IoU threshold. The final task of the *SSDMatcher* class was to find the default box with the

highest IoU for each ground truth box. The *SSDHead* class was responsible for initializing the classification and regression components of the network. The classification and regression heads, which came from the *SSDHead* class, predicted each object's class and location in every feature map. Each level of the feature map used a separate convolution to estimate class logits, which are the classes of the last layer in the neural network before applying an activation function, and bounding box locations. The number of predictions each head made per feature map level depended on the number of default boxes ($N = 8,732$ per class) and the sizes of the feature maps.

The *SSDFeatureExtractorVGG* class reconfigured and enhanced the standard VGG (Simonyan and Zisserman, 2015) backbone by including a series of four additional convolutional layers on top of the original VGG configuration. The VGG16 backbone accepted an input image of size (3, 300, 300), where (C, H, W) corresponded to image channels, image height, and image width. The image was then passed through the standard VGG configuration, followed by four additional convolutional layers. Specifically, each additional convolutional layer contained two convolutions of varying-sized feature maps and kernels. Feature map sizes across the four extra convolutional layers included [128, 256, 512, 1,024], while kernel sizes for each convolution were either 1 or 3. Each convolution was followed by a ReLU activation function (Agarap, 2019), which performed a nonlinear transformation of each neuron's weights to derive a value of either 0 (i.e., negative weights) or 1 (i.e., positive weights). Moreover, the ReLU activation helps overcome the vanishing gradient problem associated with training deep neural networks (Liu et al., 2021).

The final layer in the VGG16 architecture is a fully connected layer containing the configured *SSDHead* class with an output size of 1. Similar to other detection models, a set of arguments parameterized the SSD algorithm. For example, the mandatory parameters were the backbone responsible for estimating the feature maps (i.e., VGG16), the anchor generator, which

was a configured instance of the *DefaultBoxGenerator* class, the dimensions for resizing images, and the number of classes for the classification task (including background class). Lastly, the SSD algorithm reused the standard *BoxCoder* class used by other detection models, which encoded and decoded the bounding boxes and used the same prior variances as the original SSD implementation.

Two core methods of my SSD implementation were the loss computation and forward methods. Since the object detection program's objective was to localize and classify detected objects, two types of loss—bounding box regression and classification loss—must be considered. Thus, the L2-Norm loss was used to calculate bounding box offset, while the standard categorical cross-entropy loss with hard-negative sampling (Robinson et al., 2021) was used to calculate classification loss. The total loss value was computed by summing the bounding box regression and classification losses. As in all detection models, the forward method passed the input images through the VGG16 backbone to get feature maps. The feature maps were then passed through the head to get the predictions, the stage at which default boxes were generated.

If the model was in training mode, the forward method estimated the IoUs of the default boxes with the ground truth, used the *SSDMatcher* class to produce matches, and finally estimated the losses by calling the loss computation method. If the model was in evaluation mode, the model first selected the best detections by keeping only the ones that passed the default PyTorch score threshold (i.e., 0.01), selected the most promising boxes, and then ran non-maximum suppression (Hosang et al., 2017) to clean up and retain only the best predictions. Finally, I post-processed the predicted bounding boxes to resize them to the original image size. The code for defining the SSD300 with the VGG16 backbone model can be found in Appendix A.2.

2.2.6. Bayesian Hyperparameter Optimization using Ray Tune

Hyperparameters are the model parameters that cannot be learned during machine learning training—they must be defined at the start of model training and directly affect training outcomes (Yu & Zhu, 2020). As such, hyperparameter tuning can distinguish between an average and highly accurate model. Often, simple things like choosing a different learning rate or changing a network layer size can dramatically impact model performance. Fortunately, there are tools for distributed hyperparameter tuning. I selected Ray Tune (Liaw et al., 2018) as the hyperparameter tuning environment because it included the latest hyperparameter search algorithms, integrated well with PyTorch workflows, and natively supported distributed training. The general process for hyperparameter tuning with Ray Tune included four main steps.

First, the custom PyTorch data loaders were wrapped in a function that took two paths: one path to the folder containing all the images and another to the CSV containing all the corresponding annotation data. Next, Ray Tune needed to know which model hyperparameters were configurable. The stochastic gradient descent (SGD) optimization algorithm with momentum (Ruder, 2017) was used as the iterative method for optimizing the objective function (i.e., maximizing validation precision) during the training process. Accordingly, the learning rate, momentum, weight decay, step size, and gamma hyperparameters of the SGD optimizer and step scheduler were configurable, while batch size remained constant at 16.

I then wrapped the PyTorch training script in a function that took a dictionary containing the configurable hyperparameters and their corresponding search space. The CSV file and image folder paths containing all the data were also required inputs for the PyTorch training wrapper function. The search function used for hyperparameter tuning was the Bayesian Optimization search algorithm (Turner et al., 2021). Finally, I compiled the data loading, training, and search

functions into one main function, which allowed me to define the number of trials to compare and how many epochs to monitor each trial for during tuning. I compared 30 trials for up to 30 epochs each during hyperparameter tuning until the configuration that provided the highest validation precision was found. The code for Bayesian hyperparameter optimization with Ray Tune can be found in Appendix A.3.

2.2.7. Training SSD300 with VGG16 Backbone

After determining which combination of hyperparameters provided the highest validation precision given the custom dataset and network configuration, I stored the optimized hyperparameters (Table 2.1) as variables at the beginning of the training program. Following the methods outlined in Smith et al. (2018), I looped through a list of dictionaries specifying batch size, number of training epochs, and number of batches to accumulate gradients for each of the three training steps (Table 2.2). Incrementally increasing batch size throughout the training process can yield notable improvements in model performance while significantly reducing training costs (Smith et al., 2018), and I decided to construct the training schedule so that batch size quadrupled at each of the three training steps. However, the computer's GPU memory maxed out after the first training step at a batch size of 64, so it was necessary to implement the gradient accumulation technique (Hermans et al., 2017) during the training process to simulate larger batch sizes for the second and third steps. The Python source file containing functions for gradient accumulation-enabled training in PyTorch can be found in Appendix A.4.

Table 2.1. The optimal combination of hyperparameter values used to train SSD300 with VGG16 on the amended dataset containing both UAV and on-the-ground imagery of non-breeding waterfowl. These values were determined using a Bayesian Optimization search algorithm before training the model.

Gamma	Learning Rate	Momentum	Step Size	Weight Decay
0.39259	0.01002	0.75711	21.8483	0.00465

Table 2.2. The training schedule used for training SSD300 with VGG16 backbone on the image dataset containing UAV imagery of non-breeding waterfowl. GPU memory maxed out during training at a batch size of 64. Therefore, the gradient accumulation optimization technique was employed to simulate large batch training. For example, a batch size of 64 across 16 gradient accumulations is equivalent to training with a batch size of 1,024.

Step	Accumulations	Batch Size	Epochs
0	1	16	15
1	1	64	15
2	4	256	10
3	16	1,024	5

The main training function is as follows. Instead of random weights, I initialized the SSD300 with VGG16 using the weights from two pre-trained networks. First, the SSD300 algorithm was initialized using the COCO V1 (Lin et al., 2015) weights. Second, the VGG16 backbone was initialized using the ImageNet (Russakovsky et al., 2015) weights. Initializing models using pre-trained weights is referred to as fine-tuning transfer learning, and it drastically decreases training times and required dataset sizes by using an already good set of weights that were learned from an extensive dataset, such as COCO or ImageNet, and then fine-tuning those weights for a specific task (Ouyang et al., 2016). After initializing the model, I created three separate dataset instances—training, validation, and testing—and loaded them into PyTorch following the methods outlined in Section 2.2.4. I used a training, validation, and testing split of 80, 15, and 5%, respectively (Jangblad, 2018). To subset the data, I stored the image indices in a random order list, then indexed the indices list to subset the first 80% percent as the training dataset, the next 15% as the validation dataset, and the remaining 5% as the testing dataset.

I then moved the model from the CPU to the GPU for faster image processing and network training speeds. After the model was moved to the GPU, the SGD with momentum optimizer was constructed using the model parameters, learning rate, momentum, and weight decay values. The last step before beginning the main training loop, which iterates through the defined number of epochs in the training steps list (Table 2.2), was constructing a learning rate scheduler using the optimizer, step size, and gamma values as parameters. I used PyTorch's *StepLR* class as the learning

rate scheduler, which decayed the learning rate of each parameter group by gamma every step size number of epochs.

The main training loop iterated through a range of [0, number of training epochs]. It consisted of two main functions—*train one epoch* and *evaluate*—which were imported from the Python source file in Appendix A.4. The *train one epoch* function trained the model on the custom dataset for one epoch. It was parameterized by the training model, optimizer, data loader (i.e., training set), number of batches to accumulate gradients before updating weights, the device that training took place on (i.e., GPU), and number of training epochs. Additionally, I periodically monitored the training process outputs (e.g., training and validation loss) to ensure the model converged as anticipated.

After training for one epoch, the *evaluate* function evaluated the training performance on the validation dataset using the mean average precision (mAP) and mean average recall (mAR) metrics, which provided an unbiased estimate of model fit for tuning model hyperparameters in real time. Precision is the ratio of true positives to total positives (i.e., true plus false), while recall is the ratio of true positives to the total number of relevant objects. Precision can be considered the model's ability to accurately identify relevant objects, while recall can be considered the model's ability to detect all relevant objects. The *evaluate* function was parameterized by the model for evaluation, data loader to use (i.e., validation set), and device to evaluate the validation set on (i.e., GPU). After finishing the evaluation for that epoch, model weights were saved, and training commenced for the next epoch. This process repeated until the defined number of epochs was met. The code for training SSD300 with a VGG16 backbone can be found in Appendix A.2.

2.2.8. Evaluating Performance on Test Dataset

After the model converged and training was completed, I tested the model performance on a subset of data that the model had not yet seen—the test dataset—to gain an unbiased estimate of model performance. Specifically, I configured an instance of TorchMetrics' (Detlefsen et al., 2022) *MeanAveragePrecision* class to calculate mAP at minimum IoU thresholds of 0.5 and 0.75 and, additionally, mAR at 100 maximum detections for each image in the test dataset using model predictions and ground truth features. A minimum IoU threshold of 0.5 would disregard any predicted box with a ratio of intersection (i.e., overlap) to union (i.e., total combined area) less than 50% relative to the ground truth box. An IoU score of 1 indicates perfect overlap between the predicted and ground truth boxes, while 0 means no overlap between boxes.

In addition to total precision and recall, I calculated the precision at minimum IoU threshold of 0.75 and recall at 100 maximum detections metrics per class to get a more detailed report of each non-breeding waterfowl species' performance. Each of these metrics and the image name were then stored in a Pandas data frame and averaged across every image in the test dataset to get the final estimates of model performance. The code for performing inference on the test dataset can be found in Appendix A.5.

2.2.9. Incorporating Citizen Science with Object Detection

In addition to the UAV field imagery, I received 1,001 professionally taken high-resolution photographs of non-breeding waterfowl in the LMAV, which not only boosted dataset size by nearly 50% but also allowed for assessing whether amending the field dataset with high-resolution on-the-ground imagery improves model performance for UAV image detection. The methods outlined in Sections 2.2.2 through 2.2.8 were repeated for the professional, high-resolution images containing non-breeding waterfowl. Specifically, all on-the-ground imagery was annotated using

DARWIN and exported in JSON format. The on-the-ground image dataset was then read into Python and filtered to include the same classes as in the filtered UAV image dataset. Once filtered, data were pre-processed and loaded into a custom PyTorch Dataset.

The process for combining the on-the-ground image dataset with the UAV image dataset is as follows. The test dataset referenced in Section 2.2.8 remained the same—it contained only the UAV images used to test the model's performance trained on the non-amended dataset. The remaining 95% of UAV images (i.e., the training and validation datasets) were combined with the full on-the-ground image dataset and subset into respective training and validation datasets using 80 and 20% of the amended image dataset, respectively. The same training approach was implemented as with the UAV image dataset; however, I reused the same hyperparameter values used for training the model on the UAV image dataset (Table 2.1). I then trained SSD300 with a VGG16 backbone on the amended dataset following the same training schedule as the strictly UAV imagery dataset (Table 2.2).

The SSD model was trained across 45 epochs using gradient accumulation to simulate large batch training up to batch sizes of 1,024. The initial batch size was 16 and quadrupled at each step—the 15th, 30th, and 40th epochs, respectively. After the training, I computed the final performance metrics on the test dataset to get performance estimates for the model trained on the amended image dataset. It was necessary to ensure that the test dataset contained no on-the-ground imagery to assess any improvement in UAV detection accuracy. I then compared the accuracy metrics outlined in Section 2.2.8 for both models—UAV and UAV plus ground imagery—to see if any notable differences existed in the models' ability to detect non-breeding waterfowl species in UAV imagery.

2.3. RESULTS

2.3.1. Data Annotation

The image annotation efforts for the field-collected UAV imagery resulted in 11,283 total wildlife detections (Figure 2.3). Of the total detections, 10,216 were non-breeding waterfowl, 919

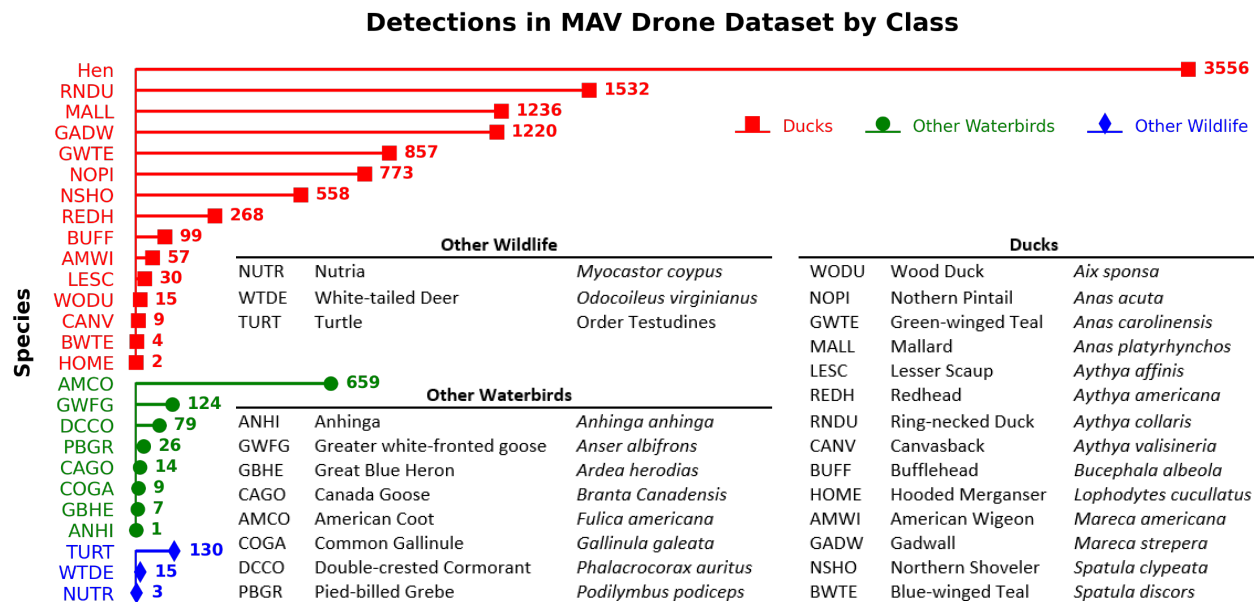


Figure 2.3. Manual counts derived from data annotation efforts. Species are grouped into three guilds: ducks, other waterbirds, and other wildlife. I annotated 11,283 detections across 26 different classes in 2,051 UAV images.

were other waterbirds, and 148 were other wildlife. I detected 26 classes between all 2,051 images in the dataset, with class detection frequencies ranging from one to 3,556. After filtering the UAV image annotation dataset to only include classes with more than 200 detection instances, nine classes remained in the dataset. However, despite 3,556 detections in the 'Hen' class, I removed them from the dataset before training to limit the amount of uncontrolled noise introduced into the training process. The final annotation dataset contained seven species of non-breeding waterfowl and one species of other waterbirds—American coot (*Fulica americana*), gadwall (*Mareca strepera*), green-winged teal (*Anas carolinensis*), northern pintail (*Anas acuta*), northern shoveler (*Spatula clypeata*), mallard (*Anas platyrhynchos*), redhead (*Aythya americana*), and ring-necked duck (*Aythya collaris*)—for a total of 7,100 detections.

Similarly, the image annotation efforts for the high-resolution on-the-ground imagery (Figure 2.4) resulted in 3,947 additional wildlife detections. Of the total detections, 3,945 were non-breeding waterfowl, and two were other waterbirds. I detected 18 classes in the on-the-ground image dataset, with class detection frequencies ranging from one to 1,112. After filtering, the on-the-ground image dataset was reduced to 1,869 detections to include only the same eight classes as the filtered UAV image dataset.

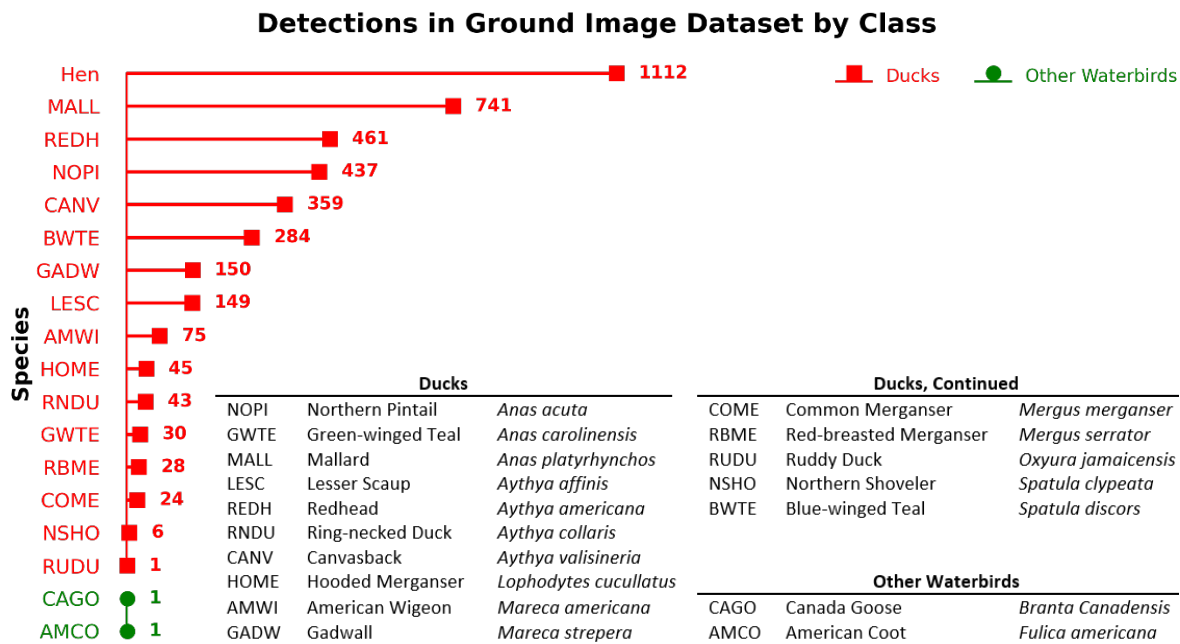


Figure 2.4. Manual counts derived from data annotation efforts on the ground imagery dataset. Species are grouped into two guilds: ducks and other waterbirds. I manually annotated 3,947 detections across 18 different classes in 1,001 images captured by professional photographers on the ground.

2.3.2. Inference with Trained Model

After training the SSD300 with VGG16 backbone on the UAV image dataset, I tested model performance (Table 2.3) by comparing model predictions (Figure 2.5) on the test dataset—which the model had not seen during training or validation phases—and the ground truth data. The mean precision at an IoU threshold of 0.5 was 99.1% (images ranged from 98.4 to 100%), while the mean precision at an IoU threshold of 0.75 was 92.8% (images ranged from 81.9 to 100%) when averaging all images in the test dataset. That is, the model correctly identified 99.1 and 92.8%

of everything it detected with a ratio of intersection to union greater than or equal to 0.5 and 0.75, respectively, when averaged across the entire test dataset. Similarly, the mean recall at 100 maximum detections for images in the test dataset was 82.9% (images ranged from 75.8 to 89.2%). In other words, the model correctly detected 82.9% of the total relevant objects in the test dataset when making up to 100 predictions per image.

Table 2.3. Results from inference on the UAV image dataset for the model trained exclusively on UAV images. mAP at IoU thresholds of 0.5 and 0.75 and mAR at 100 maximum detections are included. Mean, minimum, and maximum values are given as percentages for the entire test dataset to show variance.

	mAP 50	mAP 75	mAR 100
Mean	99.1	92.8	82.9
Minimum	98.4	81.9	75.8
Maximum	100	100	89.2

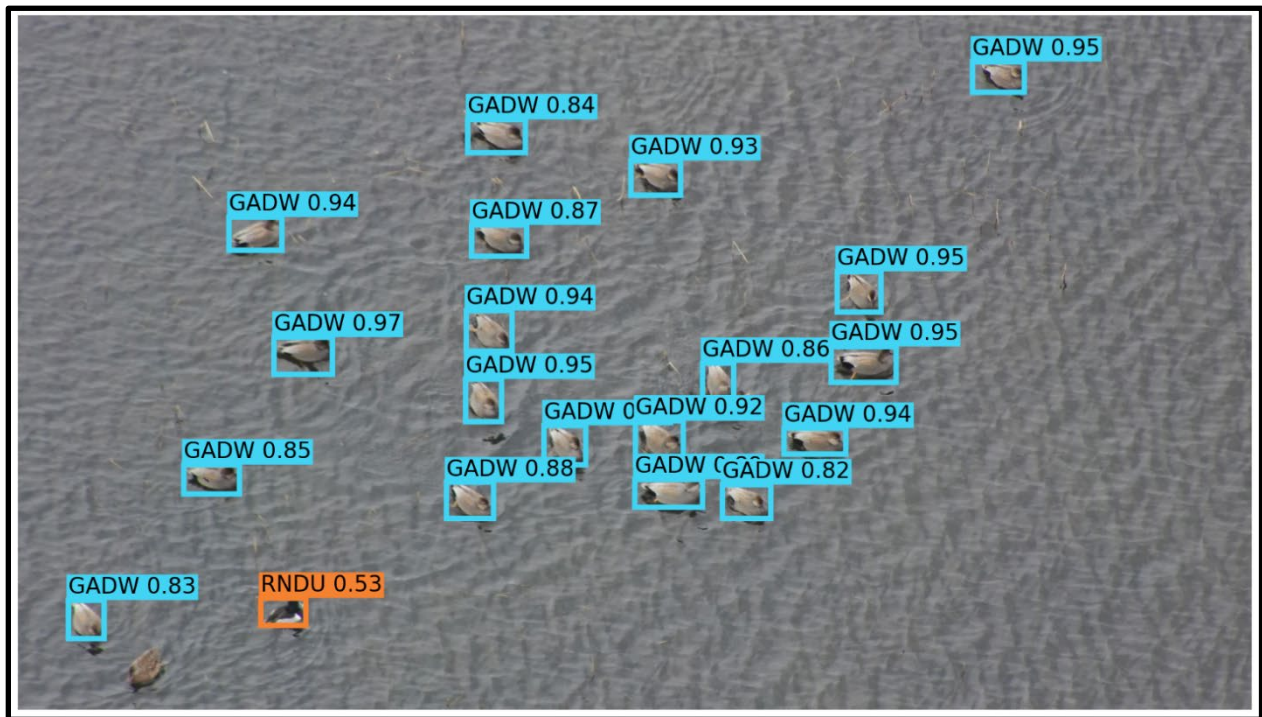


Figure 2.5. Sample predictions on an image from the test dataset using the model trained exclusively on UAV imagery. Predictions have been post-processed using non-maximum suppression and filtered to remove any predictions with confidence scores below 0.5. The model correctly predicted the locations and classes of 19 gadwall and one ring-necked duck. The 'Hen' class was removed from the dataset, so the model was correct not to identify it.

I also computed precision (IoU threshold of 0.75) and recall (100 maximum detections) values for each class (Table 2.4) in the UAV image test dataset. The per-class precision ranged

from 65.3% for northern pintail to 87.4% for northern shoveler. Similarly, the per-class recall ranged from 69.7 to 88.6% for northern pintail and shoveler. The per-class precision and recall values were calculated by averaging each metric across every class instance in the test dataset. The per-class metrics in Table 2.4 can potentially fall outside the per-image ranges reported in Table 2.3 because the per-image ranges are calculated by finding precision for each class and then taking a weighted average across all classes in the image. As such, class imbalances in each image could potentially result in per-class precision and recall values that fall below the per-image range when averaged across the test dataset. Lastly, I had no instances of redhead in the test dataset.

Table 2.4. Per-class metrics from inference on the test dataset. Average mAP at IoU threshold of 0.75 and mAR for each class across the test dataset are included as percentages.

	AMCO	GWTE	MALL	NOPI	NSHO	RNDU	GADW
mAP	86.1	84.8	82.5	65.3	87.4	80.4	84.5
mAR	87.4	87.4	83.8	69.7	88.6	83.2	86.9

To assess whether amending the UAV image dataset with high-resolution on-the-ground imagery improves model performance for detecting non-breeding waterfowl in UAV imagery, I re-trained the SSD300 with the VGG16 backbone model on the amended image dataset. After training the model, I again tested model performance (Table 2.5) by comparing model predictions on the test and ground truth data. It is important to note that the test dataset contained only UAV images. The mean precision at an IoU threshold of 0.5 was 22.2% (images ranged from 7.1 to 100%), while the mean precision at an IoU threshold of 0.75 was 12.7% (images ranged from 3.2 to 50.5%). Similarly, the mean recall at 100 maximum detections for samples in the test dataset was 22.6% (images ranged from 16.4 to 55.0%).

Table 2.5. Results from inference on the UAV image dataset for the model trained on the amended dataset. mAP at IoU thresholds of 0.5 and 0.75 and mAR at 100 maximum detections are included. Mean, minimum, and maximum values are given as percentages for the entire test dataset to show variance.

	mAP 50	mAP 75	mAR 100
Mean	22.2	12.7	22.6
Minimum	7.1	3.2	16.4
Maximum	100	50.5	55.0

Once again, I computed precision and recall values for each class (Table 2.6) in the amended image dataset. The per-class precision ranged from 0% for northern pintail and shoveler to 48.4% for gadwall. Similarly, the per-class recall ranged from 0 to 60% for northern shoveler and gadwall, respectively.

Table 2.6. Per class metrics from using the model trained on the amended dataset to perform inference on the test dataset. Average mAP at IoU threshold of 0.75 and mAR for each class across the test dataset are included as percentages.

	AMCO	GWTE	MALL	NOPI	NSHO	RNDU	GADW
mAP	8.5	6.1	8.9	0	0	20.2	48.4
mAR	18.0	13.9	11.9	2.5	0	52.3	60

2.4. DISCUSSION

2.4.1. Summary of Main Findings

The results of this study underscore the effectiveness of the SSD300 with VGG16 model trained on strictly UAV imagery compared to the model trained on an amended dataset with high-resolution ground imagery. Specifically, the precision at an IoU threshold of 0.5 for the UAV image dataset was an impressive 99.1%, even after just 45 training epochs, whereas the amended image dataset yielded a significantly lower precision of 22.2%. Additionally, the recall at 100 maximum detections for the UAV image dataset reached 82.9%, while the amended image dataset achieved a recall of only 22.6%. Remarkably, despite the amendment of the training dataset with an additional 1,869 annotations from high-resolution imagery, the model trained exclusively on UAV imagery outperformed its counterpart by almost 77% in precision at an IoU threshold of 0.5.

In evaluating per-class precisions and recalls at an IoU threshold of 0.75, notable disparities emerged between the model trained on UAV imagery and the model trained on the amended dataset. The former exhibited superior performance across various non-breeding waterfowl species, with per-class precision ranging from 65.3% for northern pintail to 87.4% for northern shoveler. In contrast, the model trained on the amended image dataset displayed considerably lower performance, with per-class precision ranging from 0% for northern pintail and shovelers to 48.4% for gadwall. Furthermore, per-class recall at 100 maximum detections for the model trained on strictly UAV imagery ranged from 69.7% for northern pintail to 88.6% for northern shoveler. In comparison, the model trained on the amended image dataset struggled, with per-class recall ranging from 0% for northern shoveler to 60% for gadwall. These results unequivocally indicate that amending a UAV image dataset with extremely high-resolution ground imagery does not enhance non-breeding waterfowl UAV detection performance and leads to significantly poorer results.

The model trained on the amended dataset, which included very high-resolution on-the-ground images alongside UAV imagery, failed to improve UAV detection performance for several reasons. First, the nature of the images in the ground image dataset differed significantly from those in the UAV dataset, as many featured in-flight waterfowl, which were scarce in the UAV dataset. Second, the substantial difference in resolution between UAV and ground images likely exacerbated the detection challenges for UAV imagery. While it cannot be definitively concluded that amending the training dataset with non-UAV imagery is ineffective, it is recommended that future attempts consider using on-the-ground imagery captured with a resolution similar to the UAV sensor (e.g., 15 MP). Moreover, it is critical that the target features in the ground dataset closely resemble those in the UAV dataset for effective amendment. Using a dataset primarily of

in-flight non-breeding waterfowl to amend a UAV image dataset primarily featuring flightless non-breeding waterfowl proved unsuccessful for this study. Nevertheless, this study firmly establishes that training the SSD300 with the VGG16 backbone model exclusively with UAV imagery provides impressive results using this framework, even in cases where imagery is limited.

The variations in detection performance among different species of non-breeding waterfowl cannot be attributed solely to sample size effects but are likely influenced by data quality and occlusion issues. While it was observed that northern pintail had more occurrences in the UAV image dataset than northern shoveler (Figure 2.3), the latter outperformed the former by over 22% precision, suggesting that the differences in performance are not exclusively due to sample size variations. Instead, the disparities are predominantly related to data quality, particularly the occlusion effect. For example, a substantial portion of northern pintail instances in the UAV image dataset suffered from occlusion, where only a fraction of the body was visible due to feeding or being partially out of frame. This data quality issue noticeably impacted model performance for the northern pintail class, as the SSD300 with VGG16 struggled to identify partially occluded pintails during inference. Additionally, it is likely that a combination of habitat and atmospheric conditions, rather than habitat quality alone, contributes to the differences in species performance. Therefore, the capacity for machine-learning algorithms to correctly classify waterfowl may in fact depend on species-specific behaviors and habitat use.

2.4.2. Drawbacks and Advantages of PyTorch

The custom PyTorch approach, while yielding impressive results, had its challenges. The most significant drawback was the time invested in learning the methods, writing thousands of lines of code, and debugging everything myself, as compared to using proprietary software like ArcGIS Pro. However, the PyTorch approach offered several notable advantages that outweighed

the drawbacks—including unmatched flexibility in pre-processing data, applying transformations, modifying model architecture, and fine-tuning hyperparameters while tailoring the model for non-breeding waterfowl detection. PyTorch models' compatibility and comparability streamlined the assessment of different approaches, ensuring the chosen method was well-suited. Additionally, PyTorch's efficiency in handling computer vision tasks minimized computational costs and expedited the processing of complex imagery from UAV surveys.

The success of the approach employed in this study can be attributed to several key factors and strategies implemented during the development of the object detection model. Enhancing the standard SSD300 VGG backbone with a deeper architecture contributed significantly to model performance, although it is worth noting that there is a balance to strike in terms of model depth, as deeper models require longer training times and struggle with challenges like the vanishing gradient problem. Initialization of the model's regression and classification heads using weights from pre-trained models on COCO and ImageNet datasets expedited the learning process. Furthermore, fine-tuning, a transfer learning technique, was applied during training by freezing all pre-trained layers of the feature extractor except for the last, which allowed the model to adapt to specific tasks even with limited training data quickly. The use of a Bayesian optimization search algorithm (Turner et al., 2021) for hyperparameter tuning, normalization of the custom dataset using mean and standard deviation values from the ImageNet dataset, accumulation of gradients (Hermans et al., 2017) to simulate large batch training (Smith et al., 2018), and post-processing with non-maximum suppression (Hosang et al., 2017) all played crucial roles in improving model performance as well.

2.4.3. Conclusion

The original hypothesis of this thesis posited that UAV-based waterfowl surveys are suitable for providing necessary site-level metrics to inform and monitor conservation delivery programs, and that deep learning-based methods for object detection are a promising means for processing the vast amounts of complex imagery collected during UAV surveys. This study validates the original hypothesis by demonstrating that UAV-based surveys, in combination with the developed deep learning-based object detection framework, offer a powerful solution for effectively surveying non-breeding waterfowl in structurally complex and difficult-to-access habitats such as the MAV.

The functional deep learning-based object detection framework developed in this study can be operationalized by waterfowl managers and researchers alike. This tool provides an efficient and cost-effective means for counting waterfowl on project sites, enhancing the ability to assess waterfowl response to wetland restoration efforts, and enabling the estimation of site-level metrics such as species diversity and abundance. When combined with other data streams of interest, such as time since restoration or proximate metrics of habitat quality, the analysis framework developed here streamlines cause-and-effect evaluations of waterfowl habitat use to inform management best practices.

2.4.4. Future Directions

The current state of the 'Drone Duck Detection' tool is a stand-alone web app (<https://drone-duck-detection.uc.r.appspot.com/>) that can effectively and efficiently detect seven species of male non-breeding waterfowl—green-winged teal, mallard, northern pintail, northern shoveler, ring-necked duck, and gadwall—and one species of other waterbirds—American coot—from UAV imagery. Waterfowl managers and researchers can use this web application to quickly analyze UAV

images containing the eight species of waterbirds that the final model was trained to detect. For example, managers can follow the UAV-based survey protocol I developed to survey restored wetland sites in the LMAV for non-breeding waterfowl using a UAV equipped with a camera sensor with a similar resolution (i.e., 20 MP) as used in this study. After surveying a restored wetland site, managers can upload their images to the Drone Duck Detection web app, which then deploys the final object detection model to identify and locate the species of waterbirds it was trained to detect. The output is an image containing predicted bounding boxes, species labels, and confidence scores. If the survey protocol outlined in Chapter 1 is followed, the methods provide managers with a framework for estimating relative abundance and species diversity at the site level. Critically, the UAV pilot or image analyst need not have any experience in waterfowl identification, meaning that such surveys could be outsourced to contractors to evaluate duck habitat use at landscape scales. When combined with other data streams of interest, this tool can be used to inform future management endeavors and refine the conservation delivery process by determining which methods of restoration are most productive from a wintering waterfowl standpoint.

Although the Drone Duck Detection tool can accurately detect eight species of waterbirds when making predictions on UAV imagery of similar resolution and scale as the imagery used to train the deep learning-based object detection model, there is still ample room for improvement. For example, this study's UAV image training dataset contained 26 different wildlife species but only had sufficient training data for eight species. The number of species the Drone Duck Detection application can detect can be increased by collecting more training data—solicited duck pics—for species beyond the eight currently included. Furthermore, the training dataset could be refined by replacing sub-par imagery (e.g., blurry or occluded objects, out-of-focus images) with higher-

quality images and re-training the model using the deep learning-based object detection framework outlined in Chapter 2. Increasing the quality of the data that the model is trained on should translate to increased precision, recall, and confidence score values when evaluating the final model performance on the test dataset.

Moreover, adding UAV imagery collected with camera sensors other than the one employed in this study to the training dataset will increase model transferability by diversifying the image resolutions and scales that the model is trained to detect. In theory, a model trained on a wider array of image resolutions and scales should generalize better, and therefore be less dependent on the specifications of the single camera employed in this study—which alleviates the financial burden of purchasing the highest-quality sensor on the market. More training data (of higher quality) generally increases model robustness and generalizability.

After spending the past couple of years comparing various deep learning-based algorithms for UAV-based monitoring of non-breeding waterfowl, I am confident that video-based monitoring is the most promising path forward if UAV surveys and automated object detection is to be applied at scale across the North American flyways. One major pitfall of the image-based object detection program I developed is the risk of double-counting individuals, which introduces errors in relative abundance estimates. However, this error can be corrected in a video-based object detection system by dropping anchors on each newly detected instance and then tracking each anchor throughout the video. For example, the Channel and Spatial Reliability Tracker (CSRT) for object tracking and counting (Dirir et al., 2021) uses deep learning concepts and correlation filters for object counting and tracking in video-based object detection systems and can be employed to reduce counting errors. As such, when compared with image object detection systems, video object detection systems minimize the error with which objects are counted, providing the most reliable

relative abundance and diversity estimates. Conveniently, a model trained on UAV images can be deployed to make video predictions because a video is simply a container of images in sequential order written at a specific frame rate. The code for performing inference on a video dataset with the model trained on imagery can be found in Appendix A.6.

In order to scale this framework across each of the North American flyways, I strongly recommend a collaborative effort among the JVs to assemble and contribute annotated UAV images and videos containing sufficient training data for all species of interest to a nationwide database, which can then be used to train a model using the framework I developed for deep-learning based object detection. An object detection model fully trained to recognize all species of North American waterfowl would be an incredibly valuable tool. For example, UAV images (or videos) could be collected as part of pair surveys such as the four square-mile surveys conducted by the USFWS Habitat and Population Evaluation Team, or the May breeding population and habitat survey visibility correction transects. This may be especially relevant given new foot-access restrictions on private land for such surveys. Additionally, if the object detection model can be trained to speciate female waterfowl—which is certainly feasible, given that humans can do it—then UAV surveys for duck broods could be conducted at scale to evaluate duck productivity in reference to any particular management question of interest. There are clearly numerous research and management applications that would be made possible through the development of a fully trained object detection model that extends the first version developed here.

APPENDIX A

A.1. Jupyter Notebook containing the code used to load and clean the annotation dataset

https://github.com/ZackLoken/MAV_Dronework_ImageOD/blob/main/SSD_PyTorch/Annotation_PreProcessing.ipynb

A.2. Jupyter Notebook containing the code used to pre-process the image and annotation datasets and train SSD300 with VGG backbone in a PyTorch machine learning framework

https://github.com/ZackLoken/MAV_Dronework_ImageOD/blob/main/SSD_PyTorch/SSD_VGG_PyTorch_Training.ipynb

A.3. Jupyter Notebook containing code for hyperparameter tuning with Ray Tune using a Bayesian optimization search algorithm in a PyTorch machine learning framework

https://github.com/ZackLoken/MAV_Dronework_ImageOD/blob/main/SSD_PyTorch/SSD_VGG_PyTorch_HyperparameterTuning.ipynb

A.4. Python source file containing functions for gradient accumulation enabled object detection training and evaluation in PyTorch

https://github.com/ZackLoken/MAV_Dronework_ImageOD/blob/main/SSD_PyTorch/engine_gradientAccumulation.py

A.5. Jupyter Notebook containing code for performing inference on an image dataset in PyTorch

https://github.com/ZackLoken/MAV_Dronework_ImageOD/blob/main/SSD_PyTorch/SSD_VGG_PyTorch_Predictions.ipynb

A.6. Jupyter Notebook containing code for performing inference on a video dataset in PyTorch

https://github.com/ZackLoken/MAV_Dronework_ImageOD/blob/main/SSD_PyTorch/SSD_VGG_PyTorch_Predictions_Videos.ipynb

LITERATURE CITED

- Aburasain, R. Y., Edirisinghe, E. A. & Albatay, A. Drone-based Cattle Detection using Deep Neural Networks. in Proceedings of SAI Intelligent Systems Conference 598–611 (2021).
- Agarap, A. F. Deep Learning using Rectified Linear Units (ReLU). Preprint at <http://arxiv.org/abs/1803.08375> (2019).
- Andersson, K., Davis, C. A., Harris, G. & Haukos, D. A. An Assessment of Non-breeding Waterfowl Surveys on National Wildlife Refuges in the Central Flyway: Waterfowl Survey Quality. *Wildlife Society Bulletin* 39, 79–86 (2015).
- Barnas, A. F., Darby, B. J., Vandeberg, G. S., Rockwell, R. F. & Ellis-Felege, S. N. A Comparison of Drone Imagery and Ground-based Methods for Estimating the Extent of Habitat Destruction by Lesser Snow Geese (*Anser caerulescens caerulescens*) in La Pérouse Bay. *PLoS ONE* 14, 8 (2019).
- Björn, L. O. & Nilsson, A. M. Thermal Emissivity: Basics, Measurement, and Biological Examples. *Bioinspired Engineering of Thermal Materials* 7, 159-174 (2018).
- Bushaw, J. D., Ringelman, K. M., Johnson, M. K., Rohrer, T. & Rohwer, F. C. Applications of an Unmanned Aerial Vehicle and Thermal-imaging Camera to Study Ducks Nesting Over Water. *Journal of Field Ornithology* 91, 409–420 (2020).
- Bushaw, J. D., Ringelman, K. & Rohwer, F. Applications of Unmanned Aerial Vehicles to Survey Mesocarnivores. *Drones* 3, 28 (2019).
- Bushaw, J. D., Terry, C. V., Ringelman, K. M., Johnson, M. K., Kemink, K. M., & Rohwer, F. C. Application of Unmanned Aerial Vehicles and Thermal Imaging Cameras to Conduct Duck Brood Surveys. *Wildlife Society Bulletin* 45(2), 274-281 (2021).
- Chabot, D. & Bird, D. M. Evaluation of an Off-the-shelf Unmanned Aircraft System for Surveying Flocks of Geese. *Waterbirds* 35, 170–174 (2012).

- Chabot, D. & Bird, D. M. Wildlife Research and Management Methods in the 21st Century: Where do Unmanned Aircraft Fit in? *Journal of Unmanned Vehicle Systems* 3, 137–155 (2015).
- Chabot, D. & Francis, C. M. Computer-automated Bird Detection and Counts in High-Resolution Aerial Images: A Review. *Journal of Field Ornithology* 87, 343–359 (2016).
- Conroy, M. J., Goldsberry, J. R., Hines, J. E. & Stotts, D. B. Evaluation of Aerial Transect Surveys for Wintering American Black Ducks. *The Journal of Wildlife Management* 52, 694 (1988).
- Costa, L. da F. Further Generalizations of the Jaccard Index. Preprint at <http://arxiv.org/abs/2110.09619> (2021).
- Dirir, A., Ignatious, H., Elsayed, H., Khan, M., Adib, M., Mahmoud, A. & Al-Gunaid, M. An Advanced Deep Learning Approach for Multi-Object Counting in Urban Vehicular Environments. *Future Internet* 13 (12), 306 (2021).
- Detlefsen, N., Borovec, J., Schock, J., Jha, A., Koker, T., Di Liello, L., Stancl, D., Quan, C., Grechkin, M. & Falcon, W. TorchMetrics – Measuring Reproducibility in PyTorch. *Journal of Open Source Software* 7 (70), 4101 (2022).
- Dundas, S., Vardanega, M., O'Brien, P. & McLeod, S. R. Quantifying Waterfowl Numbers: Comparison of Drone and Ground-based Survey Methods for Surveying Waterfowl on Artificial Waterbodies. *Drones* 5, 5 (2021).
- Eggeman, D. R. & Johnson, F. A. Variation in Effort and Methodology for the Midwinter Waterfowl Inventory in the Atlantic Flyway. *Wildlife Society Bulletin* 17, 227–233 (1989).
- Ellis-Felege, S. N., Stechmann, T., Hervey, S., Felege, C. J., Rockwell, R. F. & Barnas, A. F. Nesting Common Eiders (*Somateria mollissima*) Show Little Behavioral Response to Fixed-wing Drone Surveys. *Drone Systems and Applications* 10, 1-14 (2021).
- Evans-Peters, G. Assessing Biological Values of Wetland Reserve Program Wetlands for Wintering Waterfowl. Thesis. Oregon State University. (2010).

- Fleming, K. K., Brasher, M. G., Humburg, D. D., Petrie, M. J. & Soulliere, G. J. Derivation of Regional, Non-breeding Duck Population Abundance Objectives to Inform Conservation Planning. (2017).
- Francis, R. J., Lyons, M. B., Kingsford, R. T. & Brandis, K. J. Counting Mixed Breeding Aggregations of Animal Species Using Drones: Lessons from Waterbirds on Semi-automation. *Remote Sensing* 12(7), 1185 (2020).
- Gilbert, A. D., Jacques, C. N., Lancaster, J. D., Yetter, A. P. & Hagy, H. M. Disturbance Caused by Aerial Waterfowl Surveys During the Nonbreeding Season. *The Journal of Wildlife Management* 84, 1063-1071 (2020).
- Gilbert, A. D., Jacques, C. N., Lancaster, J. D., Yetter, A. P. & Hagy, H. M. Visibility Bias of Waterbirds During Aerial Surveys in the Nonbreeding Season. *Wildlife Society Bulletin* 45, 6-15 (2021).
- Herbert, J. A., Chakraborty, A., Naylor, L. W. & Krementz, D. G. Habitat Associations of Wintering Dabbling Ducks in the Arkansas Mississippi Alluvial Valley: Implications for Waterfowl Management Beyond the Mallard. *Wildlife Biology* 1, (2021).
- Heusmann, H. W. Let's Get Rid of the Midwinter Waterfowl Inventory in the Atlantic Flyway. *Wildlife Society Bulletin* 27, 559–565 (1999).
- Hayes, M. C., Gray, P. C., Harris, G., Sedgwick, W. C., Crawford, V. D., Chazal, N. & Johnston, D. W. Drones and Deep Learning Produce Accurate and Efficient Monitoring of Large-scale Seabird Colonies. *Ornithological Applications* 123, 1-16 (2021).
- He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. in *Proceedings of the IEEE conference on computer vision and pattern recognition* 770–778 (2016).
- Hermans, J., Spanakis, G. & Möckel, R. Accumulated Gradient Normalization. Preprint at <http://arxiv.org/abs/1710.02368> (2017).

- Hong, S., Han, Y., Kim, S., Lee, A. & Kim, G. Application of Deep-learning Methods to Bird Detection Using Unmanned Aerial Vehicle Imagery. *Sensors* 19(7), 1651 (2019).
- Höpe, A. Diffuse Reflectance and Transmittance. *Experimental Methods in the Physical Sciences* 46, 179-219 (2014).
- Hosang, J., Benenson, R. & Schiele, B. Learning non-maximum suppression. Preprint at <http://arxiv.org/abs/1705.02950> (2017).
- Ivošević, B., Han, Y., Cho, Y. & Kwon, O. The Use of Conservation Drones in Ecology and Wildlife Research. *Journal of Ecology and Environment* 38, 113-118 (2015).
- Jangblad, M. Object Detection in Infrared Images using Deep Convolutional Neural Networks. Master's Thesis. Uppsala University, Uppsala, Sweden (2018).
- Killgore, K. J., Todd, S., Fischer, R., Hoover, J., Harrison, A., Hartfield, P., Biedenharn, D. & Kleiss, B. Conservation Plan for the Interior Least Tern, Pallid Sturgeon, and Fat Pocketbook Mussel in the Lower Mississippi River. (2014).
- Lancaster, J. D., Wilson, B. C. & the Gulf Coast Joint Venture Waterfowl Working Group. Gulf Coast Joint Venture Wintering Waterfowl Population and Habitat Objective Model Refinement. (2021).
- Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E. & Stoica, I. Tune: A Research Platform for Distributed Model Selection and Training. Preprint at <http://arxiv.org/abs/1807.05118> (2018).
- Lin, T-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L. & Dollár, P. Microsoft COCO: Common Objects in Context. Preprint at <http://arxiv.org/abs/1405.0312> (2015).
- Linchant, J., Lisein, J., Semeki, J., Lejeune, P. & Vermeulen, C. Are Unmanned Aircraft Systems (UASs) the Future of Wildlife Monitoring? A Review of Accomplishments and Challenges. *Mammal Review* 45, 239–252 (2015).

- Liu, M., Chen, L., Du, X., Jin, L. & Shang, M. Activated Gradients for Deep Neural Networks. Preprint at <http://arxiv.org/abs/2107.04228> (2021).
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C-Y. & Berg, A. C. SSD: Single Shot MultiBox Detector. in vol. 9905 21–37 (2016).
- Maldague, X. P. Theory and Practice of Infrared Technology for Nondestructive Testing. Wiley, 2001.
- Masto, N. M. Evaluation of Aerial Surveys to Monitor Fall-Winter Waterbird Populations in South Carolina. Thesis. Clemson University. (2019).
- McKellar, A. E., Shephard, N. G. & Chabot, D. Dual Visible-thermal Camera Approach Facilitates Drone Surveys of Colonial Marshbirds. *Remote Sensing in Ecology and Conservation* 7(2), 214-226 (2020).
- Nakaguchi, V. M. & Ahamed, T. Development of an Early Embryo Detection Methodology for Quail Eggs Using a Thermal Micro Camera and the YOLO Deep Learning Algorithm. *Sensors* 22(15), 5820 (2022).
- National Conservation Easement Database [NCED]. Complete U.S. NCED Dataset. https://site.tplgis.org/NCED/data_downloads/2022/NCED_06232022.gdb.zip (2022). Accessed 29 June 2023.
- National Fish and Wildlife Foundation [NFWF]. Business Plan for Lower Mississippi Alluvial Valley. https://www.nfwf.org/sites/default/files/2019-12/lmav-business-plan_0.pdf (2019). Accessed 7 June 2023.
- Natural Resources Conservation Service [NRCS]. Conservation Programs. <https://www.nrcs.usda.gov/programs-initiatives/wre-wetland-reserve-easements> (2023). Accessed 28 June 2023.

- Osco, L. P., Marcato Junior, J., Marques Ramos, A. P., De Castro Jorge, L. A., Fatholahi, S. N., De Andrade Silva, J., Matsubara, E. T., Pistori, H., Gonçalves, W. N. & Li, J. A Review on Deep Learning in UAV Remote Sensing. *International Journal of Applied Earth Observation and Geoinformation* 102, 102456 (2021).
- Oswalt, S. N. Forest Resources of the Lower Mississippi Alluvial Valley. SRS-GTR-177 <https://www.fs.usda.gov/treearch/pubs/43960> (2013).
- Ouyang, W., Wang, X., Zhang, C. & Yang, X. Factors in Fine-tuning Deep Model for Object Detection. Preprint at <https://arxiv.org/abs/1601.05150> (2016).
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. TorchVision Object Detection Fine-tuning Tutorial. (2017). https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaisons, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Preprint at <http://arxiv.org/abs/1912.01703> (2019).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. Scikit-learn: Machine Learning in Python. Preprint at <http://arxiv.org/abs/1201.0490> (2018).
- Petrie, M. J., Brasher, M. G., Soulliere, G. J., Tirpak, J. M., Pool, D. B. & Reker, R. R. Guidelines for Establishing Joint Venture Waterfowl Population Abundance Objectives. (2011).
- Pöysä, H., Kotilainen, J., Väänänen, V. M. & Kunnasranta, M. Estimating Production in Ducks: a Comparison Between Ground Surveys and Unmanned Aircraft Surveys. *European Journal of Wildlife Research* 64, 74 (2018).
- Raeva, P. L., Šedina, J. & Dlesk, A. Monitoring of Crop Fields Using Multispectral and Thermal Imagery from UAV. *European Journal of Remote Sensing* 52, 192-201 (2019).

- Robinson, J., Chuang, C.-Y., Sra, S. & Jegelka, S. Contrastive Learning with Hard Negative Samples. Preprint at <http://arxiv.org/abs/2010.04592> (2021).
- Rosenberg, K. V., Dokter, A. M., Blancher, P. J., Sauer, J. R., Smith, A. C., Smith, P. A., Stanton, J. C., Panjabi, A., Heft, L., Parr, M. & Marra, P. P. Decline of North American Avifauna. *Science* 366, 120-124 (2019).
- Ringelman, K. M., Williams, C. K., Castelli, P. M., Sieges, M. L., Longnecker, R. A., Nichols, T. C. & Earsom, S. D. Estimating Waterfowl Carrying Capacity at Local Scales: A Case Study from Edwin B. Forsythe National Wildlife Refuge, New Jersey. *Journal of Fish and Wildlife Management* 9, 209-218 (2017).
- Ruder, S. An overview of gradient descent optimization algorithms. Preprint at <http://arxiv.org/abs/1609.04747> (2017).
- Sardà-Palomera, F., Bota, G., Padilla, N. & Brotons, L. Fine-scale Bird Monitoring from Light Unmanned Aircraft Systems: Bird Monitoring from UAS. *Ibis* 154, 177–183 (2012).
- Sardà-Palomera, F., Bota, G., Viñolo, C., Pallarés, O., Sazatornil, V., Brotons, L. & Gomáriz, S. Unmanned Aircraft Systems to Unravel Spatial and Temporal Factors Affecting Dynamics of Colony Formation and Nesting Success in Birds. *Journal of Avian Biology* 48, 1273–1280 (2017).
- Scholten, C. N., Kamphuis, A. J., Vredevoogd, K. J., Lee-Strydhorst, K. G. & Atma, J. L. Real-time Thermal Imagery from an Unmanned Aerial Vehicle Can Locate Ground Nests of a Grassland Songbird at Rates Similar to Traditional Methods. *Biological Conservation* 233, 241–246 (2019).
- Shahinfar, S., Meek, P. & Falzon, G. "How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics* 57, 101085 (2020).
- Shorten, C. & Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6, 60 (2019).

- Smith, S. L., Kindermans, P.-J., Ying, C. & Le, Q. V. Don't Decay the Learning Rate, Increase the Batch Size. Preprint at <http://arxiv.org/abs/1711.00489> (2018).
- Soulliere, G. J., Loges, B. W., Dunton, E. M., Luukkonen, D. R., Eichholz, M. W. & Koch, M. E. Monitoring Waterfowl in the Midwest during the Non-breeding Period: Challenges, Priorities, and Recommendations. *Journal of Fish and Wildlife Management* 4, 395–405 (2013).
- Talukdar, J., Gupta, S., Rajpura, P. S. & Hegde, R. S. Transfer Learning for Object Detection using State-of-the-Art Deep Neural Networks. in 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN) 78–83 (IEEE, 2018).
- Turner, R., Eriksson, D., McCourt, M., Killi, J., Laaksonen, E., Xu, Z. & Guyon, I. Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. Preprint at <http://arxiv.org/abs/2104.10201> (2021).
- U.S. Fish and Wildlife Service [USFWS] & Environment Canada. North American Waterfowl Management Plan. (1986).
- Weissensteiner, M. H., Poelstra, J. W. & Wolf, J. B. Low-budget Ready-to-fly Unmanned Aerial Vehicles: An Effective Tool for Evaluating the Nesting Status of Canopy-breeding Bird Species. *Journal of Avian Biology* 46, 425-430 (2015).
- Whitaker, J. M., Ringelman, K. M., Marty, J. R., Selman, W. & Linscombe, J. T. Changes in Waterfowl Abundance and Species Composition on Louisiana Coastal Wildlife Management Areas and Refuges 2004-2016. *Journal of the Southeastern Association of Fish and Wildlife Agencies* 6, 136-145 (2019).
- Williams, B. K., Koneff, M. D. & Smith, D. A. Evaluation of Waterfowl Conservation Under the North American Waterfowl Management Plan. *The Journal of Wildlife Management* 63, 417 (1999).
- Yu, T. & Zhu, H. Hyper-Parameter Optimization: A Review of Algorithms and Applications. Preprint at <http://arxiv.org/abs/2003.05689> (2020).

Zhou, M., Elmore, J. A., Samiappan, S., Evans, K. O., Pfeiffer, M. B., Blackwell, B. F. & Iglay, R. B. Improving Animal Monitoring Using Small, Unmanned Aircraft Systems (sUAS) and Deep Learning Networks. *Sensors* 21, 5697 (2021).

VITA

Zack Loken was born in Wisconsin Rapids, Wisconsin, in 1994. They graduated from the University of Wisconsin, Stevens Point, in 2019 with a B.S. in Wildlife Ecology and Management. After graduating, Zack worked a six-month stint with the National Park Service studying nesting shorebird spatial ecology on the Niobrara. Following this seasonal position, Zack worked as a ‘GIS and Remote Sensing Specialist’ for Ducks Unlimited’s Western Regional Office. During their time as a ‘GIS and Remote Sensing Specialist,’ Zack developed a strong desire to apply advanced image analysis methodologies to aerial imagery for waterfowl conservation. Zack began working with Dr. Kevin Ringelman in 2021 to study deep learning-based algorithms for object detection to automatically count, identify, and locate species of non-breeding waterfowl in UAV images and videos. Zack will finish their M.S. in Renewable Natural Resources in December of 2023 after two field seasons of surveying restored wetlands in the Lower Mississippi Alluvial Valley region of Arkansas, Mississippi, and Louisiana for wintering waterfowl. Zack is currently an ‘Imagery Analyst’ with Savanna Institute in Madison, WI.