ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ
ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12241
Τηλ: +30 210 538-1614
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
Τεχνητή Νοημοσύνη και Βαθιά Μάθηση
https://aidl.uniwa.gr/

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL &
ELECTRONICS ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND
PRODUCTION ENGINEERING
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
250, Thivon Str., Athens, GR-12241, Greece
Tel: +30 210 538-1614
Master of Science in
Artificial Intelligence and Deep Learning
https://aidl.uniwa.gr/

Master of Science Thesis

Machine Learning for Animal Detection and Recognition for European Wildlife Conservation

Student: Panagiotopoulos, Christos
Registration Number: AIDL-0027

MSc Thesis Supervisor

Patrikakis, Charalampos
Professor

ATHENS-EGALEO, September 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12241
Τηλ: +30 210 538-1614
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
Τεχνητή Νοημοσύνη και Βαθιά Μάθηση
https://aidl.uniwa.gr/

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
250, Thivon Str., Athens, GR-12241, Greece
Tel: +30 210 538-1614
Master of Science in
Artificial Intelligence and Deep Learning
https://aidl.uniwa.gr/

Μεταπτυχιακή Διπλωματική Εργασία

Χρήση Μηχανικής Μάθησης για ανίχνευση και αναγνώριση ζώων στο πλαίσιο διατήρησης της άγριας ζωής στην Ευρώπη

Φοιτητής: Παναγιωτόπουλος Χρήστος
AM: AIDL-0027

Επιβλέπων Καθηγητής
Χαράλαμπος Πατρικάκης
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΣΕΠΤΕΜΒΡΙΟΣ 2024

This MSc Thesis has been accepted, evaluated and graded by the following committee:

Supervisor	Member	Member
 Digitally signed by CHARALAMPOS PATRIKAKIS Date: 2024.12.04 11:07:33 +02'00'	PANAGIOTIS KASNESIS Digitally signed by PANAGIOTIS KASNESIS Date: 2024.12.04 10:16:19 +02'00'	Grigorios Nikolaou Digitally signed by Grigorios Nikolaou Date: 2024.12.04 12:12:18 +02'00'
Patrikakis, Charalampos	Kasnesis, Panagiotis	Nikolaou, Grigorios
Professor	Lecturer	Lecturer
Department of Electrical and Electronics Engineering	Department of Electrical and Electronics Engineering	Department of Industrial Design and Production Engineering
University of West Attica	University of West Attica	University of West Attica

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Χρήστος Παναγιωτόπουλος του Γεωργίου, με αριθμό μητρώου AIDL-0027 μεταπτυχιακός φοιτητής του ΔΠΜΣ «Τεχνητή Νοημοσύνη και Βαθιά Μάθηση» του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών και του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Η εργασία δεν έχει κατατεθεί στο πλαίσιο των απαιτήσεων για τη λήψη άλλου τίτλου σπουδών ή επαγγελματικής πιστοποίησης πλην του παρόντος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο Δηλών
Χρήστος Παναγιωτόπουλος

(Υπογραφή φοιτητή)

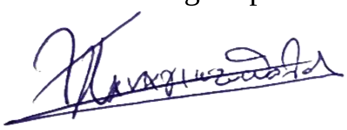
Declaration of the author of this MSc thesis

I, Christos Georgios Panagiotopoulos with the following student registration number: AIDL-0024, postgraduate student of the MSc programme in “Artificial Intelligence and Deep Learning”, which is organized by the Department of Electrical and Electronic Engineering and the Department of Industrial Design and Production Engineering of the Faculty of Engineering of the University of West Attica, hereby declare that:

I am the author of this MSc thesis and any help I may have received is clearly mentioned in the thesis. Additionally, all the sources I have used (e.g., to extract data, ideas, words or phrases) are cited with full reference to the corresponding authors, the publishing house or the journal; this also applies to the Internet sources that I have used. I also confirm that I have personally written this thesis and the intellectual property rights belong to myself and to the University of West Attica. This work has not been submitted for any other degree or professional qualification except as specified in it.

Any violations of my academic responsibilities, as stated above, constitutes substantial reason for the cancellation of the conferred MSc degree.

The author
Christos Panagiotopoulos



(Signature)

I would like to express my sincere gratitude to Theodoros Kominos for his invaluable contribution to this project. He generously provided me with the camera trap images that were essential for my research, and his expertise in zoology gave me a much better understanding of the topic. I am truly grateful for his support.

The data provided was used solely for the purpose of this thesis and will not be shared or used for any other purpose without the written permission of Mr. Theodoros Kominos.

I am also very grateful to my supervisor, Professor Charalampos Patrikakis, for his guidance and support throughout this project. His insightful perspectives and wealth of experience were invaluable in completing my research.

I would also like to extend my thanks to the other members of my thesis committee, Dr. Panagiotis Kasnesis and Dr. Grigorios Nikolaou , for their feedback and constructive criticism during the thesis process. Their contributions were valuable in refining my research and improving the quality of this thesis.

Abstract

This Master of Science thesis explores the potential of Computer Vision (CV) for wildlife research and conservation in Europe. I collaborated with Theodoros Kominos, a wildlife researcher who provided me images and video recordings from trap cameras he set in Greece at the past. The data includes various wild animals, some of which are protected in Greece, such as brown bear, wolf, chamois, and wildcat. Other species like roe deer, wild boar, red fox, European badger, other mustelids, and European hare are also present. The data also contains domestic animals (like cows, dogs, horses) and human activities (hikers, potential hunters, conservation workers, and vehicles).

The main objective was to develop a dataset of annotated images from this raw data and implement AI algorithms to analyze it. The goals were to classify and detect different animal species and identify potential threats to their habitats. Additionally, I explored how these AI techniques could enhance existing conservation efforts.

The process involved organizing and annotating the visual data provided by the researcher. I then employed computer vision techniques and trained models for accurately identifying and classifying the various subjects in the images and videos.

The study also focused on detecting potential threats to wildlife habitats by identifying forbidden or concerning activities in the monitored areas. This includes detecting unauthorized human presence, illegal hunting, or other activities that could harm wildlife or their habitats. This aspect has significant implications for proactive conservation efforts and the development of early warning systems for habitat protection, allowing for timely intervention when threats are detected.

I explored how these AI techniques could be integrated into existing conservation practices, developing user-friendly interfaces and workflows to incorporate these tools into daily operations. This integration aims to streamline data analysis, reduce manual labor, and provide more accurate and timely information for wildlife management decisions.

Furthermore, I propose potential implementations of edge hardware for use by conservationists or in areas of human-wildlife conflict.

The intended results of this research include practical AI tools and techniques for wildlife researchers and conservationists to better understand, protect, and manage biodiversity in Europe.

In conclusion, this thesis demonstrates the potential of Computer Vision in enhancing wildlife research and conservation efforts, covering a wide range of species from large mammals to smaller, less studied animals. By providing more accurate, efficient, and scalable methods for monitoring wildlife and their habitats, this work could potentially be adapted for preserving Greece's and Europe's biodiversity.

Keywords

Computer Vision, Wildlife Conservation, Object Detection, Edge AI.

Περίληψη

Αυτή η Μεταπτυχιακή Διπλωματική Εργασία διερευνά τις δυνατότητες της Όρασης Υπολογιστών (Computer Vision) στην έρευνα και τη διατήρηση της άγριας ζωής στην Ευρώπη. Συνεργάστηκα με το Θεόδωρο Κομηνό, ζωολόγο ερευνητή που μου παρείχε δεδομένα εικόνων και βίντεο από κάμερες παγίδες που είχε τοποθετήσει στο παρελθόν στην Ελληνική επικράτεια. Τα δεδομένα περιλαμβάνουν διάφορα άγρια ζώα, μερικά από τα οποία προστατεύονται στην Ελλάδα, όπως η αρκούδα, ο λύκος, το αγριόγιδο και η αγριόγατα. Άλλα είδη όπως το ζαρκάδι, ο αγριόχοιρος, η αλεπού, ο ασβός, άλλες μουστελίδες και ο λαγός επίσης εμφανίζονται. Τα δεδομένα περιέχουν επίσης οικόσιτα ζώα (όπως αγελάδες, σκύλους, άλογα) και ανθρώπινες δραστηριότητες (πεζοπόρους, πιθανούς κυνηγούς, εργαζόμενους σε εθνικά πάρκα και οχήματα).

Ο κύριος στόχος ήταν να αναπτυχθεί ένα σύνολο δεδομένων με εικόνες και να εφαρμοστούν αλγόριθμοι μηχανικής μάθησης για την ανάλυσή τους. Επιμέρους στόχοι ήταν η ταξινόμηση και η ανίχνευση διαφορετικών ειδών ζώων και ο εντοπισμός πιθανών απειλών για τους βιότοπους τους. Επίσης, σαν στόχος είχε τεθεί η ανάπτυξη και υλοποίηση τέτοιων εργαλείων για χρήση από τους εργαζόμενους για τη διατήρηση της άγριας ζωής ή σε περιοχές σύγκρουσης ανθρώπου-άγριας ζωής.

Η διαδικασία περιελάβανε την οργάνωση των οπτικών δεδομένων που παρείχε ο ερευνητής και την καταγραφή σε ηλεκτρονική μορφή της περιοχής ενδιαφέροντος σε κάθε αρχείο (annotation). Στη συνέχεια, χρησιμοποίησα τεχνικές όρασης υπολογιστών και αλγορίθμων ανίχνευσης αντικειμένων (Object detection), για να την αναπτύξη μοντέλων για την αναγνώριση, ταξινόμηση και ακριβή εντοπισμό των διαφόρων θεμάτων ενδιαφέροντος στις εικόνες και τα βίντεο.

Διερεύνησα πώς αυτές οι τεχνικές τεχνητής νοημοσύνης θα μπορούσαν να ενσωματωθούν στις υπάρχουσες πρακτικές στοχεύοντας στην απλοποίηση της ανάλυσης δεδομένων, στη μείωση της χειρωνακτικής εργασίας και στην παροχή πιο ακριβών και έγκαιρων πληροφοριών για τις αποφάσεις διαχείρισης της άγριας ζωής.

Επιπλέον, προτείνω πιθανές εφαρμογές σε συσκευές πεδίου (edge hardware). Αυτές οι συσκευές, εξοπλισμένες με μοντέλα τεχνητής νοημοσύνης, θα μπορούσαν να παρέχουν ανάλυση και ειδοποιήσεις σε πραγματικό χρόνο, επιτρέποντας την ταχεία αντίδραση σε καταστάσεις έκτακτης ανάγκης ή σε μη εξουσιοδοτημένες ανθρώπινες δραστηριότητες σε προστατευόμενες περιοχές.

Συμπερασματικά, αυτή η διπλωματική εργασία καταδεικνύει τις δυνατότητες της Όρασης Υπολογιστών και της μηχανικής μάθησης στην ενίσχυση της έρευνας και των προσπαθειών διατήρησης της άγριας ζωής, καλύπτοντας ένα ευρύ φάσμα ειδών από μεγάλα θηλαστικά έως μικρότερα, λιγότερο μελετημένα ζώα. Παρέχοντας πιο ακριβείς, αποτελεσματικές και κλιμακούμενες μεθόδους για την παρακολούθηση της άγριας ζωής και των βιοτόπων της, αυτή η εργασία συμβάλλει στη διατήρηση της βιοποικιλότητας της Ελλάδας και της Ευρώπης και θα μπορούσε ενδεχομένως να προσαρμοστεί για παγκόσμιες προσπάθειες διατήρησης.

Λέξεις – κλειδιά

Όραση Υπολογιστών, Διατήρηση της Άγριας ζωής, Ανίχνευση Αντικειμένων, Τεχνητή Νοημοσύνη αιχμής

Table of Contents

List of Tables

List of figures

Acronym Index

CV: Computer Vision

AI: Artificial Intelligence

CT(s): Camera trap(s)

UAV(s): Unmanned Aerial Vehicle(s)

IoT: Internet of Things

mAP: Mean Average Precision

IoU: Intersection over Union

YOLO: You Only Look Once

CNN: Convolutional Neural Network

NMS: Non-Maximum Suppression

INTRODUCTION

In an era of environmental challenges, the intersection of cutting-edge technology and wildlife conservation offers exciting possibilities. This Master of Science thesis explores how Computer Vision and Artificial Intelligence can make a great impact on wildlife research and conservation. This research aims to harness these technologies to aid wildlife conservationists. By analyzing trap camera footage from Greek wilderness areas, I develop tools for automatic species identification and threat detection. My work encompasses a range of animals, from bears to wolves and wildcats, as well as the detection of potentially harmful human activities. Beyond identification, this thesis explores the integration of AI-powered tools into existing conservation practices. By automating time-consuming tasks, these tools could free up valuable resources, allowing conservationists to focus on critical conservation efforts.
I also investigate the potential of edge computing devices for real-time field analysis. These smart devices could provide instant alerts, enabling rapid responses to wildlife emergencies or unauthorized activities in protected areas, a crucial capability in regions of human-wildlife conflict.

This research bridges the gap between academic study and practical application. By using user-friendly interfaces, I aim to make advanced technologies accessible to front-line conservationists.

In the following chapters, I will detail my methods, discuss challenges, and explore the implications of this work for wildlife conservation on a broader scale.

The subject of this thesis

The subject of this research is the application of computer vision techniques to analyze wildlife data collected from trap cameras in Greece, encompassing a wide range of species from large mammals like brown bears and wolves to smaller, less studied animals.
As human activities continue to encroach on natural habitats, it's becoming increasingly crucial to try robust, real-time monitoring systems that can detect and respond to threats quickly. This research is not only interesting from a technological standpoint but is timely and vital for the

preservation of Europe's biodiversity. Moreover, with climate change altering ecosystems at an unprecedented rate, there's an urgent need for tools that can track these changes and their impacts on wildlife populations efficiently. By harnessing the power of AI, we aim to provide conservationists with the means to gather and analyze data at previously unattainable scales, potentially revolutionizing how we approach wildlife management and protection.

Aim and objectives

The primary aim of this thesis is to develop and implement computer vision techniques to enhance wildlife research and conservation practices in Europe, using data collected from Greek wildlife areas as a case study. My objectives include developing a comprehensive annotated dataset from trap camera footage, implementing machine learning algorithms for species classification and threat detection, and exploring the integration of these AI techniques into existing conservation practices. I also aim to investigate the potential of edge computing devices for real-time wildlife monitoring in the field. Key research questions include optimizing computer vision for diverse European wildlife, effective methods for detecting threats to wildlife habitats, seamless integration of AI tools into current practices, and the feasibility of edge computing in remote areas.

Methodology

My approach includes a collaboration with wildlife researcher to collect and annotate trap camera footage, developing and training AI models for species classification and possible threat detection, and designing user-friendly interfaces for integration with existing conservation practices. I will also explore edge computing applications and evaluate the developed systems through test data.

Innovation

This thesis offers several innovative aspects. It provides comprehensive coverage of European wildlife species, moving beyond the typical focus on a few select animals. The research introduces a novel approach to AI-powered threat detection in habitats, enhancing our ability to protect wildlife. There's emphasis on practical integration with current conservation methods, bridging the gap between advanced technology and field application. The exploration of edge computing for real-time field analysis represents a cutting-edge approach to wildlife monitoring. Finally, the thesis takes a holistic approach, combining species identification, threat detection, and implementation strategies into a cohesive framework for wildlife conservation.

Structure

This thesis is structured into three main chapters, followed by conclusions:
Chapter 1 introduces the research topic, providing context on wildlife conservation challenges and the potential of computer vision and AI in addressing these issues.
Chapter 2 details the methodology employed in this study, including data collection methods, AI model development processes, and the integration of edge computing technologies.
Chapter 3 presents the results of the research, analyzes the performance of the developed AI models and systems, and discusses the implications of these findings for wildlife conservation practices.

The thesis concludes with a section summarizing key findings, discussing the contributions of this research to wildlife conservation, acknowledging limitations, and proposing directions for future research in this field.

1 CHAPTER 1: Wildlife Conservation and AI

This chapter introduces the core concepts and context of this thesis. It explores the current challenges in wildlife conservation, the potential of artificial intelligence and computer vision in addressing these challenges.

1.1 Wildlife Conservation in Greece

The conservation of wildlife in Europe faces numerous challenges in the 21st century. Habitat loss, climate change, and human-wildlife conflict have put significant pressure on many species, leading to population declines and increased extinction risks [1].
Greece is a prime example of the escalating challenges in wildlife conservation, particularly those involving large mammals and their interactions with human-populated areas. The wolf (*Canis lupus*), wild boar (*Sus scrofa*), and brown bear (*Ursus arctos*) frequently capture media's attention due to local concerns surrounding their presence near or within populated areas [2]-[5]. This heightened human-wildlife conflict, coupled with habitat loss and fragmentation, poses a significant threat to the long-term viability of these species [2],[4]-[5]. Meanwhile, other species like Balkan chamois (*Rupicapra rupicapra balcanica*) face additional pressures, notably illegal hunting, further jeopardizing their survival [6]. These complex and interconnected conservation challenges underscore the pressing need for effective and multifaceted strategies in Greece to ensure the coexistence of humans and wildlife.

1.2 Technology in Conservation

A plethora of emerging technologies is reshaping the field of wildlife conservation. Ranging from camera traps and drones to GPS-connected devices and sophisticated software, these advancements are transforming how zoologists conduct research and conservation efforts, both in situ (in the natural habitat) and ex situ (outside the natural habitat) [7]-[10]. This section explores the key technological tools revolutionizing wildlife conservation, examining their applications, benefits, challenges, and potential for synergy with AI and particularly Computer Vision to further enhance their impact.

1.2.1 Camera traps

Camera traps have emerged as one of the most versatile and widely used technologies in wildlife conservation over the past few decades [11]. These motion-activated devices have revolutionized our ability to study and monitor wildlife populations, particularly for elusive or rare species in remote areas.

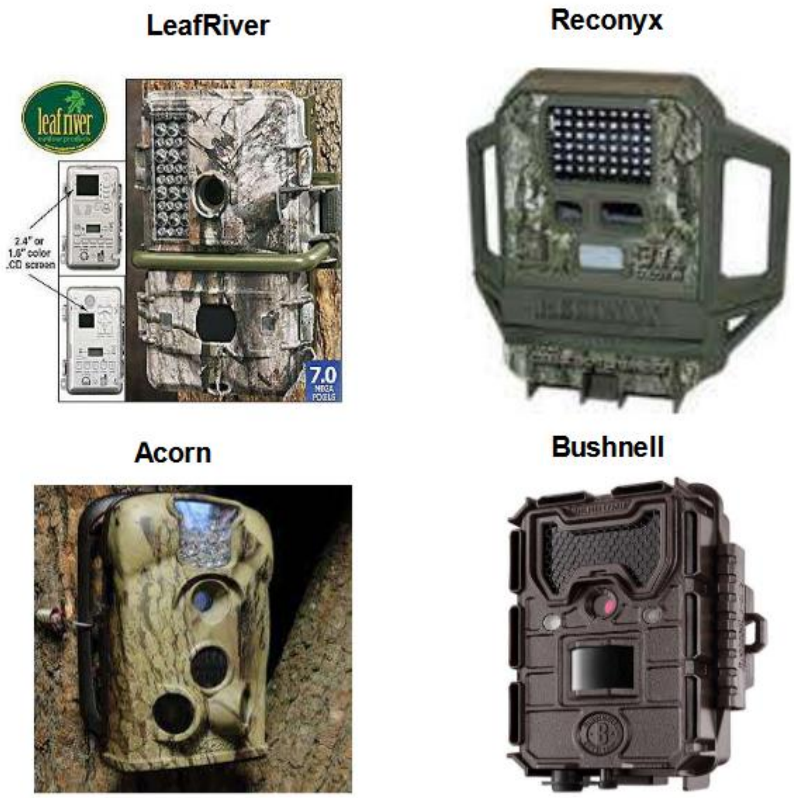


Figure 1.2 CT devices [7]

In recent years, the integration of camera traps with other technologies has further expanded their capabilities. For instance, some modern camera traps can transmit images in real-time using cellular or satellite networks, enabling rapid response to poaching activities or wildlife emergencies [12].

However, the widespread adoption of camera traps has also introduced new challenges, particularly in data management and analysis. A single camera trap study can generate thousands of images, creating a bottleneck in data processing [13]. This challenge has spurred the development of automated image analysis techniques, setting the stage for the application of artificial intelligence in wildlife conservation, which will be discussed in the subsequent section.

1.2.2 Drones and Other Technologies

Drones, or Unmanned Aerial Vehicles (UAVs), have rapidly become an invaluable tool for wildlife conservationists, offering a bird's-eye view of ecosystems and wildlife populations that were previously unattainable. Their ability to capture high-resolution imagery and videos, access remote and inaccessible areas, and conduct surveys with minimal disturbance has revolutionized various aspects of conservation research and management [10]. For example, drones have been successfully employed to monitor wildlife populations, map habitats, assess ecosystem health or detect poaching activities [14]. However, the use of drones in wildlife conservation also presents some challenges. The noise generated by drones can disturb wildlife, particularly during sensitive periods like breeding or nesting [15]. Moreover, the regulatory environment for drone use can be complex and vary across countries and regions. It is crucial for researchers to comply with all relevant regulations and obtain necessary permits before deploying drones for conservation purposes.

Beyond camera traps and drones, several other technologies are playing a crucial role in wildlife conservation. GPS tracking devices, for instance, enable researchers to monitor the movements and behavior of individual animals, providing valuable insights into their ecology and migration patterns [16]. Bioacoustics monitoring, using sound recorders to capture and analyze the soundscapes of natural habitats, helps researchers track the presence and abundance of vocalizing species, as well as detect threats like illegal logging or poaching [17].

1.3 AI in Conservation

The massive amount of data generated by these technologies has sparked an intense interest in harnessing the power of AI to extract knowledge and insights. For the first time in history, we have the potential to track and detect anomalies or patterns that have remained hidden from traditional observation methods. AI allows us to group, classify, and predict ecological phenomena with unprecedented accuracy, opening new avenues for understanding and conserving our natural world.

1.3.1 Computer Vision in Wildlife Conservation

Computer Vision (CV) has emerged as a powerful tool in wildlife conservation, offering unprecedented capabilities in analyzing visual data from various sources such as camera traps, drones, or even satellite imagery [19].

Key applications of CV in wildlife conservation may include [13], [20] - [21]:

1. Automated Species Identification: CV algorithms can rapidly and accurately identify animal species in images or videos, significantly reducing the time and effort required for manual identification.
2. Individual Animal Identification and Tracking: Advanced CV techniques can recognize and track individual animals within a species, based on unique features such as stripe patterns, spot configurations, or facial characteristics. This capability is crucial for:
    - Estimating population sizes more accurately
    - Monitoring individual animal movements and behavior
    - Studying social structures within animal groups
    - Tracking the health and life history of specific animals over time
3. Population Monitoring: By analyzing large sets of visual data and leveraging individual identification capabilities, CV can assist in estimating population sizes, demographic composition, and tracking changes over time with greater precision.
4. Behavior Analysis: CV techniques can be used to study animal behavior patterns, interactions, and movements captured in video footage, including complex social behaviors and predator-prey interactions.

5. Habitat Mapping and Monitoring: CV can analyze satellite or drone imagery to classify land cover types, detect changes in vegetation, and identify potential threats to habitats.
6. Anti-poaching Efforts: CV-enabled systems can detect human intrusions in protected areas and identify potential poaching activities in real-time.

The integration of CV with other technologies, such as edge computing and IoT devices, is pushing the boundaries of real-time wildlife monitoring and conservation efforts.

1.3.2 Other AI Practices in Conservation

While Computer Vision plays a crucial role, other AI techniques are also making significant contributions to wildlife conservation:

- Machine Learning for Predictive Modeling: ML algorithms can predict species distribution, population trends or migration [21].
- Deep Learning for Acoustic Analysis: DL models can identify and classify animal vocalizations, enabling non-invasive monitoring of biodiversity [22].

These AI practices, often used in combination with CV and other technologies, are revolutionizing our approach to wildlife conservation, enabling more efficient, accurate, and timely interventions to protect biodiversity.

1.3.3 Related works and datasets

DeepFaune

One notable initiative in wildlife monitoring techniques, particularly through the integration of camera trap technology and machine learning is the DeepFaune project, a collaborative effort involving over 50 partners across France and other European countries [29]. This project represents a significant step forward in automated species identification from camera trap images in European ecosystems.

The DeepFaune team developed a deep learning model capable of identifying 26 different species or higher taxonomic groups, with a primary focus on mammals. Their approach involved aggregating a substantial dataset of over 2 million annotated pictures and 20,000 annotated videos from various partners. This diverse dataset, collected from a wide range of habitats and contexts, provided a robust foundation for training a versatile model.

A key component of the DeepFaune pipeline is the use of MegaDetector, an open-source model developed by Microsoft [30]. MegaDetector efficiently filters images to detect the presence of animals, humans, or vehicles, serving as a crucial first step in the classification process. The DeepFaune team used MegaDetector v5a, which is based on the YOLOv5 architecture [31], to prepare their training and validation datasets.

However, to improve processing speed on CPUs, the team developed an alternative detector using YOLOv8 [32]. This custom detector, trained on a subset of their full dataset, achieved similar detection performance to MegaDetectorv5 but at a much higher speed (approximately

0.3 seconds per image compared to 2-3 seconds). This optimization demonstrates the team's commitment to creating practical, efficient tools for ecological research.

To address common challenges in machine learning applications to ecological data, the researchers implemented several innovative techniques. They employed transfer learning, starting with a ConvNext-Base model pre-trained on ImageNet 22K. To combat class imbalance, a pervasive issue in ecological datasets, they developed a novel approach combining downsampling and upsampling techniques during both training and validation phases. This method successfully prevented performance biases towards more common species.

The resulting model demonstrated impressive performance, achieving an overall balanced validation accuracy of 97.3%. Class-specific results were equally strong, with recall and precision exceeding 0.9 for most classes. Notably, the model performed well even for some species with limited training data, such as the lynx.

A key strength of the DeepFaune project is its rigorous evaluation on out-of-sample data. The model maintained high performance when tested on entirely new datasets from unseen locations, with an overall accuracy of 93.6% on individual images. This robust generalization ability is crucial for practical applications in new contexts.

The DeepFaune model (version 1.1) can identify a wide range of European fauna. The classes it can recognize, listed alphabetically, are:

- Badger
- Bear
- Bird
- Cat
- Chamois
- Cow
- Dog
- Equid
- Fox
- Genet
- Goat
- Hedgehog
- Ibex
- Lagomorph
- Lynx
- Marmot
- Micromammal
- Mouflon
- Mustelid
- Nutria
- Red deer
- Roe deer
- Sheep
- Squirrel

- Wild boar
- Wolf

This broad coverage makes the model particularly valuable for studies focusing on large mammalian communities and for quantifying human disturbance levels.

The DeepFaune project also stands out for its commitment to practical applicability. Recognizing the diverse needs of potential users, the team developed a user-friendly, cross-platform software that can run locally on standard personal computers. This approach addresses concerns about data privacy and the challenges of uploading large image sets to online platforms, making the tool accessible to a wide range of practitioners.

The DeepFaune model has been incorporated into EcoAssist, an AI platform designed to streamline the work of ecologists dealing with camera trap images [33]. EcoAssist allows users to analyze images on their local computers, using machine learning models for automatic detection and identification. This integration demonstrates the practical value of the DeepFaune model and its potential to save time for ecologists, allowing them to focus more on conservation efforts.

As the field of automated wildlife monitoring continues to evolve, the DeepFaune project serves as both a benchmark and a roadmap. It demonstrates the potential of multi-partner collaborations to create robust, widely applicable models, while also illuminating the practical considerations necessary for successful deployment in real-world ecological research and management contexts. The ongoing development of the DeepFaune model, with regular updates to improve accuracy and expand species coverage, highlights the dynamic nature of this field and the potential for continued advancements in automated wildlife monitoring.

Pytorch-Wildlife

Another significant development in wildlife monitoring is PyTorch-Wildlife [34], an open-source deep learning framework developed through a collaboration between Microsoft AI for Good and the Universidad de los Andes. This platform addresses the growing need for accessible and efficient tools in automated wildlife detection and classification.

The PyTorch-Wildlife framework emphasizes practical usability, incorporating MegaDetectorV6-compact, an optimized detection model that achieves a recall rate of 0.85 while using only one-sixth of the parameters compared to its predecessor. This efficiency makes it particularly suitable for field deployments and resource-constrained environments, achieving detection results in approximately 0.3 seconds per image on standard CPUs.

A key strength of PyTorch-Wildlife is its modular architecture, which includes a model zoo featuring various pre-trained models for wildlife detection and classification. The platform supports multiple data formats and integrates with established tools like Timelapse and EcoAssist, making it compatible with existing wildlife monitoring workflows.

The framework has demonstrated strong real-world performance through several implementations. In one application focused on genus-level classification in the Amazon Rainforest, the system achieved 92% classification accuracy across 36 different genera when operating above a 98% confidence threshold, which accounted for 90% of the analyzed data. This level of performance was maintained while requiring human validation for only 10% of the detected animals, significantly reducing manual review requirements. Similarly, in another application in the Galápagos Islands, the framework achieved 98% accuracy in classifying invasive opossums from other wildlife in video footage, processing over 491,471 videos split between training (343,053) and validation (148,418) sets.

To ensure reliable model evaluation and selection, PyTorch-Wildlife maintains a standardized leaderboard using hidden test sets from the LILA (Labeled Information Library of Alexandria) dataset. This feature allows users to assess model performance across different geographical contexts and species distributions, helping researchers select the most appropriate models for their specific monitoring needs.

The platform's development also prioritizes accessibility through:

- Cross-platform compatibility and straightforward installation processes
- Support for both local processing and cloud-based operations via Hugging Face
- User-friendly interfaces designed for researchers without extensive technical backgrounds
- Comprehensive documentation and technical support resources

PyTorch-Wildlife represents a significant step forward in making advanced wildlife monitoring technologies accessible to conservation practitioners. Its emphasis on efficiency and usability, combined with strong detection and classification capabilities, makes it a valuable tool for wildlife research and conservation efforts.

The framework continues to evolve, with ongoing development focusing on expanding species coverage and optimizing performance for various field conditions. This development includes regular updates to improve accuracy and adapt to new conservation challenges, demonstrating the platform's commitment to supporting long-term wildlife monitoring efforts.

2 CHAPTER 2: Data and Methodology

This chapter presents a comprehensive overview of the data and methodological approach employed in this study. It details the process of collecting, preprocessing, and annotating wildlife imagery data from Greece, followed by the development and implementation of machine learning models for species classification and threat detection. The chapter outlines the steps taken in dataset creation, model selection, training, and evaluation, culminating in an exploration of how these AI techniques can be applied to enhance wildlife conservation efforts. By providing a clear and detailed account of the research methodology, this chapter lays the foundation for understanding the results and implications discussed in subsequent chapters.

2.1 Data Collection and Description

The data used in this study was provided through a collaboration with Theodoros Kominos. The dataset consists of images and video recordings obtained from trap cameras set up in various locations across Greek wildlife habitats. The initial dataset comprised 3,551 camera trap images from which 2,747 images were successfully annotated. Analysis of the dataset revealed significant variability in species representation, which substantially influenced our subsequent methodological decisions.

The raw data captured a diverse range of wildlife, including:

- Protected species, including brown bear (Ursus arctos), wolf (Canis lupus lupus), chamois (Rupicapra rupicapra balcanica), and wildcat (Felis silvestris)
- Other wild animals, including roe deer (Capreolus capreolus), wild boar (Sus scrofa), red fox (Vulpes vulpes), European badger (Meles meles), other mustelids, and European hare (Lepus europaeus)
- Domestic animals: cows, dogs, horses
- Human activities: hikers, potential hunters, conservation workers, and vehicles

Figure 2.1 Images of a bear (left), image of a wolf (right)

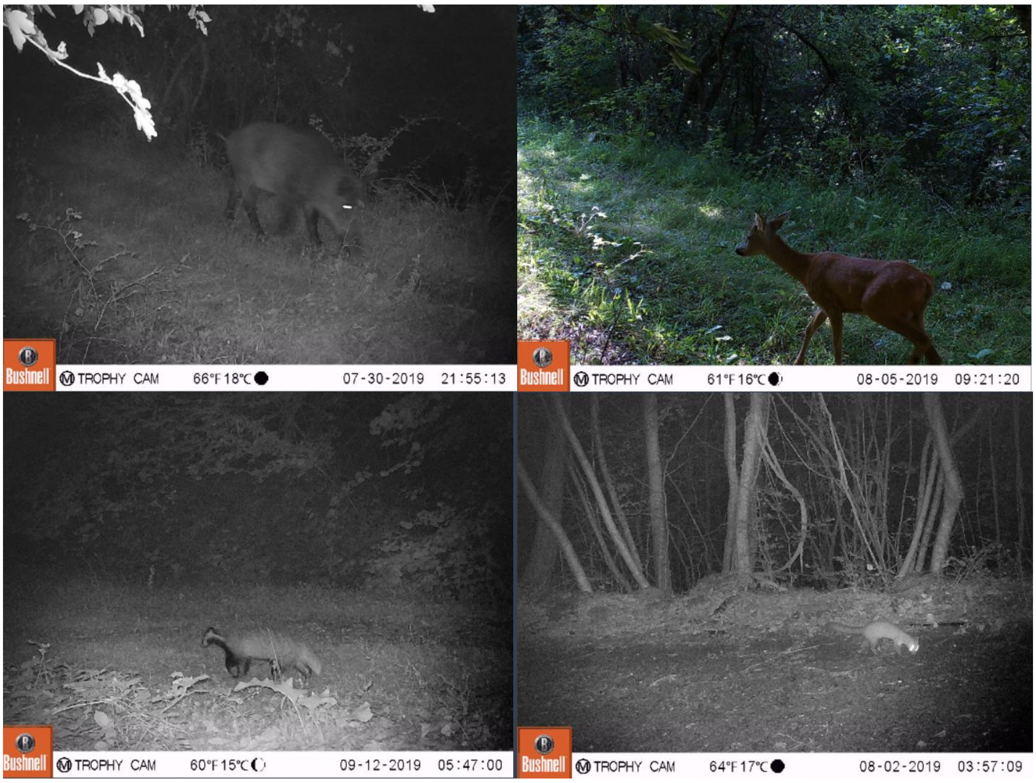


Figure 2.2 Images of wild boar, roe deer, badger and mustelid

The initial dataset included 3,551 images from which 2,747 images were annotated across these classes. The most frequently captured species in the dataset was the roe deer (Capreolus capreolus), representing 29.6% of the annotations with 814 images. This was followed by red fox (Vulpes vulpes) with 411 images (15%), wild boar (Sus scrofa) with 233 images (8.5%), and domestic cattle with 225 images (8.2%). Protected species were present in smaller numbers: Brown Bear (Ursus arctos) with 150 images, Wildcat (Felis silvestris) with 61 images, and Wolf (Canis lupus lupus) with 40 images. Human-related categories, 98 images of people and 101 of vehicles, were present with some data for distinguishing between wildlife and human activity.

It is worth mentioning the significant amount of noise that some of the samples, depending on the subject of the image, some of the captured animals' behavior vary significantly, mainly nocturnal animals are presented mostly in greyscale while mainly diurnal passive herbivores are more present during the day and stay longer times in front of the cameras (Figures 2.1-2.3). Therefore, some species are represented by less and lower quality pictures while others are captured clearly and for more instances.

2.2 Data Preprocessing and Annotation

The preprocessing and annotation phase was crucial in transforming the raw visual data into a structured, labeled dataset suitable for machine learning applications. This section details the systematic approach taken to organize, clean, and annotate the collected images and video footage. The process involved careful data organization to ensure efficient handling, followed by a meticulous annotation procedure using the Roboflow platform. These steps were essential in preparing a high-quality dataset that would serve as the foundation for subsequent model training and evaluation.

2.2.1 Data Organization

The organization phase focused on creating a systematic structure for the visual data to ensure efficient processing and analysis. A comprehensive file organization system was implemented, incorporating logical hierarchical folders and consistent naming conventions. This structure facilitated efficient data access and processing throughout the subsequent stages of the research.

The quality control process involved careful assessment of each image and video file. Through systematic review, duplicate images were identified and removed, while maintaining only the highest quality versions of similar content. Image resolution and clarity were evaluated against predetermined standards, ensuring the final dataset maintained consistent quality levels.

Video processing required particular attention to detail, as each video file needed to be converted into usable frame sequences. This process involved careful consideration of frame extraction intervals to capture relevant information while avoiding redundancy. Each extracted frame underwent quality assessment, with redundant or blurred frames being removed to maintain dataset quality.

Figure 2.3 Sample of Images included in the dataset

2.2.2 Data Annotation

The annotation process was conducted through a collaborative effort, combining the researcher's systematic approach with expert validation from Mr. Theodoros Kominos. This collaboration proved particularly valuable for challenging cases requiring specialized expertise, ensuring high accuracy and reliability in the dataset labeling.

The annotation infrastructure centered around Roboflow.com, selected for its comprehensive annotation capabilities and robust cloud-based collaboration features. The platform's version control and annotation history tracking facilitated efficient collaboration and quality assurance. Supporting tools included custom Python scripts for annotation verification and a comprehensive tracking system to monitor annotation progress.

The annotation workflow was structured to maximize accuracy and consistency. Initial annotations were performed by the primary researcher, followed by regular consultation sessions with Mr. Kominos. These sessions focused on reviewing difficult cases, verifying species identification, and resolving ambiguous examples. All challenging cases and their resolutions were thoroughly documented to maintain consistency in future annotations.

Strict annotation guidelines were established to ensure consistency across the dataset. These guidelines detailed specific protocols for bounding box placement and standardized labeling taxonomy. Special attention was given to handling complex cases such as partially visible subjects, multiple subjects in frame, and obscured or ambiguous situations. Quality assurance checkpoints were integrated throughout the process to maintain high standards.

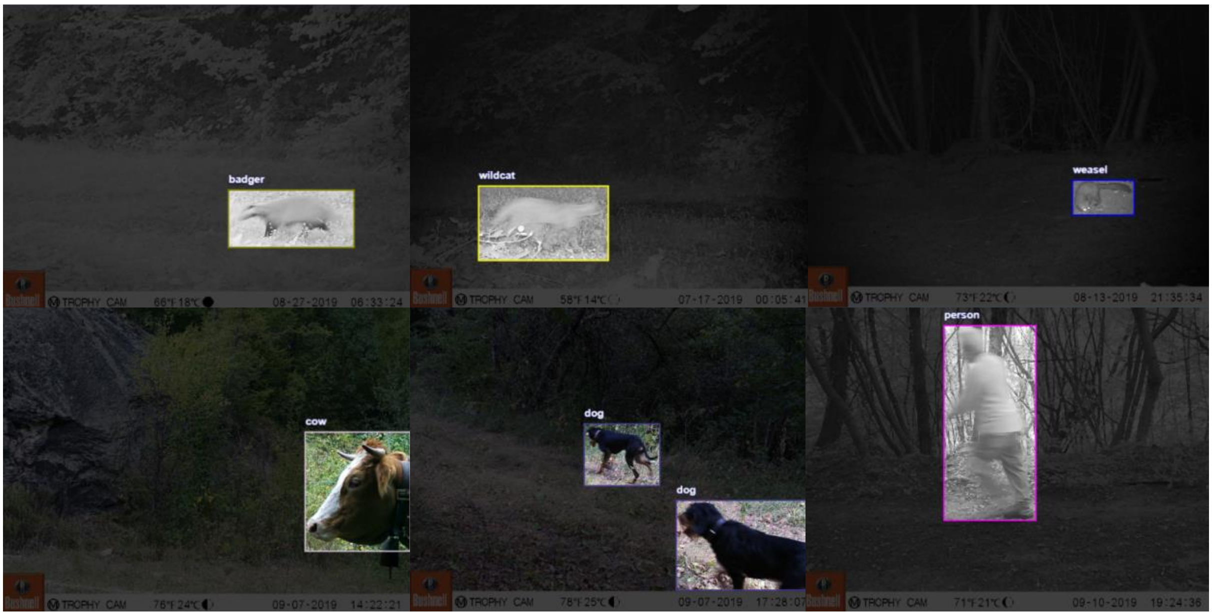


Figure 2.4 Sample of annotated images

2.3 Dataset Preparation and Model Training

Using Roboflow platform, the annotated data was processed to create a robust dataset suitable for machine learning applications. This process included:

1. Splitting the data into training, validation, and test sets. The splitting process followed these guidelines:

   Approximately 70% of images were allocated to the training set, about 20% were assigned to the validation set and the remaining 10% were reserved for the test set. This distribution ensures a sufficient number of images for training while providing adequate data for model validation and final testing.

   While the target split was 70/20/10 for training, validation, and test sets this was adjusted for classes with fewer instances. For these classes, a higher proportion of images were allocated to the training set to help the model to learn the features of these less frequent classes.

Class Name (Species/ Subfamily)	Total Images	Training set	Validation set	Test set
roeDeer (Capreolus capreolus)	814	585	150	79
fox (Vulpes vulpes)	411	295	82	34
wildBoar (Sus scrofa)	233	158	44	31
cow	225	155	43	27
weasel (Mustelinae)	215	141	44	30
hare (Lepus europaeus)	201	137	45	19
Bear (Ursus arctos)	150	104	32	14
car	101	69	25	7
person	98	69	23	6
badger (Meles meles)	64	48	11	5
wildcat (Felis silvestris)	61	45	7	9
chamois (Rupicapra rupicapra balcanica)	54	44	8	2
wolf (Canis lupus lupus)	40	26	10	4
dog	24	21	3	0
horse	20	14	6	0
hedgehog (Erinaceus europaeus)	12	8	3	1
redDeer (Cerus elaphus)	10	6	4	0
bird	7	3	3	1
gun	5	3	2	0
Squirrel (Sciurus vulgaris)	2	2	0	0
Sum of annotated	2747	1933	545	269

Table 2.1 Annotated images counts

Some of the classes were underrepresented (less than 100 images totally in the class) in the dataset, so I decided to apply data augmentation techniques to increase dataset diversity and improve model generalization.

2. Data augmentation techniques were implemented to enhance the robustness and generalization capabilities of the object detection model while addressing the common challenges associated with camera trap imagery. These techniques simulate various real-world conditions and variations that might occur in wildlife photography [36]. The following augmentation methods were applied: Augmentation methods that were applied:

    - Geometric Transformations
      Horizontal Flip: Mirror reflection across the vertical axis, simulating different animal orientations
      Random Rotation: Applied within a range of -15° to +15°, accounting for camera installation variations
      Shear Transformation: Applied at ±15° both horizontally and vertically, helping model adapt to perspective variations

- Photometric Transformations
  Brightness Adjustment: Modulation between -20% and +20% of original intensity, simulating different lighting conditions and times of day
  Gaussian Blur: Applied up to 4.9 pixels, mimicking motion blur and focus variations common in camera trap images
  Random Noise: Injection of noise affecting up to 1.96% of pixels, simulating sensor noise under low-light conditions

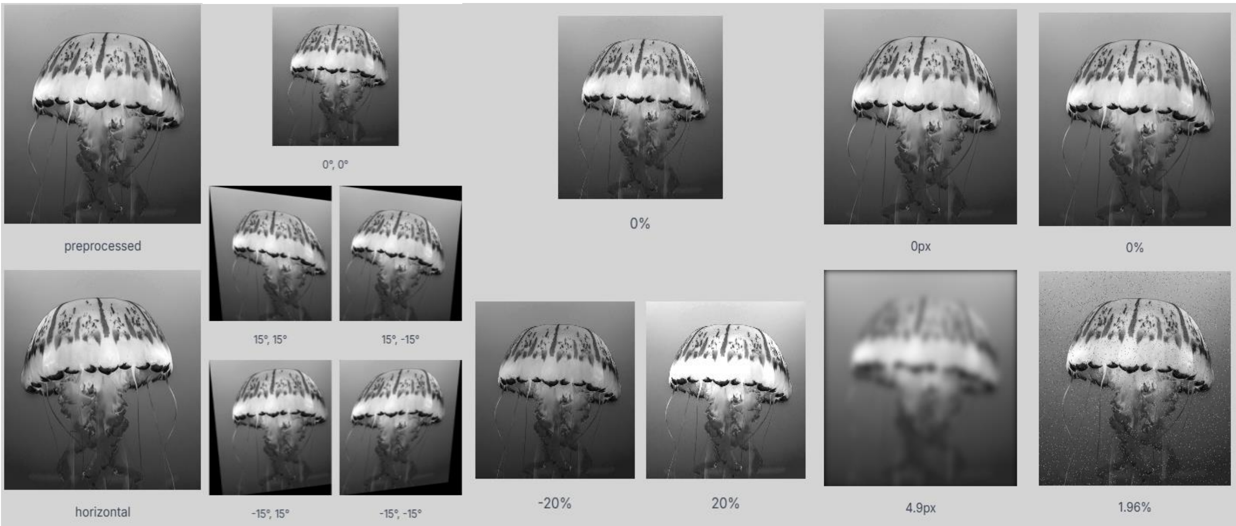


Figure 2.5 Types of augmentation techniques, from left to right
Horizontal Flip, Rotation ±15°, Brightness ±20%, Blur, Noise

To maintain dataset integrity, a comprehensive cross-validation process was implemented to verify the accuracy of annotations and ensure that augmentation procedures did not introduce artifacts or compromise the ecological validity of the images. This validation step was crucial in maintaining the quality of the training data throughout the splitting and augmentation pipeline.

2.3.1 Model Selection, Implementation and Metrics

Two main options provided by Roboflow were explored for the machine learning implementation:

1. Roboflow 3.0 Object Detection (Fast or Accurate)
   This is the recommended Model Architecture from the platform, it is claimed to provide higher rates of accuracy and competitive training times across the other available options on the platform.

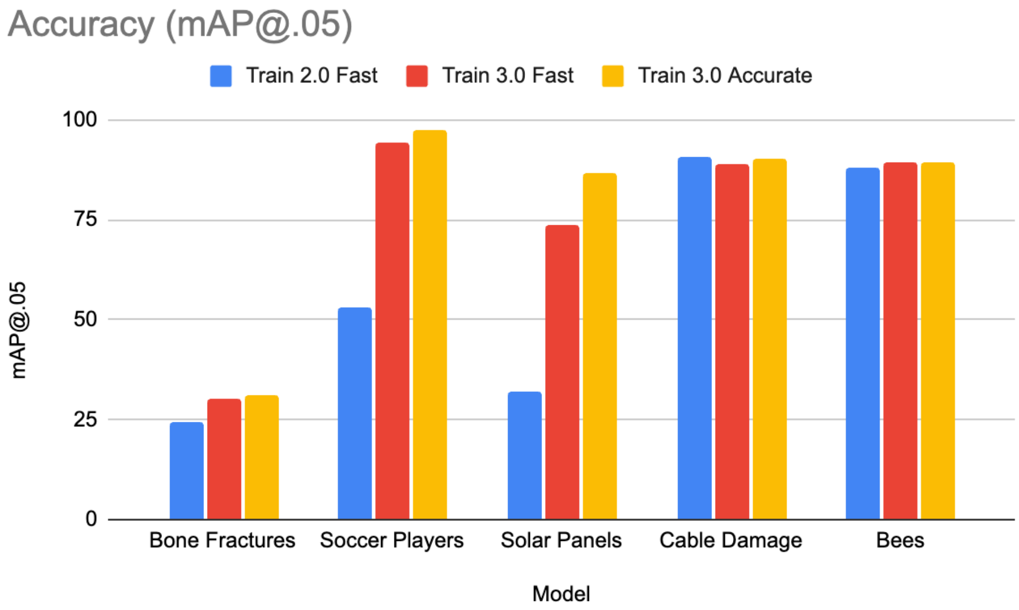


Figure 2.6 Roboflow Train 3.0 comparison to it predecessor [27]

2. YOLO-NAS (You Only Look Once - Neural Architecture Search) A cutting-edge object detection model developed by Deci AI. It distinguishes itself from previous YOLO versions by leveraging Neural Architecture Search (NAS) technology to discover an optimal architecture for object detection, leading to improved accuracy and efficiency.

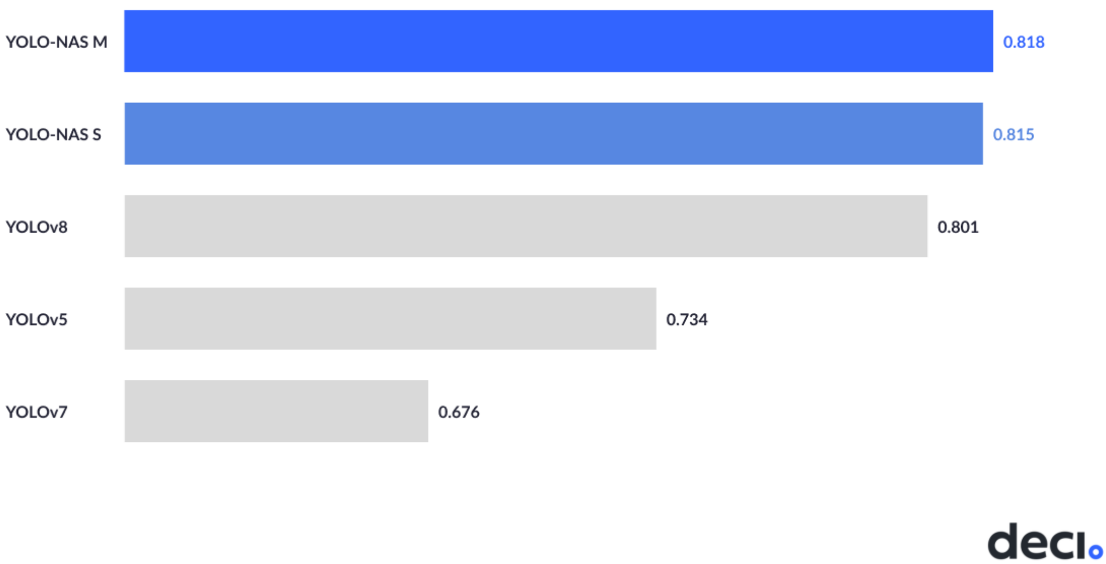


Figure 2.7 Average mAP on Roboflow-100 for YOLO-NAS vs other models [28]

YOLO (You Only Look Once)

YOLO, which stands for "You Only Look Once," is a deep learning-based real-time object detection algorithm. First introduced by Joseph Redmon et al. in 2015[31], it revolutionized object detection by introducing a single-stage detection framework that treats object detection as a regression problem. Unlike traditional two-stage detectors (like R-CNN family), YOLO divides the image into a grid and predicts bounding boxes and class probabilities simultaneously, enabling real-time detection. Various versions of YOLO have been developed over time to improve accuracy, detection speed, and robustness, especially for applications in dynamic settings .

Core Architecture and Methodology

YOLO's architecture is based on a single Convolutional Neural Network (CNN) that processes the entire image at once, rather than performing multiple region-based scans. The CNN divides the input image into an $S \times S$ grid, where each grid cell is responsible for detecting objects that fall within its boundaries. This grid-based approach allows YOLO to handle object detection in a computationally efficient manner. Each cell predicts a fixed number of bounding boxes and provides:

- Bounding Box Coordinates: Representing the location and size of the box around each detected object.
- Confidence Score: Indicating the certainty of an object being present in a bounding box.
- Class Probabilities: A vector representing the likelihood of each detected object belonging to each class.

These three elements are then combined to identify objects and assign them the most likely labels with corresponding confidence scores. YOLO uses a technique called Non-Maximum Suppression (NMS) to refine its detections by keeping the bounding box with the highest confidence score and eliminating redundant detections for the same object.

The model has evolved through numerous versions, each introducing critical improvements to enhance both detection capabilities and versatility:

YOLOv2 [37]: This second iteration introduced anchor boxes, batch normalization, and dimension clustering, significantly improving accuracy and localization precision.

YOLOv3 [38]: This version featured an improved backbone network (Darknet-53) and multiple anchor boxes for multi-scale object detection. It also introduced a more refined Spatial Pyramid Pooling (SPP) approach for better feature extraction.

YOLOv4 [35]: Developed with a focus on optimizing detection accuracy without sacrificing speed, YOLOv4 introduced Mosaic data augmentation, an anchor-free detection head, and a new loss function to improve performance on complex scenes.

YOLOv5 [41]: Developed by Ultralytics, YOLOv5 brought significant enhancements, such as hyperparameter optimization, integrated experiment tracking, and automatic export options for popular formats, making it highly accessible to developers and researchers.

YOLOv6 [39]: Open-sourced by Meituan, this version found applications in autonomous delivery systems, emphasizing computational efficiency and versatility in real-world applications.

YOLOv7 [40]: Building on previous innovations, YOLOv7 added capabilities for pose estimation on keypoint datasets (such as COCO), expanding the model's applications to broader computer vision tasks.

YOLOv8 [32]: Released by Ultralytics, YOLOv8 introduced features that improved flexibility, efficiency, and overall performance, supporting a range of vision AI tasks including classification, detection, and segmentation.

YOLOv9 [42]: This version incorporated innovations such as Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) to further enhance detection and efficiency.

YOLOv10 [43]: Developed by Tsinghua University using the Ultralytics Python framework, YOLOv10 introduced an End-to-End head architecture, which eliminated the need for Non-Maximum Suppression (NMS) and streamlined processing.

YOLOv11 [44]: The latest release from Ultralytics focuses on delivering state-of-the-art (SOTA) performance across tasks including detection, segmentation, pose estimation, tracking, and classification, enhancing its versatility across a wide range of AI applications.

This progression of YOLO versions highlights the algorithm’s adaptability and enduring relevance in fields requiring both high-speed processing and accuracy, from autonomous vehicles to surveillance and wildlife monitoring. The continual innovations within the YOLO framework exemplify the rapid advancements in deep learning for object detection.

Strengths of YOLO

- Speed: YOLO's single-pass detection allows it to run at high frame rates, making it suitable for real-time applications.
- Accuracy: YOLO achieves high accuracy with relatively simple architecture, balancing localization and classification effectiveness.
- Adaptability: It can be adapted for a range of object detection tasks, from general-purpose applications to specialized tasks.

Limitations of YOLO

- Coarse Localization: YOLO sometimes struggles with fine-grained localization in images with closely spaced objects.
- Struggles with Small Objects: Although later versions improved on this, YOLO can sometimes miss smaller objects in images, which may be a challenge when detecting small animals or distant wildlife in camera trap images.

YOLO-NAS (YOLO with Neural Architecture Search)

YOLO-NAS is a state-of-the-art object detection framework that leverages quantization-aware design and selective quantization techniques to achieve optimal performance. By quantizing the model, it maintains high accuracy while significantly reducing computational costs. This innovative approach results in a more efficient and powerful architecture compared to traditional YOLO models.
A key advantage of YOLO-NAS is its ability to adapt to diverse requirements. Unlike manually designed architectures, YOLO-NAS employs AutoML techniques to tailor its structure to specific tasks and datasets. This flexibility allows for the creation of models that are well-suited for real-time applications, resource-constrained environments, or specialized domains.
The YOLO-NAS framework incorporates advanced training methodologies and quantization techniques to enhance its overall performance. It is pre-trained on large-scale datasets like COCO, Objects365, and RoboFlow 100, making it readily adaptable for object detection tasks in production settings.

After several iterations and comparative analyses, Roboflow 3.0 was selected as the primary model for this study. The selection process involved:

1. Training both models on the prepared dataset
2. Evaluating performance metrics
3. Assessing inference speed and resource requirements
4. Considering the ease of deployment and integration with existing conservation tools

Metrics

Object detection model performance is evaluated using Mean Average Precision (mAP), mAP is a key metric used to evaluate the performance of object detection models. It provides a single score that reflects how accurately a model can identify and localize objects within images using bounding boxes.

Calculation of mAP:

Intersection over Union (IoU): IoU measures the overlap between the predicted bounding box and the actual ground truth box, with a higher IoU indicating a more accurate match. To compute mAP, an IoU threshold is first set, defining the minimum overlap required for a detection to be considered correct.

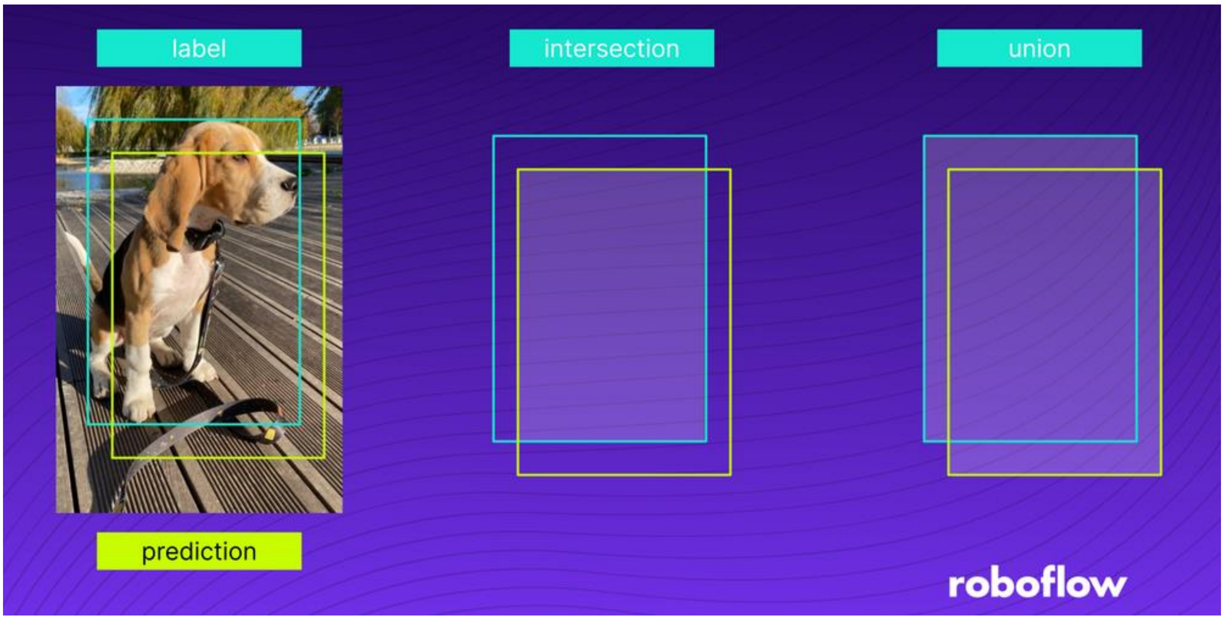


Figure 2.8 How IoU applies to an image [26]

Precision and Recall: Precision measures the proportion of predicted bounding boxes that are correct, while Recall measures the proportion of actual objects that were successfully detected by the model. Together, these metrics provide insight into the model's accuracy and its ability to detect relevant objects.

Average Precision (AP): For each class of objects, we calculate the average precision across different levels of confidence. This gives us a measure of how well the model performs for that specific class.

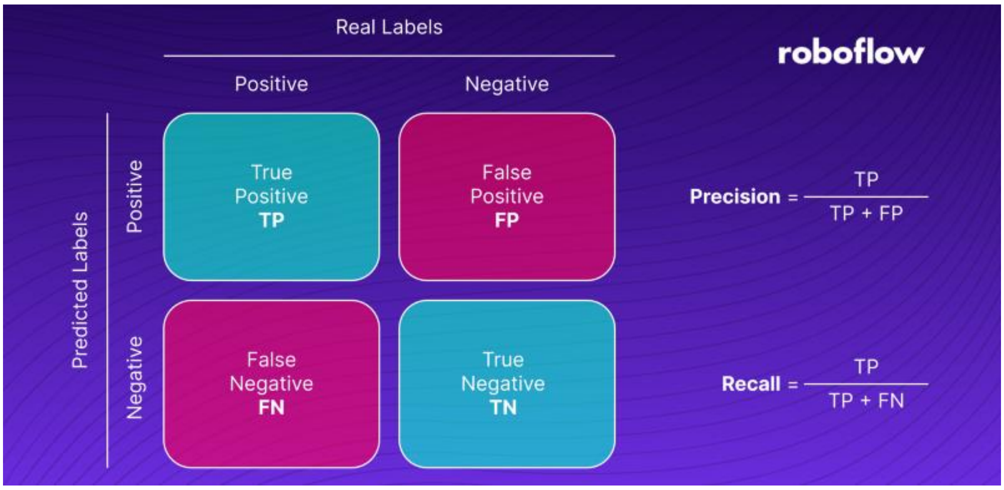


Figure 2.9 Precision and Recall [26]

Mean Average Precision (mAP): The overall Mean Average Precision (mAP) score is obtained by averaging the Average Precision (AP) values across all object classes, resulting in a single metric that represents the model's general accuracy in identifying and localizing objects within images.

Choosing the IoU Threshold:
The Intersection over Union (IoU) threshold can greatly impact the mAP score. A higher IoU threshold (e.g., 0.75) requires the model to be more precise in aligning bounding boxes with ground truth boxes for a detection to be considered correct. IoU is used during model evaluation to compare each predicted bounding box with the corresponding ground truth box, determining whether a detection meets the accuracy threshold to be considered correct. It's common to calculate mAP across a range of IoU thresholds, typically from 0.5 to 0.95, then average these scores to obtain a more comprehensive evaluation of the model's performance.

In simpler terms, if a model is trained to detect objects like cats and dogs, mAP assesses how well it can:

- Correctly identify whether an object is a cat or a dog.
- Accurately place bounding boxes around the cats and dogs in the image.

A higher mAP indicates that the model excels in both distinguishing objects and accurately localizing them, reflecting stronger overall performance.

2.3.2 Class Reduction

During the initial stages of model training, it became clear that some classes in the dataset were significantly underrepresented. This imbalance posed challenges for the model's ability to accurately detect and classify these less frequent species or objects. After conducting several training iterations and analyzing the results, a decision was made to reduce the number of classes being detected.

The primary reasons for class reduction were:

- Insufficient training data: Some classes had very few examples (less than 20 images), which is generally not enough for the model to learn robust features for accurate detection.
- Model performance: The underrepresented classes were causing decreased overall model performance, as the model struggled to differentiate between these rare classes and more common ones.
- Conservation priorities: By focusing on the more prevalent and ecologically significant species, the model could better serve immediate conservation needs.

As a result of this process, several classes with few instances were initially removed from the detection task, including dogs, horses, hedgehogs, red deer, birds, guns, and squirrels.

However, after further consideration and experimentation, a more radical simplification was implemented. The final decision was made to consolidate all animal classes into a single "animal" class. This approach was chosen for several reasons:

- Improved model generalization: By focusing on detecting the presence of any animal, the model could leverage features common to all wildlife, potentially improving overall detection rates.
- Simplified data requirements: This approach could reduce the need for species-specific labeling, according to specific use case, which can be time-consuming and requires expert knowledge.
- Broader applicability: A general "animal" detector could be more easily applied to new environments or used as a first-stage filter in a multi-stage detection pipeline.
- Focus on primary conservation goal: For many applications, simply detecting the presence of wildlife (as opposed to human activity) is the crucial first step.

This decision to use a single "animal" class aligns with successful approaches in the field, such as MegaDetector [24], which has been widely adopted for its ability to efficiently detect animals in diverse camera trap datasets. Similarly, the PyTorch Wildlife project [25] has demonstrated the effectiveness of transfer learning from general object detection models to wildlife-specific tasks.

This final reduction to a single "animal" class allowed the model to focus on the fundamental task of distinguishing wildlife from non-wildlife elements in the camera trap images. While this approach sacrifices species-specific information, it provides a robust foundation for detecting animal presence in diverse environments.

It's important to note that while the model was trained on this simplified class structure, the original dataset with detailed species classifications is preserved. Roboflow platform allows keeping several different versions of your custom dataset. Potential future expansion of the model's capabilities is possible as more data becomes available or as conservation priorities shift, potentially reintroducing multi-class species detection.

3 CHAPTER 3: Results, Analysis and Discussion

This chapter presents the outcomes of the research of this Thesis. I present the stages of the training process and iterations.

3.1 Model Performance Results

Here we present some of the train Metrics of our Roboflow 3.0 (Fast) model in detecting objects between the 20 different classes.

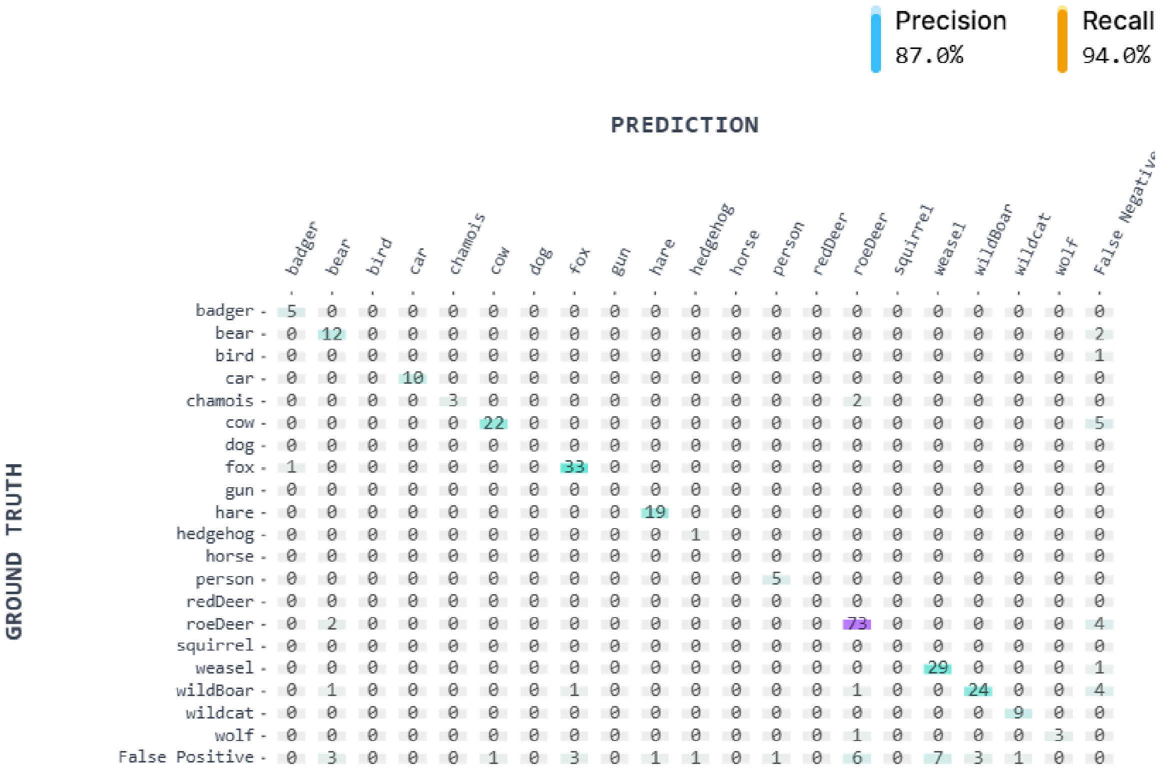


Figure 3.1 Confusion matrix of 20 class model

Average Precision by Class (mAP50) for the Test Set was the following.

Figure 3.2 Average Precision by Class (mAP50) for the Test Set

The Training Graphs of the model are the following.

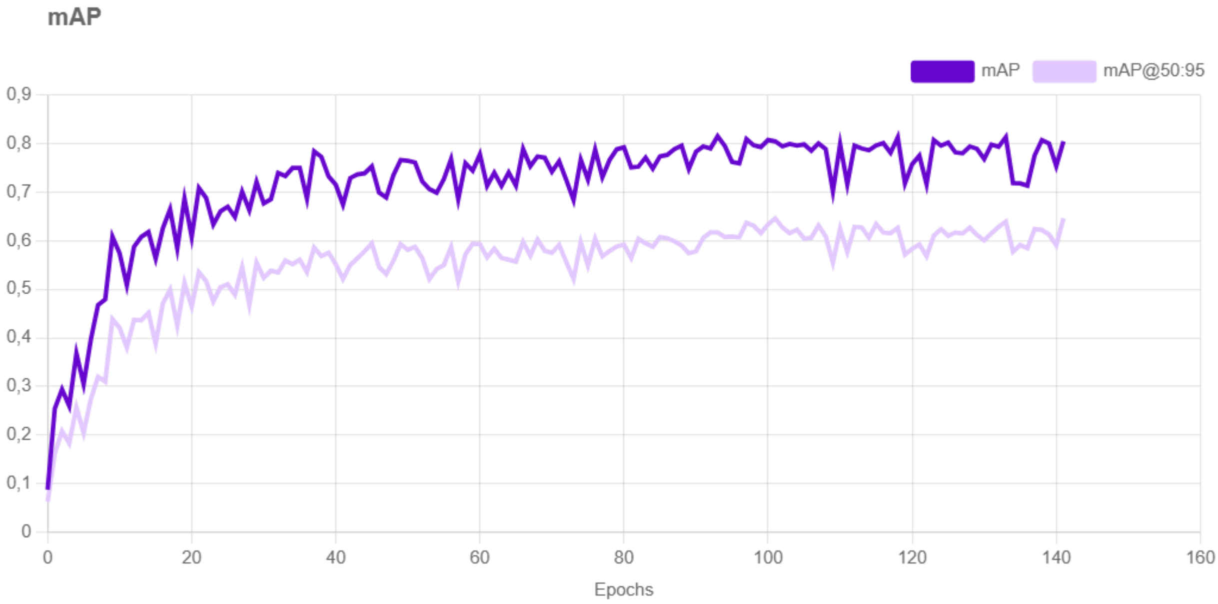
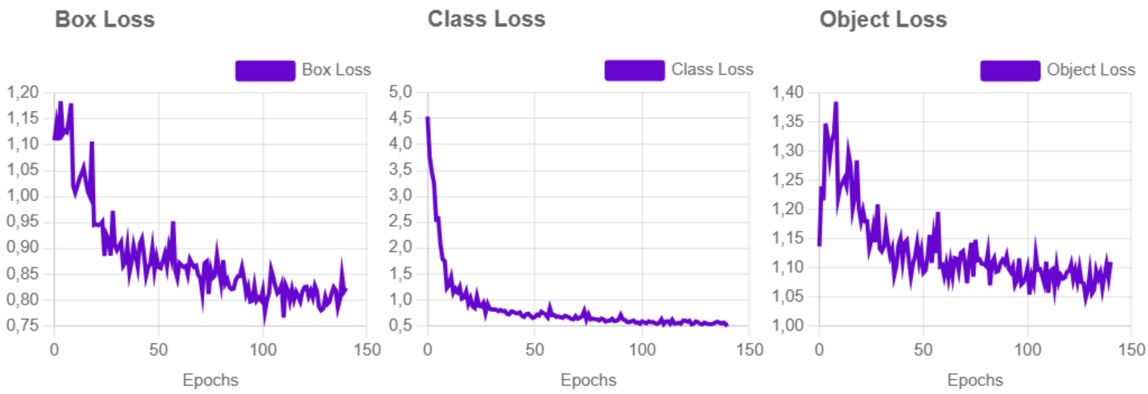


Figure 3.3 mAP (mAP@0.5) and mAP@0.5:0.95 across training of 20 class model



Figures 3.4 Training Graphs of 20 class model

I trained the model after stretching the images to 640x640, in Grayscale and filtered the null images to 15% of the total dataset.
Figure 3.4 illustrates that the model is prone to distractions from non-target objects within the images. Although it generally succeeds in detecting the presence of animals, it often misclassifies them and, in some cases, assigns multiple classes to a single object. This indicates a challenge in maintaining both specificity and accuracy, particularly when distinguishing among similar classes.

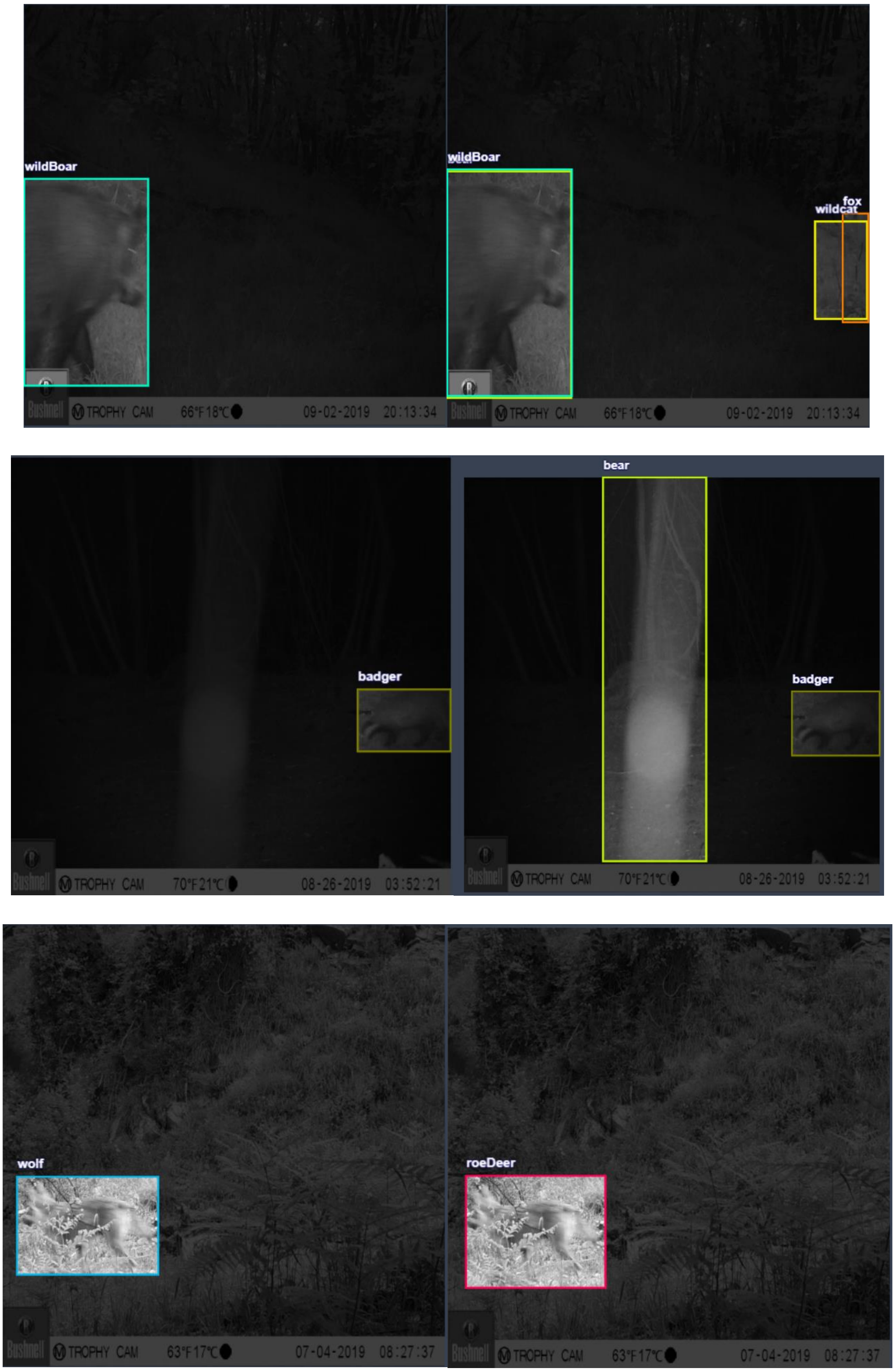


Figure 3.5 Sample of the test images, Ground Truth (left) and Model Predictions (right) (20 class model)

3.2 Class drop and Augmentation

In this model training I decided to drop some classes that were either under or over represented in the dataset and tried to develop a model that detects the incidents of interest, so I kept person and car and dropped deers, squirrel, bird, chamois, hare and hedgehog. Also, it is worth mentioning that train started using the previous model as a checkpoint.

3.2.1 Additional Augmentation steps

I also added some additional Augmentation steps to the improve the performance of the model:

- Horizontal Flip
- Rotation between -15° and +15°
- Shear ±15° Horizontal, ±15° Vertical
- Brightness adjustment between -20% and +20%
- Blur up to 4.9px
- Noise up to 1.96% of pixels

I also limited the augmented outputs per training example to 3. This dataset version contains in total 4041 images.

3.2.2 10 class Model Performance Results

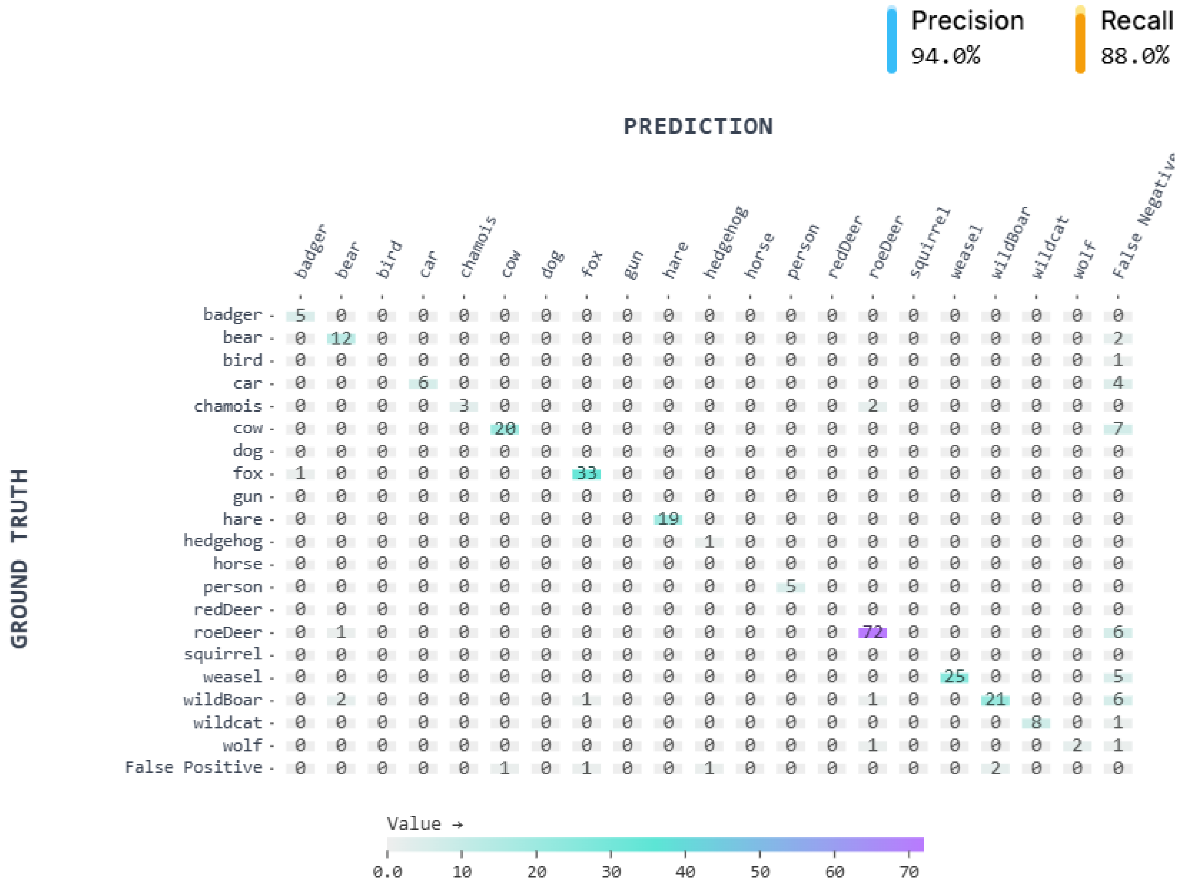


Figure 3.6 Confusion matrix of 10 class model on test dataset

Figure 3.7 Average Precision by Class (mAP50) for the Test Set of model with 10 classes

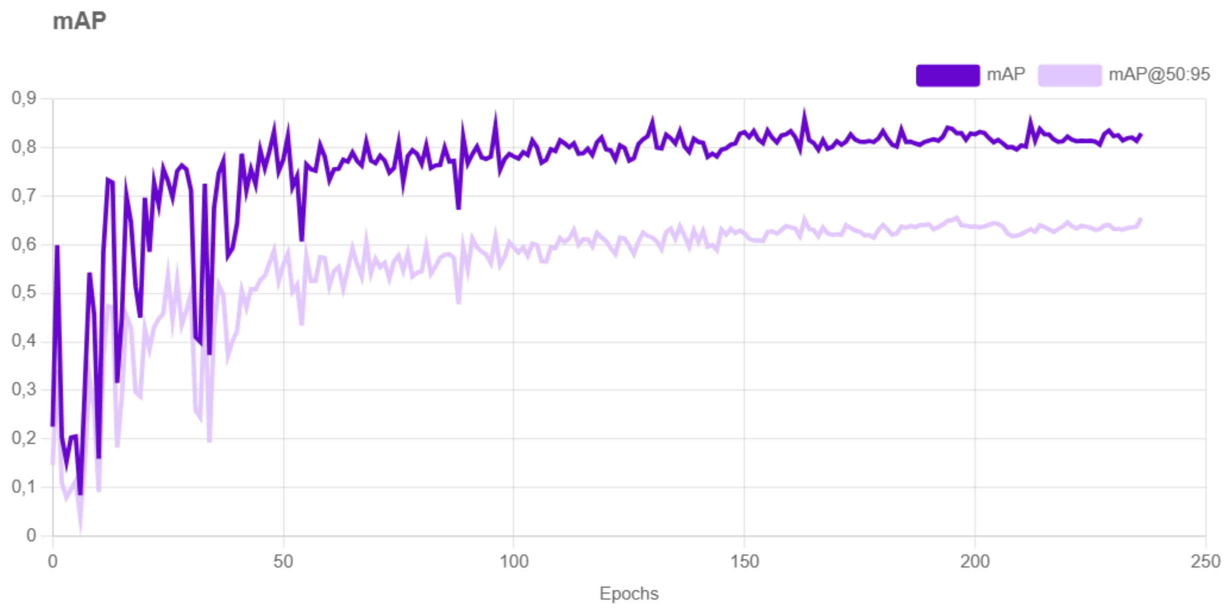
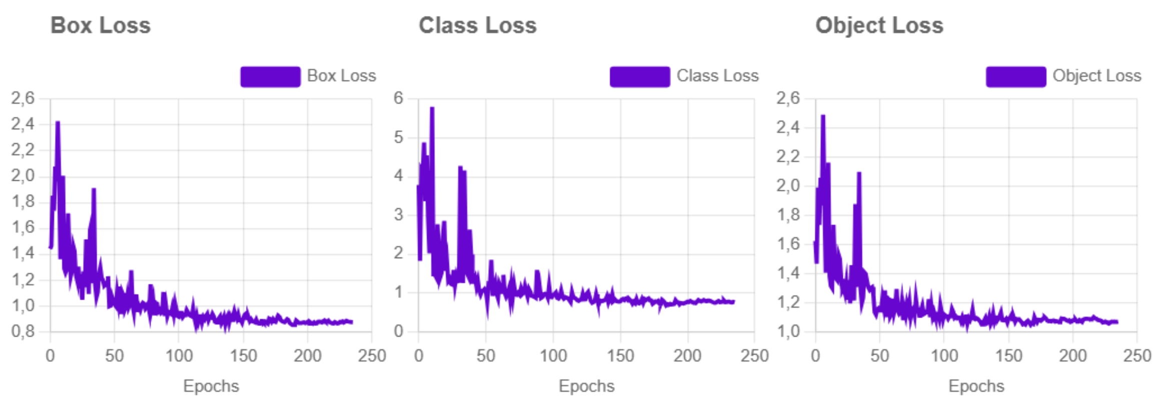
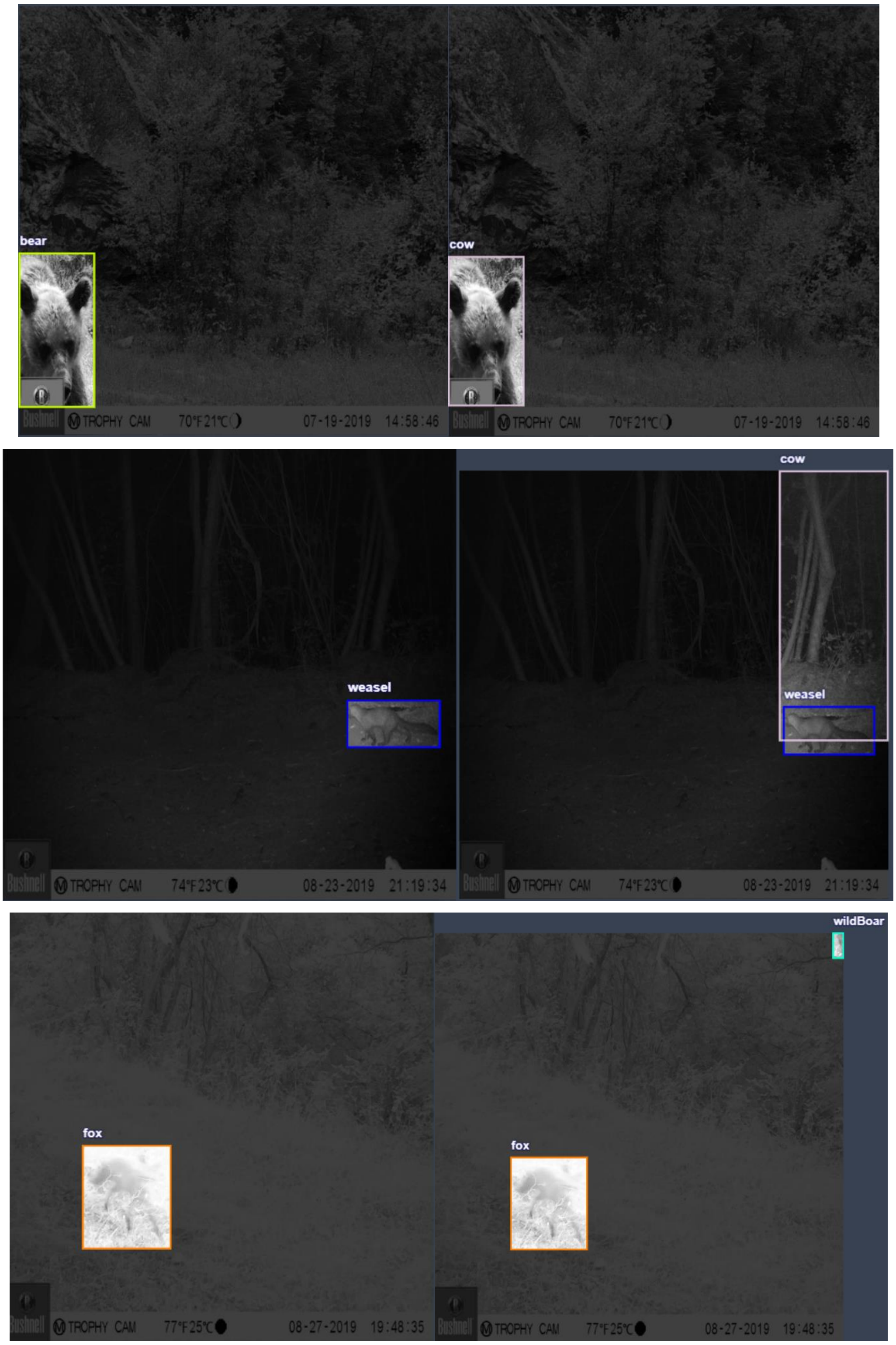


Figure 3.8 mAP (mAP@0.5) and mAP@0.5:0.95 across training of 10 class model



Figures 3.9 Training Graphs of 10 class model



Figures 3.10 Sample of the test images, Ground Truth (left) and Model Predictions (right) (10 class model)

3.3 Animal Detection model and deployment

After reducing the number of classes, I observed an improvement in the model's performance for detecting animals and accurately assigning classes. However, some erroneous detections still occurred, with the model occasionally identifying animals in background areas that contained no relevant objects. Drawing inspiration from the MegaDetector approach, I decided to train a model focused solely on detecting the presence of any animal within an image.

3.3.1 Animal detection model training

I remapped all classes that are animals to one new class, Animal.
I also dropped person, car and gun classes.
Lastly, I removed the Shear augmentation method because I noticed that it results in fault annotations, specifically there were some imageses that area of the ribbon of the cameras were in the annotated augmented results.
The dataset is contained by 6443 images and I decided that the model would start from the previous checkpoint (with the 10 classes).

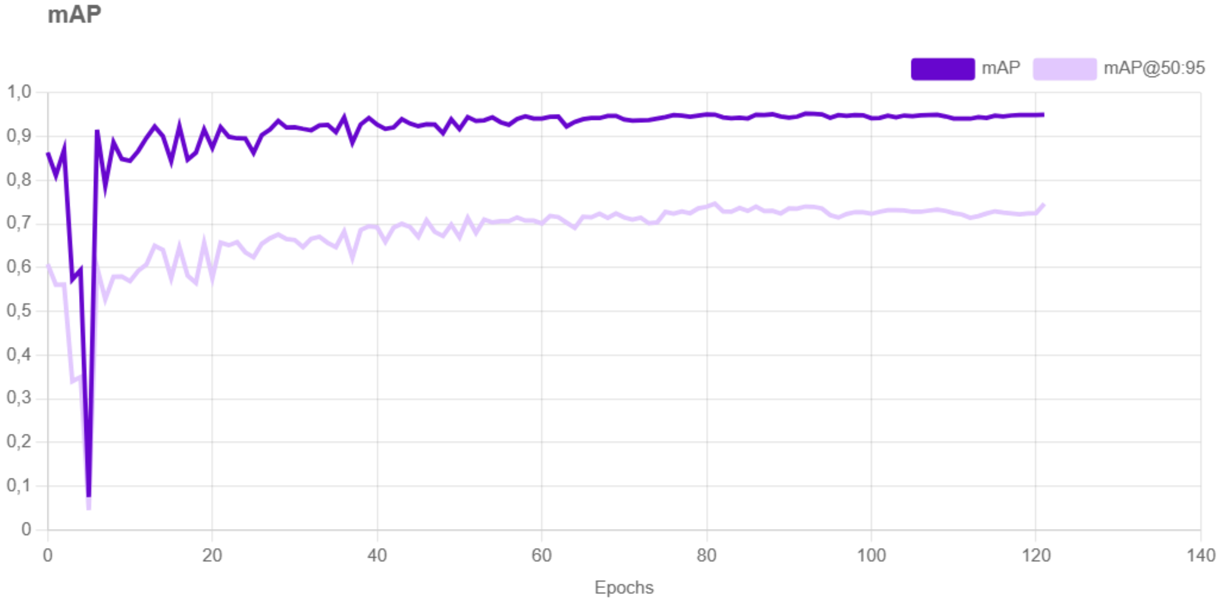
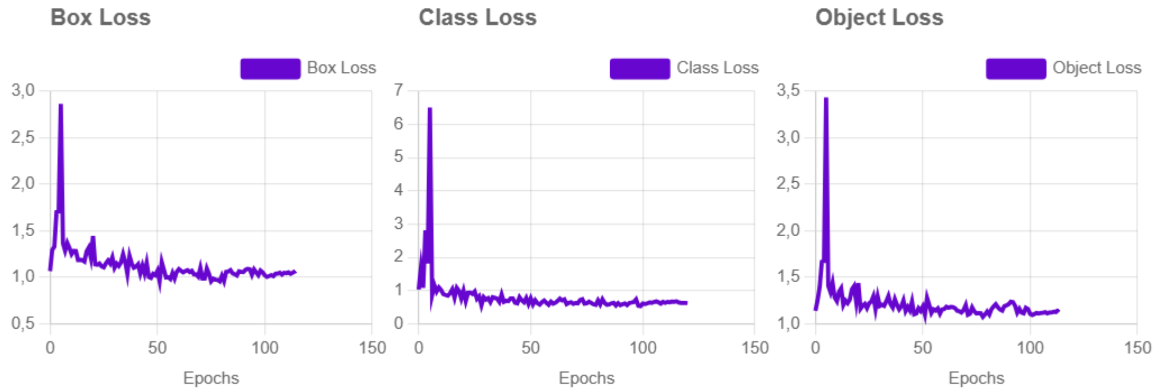


Figure 3.11 mAP (mAP@0.5) and mAP@0.5:0.95 across training (of animal detection model)



Figures 3.12 Animal detection Training Graphs

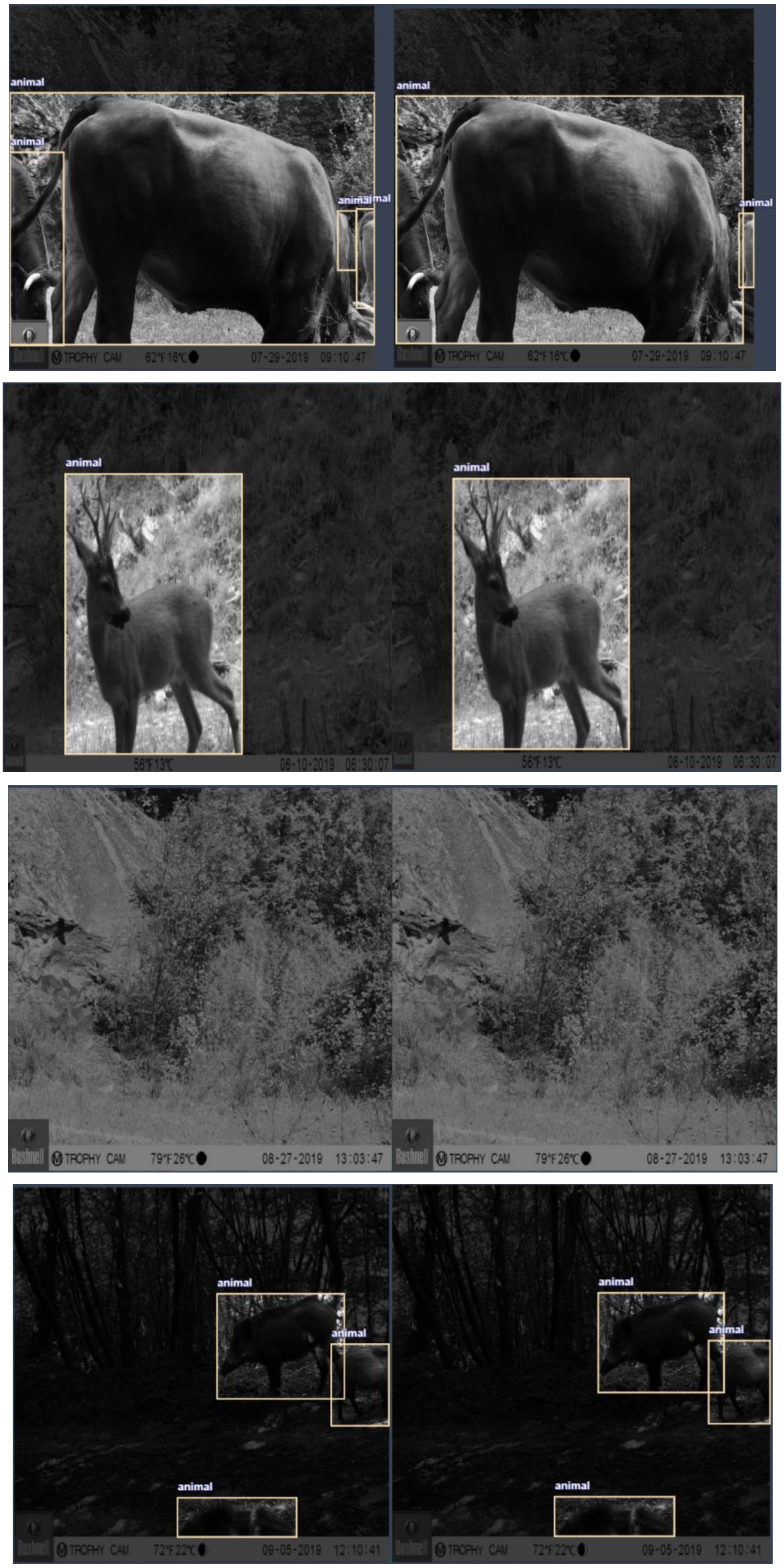


Figure 3.13 Sample of the test images, Ground Truth (left) and Model Predictions (right)

3.3.2 Testing and deployment

Roboflow platform provides several deployment options once you have trained your model.

It is available for the account user to try the trained models on videos from the web, on their mobile using their integrated camera and also on edge devices like Jetson Nano through their Hosted API.

It was a quite a surprise that the model performed quite well on range of similar content I have tried it on.
I have provided the QR codes of all the three models on the Appendix so any user may try it on their mobile phone with a camera accessing the internet. The following screenshots are from my mobile phone pointing on video footage from the web and it seems that the model generalizes in an interesting way and detects animal species that has not been exposed in the past like the bird in the next figure.

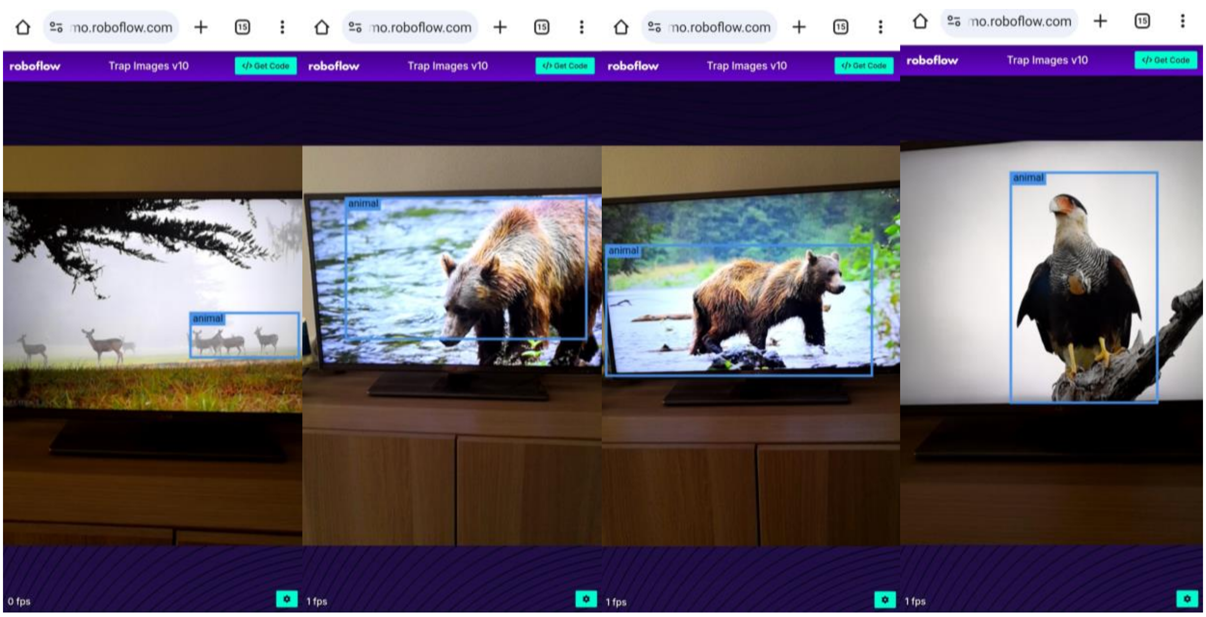


Figure 3.14 Animal detection on mobile

I also tried the inference server option hosted on a Jetson Nano Developer Kit 2GB that was provided to the students of the Master program through NVIDIA's academic Grant program.

The implementation was tested on some pictures from trap cameras that were not exposed to the model and it seem to predict in a satisfactory level.
There is a limitation for the deployment for Roboflow developer accounts as for the one that was used for this project, an internet connection is necessary for the Hosted API call, but there are alternative plans for Enterprise deployments. At the appentix there is a screenshot of the model calling to predict if there is an animal in an image with a canine taken from a camera trap during nighttime and it is executed on the mentioned edge hardware.

4 CONCLUSIONS

This thesis has explored the application of Computer Vision and Artificial Intelligence techniques in wildlife conservation, focusing on the analysis of trap camera footage from Greek wilderness areas. The research aimed to develop tools for automatic species identification and threat detection, with a particular emphasis on creating practical, accessible solutions for conservationists in the field.

4.1 Key Findings

1. **Model Development and Performance**: The study successfully developed a computer vision model capable of detecting animals in camera trap images. Through iterative refinement, including class reduction and data augmentation, the final model demonstrated robust performance in distinguishing wildlife from non-wildlife elements.
2. **Challenges in Multi-species Classification**: Initial attempts at multi-species classification faced challenges due to class imbalance and limited data for some species. This led to the strategic decision to focus on a binary classification (animal vs. non-animal), which improved overall model performance and generalizability.
3. **Data Augmentation Effectiveness**: The implementation of various data augmentation techniques, including horizontal flips, rotations, and brightness adjustments, proved effective in enhancing the model's ability to generalize across diverse image conditions.
4. **Real-world Applicability**: Testing on mobile devices and edge computing platforms (Jetson Nano) demonstrated the model's potential for real-time, in-field application, aligning with the research goal of creating accessible tools for conservationists.
5. **Generalization to Unseen Species**: Interestingly, the model showed some ability to generalize to species not present in the training data, as evidenced by successful detection of birds in mobile testing.

4.2 Implications for Wildlife Conservation

This research contributes to the field of wildlife conservation by:

- Providing a foundation for automated analysis of camera trap data, potentially saving significant time and resources in conservation efforts.
- Demonstrating the feasibility of deploying AI models on edge devices for real-time wildlife monitoring in remote areas.
- Offering a scalable approach that can be adapted to different ecosystems and conservation priorities.

4.3 Limitations and Future Work

While the study achieved its primary objectives, several limitations and areas for future research were identified:

1. **Species-specific Classification**: Future work should focus on expanding the dataset to enable reliable multi-species classification, which would provide more detailed ecological insights.
2. **Behavioral Analysis**: Incorporating video analysis could enable the study of animal behaviors, providing richer data for conservation research.
3. **Integration with Existing Systems**: Further research is needed to seamlessly integrate these AI tools with existing conservation practices and databases.
4. **Ethical Considerations**: As AI becomes more prevalent in conservation, careful consideration must be given to the ethical implications of increased surveillance of wildlife and potential unintended consequences.

In conclusion, this thesis demonstrates the potential of AI and computer vision in revolutionizing wildlife conservation practices. By providing efficient, accessible tools for wildlife detection and monitoring, this research contributes to the ongoing efforts to protect biodiversity in the face of growing environmental challenges. The successful deployment on mobile and edge devices points to a future where AI can empower conservationists with real-time insights, enabling more rapid and effective responses to threats to wildlife and their habitats.

Bibliography – References – Online sources

1. J. D. C. Linnell *et al.*, "The challenges and opportunities of coexisting with wild ungulates in the human-dominated landscapes of Europe's Anthropocene," *Biological Conservation*, vol. 244, p. 108500, Apr. 2020, doi: 10.1016/j.biocon.2020.108500.
2. A. A. Karamanlidis, M. de G. Hernando, L. Georgiadis, and J. Kusak, "Activity, movement, home range and habitat use of an adult gray wolf in a Mediterranean landscape of northern Greece," *Mammalia*, vol. 81, no. 1, pp. 95–99, Jan. 2017, doi: 10.1515/mammalia-2015-0091.
3. Y. Iliopoulos, C. Astaras, and E. Chatzimichail, "Dogs, not wolves, most likely to have caused the death of a British tourist in northern Greece," *Nature Conservation*, vol. 50, pp. 115–143, Oct. 2022, doi: 10.3897/natureconservation.50.81915.
4. A. Karamanlidis, A. Sanopoulos, L. Georgiadis, and A. Zedrosser, "Structural and economic aspects of human–bear conflicts in Greece," *URSUS*, vol. 22, pp. 141–151, Nov. 2011, doi: 10.2307/41304066.
5. M. Petridou, J. F. Benson, O. Gimenez, and V. Kati, "Spatiotemporal Patterns of Wolves, and Sympatric Predators and Prey Relative to Human Disturbance in Northwestern Greece," *Diversity*, vol. 15, no. 2, Art. no. 2, Feb. 2023, doi: 10.3390/d15020184.
6. V. Kati, C. Kassara, D. Vassilakis, and H. Papaioannou, "Balkan Chamois (Rupicapra rupicapra balcanica) Avoids Roads, Settlements, and Hunting Grounds: An Ecological Overview from Timfi Mountain, Greece," *Diversity*, vol. 12, no. 4, Art. no. 4, Apr. 2020, doi: 10.3390/d12040124.
7. A. Galanaki *et al.*, *USE OF CAMERA TRAPS IN THE MANAGEMENT OF PROTECTED AREAS: RECORD OF HUMAN ACTIVITIES IN THE RHODOPE MT NATIONAL PARK*. 2016. doi: 10.13140/RG.2.2.12764.80008.
8. C. Taklis, "Using Security Cameras, Doorbell Cameras, and Trap Cameras to Monitor the Behavior of Wild Boar (Sus Scrofa), in Liri, South East Pelion, Greece," *International Journal of Science and Research (IJSR)*, vol. 10, pp. 1091–1092, Apr. 2021, doi: 10.21275/SR21423195540.
9. K. S. Tanwar, A. Sadhu, and Y. V. Jhala, "Camera trap placement for evaluating species richness, abundance, and activity," *Sci Rep*, vol. 11, no. 1, Art. no. 1, Nov. 2021, doi: 10.1038/s41598-021-02459-w.
10. L. C. D. Dickson, S. R. B. Negus, C. Eizaguirre, K. A. Katselidis, and G. Schofield, "Aerial Drone Surveys Reveal the Efficacy of a Protected Area Network for Marine Megafauna and the Value of Sea Turtles as Umbrella Species," *Drones*, vol. 6, no. 10, Art. no. 10, Oct. 2022, doi: 10.3390/drones6100291.
11. A. F. O'Connell, J. D. Nichols, and K. U. Karanth, Eds., *Camera Traps in Animal Ecology*. Tokyo: Springer Japan, 2011. doi: 10.1007/978-4-431-99495-4.
12. J. A. Ahumada *et al.*, "Wildlife Insights: A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet," *Environmental Conservation*, vol. 47, no. 1, pp. 1–6, Mar. 2020, doi: 10.1017/S0376892919000298.
13. M. S. Norouzzadeh *et al.*, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, Jun. 2018, doi: 10.1073/pnas.1719367115.

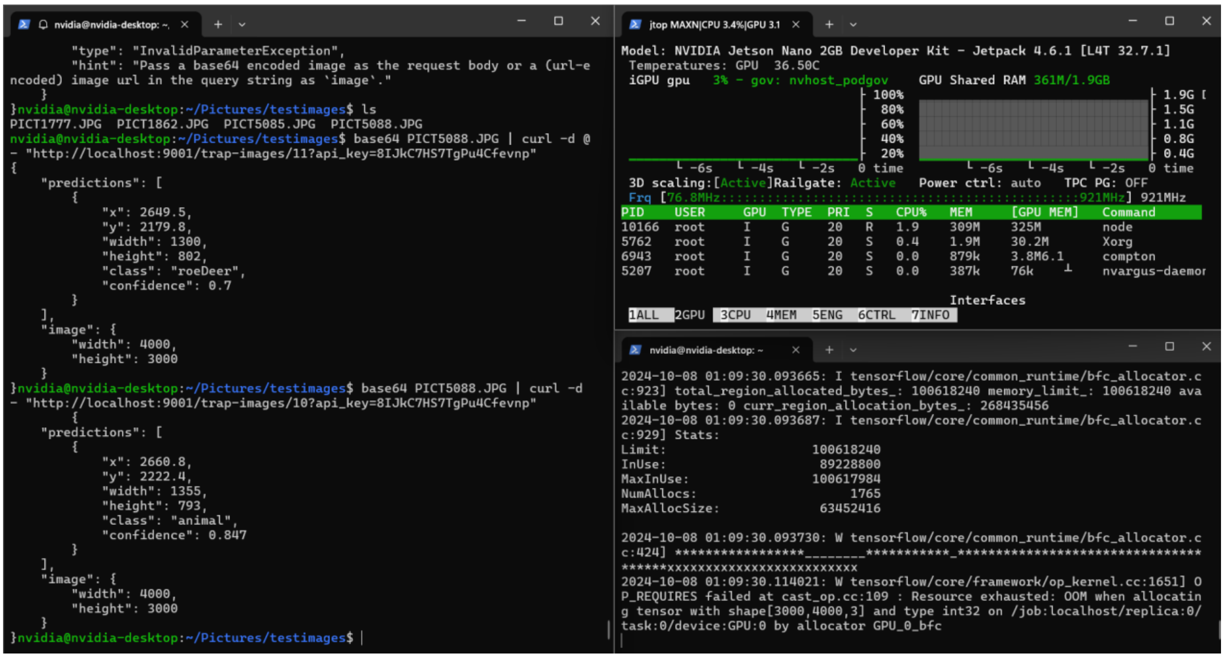
14. "Dawn of Drone Ecology: Low-Cost Autonomous Aerial Vehicles for Conservation - Lian Pin Koh, Serge A. Wich, 2012." Accessed: Sep. 30, 2024. [Online]. Available: https://journals.sagepub.com/doi/10.1177/194008291200500202
15. M. Mulero-Pázmány, S. Jenni-Eiermann, N. Strebel, T. Sattler, J. J. Negro, and Z. Tablado, "Unmanned aircraft systems as a new source of disturbance for wildlife: A systematic review," *PLOS ONE*, vol. 12, no. 6, p. e0178448, May 2017, doi: 10.1371/journal.pone.0178448.
16. R. Kays, M. C. Crofoot, W. Jetz, and M. Wikelski, "Terrestrial animal tracking as an eye on life and planet," *Science*, vol. 348, no. 6240, p. aaa2478, Jun. 2015, doi: 10.1126/science.aaa2478.
17. T. A. Marques *et al.*, "Estimating animal population density using passive acoustics," *Biological Reviews*, vol. 88, no. 2, pp. 287–309, 2013, doi: 10.1111/brv.12001.
18. R. Gibb, E. Browning, P. Glover-Kapfer, and K. E. Jones, "Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring," *Methods in Ecology and Evolution*, vol. 10, no. 2, pp. 169–185, 2019, doi: 10.1111/2041-210X.13101. [1]
19. H. J. Lynch, "Satellite Remote Sensing for Wildlife Research in the Polar Regions," *Marine Technology Society Journal*, vol. 57, no. 3, pp. 43–50, Sep. 2023, doi: 10.4031/MTSJ.57.3.1.
20. J. Wäldchen and P. Mäder, "Machine learning for image based species identification," *Methods in Ecology and Evolution*, vol. 9, no. 11, pp. 2216–2225, 2018, doi: 10.1111/2041-210X.13075.
21. B. G. Weinstein, "A computer vision for animal ecology," *Journal of Animal Ecology*, vol. 87, no. 3, pp. 533–545, 2018, doi: 10.1111/1365-2656.12780.
22. T. M. Aide, C. Corrada-Bravo, M. Campos-Cerqueira, C. Milan, G. Vega, and R. Alvarez, "Real-time bioacoustics monitoring and automated species identification," *PeerJ*, vol. 1, p. e103, Jul. 2013, doi: 10.7717/peerj.103.
23. T. Guilford *et al.*, "Migration and stopover in a small pelagic seabird, the Manx shearwater Puffinus puffinus: insights from machine learning," *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1660, pp. 1215–1223, Jan. 2009, doi: 10.1098/rspb.2008.1577.
24. "Welcome to the MegaDetector Python package — MegaDetector documentation." Accessed: Oct. 08, 2024. [Online]. Available: https://megadetector.readthedocs.io/en/latest/
25. A. Hernandez, Z. Miao, L. Vargas, R. Dodhia, and J. Lavista Ferres, *Pytorch-Wildlife: A Collaborative Deep Learning Framework for Conservation*. 2024. doi: 10.48550/arXiv.2405.12930.
26. "What is Mean Average Precision (mAP) in Object Detection?," Roboflow Blog. Accessed: Oct. 01, 2024. [Online]. Available: https://blog.roboflow.com/mean-average-precision/
27. "Announcing Roboflow Train 3.0," Roboflow Blog. Accessed: Oct. 01, 2024. [Online]. Available: https://blog.roboflow.com/roboflow-train-3-0/
28. S. Aharon et al., Deci-AI/super-gradients: 3.0.8. (Mar. 31, 2023). Zenodo. doi: 10.5281/ZENODO.7789328.
29. N. Rigoudy *et al.*, "The DeepFaune initiative: a collaborative effort towards the automatic identification of European fauna in camera trap images," *Eur J Wildl Res*, vol. 69, no. 6, p. 113, Oct. 2023, doi: 10.1007/s10344-023-01742-7.
30. S. Beery, D. Morris, and S. Yang, "Efficient Pipeline for Camera Trap Image Review," Jul. 15, 2019, arXiv: arXiv:1907.06772. Accessed: Oct. 01, 2024. [Online]. Available: http://arxiv.org/abs/1907.06772
31. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," May 09, 2016, *arXiv*: arXiv:1506.02640. Accessed: Oct. 1, 2024. [Online]. Available: http://arxiv.org/abs/1506.02640

32. Jocher, G., Qiu, J., & Chaurasia, A. (2023), “Ultralytics YOLO (Version 8.0.0)”. GitHub. https://github.com/ultralytics/ultralytics
33. P. van Lunteren, “EcoAssist: A no-code platform to train and deploy custom YOLOv5 object detection models,” Journal of Open Source Software, vol. 8, no. 88, p. 5581, Aug. 2023, doi: 10.21105/joss.05581.
34. A. Hernandez, Z. Miao, L. Vargas, R. Dodhia, P. Arbelaez, and J. M. L. Ferres, “Pytorch-Wildlife: A Collaborative Deep Learning Framework for Conservation,” Jul. 01, 2024, arXiv: arXiv:2405.12930. doi: 10.48550/arXiv.2405.12930. [1]
35. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 23, 2020, *arXiv*: arXiv:2004.10934. doi: 10.48550/arXiv.2004.10934.
36. C. Shorten and T. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” Journal of Big Data, vol. 6, Jul. 2019, doi: 10.1186/s40537-019-0197-0.
37. J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” Dec. 25, 2016, *arXiv*: arXiv:1612.08242. doi: 10.48550/arXiv.1612.08242.
38. J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement”.
39. C. Li et al., “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” Sep. 07, 2022, arXiv: arXiv:2209.02976. doi: 10.48550/arXiv.2209.02976.
40. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” Jul. 06, 2022, arXiv: arXiv:2207.02696. doi: 10.48550/arXiv.2207.02696.
41. G. Jocher, YOLOv5 by Ultralytics. (Feb. 2020). Python. doi: 10.5281/zenodo.3908559. [1]
42. C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information,” Feb. 29, 2024, arXiv: arXiv:2402.13616. Accessed: Oct. 01, 2024. [Online]. Available: http://arxiv.org/abs/2402.13616 [1]
43. A. Wang et al., “YOLOv10: Real-Time End-to-End Object Detection,” Oct., 2024, arXiv: arXiv:2405.14458. Accessed: Oct., 2024. [Online]. Available: http://arxiv.org/abs/2405.14458
44. Jocher, G., Qiu, J. (2024), “Ultralytics YOLO11” Accessed: Oct., 2024. [Online]. Available: https://docs.ultralytics.com/models/yolo11 [1]
45. J. Terven, D.-M. Cordova-Esparza, and J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” Machine Learning and Knowledge Extraction, vol. 5, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.

Appendix A

Deployment on NVIDIA Jetson Nano Developer 2GB Developer Kit

Tested on a new image provided, that did not exist in the dataset.



Ploting the prediction

```python
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt

image_filename = 'PICT5088.JPG'
prediction = {
    "predictions": [
        {
            "x": 2660.8,
            "y": 2222.4,
            "width": 1355,
            "height": 793,
            "class": "animal",
            "confidence": 0.847
        }
    ],
    "image": {
        "width": 1920,
        "height": 1080
    }
}
img = Image.open(image_filename)
draw = ImageDraw.Draw(img)
```

```
for pred in prediction['predictions']:
    x = pred['x']
    y = pred['y']
    width = pred['width']
    height = pred['height']

    # Calculate bounding box coordinates
    left = x - width / 2
    top = y - height / 2
    right = x + width / 2
    bottom = y + height / 2

    # Draw rectangle and label
    draw.rectangle([left, top, right, bottom], outline="red", width=3)
    draw.text((left, top - 10), f"{pred['class']} ({pred['confidence']:.2f})", fill="red")

# Show the image with the bounding box
plt.figure(figsize=(10, 8))
plt.imshow(img)
plt.axis('off')  # Hide axis
plt.show()
```

Appendix B

By scanning the following QR codes the user may test the models using their smartphone camera

Animal detection model:

10 class model:

20 class model