NOVA IMS Information Management School

MDSAA

Master Degree Program in
Data Science and Advanced Analytics

Wild animals detection in Camera Trapping images

A Machine Learning approach

Carolina Filipa Abreu Sá

Project Work

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

Wild animals detection in Camera Trapping images

A Machine Learning approach

by

Carolina Filipa Abreu Sá

Project Work presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science

Supervised by

Vitor Duarte dos Santos, PhD; Vitor Rodrigues

October, 2023

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisboa, 11/10/2023

ACKNOWLEDGEMENTS

First of all, I would like to say a big thank you to my Thesis supervisor, Professor Vitor Duarte dos Santos from NOVA IMS. Professor Vitor, you not only gave me a fascinating topic for my Master Thesis but also helped me every step of the way. Your dedication, patience, and the warmth you showed during our meetings made this experience truly special. I can't thank you enough for your guidance and mentorship.

I'm also grateful to Vitor Rodrigues and the INM team for this amazing opportunity.

My family has always stood by my side, guiding me through life and supporting me. To my Mom, Dad, and Grandmother, your love and motivation mean a lot to me.

A special thanks to Tiago for providing me with that extra push when I needed it the most and for being a constant presence in my life.

And to all my friends who played a part in shaping my academic journey, your contributions were essential, and I'm extremely grateful for your support.

Thank you all.

ABSTRACT

In the field of wildlife conservation and research, Camera Trapping (CT) images have become invaluable tools. However, the sheer volume of images, often captured in challenging conditions, presents a significant obstacle to accurately identifying animals. This Thesis proposes to develop an effective solution for identifying animals in CT images through the use of Machine Learning (ML) algorithms, within the context of the “Sistema de Armadilhagem Fotográfica e Análise Inteligente” (SAFArI) project. It addresses the unique challenges of processing images from the wild, by incorporating a comprehensive review of existing literature, with a specific emphasis on identifying the most suitable ML methodologies for analysing CT images. Based on this review, extensive exploration of various pre-existing object detection models, such as Faster Region-Based Convolutional Neural Network (Faster R-CNN) and You Only Look Once (YOLO), is conducted, considering the unique features of each model, and incorporating movement detection methods like Background Subtraction (BS). Benchmarking becomes essential as this study seeks to evaluate the performance of each model, providing valuable insights into their efficacy. It is within these benchmarks that a path toward a custom-tailored architecture for the SAFArI project emerges. It is vital that the developed model not only demonstrates effectiveness but also integrates seamlessly with the project's objectives, contributing to advancements in wildlife research and conservation.

KEYWORDS

Camera-trapping; Image Processing; Deep Convolutional Neural Networks; Machine learning

Sustainable Development Goals (SDG):

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
BS	Background Subtraction
CRISP-DM	Cross-Industry Standard Process for Data Mining
CT	Camera Trapping
DCNNs	Deep Convolutional Neural Networks
DL	Deep Learning
e.g.	exempli gratia
Faster R-CNN	Faster Region-Based Convolutional Neural Network
FPN	Feature Pyramid Networks
mAP50	Mean Average Precision at IoU 0.50
ML	Machine Learning
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RoI	Region of Interest
RPN	Region Proposal Network
RQ	Research Question(s)
SAFArI	Sistema de Armadilhagem Fotográfica e Análise Inteligente
SLR	Systematic Literature Review
YOLO	You Only Look Once

1 INTRODUCTION

1.1 BACKGROUND AND PROBLEM IDENTIFICATION

With the continuous development of science and technology, the application of image processing and recognition technologies has been evolving increasingly. The large volume of information collected and stored associated with Machine Learning (ML) models has given a strong incentive to the identification and analysis of images.

Nowadays several ML models are used to improve the image classification in a large variety of industries, in a way that the machine itself classifies the image in one of the created targeted classes (Naila Batool & Gilanie, 2022).

In biology, the collection of images in the field is a common practice, for which, Camera Trapping (CT) is one of the most powerful and used techniques for prospecting and identifying the presence of wildlife in their habitat. Motion-activated cameras, commonly known as Camera Trapping, take photos or videos of passing animals usually detected by motion or temperature sensors (Tabak et al., 2018). CT allows to study and understand animals' behaviours and patterns without human interference and ensuring the animal captured was not disturbed, all this in a cost-effective way (Burton et al., 2015; Kucera & Barrett, 2011).

To correctly classify CT images, it is crucial to focus on the placement and specifications of the camera first, which implies the thoughtful choice of detection zone, trigger speed, flash type and intensity, sensitivity, resolution, and robustness of the camera. Also, all the environmental factors need to be considered, such as target species and their size, climate, and the type of habitat (Rovero et al., 2013).

Moreover, the quality of the images, particularly with regards to their clarity, sharpness, and resolution, plays a pivotal role in ensuring the accurate classification of images by ML models. In simpler terms, it is essential that moving objects within the images are not blurred to ensure correct classification (Rovero, Zimmermann, Berzi & Meek, 2013).

This method can produce millions of images, in around 3 years, 225 camera traps operating continually can collect over 1.2 million sets of pictures, each set can contain 1 to 3 photos taken sequentially (Swanson et al., 2015), and every photo needs to be analysed and correctly classified. Considering the volume of images produced, doing this process manually would be understandably overwhelming, so the application of Artificial Intelligence (AI) algorithms to automate it is essential nowadays (Tabak et al., 2018).

The study that follows is referring to one of the main outcomes expected from a project called SAFArI- Sistema de Armadilhagem Fotográfica e Análise Inteligente (Photographic Trapping and Intelligent Analysis System) which will use Image Processing technologies and Deep Learning (DL) algorithms to remove image noise and categorise images into positives (image

with animals) and negatives (image without animals), while taking into account the CT challenges mentioned above.

1.2 OBJECTIVE

The goal of this study is to develop an effective solution for identifying animals in a very large quantity of photos, regardless of the image quality, within the context of the SAFArI project.

In order to achieve this goal, the following intermediate objectives were defined:

- Conduct a comprehensive review of existing literature to understand the more suitable DL methodologies to use in this specific case, having special attention to the challenges associated with analysing these images.
- Explore the different object detection models previously researched with the dataset provided, to identify the best ones for this work, always checking with the possible different features of each one and various pre-processing alternatives.
- Compare and evaluate each model's performance against benchmarks from previous research.
- Design a customised architecture tailored to the specific requirements of this project, incorporating insights gained from the model's exploration and evaluation.
- Ensure that the developed solution is not only effective but also seamlessly integrable into the SAFArI project.
- Execute tests on the model that demonstrates the highest suitability for this particular case.

1.3 EXPECTED RESULTS AND CONTRIBUTIONS

To conduct ecological and biological experiments one of the main parts is the data, besides with the volume of data that is produced these studies can become time consuming and require a lot of manual labour, which can be extremely costly financially (Norouzzadeh et al., 2018). In order to manage these resources in a more efficient manner, a DL model is necessary, through that, it is possible to store, organise and analyse this data automatically or semi-automatically. This not only guarantees a faster approach compared to traditional methods previously used, but also reduces the burdens and errors associated with analysing millions of images manually (Santangeli et al., 2022).

Through this work the biology community will have cleaner images results to conduct their own research, only receiving the images that contain valuable information for their work, in other words, they will not receive blurred images or those without an animal. Another

improvement certainly is the fact that the creation of this model can be easily adapted to other similar CT images studies.

Finally, this research can contribute to deepening the knowledge of environmental and CT image classification, as a result of the unlimited potential of Data Analytics application in the biology area, making even more evident the most effective image treatment strategies for CT images, besides encouraging the industry to invest in more automatic approaches for their future works. At the end is expected to have a model capable of detecting an animal in a photographic capture, even when only a part of the animal is visible, and independent of image quality and weather conditions.

2 LITERATURE REVIEW

Having in consideration that a literature review is one of the most important parts of any academic study, since it provides a solid support to the research, while guaranteeing the reliability of the technical and scientific quality of the work (Levy & J. Ellis, 2006). In this chapter, an extensive assessment and respective synthesis of the scientific literature was conducted, to better understand the approaches already used in the area and to summarise the main ideas and problems observed.

Although a traditional literature review can be very efficient in presenting scientific evidence, the implementation of a systematic literature review (SLR) delivers higher quality results and lower risk of bias (Tranfield et al., 2003), hence the choice to use this type of review here.

Furthermore, it would be quite challenging to draw any conclusions about this topic without this kind of method, given the enormous number of ML techniques that are now in use, and the volume of articles that are being generated in this field (Page et al., 2021).

2.1 SYSTEMATIC LITERATURE REVIEW

SLR is an organised and methodical approach to research that is used to identify, evaluate, and analyse data from a variety of sources. One of the most popular approaches to SLR is Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA), that is usually used to ensure that the review is conducted in a systematic, organised, and transparent manner (Moher et al., 2009; Page et al., 2021).

PRISMA consists of four steps that must be followed in order to conduct a successful SLR, which are:

1. **Identification -** Identify the Research Questions (RQ) that will guide the review, and these must be clear, concise statements that outline the purpose of the review. Once the RQ has been identified, it is also crucial to identify the sources of data that will be used to answer the questions.
2. **Screening -** Create a search strategy that will be used to locate relevant studies, which should include the keywords that will be used to search for studies. With that in mind, the search is conducted and the studies that are relevant to the RQ are identified.
3. **Eligibility -** Evaluate the studies and determine which ones should be included in the review, through a predetermined set of criteria.
4. **Included -** Report the results of the review.

By following the steps outlined in PRISMA, it was possible to organise an approach to the review process and ensure that it was conducted in a transparent and unbiased manner. The

outcomes that were attained and the factors that led to them are provided in the sections that follow.

2.1.1 PRISMA - Identification

The use of ML in CT images is a rapidly growing field of study. With the appearance of powerful computing systems and the increasing availability of data, researchers are exploring the potential of ML to improve the accuracy and efficiency in classifying CT images (Norouzzadeh et al., 2018). This review will explore the state of the art regarding the utilisation of ML in CT images, and seek to answer the following RQ:

RQ1 - What is the current state of research in this area?

RQ2 - What ML techniques are currently being used to solve this problem?

RQ3 - What are the advantages and disadvantages of using ML in this context?

By exploring these questions, this research will provide a comprehensive overview of the current state of the art regarding the utilisation of ML in CT images. In order to answer these RQ, a search on the following scientific information databases was conducted in December 2022:

Table 1- Systematic Review's Resource Databases and URLs

DataBases	DataBases URL
Scopus	https://www.scopus.com/search/form.uri?display=basic#basic
ScienceDirect	https://www.sciencedirect.com/
Web of Science	https://www.webofscience.com/wos/woscc/basic-search

Using three separate databases for research is important to ensure that all relevant data is included in a study. These databases were chosen for their well-established and multi-disciplinary research platforms, as well as their ability to stay up to date with the latest research.

2.1.2 PRISMA - Screening

The development of Section 1.1 has been instrumental in helping to better understand the topic being study, which, as a consequence, allows comprehension of the words that assuredly represent this topic. These keywords are divided by type in the table below.

Table 2- Systematic Review's Keywords

Biology Terms	AI Terms	RQ Terms
Camera trap*	Machine learning	Current status
Photo-trapping		
Animal movement	Artificial intelligence	
Animal detection		
Remote cameras	Deep learning	Issues
Wildlife cameras		
Wildlife survey methodology	Computer image processing and recognition technology	Problems
Conservation technology		
Wildlife monitoring	Neural Networks	Models
Wildlife ecology		

The keywords chosen are listed in the table above, separated by the area of study (AI), the application area (Biology), and finally the terms that are most helpful in answering the RQs.

Many of these words are synonyms, as one can easily confirm, which helps to understand the various ways this matter can be approached. By extracting the most commonly used words when discussing this topic, it is possible to gain a greater insight to discuss it in a more effective manner.

Through these keywords was developed the search string that follows:

```
("Camera trap*" OR "Photo-trapping" OR "Animal movement" OR "Animal
detection" OR "Remote cameras" OR "Wildlife cameras" OR "Wildlife
survey methodology" OR "Conservation technology" OR "Wildlife
monitoring" OR "Wildlife ecology") AND
("Machine learning" OR "Artificial intelligence" OR "Deep learning" OR
"Computer image processing and recognition technology" OR "Neural
Networks") AND
("Current status" OR "Issues" OR "Problems" OR "Techniques" OR
"Models").
```

2.1.3 PRISMA - Eligibility

In order to assess the quality of the articles returned by the search string, a set of criteria containing elements to be determinant and irrelevant to the research was constructed and is revealed below.

Table 3- Systematic Review's Inclusion and Exclusion criteria

Inclusion Criteria	Exclusion Criteria
Articles in English	Duplicated articles
	Articles not in English
Articles between 2018 and 2023	Articles before 2018
	Articles more focus in the Biology than AI (e.g.: study behaviour, geographic movement, etc)
Articles with use of AI models to process and classify CT images	Articles in the scope of this work but employ additional methods to arrive at the same conclusions (e.g.: temperature sensors, distance measures, etc)
	Articles in the same scope but using audio or text instead of images.
	Articles in the same scope but that uses AI for different objectives (e.g.: counting animals, detecting their behaviours, etc)

Due to the exponential expansion of AI over the past several years and the rapid advancements in it, it was determined that only articles published after 2018 were suitable for this task, and their scope had to be extremely close to the one being developed. Additionally, articles that were inaccessible or those that were written in languages other than English weren't appropriate for the review, as were publications whose purpose differed from the one being created.

2.1.4 PRISMA - Included

The research process was extensive and thorough, with a total of 1047 results found, after the creation of the search string and its insertion in the selected databases. Early on in the screening phase, the duplicates were removed, leaving 1012 of the 1047 initially found. Subsequently, through database filters, all articles published before 2018 or that weren't in English were eliminated, maintaining a total of 768 results. After reading all the abstracts, it was determined to maintain 213 articles based on the remaining inclusion and exclusion criteria. Finally, 19 of those articles were chosen after reading them in their entirety.

Below, is possible to find a workflow of all the phases of PRISMA and the results each of them produced.

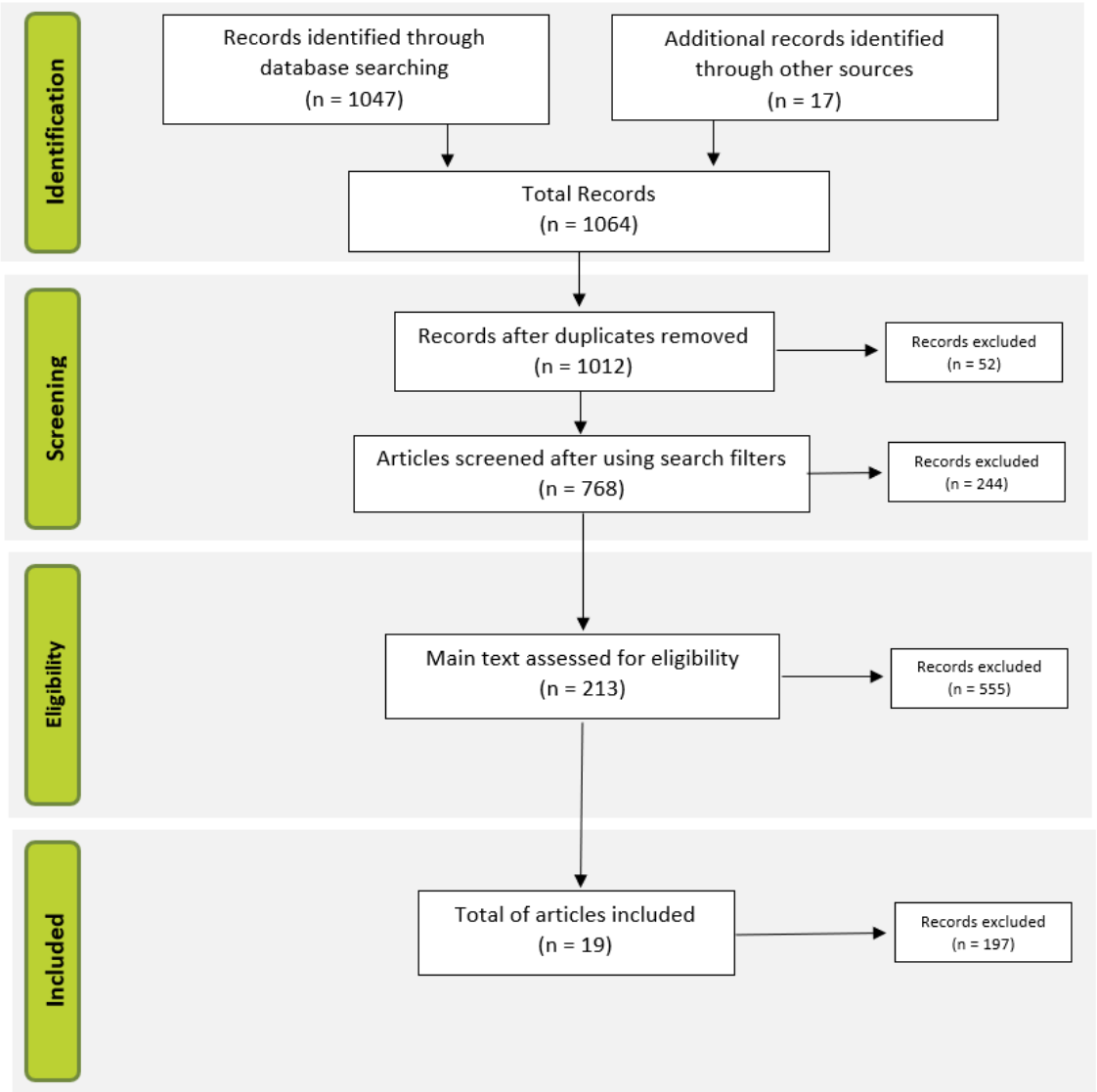


Figure 1- PRISMA execution

As mentioned, the output of this research was 19 articles, with their findings being presented in the accompanying table, which provides a comprehensive overview of the articles.

Table 4- PRISMA included articles summary.

#	Author	Year	Article Name	Main Findings
1	Ahmed et al.	2019	Semantic region of interest and species classification in the deep neural network feature domain	Develops an animal detection and species classification method for camera-trap images with highly cluttered scenes. Uses the Deep Neural Network trained for animal-background classification to analyse the input image and construct a semantic region representation using k-mean clustering and graph cut in the Deep Neural Network domain.
2	Banerjee et al.	2022	Sieving Camera Trap Sequences in the Wild	Experiments with three different approaches to eliminate empty images and detect animals, with and without a sequence analysis of each frame. These approaches are ViT, Faster R-CNN and DETR.
3	Conway et al.	2021	Frame-by-frame annotation of video recordings using deep neural networks	Provides useful suggestions of possible solutions and factors to be aware of in this type of study, while experimenting with different approaches to this problem: classifying each image independently and classifying a sequence of images at a time.
4	Cunha et al.	2021	Filtering Empty Camera Trap Images in Embedded Systems	Analyses and compares the performance of models developed specifically for low computational power devices, in CT images with the objective of detecting an animal.
5	Dhillon & Verma	2019	Convolutional neural network: a review of models, methodologies and applications to object detection	Does a complete review of DL problems and shows how it contributes to society.

6	Islam & Valles	2020	Identification of Wild Species in Texas from Camera-trap Images using Deep Neural Network for Conservation Monitoring	Identifies very well the problems in CT images, while giving a good description on how to act when facing this type of problem.
7	Kellenberger et al	2019	Best Practices to Train Deep Models on Imbalanced Datasets - A Case Study on Animal Detection in Aerial Imagery	Addresses recommendations on how to train imbalanced datasets, like the one that is being studied, and demonstrates it on a similar dataset.
8	Kellenberger et al.	2020	AIDE: Accelerating image-based ecological surveys with interactive machine learning	Exemplifies a good solution for an application, called AIDE, similar to the one that this model is being integrated in.
9	Lee et al.	2022	Improved Monitoring of Wildlife Invasion through Data Augmentation by Extract-Append of a Segmented Entity	Proposes a new augmentation technique that can be useful to treat imbalanced datasets.
10	Meena & Agilandeeswari	2019	Stacked Convolutional Autoencoder for Detecting Animal Images in Cluttered Scenes with a Novel Feature Extraction Framework	Introduces stacked convolutional autoencoders (SCAE).
11	Norouzzadeh et al.	2020	A deep active learning system for species identification and counting in camera trap images	Suggests possible solutions to the more challenging problems of extracting information from Camera Trap images.
12	Riechmann et al.	2022	Motion vectors and deep neural networks for video camera traps	Uses a promising filtering of the images, applying both movement detection and object detection.

13	Ukwuoma et al.	2022	Object detection from dynamic scene using joint background modelling and fast deep learning classification	Proposes effective dynamic background modelling with ML to develop an algorithm for human-animal detection.
14	ULAŞ TEKELİ & YALIN BAŞTANLAR	2019	Elimination of useless images from raw camera-trap data	Focuses on solving the problems and challenges from realistic scenarios of CT images. Proposes an approach combining Convolutional Neural Networks and Background Subtraction (BS) methods together.
15	Wei et al.	2020	Zilong: A tool to identify empty images in camera-trap data	Creates the Zillong software, a non-ML approach to identify empty images among Camera-Trap ones.
16	Xi et al.	2021	Image Filtering and Labelling Assistant (IFLA): Expediting the analysis of data obtained from camera traps	Describes Image Filtering and Labelling Assistant (IFLA), a program they developed to assist biologists and ecologists in identifying animals amongst all images, without the usage of ML algorithms.
17	Yang et al.	2021	A systematic study of the class imbalance problem: Automatically identifying empty camera trap images using convolutional neural networks	Explores the imbalance problem of Deep Convolutional Neural Network (DCNN) model, doing experiments around this topic, and giving the best suggestions to treat the type of problem based on what is more adequate for other problems.
18	Yang et al.	2021	An automatic method for removing empty camera trap images using ensemble learning	Explores an ensemble learning approach with DCNN models testing with a small dataset of balanced and imbalanced training sets.
19	Yang et al.	2021	An Adaptive Automatic Approach to Filtering Empty Images from Camera	Proposes an adaptive incremental training method with a DCNN model (AlexNet model) in a small-scale dataset to study the possibility to train this type of model in a common PC, instead of a complex computational platform.

Following the PRISMA methodology, it was then necessary to analyse the findings of this research, so it was imperative to examine each of the included articles in order to determine the primary contribution of each work and determine the answers to the RQ. The sections that follow present the conclusions, obtained from the reviewed articles.

2.2 CURRENT STATE OF RESEARCH IN THIS AREA

The introduction of digital cameras has revolutionised the way humans capture and store data. Thanks to the increased technical capabilities of digital cameras, large amounts of data can be easily stored (Xi et al., 2021). Consequently, Camera Traps have become a popular non-invasive technique for studying wildlife, particularly for monitoring terrestrial vertebrates (Wei et al., 2020).

In recent decades, the use of this method has increased dramatically, as it provides a wealth of data to researchers. However, the total amount of images that Camera Traps produce can be challenging and demanding for research teams to manually view and classify (Wei et al., 2020).

Most of the captured images are triggered by fluctuations of light or vegetation moving with the wind, while including no animals. In addition, some small animals are difficult to recognize, and others move too fast to be captured. These problems force the quantity of images to be much larger than needed which, consequently, makes the processing of images laborious and error prone (Xi et al., 2021). So, eliminating empty images becomes an essential step for managing large Camera Trap datasets and obtaining accurate results (Norouzzadeh et al., 2020).

Overall, Camera Traps have become a powerful tool for wildlife studies, allowing researchers to collect large amounts of data quickly and efficiently. However, the sheer number of images that Camera Traps produce can be challenging and demanding for research teams to manually view and classify. To address these issues, investigators have developed automated solutions to help in the classification of images. By applying ML algorithms to the datasets, researchers can identify and classify animals in the images with a high degree of accuracy. This allows researchers to process large datasets quickly and efficiently, as well as identifying animals that may be difficult to detect with the naked eye (Norouzzadeh et al., 2020).

2.3 MACHINE LEARNING TECHNIQUES CURRENTLY BEING USED TO SOLVE THIS PROBLEM

One may infer from the literature review that there is a lot of research being done on the use of AI techniques in the field of CT. Literature also implies that not all AI techniques are being deployed and that some technologies and techniques are more widely used.

In recent years, DCNNs have been one of the most used models to detect and classify animal species in images and videos (Yang et al., 2021). The most often used DCNNs are OverFeat, AlexNet, Region Proposal Network, You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), Residual Neural Network (ResNet), and Faster Region-Based Convolutional Neural Network (Faster R-CNN). Because it has been successful on numerous datasets and has abundant documentation and source codes, Faster R-CNN is the most well-liked of them (ULAŞ TEKELİ & YALIN BAŞTANLAR, 2019; Yang et al, 2021). To illustrate this point, it is considered the work of Banerjee et al. (2019), who employed the Faster R-CNN model in their research efforts. Similarly, ULAŞ TEKELİ & YALIN BAŞTANLAR (2019) also adopted the Faster R-CNN model and achieved an average accuracy of 90.2% for distinguishing and retaining the correct images.

In 2020, Kellenberger et al. predicted that the accuracy of the ResNet would increase significantly as more layers may be added during training. At the time, ResNet was one of the most popular architectures for image classification, including in ecological applications. Another potential solution suggested was RetinaNet, which is an evolution of Faster R-CNN, with a sequence of layers called 'Feature Pyramid Network' (FPN) and 'Focal Loss', that enables very accurate object detection by acquiring both high-resolution and semantically meaningful characteristics for each point in the image, while lowering the penalty for correct predictions with confidence that isn't perfect but is still high enough to make the model more resistant to datasets with significant class imbalances (Kellenberger et al., 2020).

However, a common problem with DCNNs models is the class imbalance of the training set, which affects the model's performance. To overcome this, Yang et al. (2021) proposed an ensemble learning approach, and while prior research had not previously demonstrated the method's efficacy in improving accuracy within this specific problem, this study provided evidence of its value, especially in the context of small labelled datasets.

The ensemble learning approach consists of combining different DCNNs models, which in the Yang et al. (2021) case study, combined AlexNet, Inception and ResNet models all trained in balanced and unbalanced sets. The choice of these 3 models was because there are two core factors in defining a good ensemble classification system: the accuracy of individual classifiers and the diversity among classifiers.

Also, to train on an imbalanced dataset, Kellenberger et al. (2019) proposed some recommendations: Curriculum Learning, Hard Negative Mining, Border Class, and Class Weighting. Curriculum learning means to sample the training images so that they always contain at least one animal. Hard Negative Mining is to amplify the weights of the four most confidently predicted false alarms in every training image. Border Class labels the 8-neighbourhood around true animal locations with a third class to make the Convolutional Neural Network learn to treat the surroundings of the animals separately. Finally, Class Weighting balances the gradients during training with constant weight corresponding to the inverse class frequencies observed in the training set.

To overcome the difficulty of only extracting the images that contain animals from CT images through AI, it is possible to use two different approaches: the Movement Detection, and the Object Detection, which in this specific case can be called Animal Detection, and is the approach discussed so far.

Since most of these images are taken in sequences of an average of three photos per sequence spaced at close intervals, it is possible to detect movement by comparing the photos that make up the sequence (Ahmed et al., 2019). This is called Movement Detection, which is often used as it is faster and more suited to real-time applications.

Due to this, it is possible to apply this approach on the CT images, which can speed up the process of spotting an animal so long as it maintains movement from one subsequent image to the next. One of the most prominent methods of movement detection is Background Subtraction (BS) method, since in these sequences the background is always the same (ULAŞ TEKELİ & YALIN BAŞTANLAR, 2019). However, naturally occurring small differences between frames can result in noise even in the absence of movement, which can wrongly classify images.

For Animal Detection, usually a DCNN is used, on which Riechmann et al. (2022) proposed YOLOv4 and Faster R-CNN, preferring to use YOLOv4 since it is a single-stage object detector with relatively low computational requirements and competitive performance when compared to the state of the art. Banerjee et al. (2022) experiment with models like Faster R-CNN, and less used models in this context like DEtection TRansformer (DETR) and Vision Transformer (ViT), from where ends up proposing DETR, a relatively recent paradigm of detectors which reason about the relations of the animals with the global image content.

In conclusion, in all of the literature it's seen that is traversal to most of the challenges of identifying empty frames from CT images, that the most used models are Faster R-CNN and YOLO, always emphasizing the use of a movement detection method to complementarity provide more accurate results. It's imperative to note that YOLO in particular has received considerable attention for its relative novelty and low computational requirements, presenting an enticing prospect for this project, alongside Faster R-CNN. Nevertheless, the ultimate selection of a model depends on the specific use-case, and the variety of available models provides a good starting point.

2.4 Advantages and disadvantages of using Machine Learning in this context

The use of ML in the detection of empty images and animals in CT images has both advantages and disadvantages. On the one hand, ML can be used to improve the efficiency and accuracy of animal detection in Camera Trap sequences by automatically identifying the presence of animals. This can reduce the time and cost needed for manual review and annotation, thereby allowing for more comprehensive large-scale environmental studies. Additionally, ML can also be used to detect animals in challenging scenarios, including situations where animals are

camouflaged, motion blur occurs, obstacles obstruct the view, lighting is poor or fluctuating, or when the animal is too close or too far away (Banerjee et al., 2022).

On the other hand, there are some drawbacks to using ML in empty frame removal and animal detection. One of the primary challenges is the large amount of data required to train the models, that must include images of animals in various poses, lighting conditions, and camera angles, which can be difficult to obtain (Kellenberger et al., 2020). Additionally, to achieve better and more accurate results, the data must be labelled in order for the models to be trained, which can be a time-consuming and costly process (Islam & Valles, 2020).

The dataset can be expanded by using data augmentation, which is a technique that allows to increase the number of images in the dataset through the blend of backgrounds and objects from the existing images (Lee et al., 2022). This can also be used to balance the number of empty and animal images in the dataset. However, Yang et al. (2021) noted that it was not possible to train these models in large datasets without the use of powerful computational resources.

Another potential disadvantage of using ML for empty frame detection is that the models may be prone to overfitting. If the models are not trained on a large enough dataset, they may be unable to generalise to new images and produce inaccurate results. Additionally, the models may be unable to identify animals in difficult or low contrast images, leading to false negatives (Lee et al., 2022).

Given the difficulty of producing labels to train detectors, Cunha et al. (2021) also stated that classifiers can achieve good performance on empty picture detection. However, the performance of these classifiers depends on the number of images available for training, as well as on specific factors that can vary from dataset to dataset.

Another important thing to take in account was Conway et al. (2021) that noted the importance of randomly allocating contiguous sequences of frames to training, validation, and test datasets, rather than randomly allocating the frames themselves. This causes sequences to be broken up, losing potentially important information, and also results in extremely similar images appearing in both the training and test sets.

Overall, the use of ML in animal detection can be a powerful tool for extracting valuable information from Camera Trap sequences quickly and accurately. However, there are some drawbacks to consider, such as the need for large amounts of labelled data and the overfitting that the models may be prone to. Therefore, it is crucial to weigh the advantages and disadvantages of using ML in this context and explore possible solutions to these problems.

3 METHODOLOGY

As mentioned, the aim of this study is to detect animals in CT images as accurately as possible. This is accomplished through the use of Data Science models, and as with any good Data Science project, it is important to follow well-defined steps to do so. As a result, a Cross-Industry Standard Process for Data Mining (CRISP-DM) was used, and it is explained below.

3.1 CRISP-DM

CRISP-DM is a process model that provides the steps needed in a Data Science project. This comprehensive approach provides a structured framework for understanding the data, creating models, and evaluating results, and has been used in real world scenarios to organise projects in a logical, straightforward way (Shearer et al., 2000).

The CRISP-DM process consists of six steps that form a cycle, as shown and explained below.

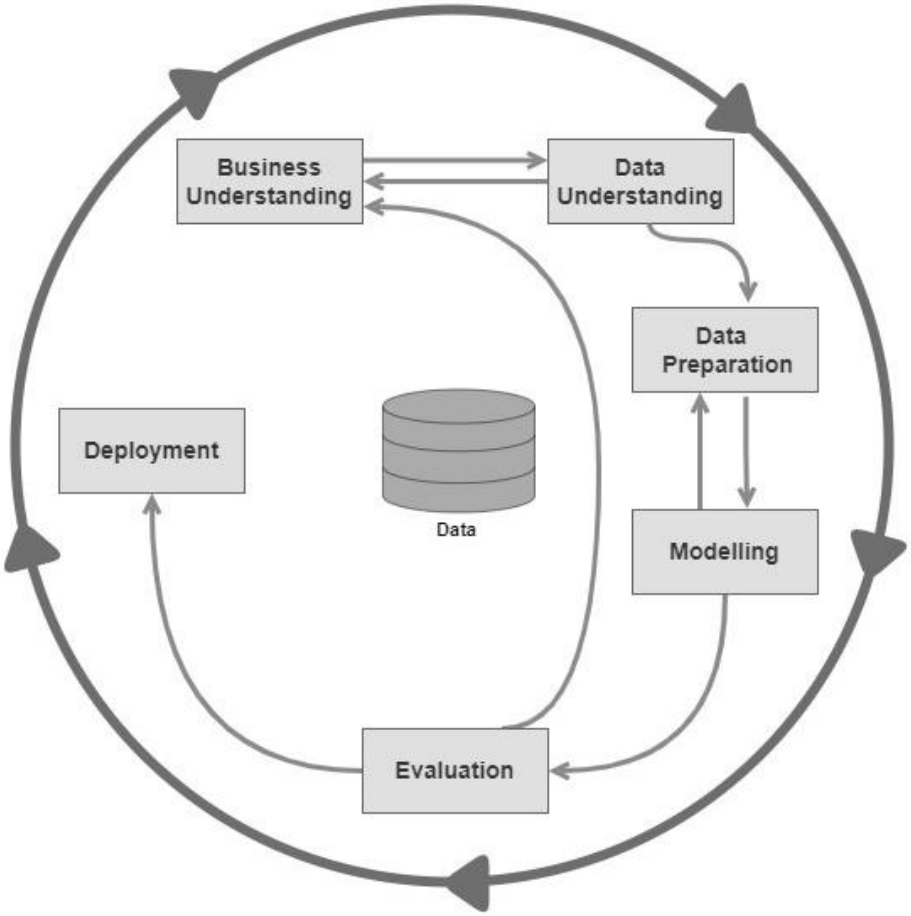


Figure 2- CRISP–DM Methodology

1. **Business Understanding-** it is the first step of the CRISP-DM process and includes understanding the project's goals and objectives, as well as the data available and the

requirements for the project. In order to guarantee that the project is properly scoped, and that the data is collected in the most effective and efficient manner, this stage is crucial.

2. **Data Understanding-** requires the collection and exploration of the data to gain an understanding of the data, including the type and structure of the data, and any potential issues with it. This step is important for evaluating the quality of the data and for understanding the relationships between the data points.
3. **Data Preparation-** involves cleaning and transforming the data to make it suitable for analysis. This includes removing any outliers or inconsistencies in the data, as well as transforming the data into the appropriate format for the analysis.
4. **Modelling-** involves selecting and applying modelling techniques to the data, to then test it in a previously generated test design.
5. **Evaluation-** implies evaluating the results of the model to determine if it is accurate and effective. This step is crucial to confirm that the model is operating as predicted and producing reliable results. At the conclusion of this process, if the model produces positive results, a business choice is taken regarding whether to adopt the model or not.
6. **Deployment-** it is the final step and includes deploying the model to be used in production. This step is necessary to guarantee that the models are available and usable in production.

3.2 RESEARCH STRATEGY

Given that the CRISP-DM is more commonly used in industry and that its steps are tailored to such environments, it was modified to become more Thesis-appropriate, allowing the incorporation of its essential steps to this research context.

In order to reach its purpose, which is to more precisely identify animals in CT images, this project used a systematic approach divided into three main phases: Exploration; Design and Development; and Conclusion. The diagram that follows illustrates how each phase is broken into several iteration steps that incorporate every CRISP-DM step.

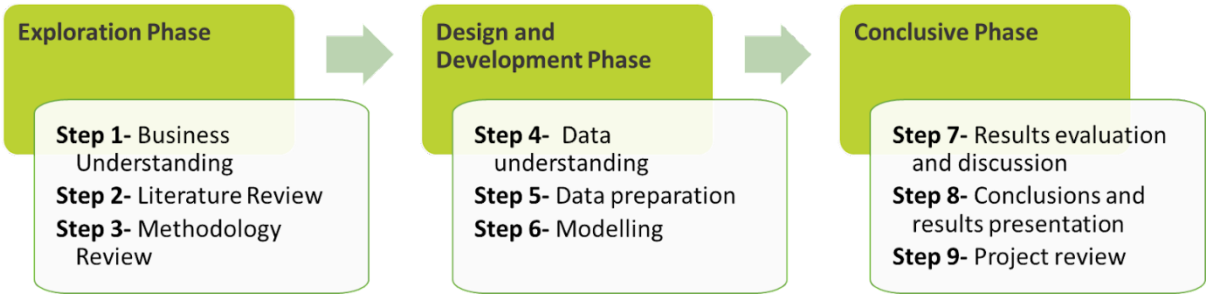


Figure 3- Methodology proposal

The Exploration Phase involves assessing the situation and determining the business objectives, which, for this particular task, are to detect animals in CT images. To gain a better understanding of the problem, three RQs are formulated in the Literature Review step: What is the current state of research in this area? What ML techniques are currently being used to solve this problem? And what are the advantages and disadvantages of using ML in this context? Answering these questions through the use of the SLR method, PRISMA, is essential in order to identify the best approach to take when trying to solve this problem.

The Exploration Phase is followed by the Design and Development Phase. During this phase, data is collected, analysed and understood in order to prepare it for the next step. Next, is data preparation which includes ensuring that there are an adequate number of labelled images to facilitate effective model training and confirming that all annotations are in the correct format. Additionally, it is essential to certify that all images have the same format and are resized according to the model's proposed specifications.

Beyond this, the data preparation stage also involves securing that all images are grouped by camera and then by bursts, in order to guarantee that all groups of photos have the same background to facilitate the experiment of implementing a movement detection model. Finally, the data must be separated into train, test, and validation sets.

Once the data is prepared, the next step is to apply a modelling technique taking into account the business objectives identified in the Exploration Phase. In this case, two approaches were explored: First applying an Object Detection model, such as Faster R-CNN and YOLO, independently to detect animals in the CT images. The performance of the Object Detection model was evaluated based on its ability to accurately identify animals. Second, a combination of a Movement Detection model as BS, and an Object Detection model, such as Faster R-CNN and YOLO, was applied. It is important to compare the results of the Object Detection models in order to understand which one performs the best, and consequently will be employed in the outcome. Furthermore, to avoid false negatives, an image must receive votes from both Object and Movement Detection techniques in order to be eliminated.

To evaluate the performances of the models and approaches used and determine the most effective one, several evaluation metrics were employed, offering distinct perspectives on the Object Detection models' performance.

The first metric used was the Mean Average Precision at IoU 0.50 (mAP50), which is a widely adopted measure for evaluating Object Detection performance in images and measures the average precision of the models' predictions at an IoU threshold of 0.50. It considers both precision and recall, providing an overall assessment of the models' ability to accurately detect objects. Higher mAP50 values indicate superior performance in object detection.

The second metric employed was Accuracy, a commonly used evaluation measure that measures the overall correctness of the models' predictions. Accuracy calculates the ratio of correctly predicted instances (true positives and true negatives) to the total number of instances. A higher Accuracy score indicates a greater proportion of correct predictions made by the models.

The third metric, Omission Error, also known as the false negative rate, quantifies the rate at which the models fail to detect true positive instances. This evaluation metric has been widely employed in similar research contexts, such as in the study conducted by Yang et al. (2021), showcasing the proportion of true positive instances that were missed by the models. Lower Omission Error values reflect improved performance with fewer missed detections.

Additionally, the time per epochs to train each model was also taken into account to assess the efficiency of the models.

The Conclusive Phase represents the final stage of the process, involving an extensive discussion and evaluation of the proposed framework based on these evaluation metrics to identify the best approach and establish implementation guidelines. It also includes obtaining the conclusions of the project as well as future recommendations. Finally, the entire project was reviewed.

3.3 TOOLS

This section describes the tools used throughout the project, categorised according to the different phases of the methodology followed. It is important to note that the entire project was developed using the Python programming language.

3.3.1 Exploration Phase

During the Exploration Phase, the primary goal was to define the problem and establish the project's objectives. To manage and organise the research articles reviewed during this initial phase of the project, Mendeley served as a valuable tool.

3.3.2 Design and Development Phase

During this phase, data was acquired, explored, cleaned, and transformed before being trained and evaluated with ML models.

To familiarise with the CT dataset and understand the problem at hand, Microsoft Excel and Visual Studio were used. Microsoft Excel allowed for organisation and analysis of the dataset. While Visual Studio was used to write and organise the initial code, specially to split data, and correctly associate the labels to the images. Also, some images need their labels to be corrected, for which the https://www.makesense.ai/ platform was used to annotate the CT images again.

Additionally, the free version of Google Colab was employed for this study, making use of Google Drive to store data. This approach facilitated access to cloud-based GPUs, allowing for experimentation with various ML models. The tracking and logging of experiments and model performance were managed through MLFlow.

3.3.3 Conclusive Phase

During the Conclusive Phase, the model performance was evaluated and determined if it met the objectives. Google Colab was used to generate performance metrics and generate reports. MLFlow helped to visualise the performance of the models over time and identify any potential issues. Finally, Microsoft Excel was employed to analyse and summarise the results.

4 CLASSIFICATION OF CAMERA TRAPPING IMAGES WITH MACHINE LEARNING MODELS

This chapter aims to provide a comprehensive overview of the object detection process designed for identifying wild animals in a real-world database. The primary goals are to establish a replicable approach that can be applied to this specific task, and evaluate the effectiveness of two widely recognized algorithms, YOLO and Faster R-CNN, as recommended in the Literature Review.

To achieve these goals, the chapter is structured as follows. The first section, Project Context, provides an introduction to the overall project, highlighting the architecture of the project, its anticipated outcomes and the module of the project that is being developed in this Thesis. The next section, Data Understanding, explores the dataset, examining image characteristics, class distribution and potential challenges. Following that, the Data Preparation section outlines the steps taken to pre-process the dataset to ensure optimal performance during model training. Finally, the Modelling section presents the implementation details of YOLO and Faster R-CNN, both with and without BS, explaining its architectures, training procedures and results obtained.

4.1 PROJECT CONTEXT

As stated in Section 1.1, the focus of this Thesis is centred on the development of one of the anticipated goals of the SAFArI project - Sistema de Armadilhagem Fotográfica e Análise Inteligente (Photographic Trapping and Intelligent Analysis System), which is developed by four entities: BIOTA - Estudos e Divulgação em Ambiente, Lda.; INM - Innovation Makers; FCUL - Faculdade de Ciências da Universidade de Lisboa; and GRUPO LOBO - Associação para a Conservação do Lobo e do seu Ecossistema.

This project intends to create a support system for the work of ecologists in the form of an application that will allow the collection of images of wild animals in different environments and geographies, which can be used for various purposes, such as biodiversity conservation projects, environmental impact assessment, ecosystem stability monitoring, among others. Image Processing and DL technologies will be used for this purpose, allowing the identification of an animal present in the images, its species, the counting of individuals by species, the characterization of their behaviour, and the detection of behavioural changes.

This type of output will speed up the work of ecologists studying biodiversity, providing a way to reduce the time that is typically spent observing, sorting, and analysing photographic images of animals, which could instead be allocated to other tasks such as the elaboration of study reports either for impact assessment purposes or for scientific study purposes.

For the entirety of this project, the following outcomes are anticipated:

1. Screening of images automatically with an efficiency greater than 90%.
2. Automatically identifying species, allowing the identification of roughly 75 species of vertebrates present in the Portuguese natural ecosystem with a success rate of more than 90%.
3. Automatically counting species with a success rate of more than 90%.
4. Behaviour analysis with an efficiency of more than 85%, which includes the detection of behavioural indicators and modification, as well as the availability of external data that contributes to inferences about factors that lead to those alterations.

The architecture that is shown below was defined in order to accomplish these goals.

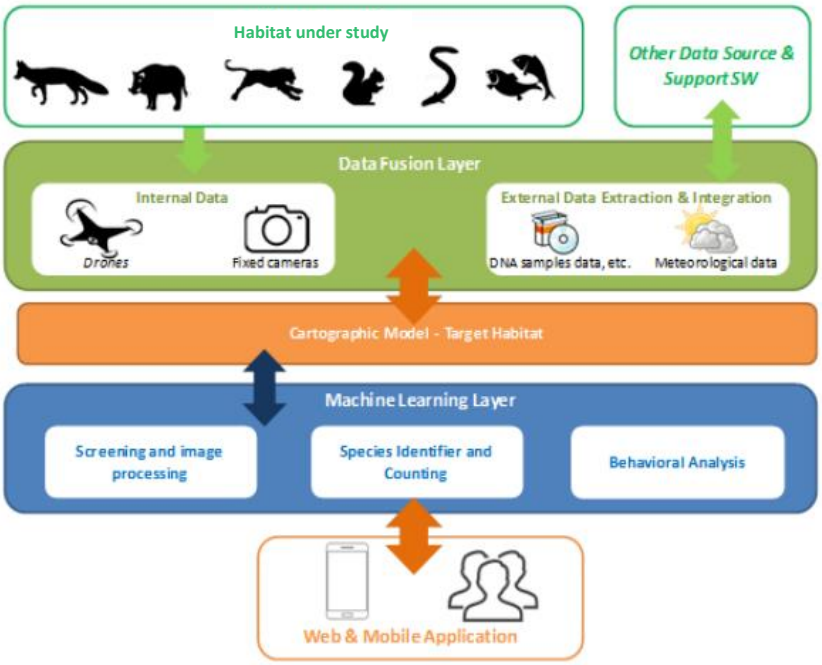


Figure 4- SAFArI solution architecture

However, this Thesis will be exclusively focused on the Screening and image processing module located in the Machine Learning Layer of Figure 4, where image processing and screening takes place and, therefore, is responsible for image noise removal.

In order to make their extraction easier, the study consists of testing which set of operations will best suit a simplification of the target objects contained in the images. Subsequently, it is intended that, after being processed and with less noise, the images are sorted into positives (images with the presence of animals) and negatives (images without animals), so that only images with the presence of validated animals are analysed in greater detail. On the other hand, the solution must be able to provide the user with the degree of confidence in the sorting performed. Thus, if the degree of confidence shown is not considered adequate, the

user will be able to manually sort the contents, helping the solution itself to evolve and learn to better classify the images.

In this context, it should be noted that the Screening and image processing module will be the layer that will compare the results obtained by the solution with results previously validated by the user. These results will confirm the successful or unsuccessful classification by the solution and will have a preponderant weight in the next classifications to be made by the solution.

4.2 DATA UNDERSTANDING

This particular project is based on data sourced from the CRLI - Centro de Recuperação do Lobo Ibérico (Center for the Recovery of the Iberian Wolf) and several monitoring projects developed by BIOTA, where images are captured through a variety of CT devices. These cameras are strategically placed across different locations in Portugal, with the intention of capturing a wide range of animals that are representative of the Portuguese fauna.

It is important to note that all images used in this project are the exclusive property of BIOTA, and are private, having been employed with explicit authorization from the company.

The dataset comprises approximately 600,000 images, taken between November 2006 and July 2020, portraying a diverse range of animal species. However, it should be noted that only 3067 of these images had been categorised by ecologists, so only these images were selected to train, validate, and test the models.

To ensure an appropriate division, the selected categorised images were divided into the following sets: 70% for training, 10% for validation, and 20% for testing. It is worth noting that, in line with YOLO's recommendation, only 10% of the images were designated as background images, devoid of any animals or objects of interest. These distributions of images can be seen in the subsequent table.

Table 5- Division of the dataset

Images	With Animals	Without Animals	Total
Train	1933	214	2147
Validate	277	30	307
Test	552	61	613
Total	2762	305	**3067**

The CT images were labelled by specialised ecologists with expertise in wildlife identification, who manually annotated each image, considering the diverse range of wildlife species commonly found in the fauna of Portugal.

In the subsequent table, detailed information about the labels created by these ecologists is provided, offering a comprehensive overview of the dataset's wildlife species. This includes the label itself, the corresponding animal name (Common Name), and the scientific name (Scientific Name), along with the total number of images that were labelled with each specific label.

Table 6- Overview of the ecologist generated labels

Label	Common Name	Scientific Name	Number of images
0	Vaca	Bos taurus	109
1	Cão	Canis lupus familiaris	203
2	Lobo Ibérico	Canis lupus signatus	231
3	Cabra	Capra aegagrus hircus	222
4	Corço	Capreolus capreolus	272
5	Burro	Equus asinus	12
6	Cavalo	Equus caballus	9
7	Ouriço	Erinaceus europaeus	4
8	Gato	Felis catus	31
9	Gato-bravo	Felis silvestris	51
10	Geneta	Genetta genetta	87
11	Saca-rabos	Herpestes ichneumon	21
12	Lebre	Lepus	199
13	Lince Ibérico	Lynx pardinus	146
14	Fuinha	Martes foina	201
15	Texugo	Meles meles	3
16	Doninha	Mustela	45
17	Coelho	Oryctolagus cuniculus	31
18	Ovelha	Ovis aries	37
19	Esquilo	Sciuridae	340
20	Javali	Sus scrofa	481
21	Raposa	Vulpes vulpes	31

The decision to keep the names of the animals in Portuguese despite the Thesis being written in English is based on the fact that some of these animals are part of the Portuguese fauna, and their translations could potentially lead to confusion or inaccuracies. However, recognizing the need for broader comprehension, a scientific name column has been added to the table for reference. Additionally, translations will be provided if considered necessary.

It is worth noting that the capturing of such images was activated solely through motion sensors, which results in a significant proportion of the images captured being empty, often triggered by environmental factors such as wind, swaying vegetation, or other non-animal movements (Figure 5- (a)). Additionally, these cameras function continuously, capturing images at various times of the day, which introduces variations in lighting conditions (Figure 5- (b)).

Another significant obstacle in the object detection process is the presence of animals that are camouflaged within their surroundings (Figure 5- (c)), which can make their detection and recognition a complex task, even for sophisticated object detection algorithms such as the ones being used. Furthermore, the CT may sometimes capture partial cuts of animals due to various reasons such as the position of the animal within the camera frame or movement during image capture (Figure 5- (d)).

Presented below are some images that are examples of these situations, extracted from the dataset.

(a) Empty image

(b) Low quality image

(c) Camouflaged animal

(d) Partial cut animal

Figure 5- Various scenarios of the dataset (Image Source: BIOTA, 2023)

Understanding and addressing these factors is crucial for developing robust and reliable object detection models tailored to the specific challenges posed by the dataset.

4.3 DATA PREPARATION

The preparation of the data for model training is covered in more detail in this section. While using the YOLO and Faster R-CNN models, an appropriately labelled dataset must be used to train, validate, and test it. In order to label an image, the objects of interest within the images must be identified and annotated with bounding boxes, which gives the models the information they need to identify the object by describing where it is in the image.

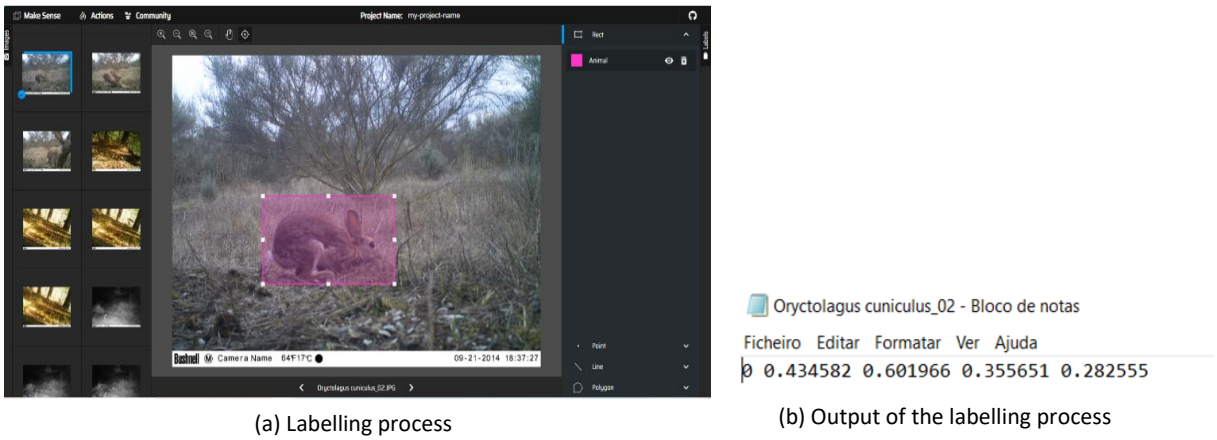
As discussed in Section 4.2, the labelling process for this project was conducted by ecologists. However, upon careful analysis of the images, it was identified that certain images had incorrect labels. For instance, Figure 6-(a) was mistakenly labelled as a Cão (Dog) instead of Texugo (Badger). Additionally, some images lacked bounding box annotations, as shown in Figure 6-(b).

(a) A Badger mistaken identity as a Dog

(b) Inconsistent labelling of the Dog and Goats in the image

Figure 6- Examples of incorrect labelled images

To address this issue, the specific images were re-labelled using the https://www.makesense.ai/ platform, which is a web-based annotation tool that lets users annotate images and produce training datasets for a variety of computer vision applications (Figure 7- (a)). The annotations were then saved in a format compatible with the YOLO model requirements, as shown in Figure 7- (b).



(a) Labelling process

(b) Output of the labelling process

Figure 7- Labelling example using https://www.makesense.ai/ platform.

In addition to re-labelling the specific images for YOLO, this newly labelled dataset was used to create similar annotations for the Faster R-CNN model. To achieve this, the annotations were converted into a JSON file format, as it is the format that the Faster R-CNN model can read and process efficiently. The JSON file contains the necessary information about the object's bounding boxes and its respective labels in one place, enabling the Faster R-CNN model to accurately detect and classify the objects in the images.

Additionally, in the context of this study, the primary objective is to accurately identify individual animals within the CT images, rather than focusing solely on species classification. Considering this objective and recognizing the importance of having a sufficient number of labelled images per object class to achieve optimal model performance, a decision was made to group the labels first by taxonomic family and then by visual similarities. As demonstrated in Table 6, it is evident that some of these labels lack a sufficient number of images, emphasising the importance of this approach.

By grouping the labels according to the taxonomic family, it is possible to capture characteristics and visual traits shared among species that belong to the same family. This approach enables the model to learn and recognize common features across related species, enhancing its ability to accurately identify animals within the CT images. The table below shows the animal species grouped by family.

Table 7- Animal grouped by family.

Grouping by Family
Bovidae
Cabra, Ovelha, Vaca
Canidae
Cão, Lobo, Raposa
Cervidae
Corço
Equidae
Burro, Cavalo
Erinaceidae
Ouriço
Felidae
Gato, Gato-bravo, Lince
Herpestidae
Saca-rabos
Leporidae
Coelho, Lebre
Mustelidae
Doninha, Fuinha, Texugo
Sciuridae
Esquilo
Suidae
Javali
Viverridae
Geneta

Furthermore, by considering visual similarities, it is possible to consider variations in appearance, coloration, and body structure that may exist within the same family, or even among animals from different families that share resemblances in their features.

The results of this label grouping approach are summarised in the table below, which showcases the newly created labels, the corresponding animal species assigned to each one and the total number of images by label.

Table 8- Final Labels

New label	Group of animals	Number of images
0	**Vaca, Cabra, Ovelha, Javali**	849
1	**Cão, Lobo, Raposa**	465
2	**Gato, Gato-bravo, Lince**	228
3	**Cavalo, Burro, Corço**	293
4	**Geneta, Saca-rabos, Texugo, Doninha, Fuinha, Esquilo**	697
5	**Lebre, Coelho**	230

As evident from the table above, it can be observed that Ouriço (Hedgehog) has been excluded from the label grouping approach. This decision was made due to the limited number of available images featuring Hedgehog and the absence of a clear visual similarity or taxonomic family grouping with other animals in the dataset. Since the primary objective of the study is to accurately identify animals within the images, the small number of Hedgehog images would not provide sufficient data to train and evaluate a robust model for their detection.

The decision to group the animals in this manner was primarily driven by the limited availability of images for each individual animal category. As indicated in Table 8, some groups may have a smaller number of total images compared to others, but it's important to note that many of these images contain multiple instances of the animals within them. This approach helps to maximise the utilisation of available data, ensuring that even groups with a limited number of images contribute significantly to the model's performance. For instance, the label 2 has only 228 images, but the models have been pre-trained with similar types of animals, so having this number of images does not significantly impact their performance.

With this comprehensive data preparation a solid foundation was established, providing a powerful tool for in-depth analysis, offering valuable insights for the subsequent section.

4.4 MODELLING

As mentioned in Section 3, the object detection process in this study focused on two prominent models: YOLO and Faster R-CNN. In order to enhance the precision of these models, BS was incorporated as an additional technique during the object detection process.

To provide a comprehensive understanding of the implemented approach, this sub-chapter is organised as follows. The first section provides a detailed description of the YOLO model and its implementation, as well as the results obtained through its application. The second section introduces the Faster R-CNN algorithm, explains its implementation, and presents the results achieved. Lastly, in the third section, a modification to the models is presented, incorporating BS to each of them to further improve detection accuracy.

4.4.1 YOLO

The present section provides a comprehensive examination of the functionality of the YOLO model. The YOLO (You Only Look Once) model is a state-of-the-art object detection algorithm widely used in computer vision applications. The model operates by segmenting an input image into a grid of cells and predicting bounding boxes and object probabilities for each cell. The object probabilities represent the likelihood of an object's presence in that box and how well the predicted box fits the object, while the bounding boxes denote the object's position and size within the box.

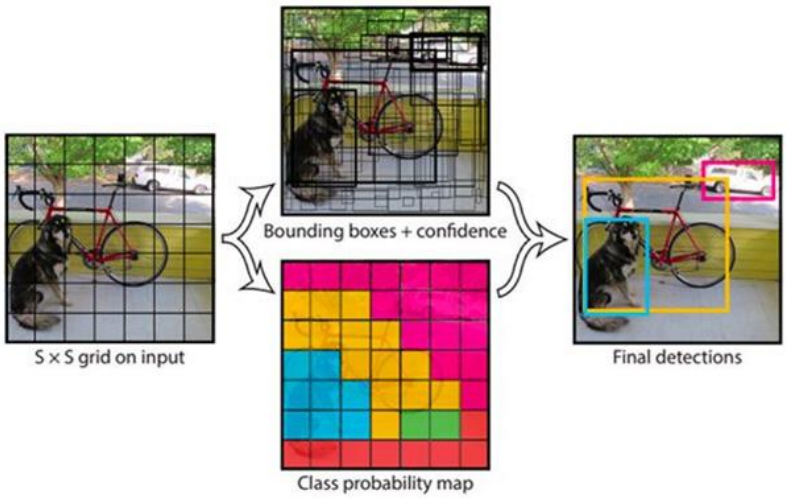


Figure 8- YOLO model demonstration (Image Source: Redmon et al., 2016)

The YOLO model uses a DCNN to process input images and predict object classes and bounding boxes. This model is composed of multiple layers, with each layer focusing on learning progressively more abstract features from the input image. At each grid cell, the final layer of the model is responsible for predicting both object classes and bounding boxes. The fact that YOLO treats frames detection as a regression issue makes it exceptionally quick and eliminates the need for a complicated pipeline that other models require (Redmon et al., 2016).

Since Redmon et al. first made YOLO available in 2015, it has been constantly evolving, with many versions being introduced to improve or address the previous versions' limitations, which subsequently improved accuracy, speed, and/or localization performance, shown on Figure 9. Currently, during the development of this Thesis, YOLO is in version 8.

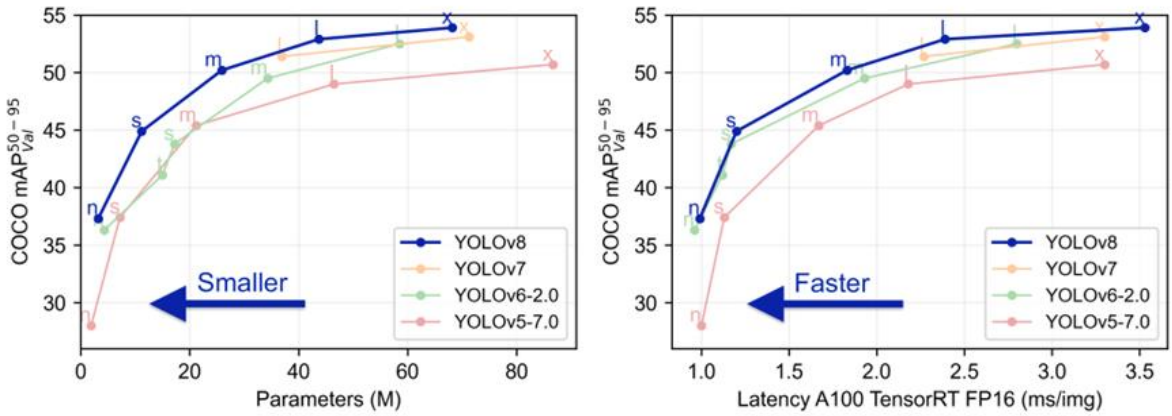


Figure 9- YOLO model versions comparison (Image Source: ultralytics, 2023)

4.4.1.1 Experiments with YOLO model trained from scratch

The following section presents the experiments conducted in this study, relating to the YOLO model. The experiments were carried out to evaluate the performance and effectiveness of different versions of the YOLO model in conjunction with different label configurations in the dataset.

Given that many of the images within the dataset contain instances of people or cars, attributed to the placement of CT devices in areas where human and vehicular activity is observed, or the need for regular maintenance and monitoring of the cameras, several experiments were conducted to investigate different approaches to labelling these images. This experiment was conducted to ensure that an image with an animal was not mistaken by an image with a person, due to their visual similarities.

The first experiment involved labelling only the animals present in the images and training the model solely for animal detection (Figure 10-(a)). Subsequently, a second experiment (Figure 10-(b)) was conducted to label both animals and people with distinct labels, allowing for the differentiation between the two object classes. Finally, a third experiment was performed, where both animals and people were labelled with a single label, "Animal," to classify them under a unified category (Figure 10-(c)).

(a) Output of the first experiment

(b) Output of the second experiment

(c) Output of the third experiment

Figure 10- Examples of the experiments taken.

The experiments were conducted using YOLOv5 and a smaller dataset comprising only 200 images, since it aims to investigate the impact of different labelling approaches on the performance and accuracy of the YOLO model in detecting animals and people in the dataset, with their results being presented in the table below.

Table 9- Labelling experiments results

# Experiment	Animal accuracy	Person accuracy	Average accuracy
1	0.721	-	0.721
2	0.635	0.995	0.815
3	0.487	0.487	0.487

Given that the primary focus of the object detection model is to accurately detect animals, the accuracy results for animal detection are of utmost importance. Although Experiment 2 returned the highest overall average accuracy (0.815), Experiment 1 demonstrated a higher accuracy specifically for animal detection. Consequently, subsequent tests and evaluations were conducted using the labelling approach employed in Experiment 1.

Following this experiment, the same dataset was evaluated with the labels from Experiment 1, using various versions of the YOLO model ranging from version 4, as proposed in Riechmann et al. (2022) study, to the latest version, version 8. However, due to compatibility issues with the CPU of the computer being used, version 4 could not be tested, as it requires a different CPU. As a result, the initial version tested was version 5.

The subsequent table presents the mAP50 results obtained in each version of YOLO, allowing for a comparison of the outcomes as the same dataset and number of epochs (epochs = 25) were utilised for training in each version, ensuring consistency in the evaluation process. It is

important to note that the model was trained from scratch, with only the label 'Animal', created in Experiment 1, as previously described.

Table 10- Comparison of the YOLO versions

	YOLOv5	YOLOv6	YOLOv7	YOLOv8
mAP50	0.721	0.191	0.413	0.004
Time to train (per epoch)	0h07	0h01	0h02	0h01

The results presented in Table 10 reveal substantial variations in mAP50 and training efficiency among the different YOLO versions. While YOLOv5 stands out as the most accurate option, it requires longer training times. In contrast, YOLOv6 and YOLOv8 demonstrated lower accuracy and relatively faster training times. YOLOv7, while not matching the accuracy of YOLOv5, offers a balance between accuracy and training time.

4.4.1.2 Experiments with YOLO model using pretrained weights

The next experiment used the complete dataset explained in Section 4.2, as outlined in Table 5, and applied it to each version of the YOLO model, consistent with the versions used in the previous section.

The models were trained using pre-trained weights from YOLO and the labels generated in Section 4.3, as outlined in Table 8. For demonstration purposes, only the command of YOLOv5 is explained, but it is important to note that the same parameters were used across all versions of the model, with appropriate modifications required by each version.

To train YOLOv5, the following command was executed:

```
!python 'yolov5/train.py' --img 640 --batch 16 --epochs 100 --data
'yolov5/data/custom_data.yaml' --weights 'yolov5/data/yolov5s.pt' --
patience 30 --save-period 1
```

The training process involved 100 epochs, employing the pre-trained weight yolov5s as it is the standard parametrization. Additionally, a patience value of 30 was set during training, which determines the number of consecutive epochs to wait for improvement before early stopping. This value helps prevent overfitting and ensures that the model converges to the best possible performance. The custom_data.yaml file was used to provide the necessary labels of Table 8, and the dataset path for the training process.

The results of this experiment exceeded the expectations, achieving the highest level of performance observed thus far. The following table presents the overall performance results of the different YOLO versions on the test images.

Table 11- Performance comparison of YOLO versions

	YOLOv5	YOLOv6	YOLOv7	YOLOv8
mAP50	0.858	0.787	0.660	0.865
Accuracy	0.864	0.868	0.770	0.877
Omission Error	0.143	0.136	0.246	0.127
Time to train (per epoch)	1h07	1h35	0h24	1h42

As mentioned, the table illustrates the comparative performance of different YOLO versions. YOLOv8 demonstrates the best results in terms of mAP50, accuracy, and omission error, indicating its superior ability to accurately detect animals in the given dataset. However, it should be noted that YOLOv8 requires a longer training time (1h42 per epoch) compared to other versions.

On the other hand, YOLOv5 provides a noteworthy alternative with a relatively faster training time of 1 hour and 7 minutes per epoch, and achieves reasonable results, with an mAP50 score of 0.858 and an accuracy of 0.864. Although YOLOv5 may not match the performance of YOLOv8, it offers a good balance between detection accuracy and training efficiency.

In contrast, YOLOv7 demonstrates a lower overall performance, with an mAP50 score of 0.660 and an accuracy of 0.770. This version exhibits reduced accuracy compared to the other versions in detecting animals in the CT images. Additionally, the relatively higher omission error of 0.246 suggests that YOLOv7 is more prone to missing animals in the detections. While YOLOv7 falls behind in performance, it is significantly shorter in the training time, taking only approximately 24 minutes per epoch.

Considering the results of YOLOv7, it may not be the most optimal choice when high accuracy is of utmost importance. However, its quick training time can make it a viable option in specific use cases that prioritise efficiency over absolute accuracy.

The results obtained from the two sets of experiments presented above showcase the significant impact that dataset variations and labelling techniques can have on the performance of the YOLO object detection model. It is important to emphasise that the outcomes of the first experiment (Table 10), where the model was trained from scratch with a smaller dataset using different labelling approaches, are quite distinct from the results of the second experiment (Table 11), which involved using the complete dataset with pre-trained weights.

These divergent results underscore the importance of tailoring the choice of the YOLO version, dataset, and labelling strategy to the specific requirements and objectives of the research. Ultimately, selecting the most appropriate YOLO version for this project should involve a thorough evaluation of factors such as detection performance, training time, and alignment

with research goals, ensuring the chosen approach effectively meets the unique project requirements.

4.4.1.2.1 Modifications

After analysing the results obtained by the model that achieved the highest performance in terms of mAP50 and accuracy, YOLOv8, some unexpected outcomes were identified. As evident from the confusion matrix below, label 0, which includes animals like Vaca (Cow), Cabra (Goat), Ovelha (Sheep), and Javali (Boar), exhibits significantly lower performance compared to the other labels. Only 50% of the classifications that should belong to label 0 are correctly identified, while 49% of those images are incorrectly classified as background.

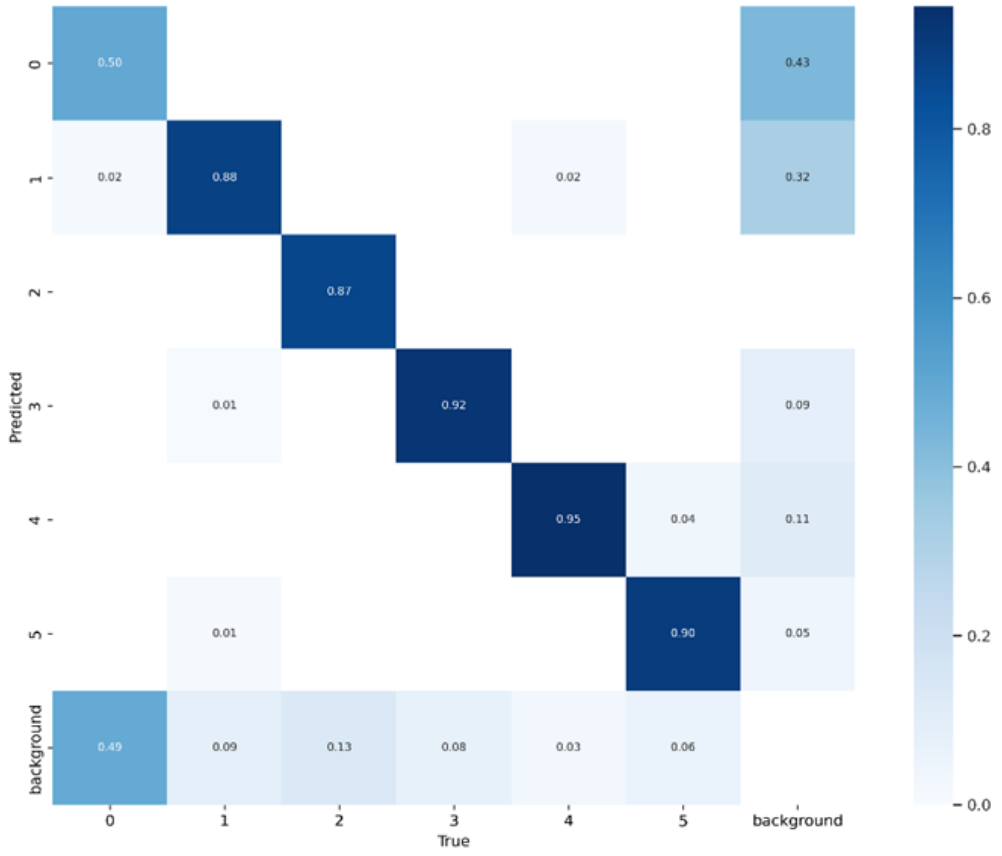


Figure 11- Confusion Matrix of the YOLOv8

This occurrence could be attributed to various factors, such as the visual dissimilarity of Javalis compared to other animals in the group, as they do not belong to the same family. Additionally, insufficient training images of label 0 might contribute to this issue. To address this, an experiment was conducted where modifications were made to the image labels, creating a new exclusive label for Javali, label 6. As a result, the groups were reorganised as follows:

Table 12- Labels of Modification 1

New label	Group of animals	Number of images
0	**Vaca, Cabra, Ovelha**	368
1	**Cão, Lobo, Raposa**	465
2	**Gato, Gato-bravo Lince**	228
3	**Cavalo, Burro, Corço**	293
4	**Geneta, Saca-rabos, Texugo, Doninha, Fuinha, Esquilo**	697
5	**Lebre, Coelho**	230
6	**Javali**	481

The YOLOv8 model was then re-trained with these new labels, and the results are shown in the table below.

Table 13- Performance of Modification 1 in YOLOv8

Modification 1	**YOLOv8**
mAP50	0.804
Accuracy	0.865
Omission Error	0.138
Time to train (per epoch)	0h35

The results of Modification 1 show a decrease in accuracy and mAP50, along with a slight increase in the omission error. The modified YOLOv8 model achieved an mAP50 of 0.804 and an accuracy of 0.865, both slightly lower than the initial performance of YOLOv8. The omission error increased to 0.138, indicating a slight decrease in sensitivity for detecting animals. Upon further evaluation, the confusion matrix was re-examined to assess the impact of the label modifications on individual classes.

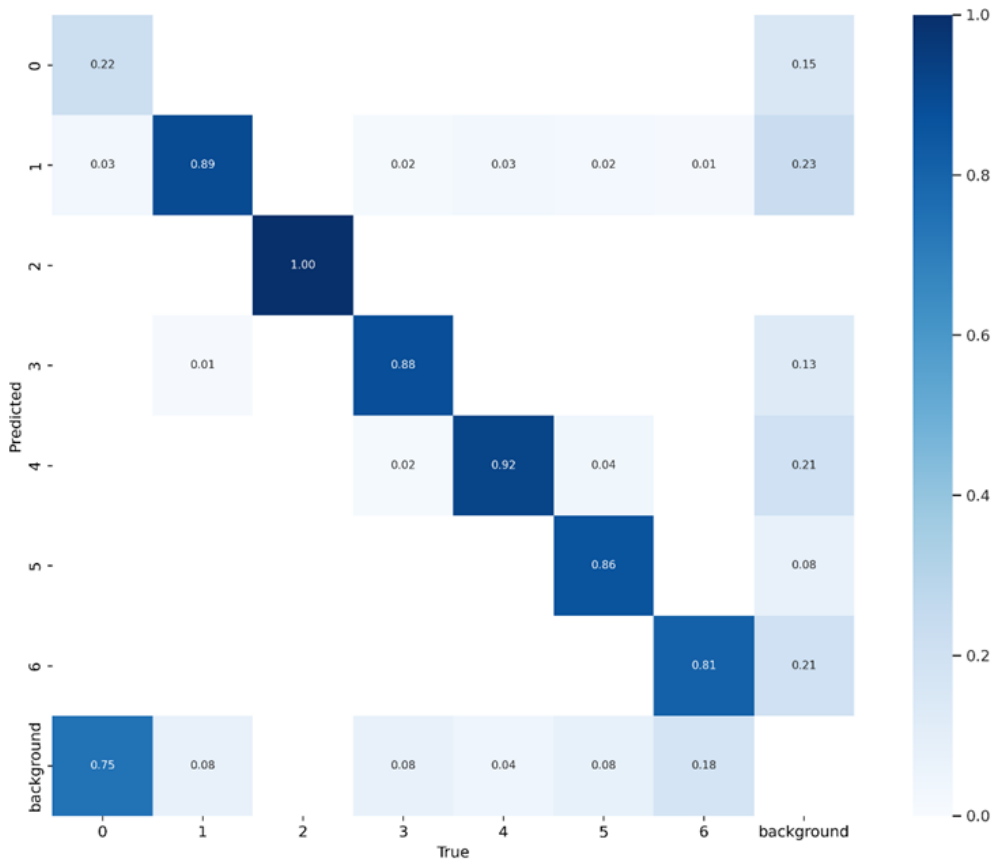


Figure 12- Confusion Matrix of Modification 1 in YOLOv8

After analysing the results from the confusion matrix with these new modifications, it can be observed that the performance of label 0 worsened, with only 22% of correct classifications. However, the new label 6 achieved a satisfactory classification rate of 81%. Based on these results, further adjustments to the labelling system were considered to investigate the concern further. Therefore, a new label exclusively for Cabra and Ovelha, label 7, was created, considering that these two animals share closer visual similarities. The revised labels are as follows:

Table 14- Labels of Modification 2

New label	Group of animals	Number of images
0	**Vaca**	109
1	**Cão, Lobo, Raposa**	465
2	**Gato, Gato-bravo Lince**	228
3	**Cavalo, Burro, Corço**	293
4	**Geneta, Saca-rabos, Texugo, Doninha, Fuinha, Esquilo**	697
5	**Lebre, Coelho**	220
6	**Javali**	481
7	**Cabra, Ovelha**	259

Consequently, the results of the second modification will be presented and analysed below:

Table 15- Performance of Modification 2 in YOLOv8

Modification 2	**YOLOv8**
mAP50	0.842
Accuracy	0.812
Omission Error	0.104
Time to train (per epoch)	0h37

After examining the results presented in the table above, a comparison between the outcomes of Modification 2 and Modification 1 reveals interesting insights. In Modification 2, the YOLOv8 model achieved an improved mAP50 of 0.842, indicating a better overall detection performance. However, there was a trade-off with accuracy, which decreased to 0.812. The omission error, on the other hand, decreased to 0.104, signifying a more sensitive detection of animals.

Continuing the analysis of the results, the associated confusion matrix demonstrated that the performance of label 0, Vaca, improved significantly. On the other hand, the group containing Cabra and Ovelha exhibited suboptimal results, with only 25% of correct classifications. This observation indicated challenges in effectively distinguishing these two animals within the same label, as 75% of the classifications were confused with the background, as is evident in the confusion matrix below.

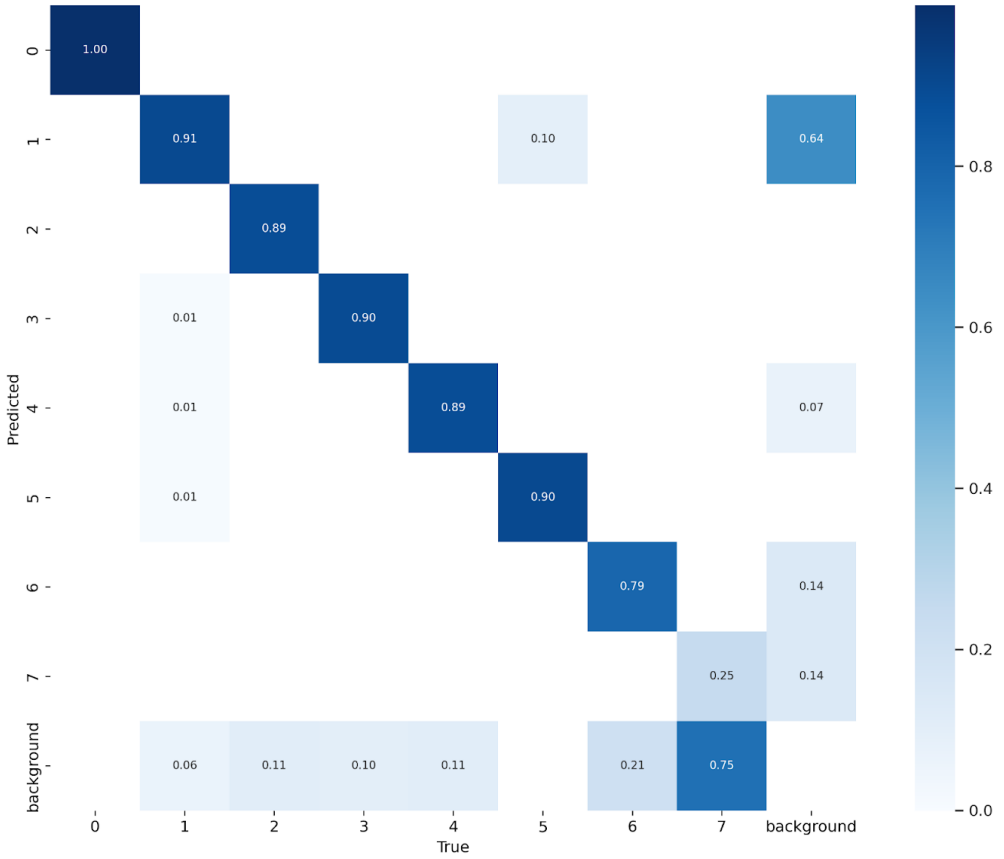


Figure 13- Confusion Matrix of Modification 2 in YOLOv8

Considering the observed trends, the possibility of a third modification that involves separating Cabra and Ovelha into distinct labels becomes apparent. It is reasonable to assume that implementing this modification could lead to more favourable results, particularly in accurately classifying these two animals.

These adaptations emphasised the potential benefits of classifying animals by species rather than groups. However, due to the limitations of the available data in this experiment, a more comprehensive exploration was not feasible. Despite these complexities, the overarching conclusion remains that YOLOv8, in its original configuration without these modifications, achieved superior overall classification performance.

4.4.2 Faster R-CNN

This section provides a comprehensive analysis of the Faster R-CNN (Faster Region-based Convolutional Neural Network) model, a state-of-the-art object detection algorithm widely employed in computer vision applications. ULAŞ TEKELİ & YALIN BAŞTANLAR (2019) and Banerjee et al. (2022) conducted experiments comparing different object detection models, and both studies concluded that Faster R-CNN demonstrated superior performance and

accuracy compared to other models tested. As a result, Faster R-CNN was chosen as the object detection model for their research, ensuring reliable and precise animal detection.

Faster R-CNN, developed by Shaoqing Ren et al., is a highly acclaimed object detection model that has revolutionised the field of object detection in images and videos. The model showcases remarkable accuracy in real-time object detection applications while maintaining impressive computational efficiency. Therefore, it is crucial to introduce and comprehensively explain Faster R-CNN, along with its underlying components and operational mechanism.

As illustrated in Figure 14, Faster R-CNN consists of two main components: the Region Proposal Network (RPN) and the Fast R-CNN. The RPN generates region proposals by sliding a small network over the convolutional feature map, creating potential object bounding boxes that are subsequently refined and classified in the subsequent stages. The Fast R-CNN takes these proposals and extracts features using a Region of Interest (RoI) pooling layer. These features are then fed into fully connected layers to predict the object class and refine the bounding box coordinates (Ren et al., 2015).

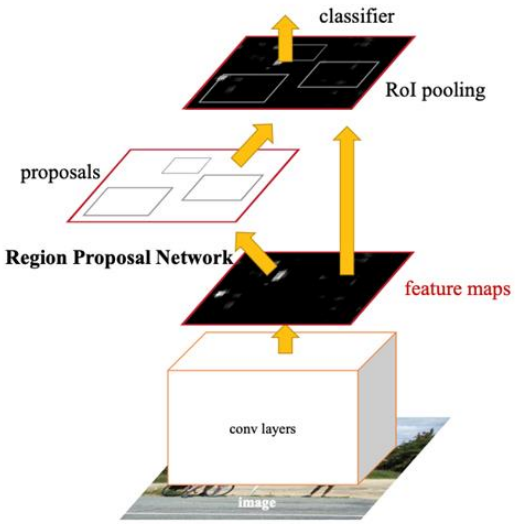


Figure 14- Faster R-CNN main components (Image Source: Ren et al., 2017)

Compared to previous object detection models including R-CNN and Fast R-CNN, Faster R-CNN introduces the innovative concept of RPN, significantly improving the efficiency of the detection process. By sharing convolutional layers between the RPN and Fast R-CNN, the model can generate region proposals and classify objects in a single forward pass, making it faster and more accurate.

It is worth mentioning that Faster R-CNN has undergone various iterations and improvements since its introduction, with subsequent versions aiming to enhance accuracy, speed, and localization performance. Researchers have introduced improvements such as Feature Pyramid Networks (FPN) and Region-based Fully Convolutional Networks (R-FCN) to further refine the model's capabilities.

Further elaboration will be provided subsequently regarding a detailed explanation of the implementation of Faster R-CNN, followed by the presentation of the obtained results.

4.4.2.1 Model Architecture and Results

In the context of this Thesis, the default version of Detectron2 for Faster R-CNN was used. The model, represented as faster_rcnn_R_50_FPN_3x, is built on top of a ResNet-50 backbone (R_50), and employs a FPN that allows the model to effectively use multi-scale features for detecting objects of various sizes.

Detectron2 is a library supported by Facebook, which follows previous versions that are now deprecated, and it comes pre-trained on the COCO Dataset. It can be fine-tuned on custom datasets using pre-trained models, as is the case in this research, addressing the challenges faced when transitioning from research to production.

The training of the model was performed using the labels generated in Section 4.3, as outlined in Table 8. The following configurations were applied in the training of the model:

```
cfg = get_cfg()
cfg.merge_from_file("detectron2/configs/COCO-
Detection/faster_rcnn_R_50_FPN_3x.yaml")
cfg.DATASETS.TRAIN = ("my_dataset_train",)
cfg.DATASETS.TEST = ("my_dataset_val",)
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = "detectron2://COCO-
Detection/faster_rcnn_R_50_FPN_3x.yaml"
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 3000
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 64
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 6
```

The model was trained with specific hyperparameters, including a batch size of 2, a learning rate of 0.00025, and a maximum of 3000 iterations during training. Additionally, the RoI heads were designed to handle 64 bounding box proposals per image. The model was customised to classify objects into 6 different classes specified in Table 8.

The decision to use 3000 epochs during training is influenced by a balance between resource availability and convergence to an optimal solution. Too few epochs might result in underfitting, where the model hasn't learned enough from the data, where too many epochs can lead to overfitting, and where the model becomes too specialised to the training data and doesn't generalise well to new data. In this context, after experimentation and considering the dataset's complexity, 3000 iterations were deemed an appropriate number of epochs to allow the model to learn meaningful patterns from the data without overfitting.

The performance of the Faster R-CNN model was evaluated on the test provided in the Data Preparation Section. Once again, quantitative evaluation metrics such as mAP50, accuracy and omission error were employed and are presented in the table below:

Table 16- Performance of Faster R-CNN

	Faster R-CNN
mAP50	0.293
Accuracy	0.703
Omission Error	0.089
Time to train (per epoch)	0h03

As shown in the table, Faster R-CNN achieves an mAP50 score of 0.293, which indicates a reasonable level of detection, accompanied by an accuracy score of 0.703. The omission error, quantified at 0.089, highlights a relatively low rate of missed animal detections, solidifying the model's competence in minimising false negatives.

A notable attribute of Faster R-CNN is its rapid training time, requiring just 3 minutes per epoch. This efficient training process contributes to its suitability for scenarios where timely results are essential.

Interestingly, the mAP50 result of Faster R-CNN contradicts expectations set by the literature review, suggesting potential for further investigation into the reasons behind its lower mAP50 score in the context of this specific dataset.

Moreover, it's essential to recognize that while the omission error remains notably low, this outcome could potentially stem from an inadequacy of empty images (images without an animal) within the model's training dataset, which might result in their inadvertent exclusion.

When assessing the performance of Faster R-CNN, it's evident that the model strikes a balance between accuracy and training efficiency. While it’s mAP50 score and accuracy figures are promising, they might not reach the levels achieved by some YOLO versions. Given that the achieved performance was lower than anticipated and considering the already competitive results obtained from YOLO variants, it was deemed unnecessary to undertake training the Faster R-CNN model with the various image label modifications implemented for the YOLO models in the Modifications Section. This strategic decision was made with the aim of optimising resource allocation and research focus, acknowledging that the performance gap might be due to the intrinsic characteristics of the Faster R-CNN framework.

Ultimately, the selection of the most suitable model, whether it be YOLO or Faster R-CNN, should be informed by a thorough consideration of the specific research goals, detection accuracy requirements, and training time constraints.

4.4.3 Background Subtraction

Background subtraction (BS) is a fundamental technique in computer vision that aims to separate foreground objects from the background in video or image sequences. Its primary goal is to detect and extract moving objects, which can be particularly useful for detecting moving animals in CT images. By subtracting the static background from the current frame, the algorithm can identify regions where there is a significant change, indicating the presence of a moving object, such as an animal.

It is worth mentioning that BS is used in this Thesis as a complementary technique alongside Object Detection models such as YOLO or Faster R-CNN. The results obtained from BS are combined with the outputs of the Object Detection models to try to enhance the accuracy and reliability of the overall Animal Detection system.

In this Thesis, the implementation of BS incorporates concepts and techniques described in the OpenCV documentation. The documentation provides additional information, insights, and usage examples, which can be found at the following link: https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html. It is important to note that while the code provided in the documentation serves as a reference, there may be slight differences in the code used in this Thesis.

The implementation process involves several steps. Initially, the first frame of the image sequence is used as the initial background, capturing the static background without any moving objects. The subsequent frames are then compared with the initial background to detect changes, through the use of a foreground mask. The algorithm updates the background model by considering the current frame and its similarity to the existing background.

The foreground mask mentioned above is a binary image that identifies the pixels belonging to moving objects in the scene. It is obtained by subtracting the current frame from a background model that represents the static portion of the scene. By subtracting the background model from the current frame, the algorithm identifies regions with significant changes, which correspond to the foreground objects or moving elements in the scene. However, these extracted foreground regions may contain noise, requiring the use of contour analysis to improve object detection accuracy. In the context of CT images, this analysis helps to eliminate images with artefacts such as wind movement or trees.

The subsequent sub-sections will provide a detailed explanation of the code employed, followed by an in-depth analysis of the obtained results.

4.4.3.1 Code Explanation

As mentioned, the BS technique in this Thesis is implemented using the OpenCV library in Python. The code processes a sequence of images obtained from CT, starting by organising

the images into these sequences based on a specified time interval, which in this case is 10 minutes. Each sequence represents a potential event where a moving object might be present.

The BS technique is then applied to each sequence of images. The code loads the first image in each sequence as the initial background and subsequently compares the remaining images to this background model. It computes the absolute difference between the current image and the background, applies a threshold to obtain a binary motion mask, and performs morphological operations to refine the mask. Contours are extracted from the motion mask, and objects are detected based on these contour areas.

It is important to note that the BS is only used when a burst of images contains more than one image. For single images, only the results obtained from the Object Detection models, such as YOLO or Faster R-CNN, are used, given that comparing the background of a single image is not feasible. Figure 15 provides an example of the results obtained through this method, where Figure 15- (a) and Figure 15- (b) are a sequence of images within the same burst, and Figure 15- (c) represents the motion mask obtained through BS for these two images. The white pixels in the foreground mask image indicate the differing pixels between the two images, through which the algorithm then verifies if these differing pixels form an area larger than a specified threshold. If they do, a "True" value is assigned to the sequence, as demonstrated in Figure 15- (d).

(a) First image of the sequence

(b) Second image of the sequence

(c) Foreground mask of the sequence

Folder	Images_names	Outputs	Object Detected
2013-04-16_18-19-29	set_bgl1_1566.jpg, set_bgl1_1586.jpg	True	VERDADEIRO

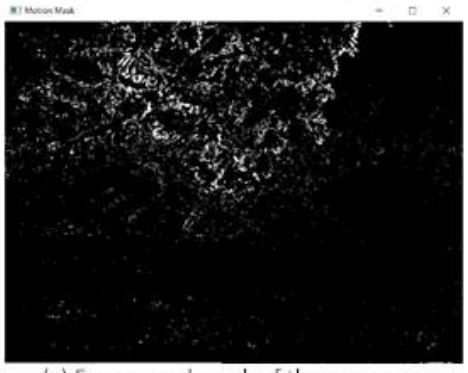
(d) Results obtained through the BS

Figure 15- Example of the results obtained through BS in an image with an animal.

Furthermore, to ensure accurate detection, only images within a contour area larger than 200 pixels are considered significant enough to indicate the presence of an animal. This approach effectively addresses the challenge posed by images where the only difference is the movement of clouds or trees, ensuring they are classified as empty instead of mistakenly identifying them as containing an animal. Figure 16 showcases an example of such images, where Figure 16- (a) and Figure 16- (b) represent a sequence of images within the same burst, and Figure 16- (c) is the foreground mask obtained through the BS for these two images. Since the differing pixels between the images are more spread out, the images are considered "False", as shown in Figure 16- (d).

(a) First image of the sequence

(b) Second image of the sequence



(c) Foreground mask of the sequence

Folder	Images_names	Outputs	Object Detected
2016-11-30_07-23-09	11300059.JPG, 11300060.JPG	False	FALSO

(d) Results obtained through the BS

Figure 16- Example of the results obtained through BS in an image without an animal.

Finally, the results obtained are then saved to an Excel file for further comparison with the object detection models. The saved results include the sequence folder, the image names of this folder, the outputs of the BS algorithm indicating the presence or absence of objects in each image, and a final flag indicating if any object was detected in the sequence. Examples of these results are presented in Figure 15- (d) and Figure 16- (d).

Additionally, the choice of a contour area threshold of 200 pixels was determined through experimentation to identify the optimal value for this scenario. The table below presents the results obtained with different contour area thresholds in the images specifically separated for testing and can be compared against the actual values found in Table 5, line 3 in the Data Understanding Section.

Table 17- Results of contour area threshold experimentation

Contour Area (pixels)	True	False	None
100	269	146	198
200	257	158	198
300	246	169	198
400	239	176	198

It is noteworthy that the "None" values represent images that are alone in a burst and thus are not included in the results obtained through BS. Moreover, accuracy values were calculated to evaluate the performance of different contour area thresholds, as shown below.

Table 18- Accuracy evaluation of different contour area thresholds

Contour Area (pixels)	Accuracy
100	0.662
200	**0.672**
300	0.660
400	0.643

Based on these results, the contour area threshold of 200 pixels demonstrates a relatively higher accuracy compared to other thresholds. This validates the effectiveness of the chosen value in distinguishing significant movements indicative of animals while minimising False Positive detections.

These findings emphasise the role of BS as a valuable tool for Movement Detection but given that it may not be as effective as an Object Detection model on its own, it should be used as a supplementary approach to complement these models.

4.4.3.2 Results obtained

As mentioned previously, BS was used as an additional tool to verify and compare the results obtained with the YOLO and Faster R-CNN. This sub-section will present these results along with the challenges encountered when working with BS in CT images.

The results of the BS that were saved in an Excel file were then grouped together with the results saved from the Object Detection models. A combination of the outputs from both methods was used to determine the final classification of each image. If both methods output a False result or, in the case of single images in a burst, if the BS method output a None result and the Object Detection method output a False result, the image was considered False,

indicating the absence of an animal. Otherwise, the image was considered True, indicating the presence of an animal. This approach was chosen to minimise False Negatives, as these images containing animals are rare and valuable, as previously mentioned in Section 3.2.

With these results grouped, then to evaluate the performance of each method, the accuracy, and omission error were calculated. The results obtained through the combined approach of YOLO with BS are presented in in Table 19.

Table 19- Performance comparison of YOLO versions with BS

	YOLOv5 + BS	YOLOv6 + BS	YOLOv7 + BS	YOLOv8 + BS
mAP50	0.858	0.787	0.660	0.865
Accuracy	0.871	0.875	0.815	0.882
Omission Error	0.066	0.063	0.131	0.055
Time to train (per epoch)	1h07	1h35	0h24	1h42

As evident from the results, the combination of YOLO with BS achieves improved performance compared to using YOLO alone, particularly in terms of accuracy and omission error. This enhancement is evident across all versions of YOLO with BS, showcasing the effectiveness of incorporating the BS technique. It is worth noting that the time to train (per epoch) and the mAP50 values remain consistent with those achieved by YOLO alone, as these metrics are solely calculated based on the YOLO model and remain unaffected by the integration of BS.

When comparing the different combinations of YOLO with BS, YOLOv8 + BS achieves the highest accuracy (0.882) and the lowest omission error (0.055), so stands out as the top-performing combination.

Following closely behind is YOLOv6 + BS, which achieves a respectable accuracy of 0.875 and an omission error of 0.063. This combination demonstrates improved performance compared to YOLOv5 by a small margin, highlighting its improved detection capabilities when aided by the BS technique.

On the other hand, YOLOv7 + BS exhibits lower overall performance, with an accuracy of 0.815 and a higher omission error of 0.131 compared to the other combinations. However, it compensates again by offering a significantly shorter training time of 24 minutes per epoch. Importantly, even though YOLOv7 + BS exhibits lower performance compared to other combinations, it still outperforms YOLOv7 alone.

The results of the combination of Faster R-CNN with the BS technique are presented in Table 20.

Table 20- Performance of Faster R-CNN with BS

	Faster R-CNN + BS
mAP50	0.293
Accuracy	0.436
Omission Error	0.567
Time to train (per epoch)	0h03

The outcomes presented in the table reveal that the integration of the BS technique with Faster R-CNN doesn't yield as notable improvements as observed with YOLO.

Again, it is worth highlighting that while the addition of the BS technique influences the detection process, the metrics directly associated with the Faster R-CNN model, such as mAP50 and the time to train, remain the same.

However, the changes in accuracy and omission error showcase that the benefits of BS are not uniformly present across all Object Detection architectures. While YOLO models exhibit enhanced performance when combined with BS, the impact on Faster R-CNN performance is less pronounced.

In conclusion, the incorporation of BS as a complementary technique generates distinct outcomes when combined with different Object Detection models. Although the incorporation of BS alongside YOLO demonstrated its potential in enhancing performance leading to improved accuracy and more reliable animal detection, its integration with Faster R-CNN reveals that, in certain contexts, the anticipated benefits of technique fusion did not occur, and the performance exhibited a decline instead of improvement. These results emphasise the importance of recognizing how each technique interacts with the underlying models and data characteristics, and the need for an approach that tailors integration to the unique characteristics of each model and dataset.

5 DISCUSSION

In this chapter, the findings of the study obtained in the Modelling (Section 4.4) are discussed, focusing on the performance of YOLO, Faster R-CNN, and the integration of the BS technique. The discussion aims to elucidate the implications of these results, provide a deeper understanding of the factors influencing the performance of these object detection models, and explore observations and implications arising from these findings.

To facilitate a clearer presentation of the information, this discussion is divided into distinct topics where the results of different experiments are discussed to provide a comprehensive understanding of the performance and implications of the object detection models used.

Experiments with YOLO object detection model trained from scratch

Foremost, it is essential to emphasise that the experiments involving training models from scratch (Section 4.4.1.1) not only demonstrated the limited effectiveness of training from scratch compared to transfer learning with pre-trained weights, but also indicated that distinguishing between people and animals in the labelling had a minimal impact on the results. The findings suggested that, despite the visual similarities between them, the YOLO model's ability to differentiate between the classes did not significantly improve. This implies that factors such as the quantity and quality of the training data and the choice of YOLO version play a more critical role in class discernment than the specific labelling strategy employed. Therefore, the preference should lean towards transfer learning, as it not only enhances performance but also conserves valuable time and resources.

Additionally, a comparative analysis of the results across different YOLO versions done when trained from scratch, as presented in Table 10 of Section 4.4.1.1, highlights significant disparities in performance metrics, particularly in the mAP50, which can be attributed to several factors. One of them is the fact that each YOLO version employs distinct architectural improvements and hyperparameter settings, which can significantly impact their detection capabilities. Also, hardware specifications and computational resources may play a role, as some YOLO versions require more powerful GPUs for optimal training. In conclusion, this divergence in results points out the multifaceted nature of factors affecting the outcomes, which emphasise the importance of comprehensive consideration in model selection and training for optimal results.

Modifications of the image labels

Exploring the adjustments and refinements made to image labels in Section 4.4.1.2.1, a notable challenge emerged regarding animal labelling. Despite some animals belonging to the same family or sharing similar physical characteristics, they often posed a challenge for the

model when it came to categorising them under a single label. Therefore, the preferable approach would have been to categorise animals by species, whenever circumstances allowed, however, this was not feasible within the scope of this project due to the limited number of labelled images available for certain species.

Finally, it is possible to say that this exploration highlights the significance of comprehensive and diverse annotated datasets for successful model training and classification.

Comparative performance of the models with and without BS

In this topic, the comparative performance assessment of various YOLO versions, along with Faster R-CNN, in both their original configurations and with the incorporation of the BS technique, will be explored.

One key aspect to consider when evaluating the results is the inherent difference in computational effort between YOLO and Faster R-CNN, as each algorithm comes with its own set of recommended parameters, leading to substantial variances in training times and resource requisites. YOLO models tend to have longer training times, with recommended parameters, primarily due to their more extensive model architectures. Contrariwise, Faster R-CNN, being a region-based approach, typically requires fewer resources due to its two-stage detection process.

Taking this into account, the table below summarises the key performance metrics for the various YOLO models tested, both with and without the integration of the BS technique, as well as the results for Faster R-CNN with and without BS, all derived from the same dataset:

Table 21- Comparative Performance Metrics of the Models

Model	mAP50	Accuracy	Omission Error	Time to train (per epoch)
YOLO v5	0.858	0.864	0.143	1h07
YOLO v6	0.787	0.868	0.136	1h35
YOLO v7	0.660	0.770	0.246	0h24
YOLO v8	0.865	0.877	0.127	1h42
YOLOv5 + BS	0.858	0.871	0.660	1h07
YOLOv6 + BS	0.787	0.875	0.630	1h35
YOLOv7 + BS	0.660	0.815	0.131	0h24
YOLOv8 + BS	0.865	0.882	0.550	1h42
Faster R-CNN	0.293	0.703	0.890	0h03
Faster R-CNN + BS	0.293	0.436	0.567	0h03

Among all the YOLO versions tested, YOLOv8 emerged as the top performer, demonstrating the highest accuracy and the lowest omission error rates. However, this superior performance was accompanied by longer training times. YOLOv5 offered an appealing alternative, striking a commendable balance between detection precision and training efficiency. Contrariwise, YOLOv7, while boasting quicker training times, exhibited reduced overall performance, characterised by diminished accuracy, and elevated omission errors.

Moreover, it is worth noting that YOLOv7 is developed by the same authors as YOLOv4, just as YOLOv5 is related to YOLOv8. This familial relationship may have influenced the results, since YOLOv5 and YOLOv8 demonstrate similar performance. This leads to considering the possibility that YOLOv4, if subjected to testing, might exhibit results closer to YOLOv7, suggesting that while YOLOv4 was initially recommended in the literature review, later YOLO versions may have the potential to achieve even better performance.

Regarding YOLO combined with BS, substantial improvements were observed across all YOLO versions, marked by heightened accuracy and reduced omission errors. Remarkably, YOLOv8 + BS and YOLOv6 + BS emerged as the most effective combinations, showcasing high accuracy and minimal omission errors.

Faster R-CNN delivered respectable results alongside rapid training times. Nevertheless, the unexpectedly lower mAP50 score raised intriguing questions about its performance in this specific dataset, potentially linked to the scarcity of negative images in its training set.

Contrary to the expectations set by prior studies, such as ULAŞ TEKELİ & YALIN BAŞTANLAR (2019), where Faster R-CNN is portrayed as a robust two-class classifier suitable for scenarios with animals not represented in the training set, these results formed a slightly different scenario. While Faster R-CNN did exhibit a reasonable level of detection, as evidenced by its accuracy, it fell short in handling animal classes not explicitly present in the training data. This unexpected restriction became apparent when comparing Faster R-CNN performance with that of YOLO, which outperformed it in this aspect. These findings suggest that, despite its capabilities, Faster R-CNN might require further adaptations or an expanded training dataset.

Lastly, when the BS technique was combined with Faster R-CNN, the model yielded less substantial improvements, as evidenced by lower mAP50 scores and accuracy, alongside higher omission errors. These results emphasise the importance of carefully considering the integration of techniques like BS, as their impact can vary depending on the model architecture.

With these comprehensive insights into the performance of various object detection models, it is time to delve into the process of selecting the most suitable model for this particular project.

Optimal model selection for this project

As previously mentioned, the results obtained in this study provide valuable insights into the performance of various object detection models, particularly YOLO versions and Faster R-CNN, both in their original configurations and with the integration of the BS technique. The project's initial objective was to achieve screening efficiencies exceeding 90%, as outlined in the Project Context (Section 4.1), therefore, the selection of the most suitable model for implementation is in line with this objective.

Among these models, the YOLOv8 + BS combined model emerges as the preeminent selection, as it demonstrates exceptional performance with an accuracy score of 0.882 and an omission error of merely 0.055. However, it is essential to further investigate the reasons behind this standout performance and speculate on its implications.

The experience gained from this comparative evaluation confirmed that YOLOv8, while demonstrating superior accuracy, did fall short in terms of training time when compared to some of its counterparts, particularly YOLOv5 and YOLOv7. This could be attributed to YOLOv8's more extensive model architecture and possibly more demanding computational requirements.

Moreover, the application of the BS technique, which was found to significantly enhance the performance of various YOLO versions, further strengthened the recommendation for choosing the YOLOv8 + BS combination. Therefore, if the primary goal is to maximise accuracy

and minimise omission errors, the BS technique appears to be a worthwhile addition to the YOLOv8 model.

In conclusion, the YOLOv8 + BS model, while it may require slightly more time for training compared to the other models, stands as the optimal choice, consistently outperforming its counterparts and aligning closely with the project's initial objectives. Its exceptional accuracy and minimal omission errors make it a compelling choice, solidifying the author's recommendation for implementation in this project.

6 CONCLUSIONS

In the final chapter of this Thesis, three crucial aspects will be addressed to bring this comprehensive study to its completion. First and foremost, the key findings, methodologies, achieved objectives and outcomes will be encapsulated in the synthesis of the work developed, offering a comprehensive overview of the research conducted. Secondly, the limitations and constraints encountered during this study will be reviewed. Lastly, recommendations for future work will be proposed, to guide subsequent research efforts in this domain. Collectively, these components mark the culmination of this research.

6.1 SYNTHESIS OF THE DEVELOPED WORK

As outlined in the Introduction chapter, the primary objective of this study was to devise a robust solution for the automated identification of animals in CT images, a task that has traditionally been labour-intensive and error-prone when done manually. The need for such automation becomes apparent when considering the sheer volume of images produced by CT devices, often numbering in the millions, which makes manual analysis impractical. This Thesis not only fulfilled this goal but also achieved the defined intermediate objectives, ultimately evolving into a practical and usable solution for the SAFArI project.

As part of these objectives, the study conducted an extensive SLR and rigorous exploration of various DL methodologies, meticulously examining the suitability of different models for the task at hand. An in-depth comparison and evaluation of these models, both among themselves and against benchmark results derived from prior research, ensured a well-informed decision-making process. The preceding Discussion chapter provides a comprehensive analysis of the evaluation and experiments made during the modelling process, with a primary focus on the performance of the models recommended in the literature review: YOLO, Faster R-CNN, and the integration of a movement detection method, the BS. Among these, the YOLOv8 + BS combination model clearly emerges as the optimal choice for projects that prioritise accuracy and seek to minimise omission errors, such as the SAFArI project. Considering the extensive analysis conducted, the study culminated in the development of a specialised architecture designed to identify animals in CT images.

The contributions of this research to the field of wildlife conservation and biology are substantial. By automating the labour-intensive task of image analysis in this study, it ensures that the ecological and biological research community receives cleaner, more valuable data, eliminating the need to filter through countless irrelevant or blurred images.

Furthermore, the adaptable nature of the model developed here is promising for similar CT image analysis projects, thereby promoting efficiency and reducing errors in data interpretation. This approach promises to accelerate ecological and biological experiments, making them less resource-intensive and more accessible.

Considering all the research presented, it is evident that this Thesis successfully achieved its stated objectives, laying the foundation for more efficient and accurate wildlife image analysis. Additionally, the developed solution is already in practical use as part of a software demo, where the model functions effectively. This versatile tool can be readily adapted for use in various contexts, as evidenced by its application in the ongoing SAFArI project, further emphasising its usability and value in real-world scenarios.

6.2 LIMITATIONS AND CONSTRAINTS

This Thesis has successfully explored and developed a system for the detection of animals in CT images, nevertheless, it is essential to acknowledge the various limitations and constraints that influenced the research and its findings. Despite the best efforts, the scope and depth of this study were impacted by these factors, which will be discussed in this section.

One of the primary limitations faced during this research was the availability of computational resources. The detection of animals in CT images often demands extensive computing power, especially when using DL models. However, due to constraints in hardware and budget, this study was limited to a specific set of processors and GPU, which might have affected the performance and efficiency of the models. Consequently, the choice of settings for the algorithms and the scale of experiments had to be tailored to fit the available computational resources, restricting exploration of more complex models and potentially leading to suboptimal results.

As mentioned, freely online GPUs were used, such as those provided by platforms like Google Colab, however, since only a free GPUs was used, the number of experiments that could be performed was restricted. As a result, the lack of access to more powerful resources delayed the exploration of different model configurations and architectures, potentially affecting the system's accuracy and certainly its efficiency.

Another significant limitation was the scarcity of labelled images for training the models. Annotated datasets play a critical role in the success of supervised ML approaches, but acquiring comprehensive and diverse labelled images of wildlife can be challenging and time-consuming. As a result, the model's accuracy and generalisation capability may have been affected due to limited data availability.

Within the previously mentioned constraint,, another challenge became apparent: the shortage of available images required for species-level separation. As illustrated in the Modifications Section, the strategy of categorising images by species demonstrated superior effectiveness. However, due to the shortage of images suitable for this separation, it restricted the system's capacity to classify animals with finer granularity.

Moreover, the manual annotation of images for training data or the validation of model outputs introduces the potential for human error. Annotators may misidentify animals or

overlook some instances, affecting the reliability of the ground truth data used in the research. Despite a careful review of all the images before the modelling phase, as demonstrated in the Data Preparation Section, this limitation remains a consideration in the interpretation of results.

Additionally, one crucial aspect that presented a limitation in this study was the inherent variability in CT conditions. Wildlife CT images are often deployed in diverse environments, ranging from dense forests to open savannas, and encounter different weather conditions and lighting variations. The diverse settings can lead to variations in image quality, clarity, and animal behaviour, which may impact the performance of the animal detection system. While efforts were made to curate a representative dataset, the challenge of obtaining a sufficient number of images covering these diverse conditions remains, which potentially affects the model's ability to handle real-world variations effectively, particularly in scenarios involving multiple animals in varied environments.

The Thesis also faced several constraints that further limited the scope and execution of the research. One of the primary constraints was time, which restricted the ability to conduct more extensive experiments with a larger dataset and enhanced processing capabilities. Proper fine-tuning and optimisation of detection algorithms often requires extensive experimentation with various hyperparameters and architectures. However, due to time constraints, specific experiments had to be prioritised, potentially leaving unexplored avenues for improving the detection system's performance and robustness.

Despite these limitations and constraints, this Thesis has made significant progress in developing an animal detection system, and the insights gained from addressing these challenges can provide valuable guidance for the support system being created for the work of ecologists studying biodiversity.

6.3 RECOMMENDATIONS FOR FUTURE WORK

Building upon the research and insights gained from this Thesis, several key directions for future work emerge, with the aim of enhancing the accuracy and robustness of the detection system and ensuring its seamless integration within the SAFArI project.

Firstly, it is imperative to invest in advanced hardware and computing resources to support the scaling up of testing and improvement. This not only accelerates processing but also enables the accommodation of more complex DL models, thereby facilitating a comprehensive exploration of model configurations to improve system accuracy and efficiency.

Also, further investigations should explore alternative strategies for integrating BS with Faster R-CNN to improve its performance. These explorations may involve adapting the BS technique, incorporating additional pre-processing steps, and experimenting with different

hyperparameters of the Faster R-CNN model. Moreover, it could be valuable to experiment with the RetinaNet model, which is an evolution of Faster R-CNN (Kellenberger et al., 2020). RetinaNet is promising, especially when considering that the literature review also recommended YOLOv4, and the latest version, YOLOv8, outperformed its predecessors, leading to the belief that similar advancements may be achieved with the RetinaNet model.

Additionally, to enhance generalisation and mitigate overfitting, it is crucial to intensify efforts aimed at acquiring a more extensive collection of labelled images, with a specific emphasis on diversifying the dataset and increasing the number of images for species-level separation.

While this Thesis primarily focuses on the detection phase of the SAFArI architecture, future work should involve the completion of subsequent phases, including the identification of animal species, counting individuals by species, characterising their behaviour, and detecting behavioural changes. By doing so, the system can contribute to a more comprehensive understanding of wildlife populations and ecological dynamics.

Furthermore, the Modifications Section has highlighted the potential for merging the animal detection and species classification phases, particularly when a sufficient number of images are available for the classification task. Such integration has the potential to simplify system performance by combining these two tasks, leading to a more efficient execution of the SAFArI project.

Lastly, future research should prioritise testing the system on more realistic datasets, which should incorporate a more extensive collection of labelled images, including a larger number of negative samples relative to positive samples. This alignment with the real-world distribution of animal instances will provide valuable insights into the system's robustness and practical applicability.

In conclusion, the future work recommendations outlined above represent the next crucial steps in advancing the animal detection system within the SAFArI project. Implementing these strategies will lead to the completion of the ML layer of the project, thereby enabling the identification of animals within images, species categorization, individual counting by species, behavioural characterization, and behavioural change detection in CT images. Moreover, it serves as a comprehensive roadmap for guiding future research and development efforts within the SAFArI project.

REFERENCES

BIBLIOGRAPHICAL REFERENCES

Ahmed, A., Yousif, H., Kays, R., & He, Z. (2019). Semantic region of interest and species classification in the deep neural network feature domain. Ecological Informatics, 52, 57–68. https://doi.org/10.1016/j.ecoinf.2019.05.006

Banerjee, Anoushka, et al. “Sieving Camera Trap Sequences in the Wild.” Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods, 2022, pp. 470–479, 10.5220/0010919000003122.

BIOTA (2023). [Camera Trap Data Set] [Unpublished raw data].

Burton, A. C., Neilson, E., Moreira, D., Ladle, A., Steenweg, R., Fisher, J. T., Bayne, E., & Boutin, S. (2015). REVIEW: Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes. Journal of Applied Ecology, 52(3), 675–685. https://doi.org/10.1111/1365-2664.12432

Conway, A. M., Durbach, I. N., McInnes, A., & Harris, R. N. (2021). Frame-by-frame annotation of video recordings using deep neural networks. Ecosphere, 12(3). https://doi.org/10.1002/ecs2.3384

Cunha, F., Eulanda, S., Barreto, R., & Colonna, J. G. (2021). Filtering Empty Camera Trap Images in Embedded Systems. Thecvf.com, 2438–2446. https://openaccess.thecvf.com/content/CVPR2021W/MAI/html/Cunha_Filtering_Empty_Camera_Trap_Images_in_Embedded_Systems_CVPRW_2021_paper.html

Dhillon, A., & Verma, G. K. (2019). Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, 9(2), 85–112. https://doi.org/10.1007/s13748-019-00203-0

Glover-Kapfer, P., Soto-Navarro, C. A., & Wearn, O. R. (2019). Camera-trapping version 3.0: current constraints and future priorities for development. Remote Sensing in Ecology and Conservation, 5(3), 209–223. https://doi.org/10.1002/rse2.106

Islam, S. B., & Valles, D. (2020). Identification of Wild Species in Texas from Camera-trap Images using Deep Neural Network for Conservation Monitoring. 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). https://doi.org/10.1109/ccwc47524.2020.9031190

Kellenberger, B., Marcos, D., & Tuia, D. (2019). Best Practices to Train Deep Models on Imbalanced Datasets—A Case Study on Animal Detection in Aerial Imagery. Machine Learning and Knowledge Discovery in Databases, 630–634. https://doi.org/10.1007/978-3-030-10997-4_40

Kellenberger, B., Tuia, D., & Morris, D. (2020). AIDE: Accelerating image-based ecological surveys with interactive machine learning. Methods in Ecology and Evolution, 11(12), 1716–1727. https://doi.org/10.1111/2041-210x.13489

Lee, J., Lim, K., & Cho, J. (2022). Improved Monitoring of Wildlife Invasion through Data Augmentation by Extract–Append of a Segmented Entity. Sensors, 22(19), 7383. https://doi.org/10.3390/s22197383

Levy, Yair, and Timothy J. Ellis. "A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research." Informing Science: The International Journal of an Emerging Transdiscipline, vol. 9, no. 1, 2006, pp. 181–212, 10.28945/479.

Meena, S. D., & Agilandeeswari, L. (2019). Stacked Convolutional Autoencoder for Detecting Animal Images in Cluttered Scenes with a Novel Feature Extraction Framework. Advances in Intelligent Systems and Computing, 513–522. https://doi.org/10.1007/978-981-15-0184-5_44

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. Journal of Clinical Epidemiology, 62(10), 1006–1012. https://doi.org/10.1016/j.jclinepi.2009.06.005

Norouzzadeh, M. S., Morris, D., Beery, S., Joshi, N., Jojic, N., & Clune, J. (2020). A deep active learning system for species identification and counting in camera trap images. Methods in Ecology and Evolution, 12(1), 150–161. https://doi.org/10.1111/2041-210x.13504

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, 115(25), E5716–E5725. https://doi.org/10.1073/pnas.1719367115

Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., & McGuinness, L. A. (2021). PRISMA 2020 Explanation and elaboration: Updated Guidance and Exemplars for Reporting Systematic Reviews. BMJ, 372(160), n160. https://doi.org/10.1136/bmj.n160

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/tpami.2016.2577031

Riechmann, M., Gardiner, R., Waddington, K., Rueger, R., Leymarie, F. F., & Rueger, S. (2022). Motion vectors and deep neural networks for video camera traps. Ecological Informatics, 69, 101657. https://doi.org/10.1016/j.ecoinf.2022.101657

Rovero, F., Zimmermann, F., Berzi, D., & Meek, P. (2013). "Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications. Semantic Scholar. https://doi.org/10.4404/HYSTRIX-24.2-6316

Santangeli, A., Chen, Y., Boorman, M., Sales Ligero, S. and Albert García, G. (2022), Semi-automated detection of tagged animals from camera trap images using artificial intelligence. Ibis, 164: 1123-1131. https://doi.org/10.1111/ibi.13099

Shearer, C., Moss, L., Adelman, S., Herdlein, S. A., Fong, J., Wong, H. K., & Fong, A. (2000). The CRISP-DM Model: The New Blueprint for Data Mining E-Business and the New Demands on Data Warehousing Technology: The New Demands E-Commerce Places on Data Warehousing Technology Katherine Hammer Turning the Corner from Data Warehousing to Electronic C. JOURNAL, 5. https://mineracaodedados.files.wordpress.com/2012/04/the-crisp-dm-model-the-new-blueprint-for-data-mining-shearer-colin.pdf

Swanson, Alexandra, et al. "Snapshot Serengeti, High-Frequency Annotated Camera Trap Images of 40 Mammalian Species in an African Savanna." Scientific Data, vol. 2, no. 1, 9 June 2015, 10.1038/sdata.2015.26.

Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., Vercauteren, K. C., Snow, N. P., Halseth, J. M., Di Salvo, P. A., Lewis, J. S., White, M. D., Teton, B., Beasley, J. C., Schlichting, P. E., Boughton, R. K., Wight, B., Newkirk, E. S., Ivan, J. S., Odell, E. A., Brook, R. K., & Lukacs, P. M. (2018). Machine learning to classify animal species in camera trap images: Applications in ecology. Methods in Ecology and Evolution, 10(4), 585–590. https://doi.org/10.1111/2041-210x.13120

Tranfield, David, et al. "Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review." British Journal of Management, vol. 14, no. 3, Sept. 2003, pp. 207–222, onlinelibrary.wiley.com/doi/epdf/10.1111/1467-8551.00375?src=getftr, 10.1111/1467-8551.00375. Accessed 14 Dec. 2022.

Ukwuoma, C. C., Qin, Z., Yussif, S. B., Happy, M. N., Nneji, G. U., Urama, G. C., Ukwuoma, C. D., Darkwa, N. B., & Agobah, H. (2022). Animal species detection and classification framework based on modified multi-scale attention mechanism and feature pyramid network. Scientific African, 16, e01151. https://doi.org/10.1016/j.sciaf.2022.e01151

ULAŞ TEKELİ, & YALIN BAŞTANLAR. (2019). Elimination of useless images from raw camera-trap data. TÜBİTAK Academic Journals. https://journals.tubitak.gov.tr/elektrik/vol27/iss4/2/

Wei, W., Luo, G., Ran, J., & Li, J. (2020). Zilong: A tool to identify empty images in camera-trap data. Ecological Informatics, 55, 101021. https://doi.org/10.1016/j.ecoinf.2019.101021

Willi, M., Pitman, R. T., Cardoso, A. W., Locke, C., Swanson, A., Boyer, A., Veldthuis, M., & Fortson, L. (2018). Identifying animal species in camera trap images using deep learning and citizen science. Methods in Ecology and Evolution, 10(1), 80–91. https://doi.org/10.1111/2041-210x.13099

Xi, T., Wang, J., Qiao, H., Lin, C., & Ji, L. (2021). Image Filtering and Labelling Assistant (IFLA): Expediting the analysis of data obtained from camera traps. Ecological Informatics, 64, 101355. https://doi.org/10.1016/j.ecoinf.2021.101355

Yang, D., Ren, G., Tan, K., Huang, Z., Li, D., Li, X., Wang, J., Chen, B., & Xiao, W. (2021). An Adaptive Automatic Approach to Filtering Empty Images from Camera Traps Using a Deep Learning Model. Wildlife Society Bulletin, 45(2), 230–236. https://doi.org/10.1002/wsb.1176

Yang, D., Tan, K., Huang, Z., Li, X., Chen, B., Ren, G., & Xiao, W. (2021). An automatic method for removing empty camera trap images using ensemble learning. Ecology and Evolution, 11(12), 7591–7601. https://doi.org/10.1002/ece3.7591

Yang, D.-Q., Li, T., Liu, M.-T., Li, X.-W., & Chen, B.-H. (2021). A systematic study of the class imbalance problem: Automatically identifying empty camera trap images using convolutional neural networks. Ecological Informatics, 64, 101350. https://doi.org/10.1016/j.ecoinf.2021.101350

Yang, N., Wang, Z., & Wang, S. (2021). Computer Image Recognition Technology and Application Analysis. IOP Conference Series: Earth and Environmental Science, 769(3), 032065. https://doi.org/10.1088/1755-1315/769/3/032065

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

LIBRARIES REFERENCES

YOLOv4- AlexeyAB. (2023, May 6). GitHub - AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet). GitHub. https://github.com/AlexeyAB/darknet

YOLOv5- ultralytics. (2023, May 23). GitHub - ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite. GitHub. https://github.com/ultralytics/yolov5

YOLOv6- meituan. (2023, May 17). GitHub - meituan/YOLOv6: YOLOv6: a single-stage object detection framework dedicated to industrial applications. GitHub. https://github.com/meituan/YOLOv6

YOLOv7- WongKinYiu. (2023, March 4). GitHub - WongKinYiu/yolov7: Implementation of paper - YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. GitHub. https://github.com/WongKinYiu/yolov7

YOLOv8- ultralytics. (2023, May 25). GitHub - ultralytics/ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > CoreML > TFLite. GitHub. https://github.com/ultralytics/ultralytics

Faster R-CNN- facebookresearch. (2019). detectron2/configs/COCO-Detection at main · facebookresearch/detectron2. GitHub. https://github.com/facebookresearch/detectron2/tree/main/configs/COCO-Detection

Background Subtraction- OpenCV: How to Use Background Subtraction Methods. (2023). Opencv.org. https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa