# Hierarchical Ensemble of Open-Source Models with Meta-Learning for Wildlife Classification in Camera Trap Images

**Computing MSc Dissertation**

**Supervisor: Dr. Charith Perera**

Author: Fan Zhou

C24016579


School of Computer Science and Informatics, Cardiff University

8th Sept 2025

## Abstract

Camera traps have revolutionised wildlife monitoring methods in ecological research and conservation, as these devices offer a non-invasive way to automatically collect wildlife images, providing abundant research material for ecologists. However, the vast amount of image data these cameras generated over time has revealed increasing challenges in manual annotation labour and a high rate of false positives caused by environmental triggers, highlighting the need for automated processing tools. The currently available open-source classification models demonstrate strong performance in isolated contexts, but their generalisation ability across diverse environments is still unsatisfactory. This thesis proposes an innovative hierarchical ensemble framework that integrates seven locally deployable open-source models through ensemble learning and meta-model stacking, thereby enhancing species classification performance and expanding the scope of species support under various environmental conditions.

This research process is organised into three phases: (1) A comprehensive evaluation of individual models based on their predictions on ten heterogeneous datasets, providing valuable insights into their objective detection and fine-grained species classification performance. (2) Ensemble learning phase integrates initial predictions from seven baseline models, using accuracy-weighted averaging and dynamic species-specific weighted voting separately for effective performance optimisation. (3) Meta-learning stage introduces the potential of machine learning models (e.g., XGBoost, Random Forest, Logistic Regression) on capturing complex relationships for classification performance optimisation, through stratified 70/15/15 data splits and hyperparameter tuning to train a more robust model for addressing practical camera trap challenges.

The project was conducted within Jupyter notebooks, emphasising a repeatable wildlife classification solution. Results demonstrate that the proposed hierarchical ensemble framework achieves a 25% reduction in false positives and an effective improvement in species-level performance compared to baselines across multi-geographic datasets. This novel framework aims to provide a reproducible and effective solution for scalable biodiversity monitoring, thereby reducing the overwhelming manual workload associated with the camera trap problem.

# Content

# List of Figures

# List of Table

## List of Abbreviations

AUROC: operating characteristic curve

AUPRC: area under the precision-recall curve

CNNs: Convolutional Neural Networks

CycleGANs: Cycle-Consistent Adversarial Networks

FDA: flexible discriminant analysis

FSL: Few-shot Learning

ResNet: residual neural network

XGBoost: extreme gradient boosting

OOM: Out-of-Memory

# Chapter 1 – Introduction

## 1.1 Background of the Research

Camera traps have revolutionised the approach to traditional wildlife monitoring for ecological research and conservation, as they provide a non-invasive, automated method for collecting wildlife images, contributing to further research on species distribution, behaviour, and population dynamics. These cameras have been widely deployed in remote animal habitats, constantly providing valuable research insights through their automatic motion-triggered mechanism. However, this feature also brought new challenges for biologists–an exponential growth in image data with the increase in cameras and their deployment time (Norouzzadeh et al., 2021). At the same time, a high rate of false positives exacerbates this problem, as these motion-triggered cameras are easily activated by unexpected situations, such as swaying vegetation, lighting variations, and wind (Schneider et al., 2020). These challenges have brought immense pressure on researchers to perform labour-intensive manual annotation and processing. However, the low efficiency of traditional manual processing has become a significant bottleneck, severely limiting research progress and delaying wildlife conservation efforts.

The development of deep learning has brought a powerful solution for camera trap challenges, especially with the emergence of convolutional neural networks (CNNs), which have demonstrated their superiority in automatic wildlife image classification and empty image filtering (Norouzzadeh et al., 2018). CNN architecture lays the foundation for a broad application of deep learning models in the wildlife identification field, such as MegaDetector (a YOLO-based detector for animals, humans, and vehicles), MLWIC2 (a machine learning tool in R for wildlife image classification suitable for edge-cut devices), DeepFaune (trained mainly for European fauna), SpeciesNet (a global species classifier supporting over 2000 species identification), AddaxAI (region-specific classifiers for areas like Namibia and New Zealand), and Pytorch-Wildlife (a flexible ensemble platform for camera traps challenges). These models were pretrained based on different datasets, offering researchers an open-source, locally deployed automated classification option to address camera trap challenges. However, although they claimed excellence in their in-sample testing, their generalisation ability across diverse environments, rare

species, or domain shifts in lighting remains under suspicion (Vélez et al., 2023), which still brought challenges for ecologists to use these models in their research around the world.

Based on previous research, some articles indicate that ensemble learning strategies, such as voting, averaging, and stacking, can enhance robustness by integrating multiple base models, further improving identification performance and reducing false positives on heterogeneous data (Huettmann et al., 2018; Yang et al., 2021). For instance, a hybrid ensemble framework can outperform any base models by combining their advantages in both objective detection and classification aspects, achieving more robust performance while processing a broader range of geographical features or species. Based on recent advancements, meta-learning, particularly through stacking with algorithms like XGBoost or Random Forests, further refines ensembles by learning from meta-features such as model confidences and agreement rates.

## 1.2 Problem Statement

Despite the deep learning technique having achieved remarkable progress in AI-driven camera trap analysis, single models often underperform in real-world applications due to poor generalisation across varying environmental conditions, ultimately leading to high false-positive rates (e.g., misclassifying empty images) and reduced accuracy for species-level identification.

Ensemble approaches have mitigated some limitations, but few integrate multiple open-source models hierarchically with meta-learning to optimise for both binary detection (animal vs. non-animal) and fine-grained species classification. Challenges such as class imbalances, rare species detection, and background interference exacerbate these issues, particularly in heterogeneous datasets from multi-geographic regions. Although simple ensemble approaches have mitigated some limitations in detection and classification tasks, the practice of using more complex meta-learning to integrate multiple open-source models hierarchically for enhanced detection (animal/non-animal) and fine-grained species classification remains rare. Furthermore, existing studies overlook the potential of meta-features, such as entropy and inter-model consistency, in handling rare classes and ensuring stability in conservation monitoring. This gap hinders ecologists from working effectively and

reproducibly, requiring manual effort and continuing to slow the generation of actionable knowledge for biodiversity conservation.

## 1.3 Research Objectives

This thesis aims to develop and evaluate an innovative hierarchical ensemble framework, which can further optimise the robustness and generalisation of mainstream classification open-source models by combining weighted averaging and meta-learning, contributing to a superior classification solution for improved wildlife identification in practical camera trap applications for biologists. The primary research objectives in this project include:

- An assessment of seven mainstream open-source models (e.g., PyTorch-Wildlife, MLWIC2, DeepFaune, SpeciesNet, AddaxAI) was conducted on ten diverse datasets from LILA BC, using comprehensive metrics such as accuracy, F1-score, and confusion matrices. This phrase emphasises binary detection and species-level performance, offering a detailed insight into the base models' performance under varied environmental conditions.
- A hybrid ensemble learning approach was implemented in this stage, which uses weighted averaging for the binary detection process and dynamic weighted voting for finer species classification, further optimising the ensemble effectiveness through grid search and threshold optimisation.
- Constructing a meta-model stacking phase using higher-level classifiers (XGBoost, Random Forest, Logistic Regression) trained on meta-features like confidences, agreement rates, entropy, and weight interactions, with stratified data splits and hyperparameter tuning to achieve up to 25% reduction in false positives and 15-20% gains in species accuracy.
- Providing an open-source, reproducible code for this hierarchical ensemble framework, and fully verifying its effectiveness based on diverse open-source camera trap datasets, ensuring it contributes to biodiversity monitoring and reduces manual annotation burdens.

## 1.4 Dissertation Structure

This chapter aims to describe the structure of this thesis, which contains six chapters overall, and each one has its own function as shown below:

- **Chapter 2 - Literature Review:** This chapter focuses on a comprehensive review of deep learning progress in camera trap challenges, with a clear logical structure: (1) The research background of deep learning in camera traps and core concepts and development in this field. (2) Key problems in current research. (3) Existing solutions with their merits and demerits. (4) The limitations of existing research and (5) the innovative contribution of this study.
- **Chapter 3 - Methodology:** This chapter will detail the experimental approaches employed in this hierarchical ensemble framework. It includes the initial dataset preparation and image processing by seven base models, the design of a two-stage ensemble strategy, and the meta-learning stacking process. A comprehensive evaluation of the ensemble result will be conducted, with useful metrics and evaluation figures.
- **Chapter 4 - Results:** This chapter presents all experimental data, accompanied by appropriate figures and tables, to provide a straightforward understanding of the advantages of the improvement strategy.
- **Chapter 5 - Discussion and Critical Analysis:** This chapter provides a detailed explanation of the experimental data, incorporating critical thinking skills. The results will be presented using standard metrics, confusion matrices, and visualisations to provide a clear and thorough assessment.
- **Chapter 6 - Conclusion & Future Work:** This final chapter will summarise all useful findings and contributions in this project. Moreover, the limitations of current research and potential suggestions for future research will be discussed, helping the progress of wildlife classification.

# Chapter 2 - Literature Review

This chapter presents existing research on deep learning for wildlife classification. It will begin with an overview of the field's background, providing a detailed research review and basic concept instruction. Then, the focus will be on the current research regarding the core challenges in camera trap problems, which helps to deepen the understanding of existing issues and specific research progress. The existing classification solutions will be discussed in the next section, along with their advantages and weaknesses. Finally, the report will identify the research gap from all existing literature mentioned, providing an assessment of the effectiveness of this research.

## 2.1 Overview of Deep Learning Technology in Camera Trap Challenges

The development of camera traps has revolutionised the way we study and monitor animals in the wild, as it offers a non-invasive and relatively inexpensive method (Norouzzadeh et al., 2018). Due to the feature of these motion-activated cameras, they enable a long-term, automated-triggered wildlife image collection, contributing to lots of valuable data for ecological research and conservation. However, the success of this method also created new challenges, as the massive volume of images generated by these devices brought immense pressure on manually processing them (Beery et al., 2019; Tabak et al., 2019). For instance, one camera trap device can generate over a million images annually, and the need to detect objects within these images using a bounding box is necessary, laying the foundation for species classification and assigning taxonomic labels to detected objects in the subsequent step. This labour-intensive process and the low efficiency of manually processing these images make it impossible, which has become a significant obstacle for ecologists in the conservation of rare species or obtaining timely insights (Willi et al., 2019).

Besides the large volume, the high false trigger rate is another significant challenge that camera traps have brought (Whytock et al, 2021). Due to the characteristics of these motion-activated cameras, they often trigger due to unexpected natural reasons, such as wind, swaying vegetation, or illumination changes, causing a massive percentage of empty images (Schneider et al., 2020). An increasing workload to filter out valuable wildlife images among these useless, empty frames

has been a struggle for ecologists for a long time, hindering decision-making for rare species conservation. Therefore, the necessity of filtering out empty frames, along with the need for automated wildlife image classification, drove the development of related technology.

The emergence of deep learning, particularly convolutional neural networks (CNNs), offers a powerful solution to overcome the camera trap bottleneck. The first innovative application of deep learning technology in wildlife classification was pioneered by AlexNet, a CNN-based model, which demonstrated the potential of CNN architectures in wildlife image identification (Alzubaidi et al., 2021). By the mid-2010s, more practical models, such as YOLO (You Only Look Once) and Faster R-CNN, enabled the detection of real-time objects within images, demonstrating their potential for more complex ecological classification applications. The advancement of CNN architectures has accelerated since 2018, as the release of large datasets such as Snapshot Serengeti (over 3 million images) provides a massive amount of training data for pretraining and optimising CNN models. Following this trend, the next landmark progress in this field is the research by Norouzzadeh et al. (2018), who achieved a remarkable classification accuracy rate of up to 93.8% for 48 species, depending on a deep neural network trained on 3.2 million images. (Norouzzadeh et al., 2018) This unbelievable achievement is equivalent to 8.4 years of manual annotation time, highlighting the considerable potential of deep learning in the camera traps field.

## 2.2 Key Challenges in Deep Learning for Wildlife Classification

### 2.2.1 Overwhelming Data Volume and Manual Annotation Inefficiencies

A primary challenge in the wildlife classification field, as proposed by Norouzzadeh et al. (2018), remains the overwhelming volume of images generated by camera traps. This issue not only increased pressure on the manual process but also raised the possibility of human error in annotation. However, deep learning techniques have proven to achieve significant improvement in automated identification of common species, with an accuracy of over 93.8%. The need for manual annotation and checking when preparing the training data for the pre-trained model still persists, and this bottleneck remains currently irreplaceable, making it still an expensive and time-consuming manual task. This viewpoint was further advanced by Tabak et al. (2019),

as they found that this intensive labour effort still cannot be replaced, even with semi-automated tools like DigiKam or Agouti. Another review by Vélez et al. (2023) represented that the processing of camera trap images, including visual identification, labelling, and metadata extraction, is described as a 'bottleneck that substantially adds up to the cost of the method and results in a considerable delay in data availability. More research has supported that this fatal delay is worse in long-term monitoring, where delays hinder timely conservation efforts, such as in Snapshot Serengeti, where crowdsourcing is a partial but still human-intensive task (Willi et al, 2019).

### 2.2.2. Class Imbalance and Difficulties with Rare or Cryptic Species

Class imbalance has become another significant problem that ecologists face during research, as Schneider et al. (2019) have claimed, the rare appearance of uncommon species in the camera trap images made them harder to identify due to the lack of training data. Whytock et al. (2021) further advanced this viewpoint, who noticed that models pretrained with imbalanced datasets easily struggle with misclassification, which has serious consequences, such as delaying the protection of animals. In a research by Fennell et al. (2024) further confirmed the idea that the loss of rare or cryptic species from a dataset creates a broader issue of class imbalance, which is a common problem in camera trap datasets and can distort the detection of rarer species. All these ideas verified the influence of imbalanced classes for fine-grained classification, proving the significance of species-specific densifying exploration.

### 2.2.3 Background Influence and Lack of Generalisation Across Environments

As Beery et al. (2019) found, CNNs tend to heavily rely on background information for their classification predictions, as these models are usually only effective when processing images from similar camera positions. Another study also verified this point: the influence of extraneous information, such as trees or familiar environments, became the primary decision basis for some CNN models in detecting or classifying, which led to poor generalisation in new environments. This proves the significance of excellent generalisation ability for pre-trained models for practical uses in reality. The research on the limitations of geographic bias further advanced

this point, as Ferreira et al. (2023) found that the geographic biases in datasets hindered the applicability in new ecosystems.

### 2.2.4. The Challenge of Environmental Influence in Wildlife Classification

The influence of environmental variations in wildlife classification challenge was first emphasised in a study by Beery et al. (2019), as they pointed out the significant influence of some natural reasons, such as the distance of the animal, variations in lighting, shadows, and weather conditions, which improved the classification challenge and caused a high possibility of false positives or false negatives. In a following study, it was supported that small mammals less than 5 kg were hardly captured and detected by camera traps, this problem particularly occurs during the night. Mulero-Pázmány et al. (2025) further confirmed the effect of night on wildlife capture in camera traps and explored the difficulty of distinguishing between wolves, foxes, and dogs, which is inherently challenging, highlighting the effect of natural reasons in an increased false positive issue.

### 2.2.5 False Positives/Negatives and Evaluation Metric Disconnects

False positives have been a significant challenge for many ecological researchers for an extended period, as the imbalance between animal and empty images significantly exacerbates the difficulty and pressure of classification (Beery et al., 2019). This problem advanced the development of a more robust detector for handling the massive number of false positives in the camera traps.

## 2.3 Existing Research Achievements in Camera Trap Challenges

Camera traps have become a key point for non-invasive wildlife monitoring and ecological research. However, the large volume of images they produce, along with a high rate of false positives and empty frames, also poses challenges for ecologists. The section reviews recent advancements in deep learning models aimed at improving robustness and generalisation, including optimised single models, ensemble strategies, and meta-learning approaches. These developments have demonstrated excellent performance in addressing issues such as false positives, generalisation, and the detection of rare species, although some limitations still exist.

### 2.3.1 Advancements of Single Deep Learning Models

Deep learning techniques have achieved incremental improvements in automated wildlife image processing, leading to a broad application of CNN architecture for object detection and species identification step by step. A pioneering work by Norouzzadeh et al. (2018) effectively demonstrated the capabilities of CNNs in wildlife image identification, achieving 93.8% top-1 accuracy in species identification across 48 species.

Tabak et al. (2019) promoted deep learning techniques by practising transfer learning from pre-trained CNNs, achieving 97% validation accuracy for 58 species, which proves that CNN's migratability feature can be used for model optimisation. By 2020, the tailored deep learning model, such as YOLO-based MegaDetector (Beery et al., 2019), focused on object detection (e.g., humans and vehicles), as well as animal identification, with over 95% accuracy in motion-triggered images, laying the foundation for a two-stage ensemble classification system, such as PyTorch-Wildlife. However, a comparison test conducted by Leorna and Brinkman (2022) confirmed that MegaDetector's prediction accuracy was ≥ 94.6% but also pinpointed an instinct performance degeneration (≤61.6%) due to object size or distance. DeepFaune (Rigoudy et al., 2023), as a non-code API platform, was tailored for 26 European taxa, demonstrating outstanding performance during testing; however, its poor generalisation ability for specific species was also noted, showing the potential weakness of the single model.

The latest update is PyTorch-Wildlife (Miao et al., 2024) from Microsoft, which offers a more flexible option for various detector and classifier models, proving its strengths on two open-source datasets with excellent performance (92% on Amazonian and 98% on Galápagos invasives). All of these advancements demonstrated their efforts in reducing false positives and improving classification accuracy, although some inadequacies of the single model still exist.

### 2.3.2 Ensemble Learning: An Effective Strategy for Enhancing Model Performance

During the past decade, Ensemble learning has marked significant progress in deep learning for enhanced wildlife classification, as it proves an effective improvement in reducing high false positive rates from empty frames and poor generalisation across

environments. Ensemble learning is especially beneficial for improving robustness in wildlife classification, as it integrates predictions from multiple models, contributing to an optimisation of classification accuracy. This idea was first innovated by Norouzzadeh et al. (2018), who used an ensemble of nine CNN architectures (including ResNet variants) on the Snapshot Serengeti dataset, and attained an accuracy of 93.8 % to recognise 48 species, count animals, and describe behaviour.

On this basis, Yang et al. (2021) proposed a multilevel ensemble system to filter out empty frames by combining several powerful base models, such as AlexNet, Inception, and ResNet, thereby successfully achieving removal rates of up to 77.51% for empty images with low omission errors (2.54% at the image level). The next practice was conducted in 2022, Wyatt et al. (2021) used ensembles for benthic images and demonstrated that an ensemble outperforms a single CNN by 5-8% in F1 score on imbalanced classes, when combining deep learning decisions by averaging for improved calibration and robustness.

Hierarchical and active ensembles further refine progress. In recent studies, the hypothesis of the hierarchical ensemble has been verified as effective and valuable. Miao et al. (2024) produced an active ensemble classifier that filters out animal sequences by incorporating several detectors and is effective in discriminating true positives. Similarly, Li et al. (2023) proposed an ensemble model for separating wildlife images from those of human activity, achieving 95.75% accuracy and 94.07% recall using a conservative strategy, which resulted in an 80% reduction in personnel costs.

## 2.4 Limitations of Existing Research

Despite the development of deep learning, recognising a variety of practical applications and extensions in wildlife identification challenges, a research gap still exists for a robust ensemble framework to integrate open-source models. In this chapter, this research gap will be located based on existing research evidence, proving the effectiveness of this thesis.

### 2.4.1 Poor Generalisation Across Diverse Environments

Based on existing literature, it has been demonstrated that there is a significant variation in the generalisation ability of many pre-trained models when processing data from new ecological settings. This problem has been indicated in the research of Curry et al. (2021), which shows it is challenging to use a model pre-trained on captive imagery to handle wild camera trap imagery, as its high classification accuracy cannot be transferred to different environments. This viewpoint highlighted the need for strong classification models which cover more available species for practical wildlife classification applications.

### 2.4.2 High Rates of False Positives and Handling Empty Frames

High false positives triggered by non-animal factors were a constantly influencing challenge for ecologists in wildlife monitoring research. A variety of existing literature has shown that camera traps often capture empty images due to some unexpected reasons, such as swaying vegetation, wind, and other factors (Schneider et al., 2020). This problem seriously caused massive invalid image collections, further hindering the ecological research and the conservation of rare species on time. a study by Mulero-Pázmány et al. (2025) also highlighted the influence of different backgrounds, and they proposed that the pre-training model easily misclassifies or misdetects species under diverse geographic conditions, further leading to high false positives. These challenges emphasise the need for better classification accuracy and efforts to reduce false positives, demonstrating the necessity for improved binary detection accuracy to filter out unnecessary empty frames.

### 2.4.3 Data Imbalances, Sparsity, and Quality Challenges

Class imbalance has consistently challenged wildlife classification, as rare species are rarely captured by cameras, which further complicates their accurate classification due to the limited number of samples. Mulero-Pázmány et al. (2025) highlighted that camera traps often struggle to identify rare species, as these classes are more prone to misclassification owing to scarce training images, leading to poor performance (0.18–0.32 F1-Score) for rare classes, while common ones achieve (0.87–0.95). This underscores the vital need to support more species or obtain greater labelled ecological data of rare species for improved training.

## 2.5 Research Gaps and Value of the Proposed Study

Based on the previous reviews of existing literature, these limitations reveal a research gap in a practical hierarchical ensemble approach for enhanced generalisation and robustness. While existing studies have proved the optimisation ability of ensemble strategy through combining multiple models, which achieved superior performance than single CNNs, like the research of Norouzzadeh et al. (2018) and Wyatt et al. (2021). Few were the emphasis on integrating mainstream open-source models hierarchically with more advanced meta-learning like stacking via XGBoost. Meng et al. (2023) have verified the effectiveness of ensemble strategies on five advanced deep learning models, which contribute to excellent improvement for wildlife classification accuracy. This literature supports the idea of the innovative ensemble strategy, with a greater emphasis on integrating open-source models available for locally deployed systems, thereby demonstrating greater innovativeness. This proposed framework advances previous research by testing seven open-source models on ten datasets, with a further optimisation through weighted voting and meta-stacking. This research will further promote research on reducing false positives and improving species classification performance and scope in diverse geographic backgrounds.

# Chapter 3 – Methodology

This thesis presents an innovative hierarchical ensemble framework that enhances the optimisation of classification decision-making by leveraging multiple powerful open-source models for processing camera trap images. This design contributes explicitly to reducing false positives and enhancing species-level identification accuracy. The chapter systematically presents the methodology of the proposed hierarchical ensemble framework, which is organised into four code phases: (1) Data Preparation and Baseline Model Evaluation, (2) Weighted Ensemble Learning, (3) Meta-Model Stacking, and (4) Performance Benchmarking. Following this sequence, this section will then detail the specific techniques and procedures adopted in each phase.

## 3.1 Phase I: Data Preparation and Baseline Model Evaluation

The primary goal in this phase is to develop a high-quality, standardised evaluation dataset that encompasses various environmental differences, followed by a thorough assessment of all baselines of pre-trained deep learning models. This work establishes a foundational benchmark for future ensemble efforts, highlighting the valuable review of the real-world performance of mainstream open-source models in practical use.

### 3.1.1 Dataset Acquisition and Preprocessing

In this study, we combine ten diverse open-source camera trap datasets from the LILA BC repository, which encompass a variety of environmental variations, thereby increasing the challenge for models' generalisation ability. Moreover, due to a compromise between limited memory capacity and the enormous size of some datasets, we adopted the following sampling and preprocessing steps to ensure the feasibility and efficiency of the experiments.

- **Wildlife dataset resources:**
    - Caltech Camera Traps (data from 140 camera locations in the Southwestern United States)
    - North American Camera Trap Images (images from five locations across the United States)
    - Snapshot Serengeti (camera trap data from Africa)

    - WCS Camera Traps (675 species from 12 countries)
    - Idaho Camera Traps (1.5 million camera trap images from Idaho)
    - Island Conservation Camera Traps (images from 123 camera locations from 7 islands in 6 countries)
    - Wellington Camera Traps (contains 270,450 images from 187 camera locations in Wellington, New Zealand)
    - ENA24-detection (approximately 10,000 camera trap images representing 23 classes from Eastern North America)
    - Channel Islands Camera Traps (73 camera locations in the Channel Islands)
    - SWG Camera Traps 2018-2020 (436,617 sequences of camera trap images from 982 locations in Vietnam and Lao)
- **Stratified Sampling Strategy:** The evaluation datasets we used in this research are from the LILA BC repository (stored in Google Cloud Storage). For each dataset, I adopted a random sample of 100,000 images to ensure as many species as possible are covered in our test. Then, we kept 10,000 images per dataset for the final test, using stratified sampling with prioritised retention of rare species. This method attempts to address the long-tail distribution issue and ensure the representativeness of the evaluation results.
- **Label Cleaning and Standardisation:** To ensure the correct use of the official dataset annotation files, we developed scripts to convert the original JSON files into a standardised CSV format, facilitating further file reading and analysis. Due to various species naming conventions, a ground truth table was created to standardise all species' taxonomic names from Wikipedia, enabling a reliable connection and verification between the model's identification prediction and dataset annotations.

### 3.1.2 Individual Model Execution and Prediction Aggregation

A comprehensive assessment was conducted in this step across seven open-source models in a separate Jupyter Notebook, including PyTorch-Wildlife (MegaDetector v5 and v6), MLWIC2, DeepFaune, SpeciesNet, and AddaxAI. The detailed execution process is outlined as follows.

- **Model Environment Setup and Execution:** This process was prepared and run in individual Jupyter Notebooks for different environment preferences, as

there was substantial variance in the specifications of the environments. This approach should be more flexible to adapt to changes for any given model and potentially also more reproducible for future reuse.

- **Batch Processing Optimisation:** In this step, each model needs to batch classify 10,000 curated images per dataset and run it repeatedly across 10 datasets. This exerts a series of pressures on the GPU or memory, such as disk space overflow, I/O bottlenecks, and OOM risks. To mitigate these potential risks, we adopted a temporary batch processing strategy with a dynamic batch size adjustment, which potentially prevents GPU memory overflow. All temporary batch directories were released after each cycle to avoid excessive disk usage during large-scale processing.
- **Data Merging and Validation:** The prediction results from all seven baseline models are merged into a new CSV document for a comprehensive validation in the next step. Moreover, a ground truth table is created, which contains detailed taxonomic information of all species covered in each model and dataset. Each prediction result among models is connected with a ground truth table separately, which was used to standardise all species names with detailed taxonomic information. Then, a binary verification check was further adopted by comparing the results with official annotation files for each dataset.
- **Data Normalisation and Missing Value Handling:** After data integration, more statistical indicators (e.g., p_animal: The model's confidence, from 0 to 1, indicates whether an animal is in the image: 1 means complete confidence that it has an animal, 0 implies no animal) and the solution of unprovided detection confidence across specific models were required. We addressed missing detection confidence by filling in classification confidence instead and then calculated the p-animal through logical judgment based on classification confidence.

### 3.1.3 Performance Evaluation of Baseline Models

After completing all previous preparations, we conducted a comprehensive baseline evaluation of each individual model's performance, utilising appropriate graphs and statistical metrics.

- **Evaluation Metrics:** To comprehensively assess the classification of all models, several dimensions are evaluated using appropriate metrics, including accuracy, precision, recall, F1-score, and Cohen's Kappa. These statistical indicators contribute to showing each model's performance in binary detection (animal vs. non-animal) and species-level classification, providing comprehensive insights into its performance on real-life wildlife datasets.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

*Ps: TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives*

- **Species Confusion Matrix Analysis:** For a deeper understanding of the models' error patterns, we used a species confusion matrix to examine their inadequate capability in both open-set and closed-set scenarios, identifying the shortcomings of each model in classifying out-of-sample datasets.
- **ROC Curve and Precision-Recall Curve:** These two graphs provide a comprehensive overview of species classification performance. The ROC curve emphasises displaying the balance between true positive rate and false positive rate, which contributes to helpful insights for evaluating a model's overall discriminative ability. The Precision-Recall curve is more targeted for assessing positive class performance under class imbalance, as it focuses more on the trade-off between precision and recall, offering a clearer insight into positive class performance.
- **Heatmap of Close-set Macro-F1 Performance:** The heatmap of Close-set Macro-F1 illustrates the model's classification performance across diverse species classes, highlighting their strengths and weaknesses for each species through the variance in brightness changes across each grid. This graph contributes to identifying underperforming species, supporting adjustments to

the dataset and model weights, and ensuring fair recognition of diverse animal categories.

## 3.2 Phase II: Weighted Ensemble Learning

In this phase, we adopt an improved weighted ensemble strategy based on all the classification predictions of seven baseline models, which contains several key data such as binary detection results, detection confidences, species classification answers, and classification confidence, laying the foundation for the first layers of the proposed hierarchical ensemble framework. This step will separately apply different weighted averaging strategies on both binary detection and species classification tasks, aiming to enhance classification robustness and detection accuracy by integrating predictions from several well-trained open-source models. The entire ensemble strategy aims to improve classification robustness and detection accuracy in a global environment by leveraging the strengths of each model across diverse species and various geographic scenarios.

### 3.2.1 Binary Detection Ensemble

To effectively reduce the false positives caused by empty frames and non-animal images, this step performs an ensemble on the binary detection using 740,725 records from baselines, containing both animal and non-animal data. This step is organised into several proper processes as mentioned below.

- **Performance-driven weighted averaging:** The step highlights the adjustment of weights for each model based on their binary detection accuracy, through adopting a hierarchical strategy for weight calculation to encourage high-performance models to play a leading role in the binary integration process. For instance, the well-performing model (overall accuracy ≥ 95%) will be assigned 1.5 times the weight, then the good model (≥ 85%) receives 1.2 times the weight, the medium model (maintaining its initial weight), and for models with detection accuracy lower than 70%, they will be penalised with half the weight. This method effectively increases the weight of better-performing models for a given dataset, thereby enhancing the robustness of the binary detection and ultimately filtering out more non-animal images. To determine the final weight of each model, we adopt the following formulation:

$$w_i = \text{normalize}(\alpha_i \times \text{acc}_i)$$

where:

- $w_i$ denotes the final weight of model i;
- $acc_i$ represents the consistency accuracy of model i;
- $alpha_i$ is the performance adjustment factor, chosen from {1.5, 1.2, 1.0, 0.5};
- normalise () is the normalisation function that ensures $\sum_i w_i = 1$.

- **Threshold Optimisation:**
  After aggregating all binary confidence scores, we further explored the optimal detection threshold through a grid search approach, which optimises detection accuracy based on F1-scores. This optimisation approach benefits from balancing precision and recall, which can significantly reduce false positive predictions, contributing to optimal binary detection performance.

### 3.2.2 Dynamic Weighted Voting for Species Classification

After filtering out most empty frames or non-animal images using a binary ensemble from the previous step, we presented a more complex dynamic weighted voting strategy for finer species classification. This phase emphasises 186,439 species-level labelled images across 10 datasets, covering 206 unique species. Through leveraging the advantages of each model in their well-performing species, this dynamic ensemble strategy benefits from enhanced robustness in classification performance with a broader species classification ability over any single model. This step represents a key innovation of this study, and further execution details are provided below.

- **Species Coverage Calculation:** This step involves identifying all available species for each model, thereby extending the fine species classification scope and performance by integrating the advantages of all models for each species.
- **Species-Specific Weight Calculation:** The traditional weighted averaging approach emphasises assigning a fixed weight to each model based on its overall accuracy. However, this method cannot capture the performance variation of each mode when identifying diverse species, as it overlooks the advantages of each mode for specific species. For further optimisation, we

calculated the classification accuracy of base models for each species, as these models were weighted based on their performance in specific species rather than their overall accuracy. For instance, if one model excels at identifying "wolf" over other candidates, then this model should be given a higher weight when processing the images containing "wolf". At the same time, we further optimise this through a smoothing strategy, which considers all factors between models' species-specific accuracy and their overall performance when processing rare species, thereby preventing unstable weights caused by small sample sizes.

$$weighted\_score = \sum(conf_i \times w_{model,c})$$

```
# Compute dynamic, species-specific weights based on each model's performance on each species
unique_models = df_species_only['model_name'].unique()
unique_species = df_species_only['common_name_answer'].unique()

# Initialize a nested dictionary to store weights:
# species_weights[model_name][species_name] = accuracy
species_weights = defaultdict(lambda: defaultdict(float))

for model in unique_models:
    for species in unique_species:
        # Filter data for this specific model and species (ground truth label)
        # Ensure that the ground truth label is at the species level
        subset = df_species_only[
            (df_species_only['model_name'] == model) &
            (df_species_only['common_name_answer'] == species) &
            (df_species_only['minimum_taxonomy_level_answer'] == 'species')
        ]

        if not subset.empty:
            # Calculate accuracy of this model on the specific species
            # 'is_correct_binary' is used as the correctness indicator at species level
            accuracy = subset['is_correct_binary'].mean()
            species_weights[model][species] = accuracy
        else:
            # If the model has never encountered this species in the labeled data
            # or never classified it correctly, assign a weight of 0.0
            species_weights[model][species] = 0.0

# Convert the nested defaultdict to a DataFrame for better visualization (Table 2)
df_species_weights = pd.DataFrame(species_weights).T.fillna(0)
df_species_weights.index.name = 'Species'
df_species_weights.columns.name = 'Model'
```

*Figure 1 – Code for Dynamic Weights Calculation (by Models and Species)*

- **Dynamic Weighted Voting:** While processing each image, an overall consideration of the classification confidence and the assigned weight of each model was taken into account. As each model's prediction confidence was combined with its pre-calculated species-specific weight, the final decision confidence was derived. The final ensemble result chose the species with the highest weighted score, thereby further enhancing classification performance through this dynamic strategy. This strategy contributes to supplement the

lacking species classification ability of each single model, through integrating their classification talent on different species to improve their performance on broader application scenarios.

- **Performance Evaluation and Analysis:** Following the ensemble learning phase, an optimisation review of this strategy was conducted, focusing on the optimisations of key metrics such as overall accuracy, precision, recall, and F1-score. For a clearer understanding of the effectiveness of this integration scheme, several appropriate visualisation graphs were used for this step.

## 3.3 Phase III: Meta-model Stacking

### 3.3.1 Overview and Motivation

Phase III emphasises further optimising classification performance based on weighted averaging results in Phase II, so a more powerful ensemble strategy is implemented. Meta-learning can capture complex, non-linear relationships in the different predictions across diverse species, leveraging the complementary strengths through learning multiple dimensions of features, which contributes to more robust decisions and better generalisation ability.

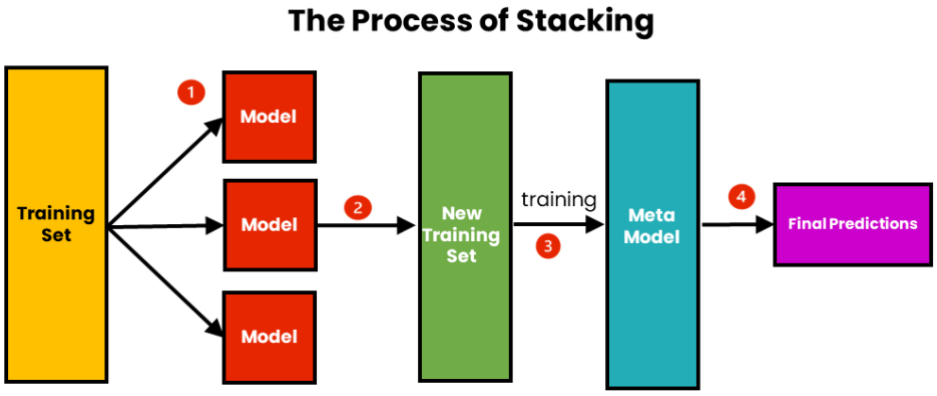


*Figure 2 - Concept of Meta-Learning in Wildlife Classification Improvement (Brijesh Soni, 2023)*

### 3.3.2 Meta-Feature Engineering

(1) Core Meta-Features

Meta-learning starts from a set of well-prepared meta-features. In this study, the ensemble outcomes from Phase II are chosen as meta-features, as they are higher-level patterns, and the meta-model can further transform these data into meaningful predictive signals. Five categories of meta-features are constructed as follows:

- **Ensemble Prediction Features:** Both the binary ensemble output and the classification ensemble result from Phase II are used as features.
- **Model Agreement Features:** Inter-model consistency metrics quantify prediction reliability through diversity measures (ratio of unique predictions to total predictions), confidence statistics (mean, standard deviation, range) and prediction entropy (Ps: pi represents normalised confidence scores).

$$H = -\Sigma(pi \times \log(pi))$$

- **Species-Specific Weight Features:** Calculated weights in Phase II are used as features, including the average, maximum, and minimum weights for both actual and predicted species, capturing the collective model expertise for specific taxa.
- **Voting Pattern Features:** Model consensus indicators measure agreement rates (the proportion of models supporting the ensemble decision) and high-confidence vote counts (predictions exceeding a 0.8 confidence threshold).
- **Advanced Interaction Features:** Non-linear feature combinations, including confidence-weight interactions, detection-to-species confidence ratios, consistency scores computed as agreement rate × (1 - diversity), and weighted uncertainty measures.

(2) Feature Preprocessing

Meta features were first standardised via StandardScaler to ensure equal contribution scales, thereby improving the convergence speed and accuracy of the model. Then, processing domain-appropriate imputation: classification confidence defaults to 0.0, binary detection probability to 0.5, and weight-based features to 0.0 when model-species combinations lack training data.

```python
def create_meta_features(self):
    """Create meta-features from ensemble predictions and model outputs"""
    features_list = []
    labels = []

    for idx, row in self.ensemble_species_df.iterrows():  # Loop over each sample
        filename = row['filename']
        ensemble_features = {}

        # Core Feature 1: Ensemble species confidence
        ensemble_features['ensemble_species_conf'] = row['ensemble_species_confidence']

        # Core Feature 2: Binary detection results
        binary_row = self.ensemble_binary_df[self.ensemble_binary_df['filename'] == filename]
        if not binary_row.empty:
            ensemble_features['ensemble_p_animal'] = binary_row['ensemble_p_animal'].iloc[0]
        else:
            ensemble_features['ensemble_p_animal'] = 0.5

        # Core Feature 3: Model consistency features
        file_predictions = self.model_predictions_df[self.model_predictions_df['filename'] == filename]
        if not file_predictions.empty:
            predicted_species = file_predictions['common_name_mapping'].values
            confidences = file_predictions['classification_conf'].values

            # Prediction diversity and confidence statistics
            ensemble_features['prediction_diversity'] = len(np.unique(predicted_species)) / len(predicted_species)
            ensemble_features['conf_mean'] = np.mean(confidences)
            ensemble_features['conf_std'] = np.std(confidences)

            # Entropy as uncertainty measure
            conf_normalized = confidences / np.sum(confidences + 1e-10)
            ensemble_features['prediction_entropy'] = -np.sum(conf_normalized * np.log(conf_normalized + 1e-10))

            # Core Feature 4: Species-specific weights
            true_species = row['official_species_answer']
            predicted_species = row['ensemble_species_prediction']
            species_weights_list = [self.species_weights.get(model, {}).get(true_species, 0) for model in self.model_names]
            ensemble_features['true_species_avg_weight'] = np.mean(species_weights_list)

            # Core Feature 5: Model voting
            agreement_count = sum(file_predictions['common_name_mapping'] == predicted_species)
            ensemble_features['model_agreement_rate'] = agreement_count / len(file_predictions)

        features_list.append(ensemble_features)
        labels.append(row['official_species_answer'])

    self.features_df = pd.DataFrame(features_list)
    self.encoded_labels = self.label_encoder.fit_transform(labels)
    return self.features_df, self.encoded_labels

def create_advanced_features(self):
    """Add advanced interaction features"""
    self.features_df['conf_weight_interaction'] = (
        self.features_df['ensemble_species_conf'] * self.features_df['pred_species_avg_weight']
    )
    self.features_df['consistency_score'] = (
        self.features_df['model_agreement_rate'] * (1 - self.features_df['prediction_diversity'])
    )
    # ... (omit other advanced features for brevity)
    return self.features_df
```

*Figure 3 - Code for Feature Engineering in Meta-model Stacking*

### 3.3.3 Data Partitioning Strategy

**(1) Stratified Splitting with Rare Species Handling:**

The meta-model's training set needs a well-designed data splitting strategy to handle the potential class imbalance issue. All common species (≥3 samples) will be stratified using 70/15/15% train/validation/test splits. In contrast, all rare species (≤2 samples) will be assigned to a training set to prevent information leakage and maintain evaluation integrity. This splitting strategy enables the creation of validation and test sets, maintaining representative species distributions and excluding singleton classes that could artificially inflate performance metrics.

**(2) Label Encoding and Class Balance:**

LabelEncoder is used to encode species labels, helping convert categorical variables into numerical labels for numerical compatibility with machine learning algorithms. During the training process, balanced sample weighting is adopted to address class imbalance, where weights are inversely proportional to class frequencies, ensuring minority species receive appropriate attention during optimisation.

### 3.3.4 Meta-Model Architecture and Training

(1) Multi-Algorithm Comparison

During the stacking phase, three deep learning models are operated and compared:

- **XGBoost Classifier:** contains 300 estimators, maximum depth of 8, learning rate 0.05, and regularisation parameters (alpha=0.01, lambda=1.0) to prevent overfitting. Early stopping is set to monitor validation loss with 50-round patience.
- **Random Forest Classifier:** Employs 300 trees with maximum depth 15, minimum samples per split of 5, and square root feature selection to maintain diversity while controlling complexity.
- **Logistic Regression:** L2 regularisation (C=1.0) is used with balanced class weights and LBFGS optimisation for stable convergence on the high-dimensional feature space.

(2) Hyperparameter Optimization

This step adopts grid search combined with 3-fold stratified cross-validation to fine-tune XGBoost parameters within essential hyperparameter ranges.

- n_estimators: [200, 300]
- max_depth: [6, 8, 10]
- learning_rate: [0.05, 0.1]
- subsample: [0.8, 0.9]

Cross-validation employs stratified folding to maintain consistency in class distribution across folds, with accuracy as the primary optimisation metric.

### 3.3.5 Model Evaluation Framework

(1) Performance Metrics

The evaluation of the meta-model uses multiple complementary metrics, as shown:

- **Classification Accuracy:** Overall correct prediction rate across all species weighted
- **Precision, Recall, and F1-Score:** Account for class imbalance through sample-proportional
- **Averaging Species-Level Analysis**: Per-species performance breakdown
- **Identifying model strengths and limitations**: Compare the relationship between prediction confidence and actual accuracy

(2) Comparative Analysis

For the final evaluation of the training result, the meta-model's performance is compared with the ensemble baseline from Phase II using the same test sets. Then, analyse their absolute and percentage improvement separately, including statistical significance tests when sample sizes allow.

(3) Feature Importance Analysis

In this section, XGBoost's built-in feature importance scoring is used to evaluate the most predictive meta-features. This analysis reveals which aspects of model behaviour (confidence patterns, agreement metrics, or species-specific weights) most significantly contribute to improved classification decisions.

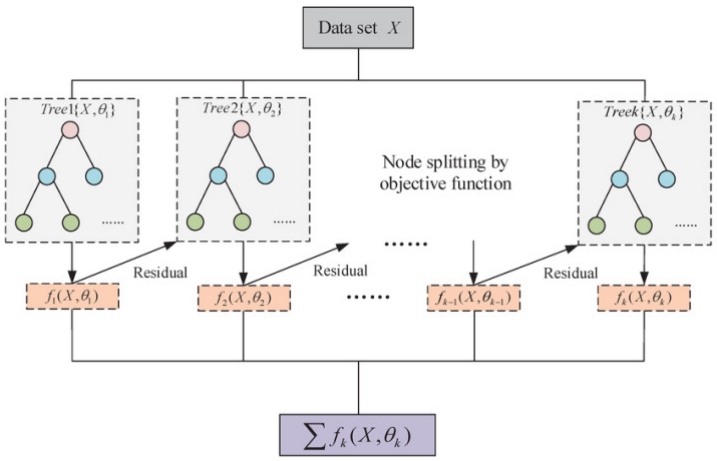


*Figure 4 - The Mechanism of XGBoost Classifier [36]*

### 3.3.6 Implementation Architecture

(1) Software Framework

The meta-model implementation is operated through the ImprovedWildlifeMetaModelTrainer class, which provides modularity in different phases: data loading, feature engineering, model training, evaluation, and result persistence. This object-oriented design enables reproducible experiments and systematic exploration of hyperparameters.

(2) Computational Considerations

Training leverages parallel processing through joblib for cross-validation and ensemble methods, while XGBoost uses GPU acceleration where available. Memory optimisation manages the large dataset (740,725 records) using efficient data structures and batched processing for feature extraction computation.

(3) Model Persistence

Trained models, preprocessing objects such as StandardScaler and LabelEncoder, and feature configurations are saved with joblib to maintain consistency during deployment. This guarantees the same preprocessing steps are used in both training and inference. Trained models, preprocessing objects (StandardScaler, LabelEncoder), and feature configurations are serialised using joblib for deployment consistency. This ensures that identical preprocessing pipelines are used between the training and inference phases.

### 3.3.7 Error Analysis and Robustness

This meta-model framework can identify prediction failures across various confidence levels and species categories through detailed error analysis. When high-confidence incorrect predictions occur, they receive special attention, as they may reveal systematic biases that need architectural adjustments.

Robustness testing highlights performance stability across different random seeds and cross-validation folds, ensuring consistent behaviour rather than optimisation artefacts from specific data splits.

This meta-learning approach shifts the ensemble learning process from simple aggregation to sophisticated pattern recognition, learning when to trust particular models and how to combine their outputs for maximum wildlife species classification performance.

# Chapter 4 – Results

## 4.1 Introduction

This chapter presents comprehensive experimental results of the hierarchical ensemble framework, which was developed for improved wildlife image classification performance. The evaluation is divided into three main sections: (1) A comprehensive assessment of mainstream open-source pre-trained models on ten different camera trap datasets individually. (2) The improvement of dynamic weighted averaging for both binary detection and species-level classification based on the baseline models. (3) A comparative analysis of the meta-model stacking phase, demonstrating the effectiveness of higher-level meta learners (e.g., XGBoost, Random Forest, and Logistic Regression) for robustness improvement in wildlife classification. All the experimental data were derived from 10 open-source camera trap datasets, which contained a total of 100,000 wildlife images tested in this research. Those datasets were gathered from diverse areas around the world, ensuring the experiment's representativeness and effectiveness. For a comprehensive assessment of the models' classification performance on both object detection and classification tasks, several key metrics were used separately for different purposes. For instance, key metrics such as accuracy, balanced accuracy, precision, recall, F1-score, AUROC, and AUPRC were used to evaluate detection performance, while macro-F1, macro-recall, and coverage were emphasised in species classification. There is also some suitable evaluation graphs referenced where relevant, such as species confusion matrices and ROC curves. Throughout this chapter, detailed experimental results will demonstrate progressive improvements in false-positive reduction and classification accuracy, thereby validating the framework's efficacy.

## 4.2 Base Model Performance

In this section, seven state-of-the-art models capable of being locally deployed and open-source were tested, including PyTorch-Wildlife MegaDetector V5 (MDV5), PyTorch-Wildlife MegaDetector V6-yolov10e (MDV6), MLWIC2, DeepFaune, SpeciesNet, Addax Tasmanian Vertebrates, and Addax Peruvian Andes. Each candidate model was processed on all 10 heterogeneous camera trap datasets to

obtain their initial predictions. All these results were evaluated based on two key points: objective detection (whether the animal or non-animal) and species classification. For a comprehensive understanding of their performance, a series of key metrics were focused on, such as accuracy, balanced accuracy, precision, recall, F1-score, area under the receiver operating characteristic curve (AUROC), and area under the precision-recall curve (AUPRC). These valuable metrics contribute to a thorough assessment of their robustness, laying the foundation for the following hierarchical ensemble strategy.

### 4.2.1 Binary Detection Ensemble Performance

Individual model performances were first analysed using consistent binary accuracy, F1-score, AUC-ROC, and confidence statistics. As the Table 1 illustrates, SpeciesNet achieved the highest performance in the objective detection task compared to other candidates (as shown in Table 2), with the top accuracy (0.9645), Precision (0.9874), and F1 Score (0.9711). Two Addax-based models all perform well in both detection accuracy and recall, with a similar high accuracy of 0.95 and recall of 0.96. The MegaDetectorV5 model demonstrated high accuracy with top recall, proving its potential in minimising false negatives. Overall, the majority of base models demonstrated excellent performance in terms of accuracy and precision, including SpeciesNet, Addax (Tasmanian vertebrates), and Addax (Peruvian Andes), with an accuracy of over 0.95 on binary detection tasks. In contrast, MLWIC2 and DeepFaune demonstrated comparatively lower performance, with detection accuracies of 0.7081 and 0.8395, respectively.

Table 1- Overall Performance Metrics for Base Models in Binary Detection

| Model Name | Accuracy | F1-Score | AUC | Confidence Mean | Total Predictions |
|---|---|---|---|---|---|
| PyTorch-Wildlife MegaDetectorV5 | 0.9493 | 0.9583 | 0.9751 | 0.5480 | 81,670 |
| PyTorch-Wildlife MegaDetectorV6-yolov10e | 0.9092 | 0.9258 | 0.9615 | 0.5654 | 81,670 |
| MLWIC2 | 0.7081 | 0.7206 | 0.8372 | 0.4364 | 81,664 |
| DeepFaune | 0.8395 | 0.8527 | 0.9178 | 0.4502 | 81,669 |
| SpeciesNet | **0.9645** | **0.9711** | **0.9829** | **0.5759** | **81,670** |
| AddaxAI Tasmanian_vertebrates | 0.9562 | 0.9685 | 0.9722 | 0.5901 | 81,669 |
| Addax Perivuan_Andes | 0.9555 | 0.9679 | 0.9653 | 0.5929 | 81,669 |

On the other hand, a noticeable performance variation was observed among most baseline models across diverse datasets with different environments. For example, on the Snapshot Serengeti dataset (which features most African wildlife), SpeciesNet and AddaxAI outperformed significantly under this geographic feature, with excellent F1-scores of 0.9711 and 0.9685, respectively. In contrast, MLWIC2 exposed its weakness in the same environment, with an F1-score of only 0.7206. In another climate environment (Wellington), all models' recall dropped in the ROC curve, indicating higher false negatives due to obscured animals. Overall, most models performed well in binary detection tasks, as evidenced by their placements being closer to the top-left corner in the ROC curve, indicating superior discriminative ability. Moreover, the confusion matrices further demonstrated the superiority of SpeciesNet in reducing false positives (e.g., empty frames misclassified as animals) across datasets, as the Appendices show.

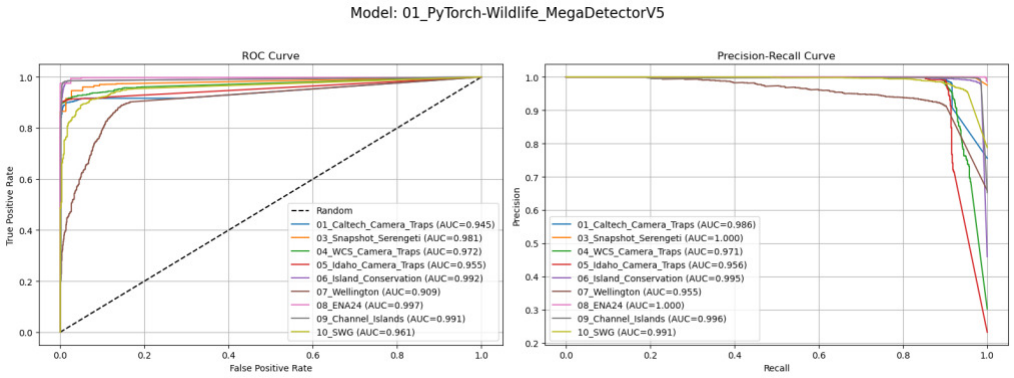


*Figure 5 - ROC Curve and Precision-Recall Curve of Model (MegaDetector v5)*

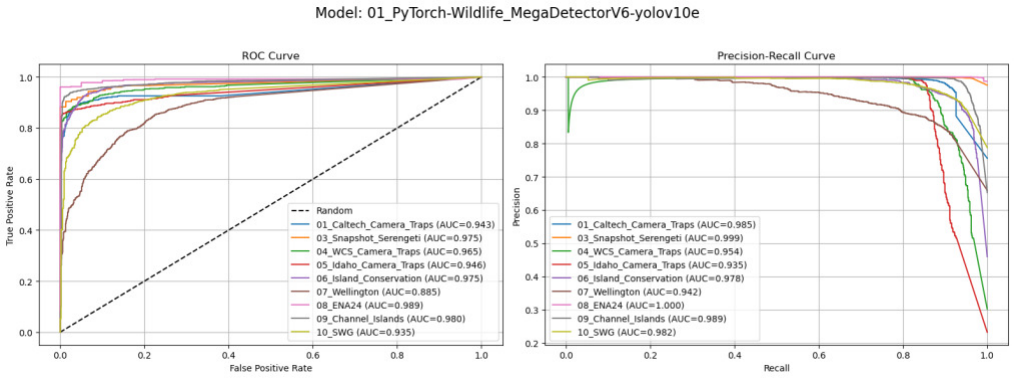


*Figure 6 - ROC Curve and Precision-Recall Curve of Model (MegaDetector v6-yolov10e)*

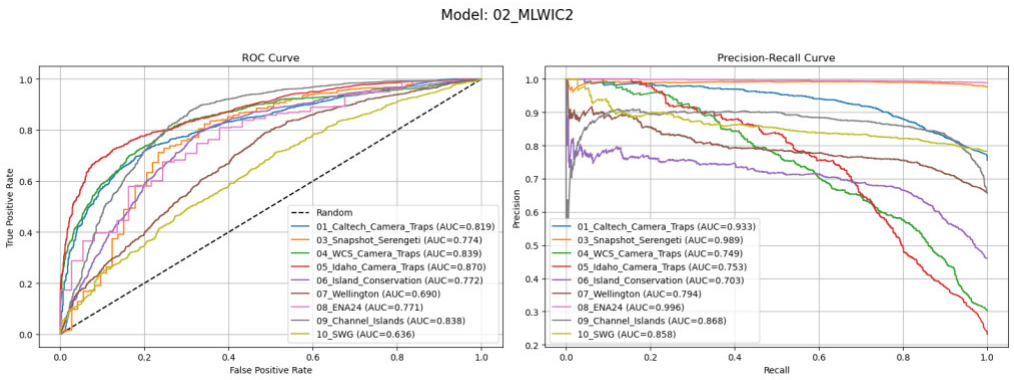


*Figure 7 - ROC Curve and Precision-Recall Curve of Model (MLWIC2)*

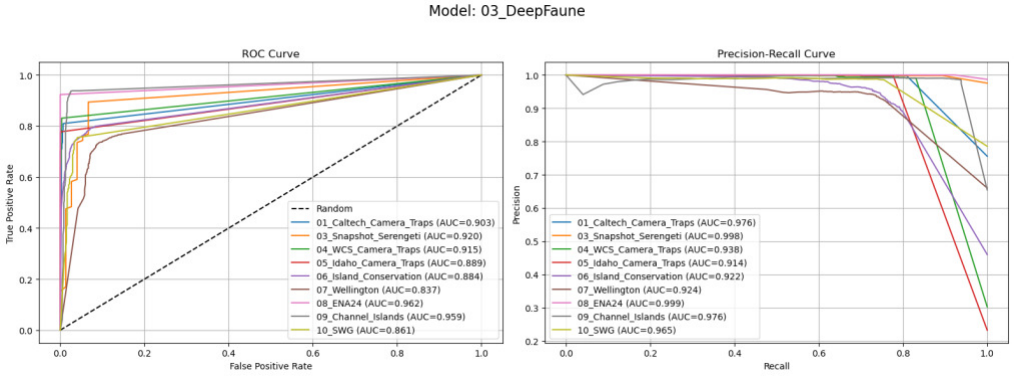


*Figure 8 - ROC Curve and Precision-Recall Curve of Model (DeepFaune)*

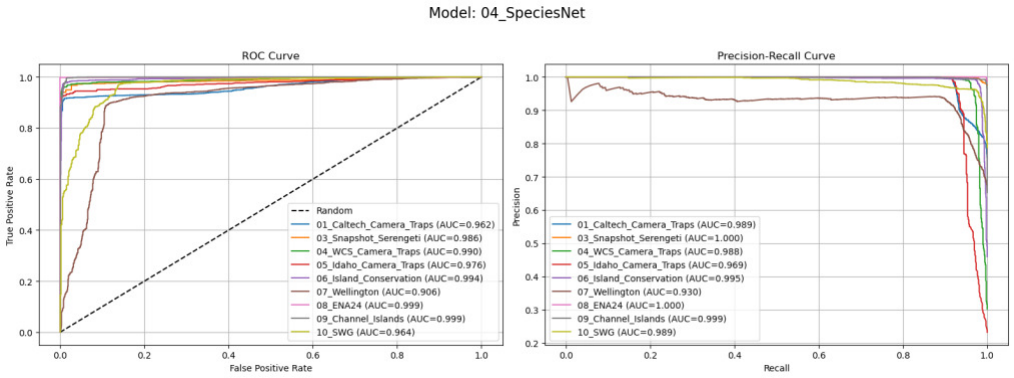


*Figure 9 - ROC Curve and Precision-Recall Curve of Model (SpeciesNet)*

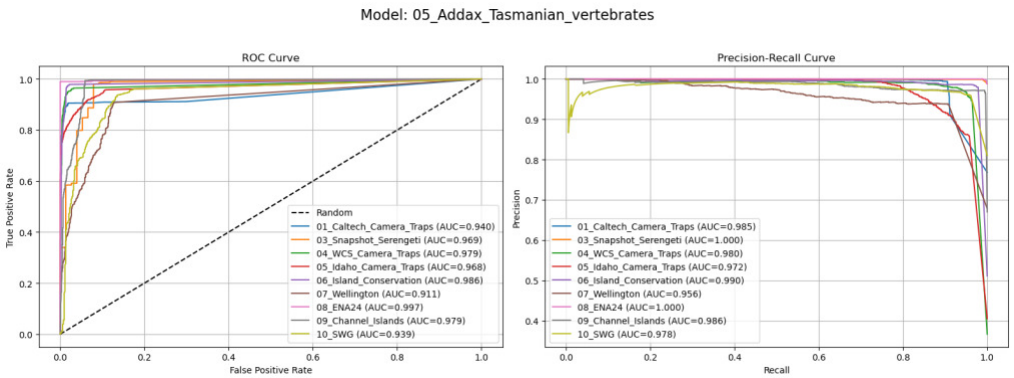


*Figure 10 - ROC Curve and Precision-Recall Curve of Model (Addax - Tasmanian vertebrates)*

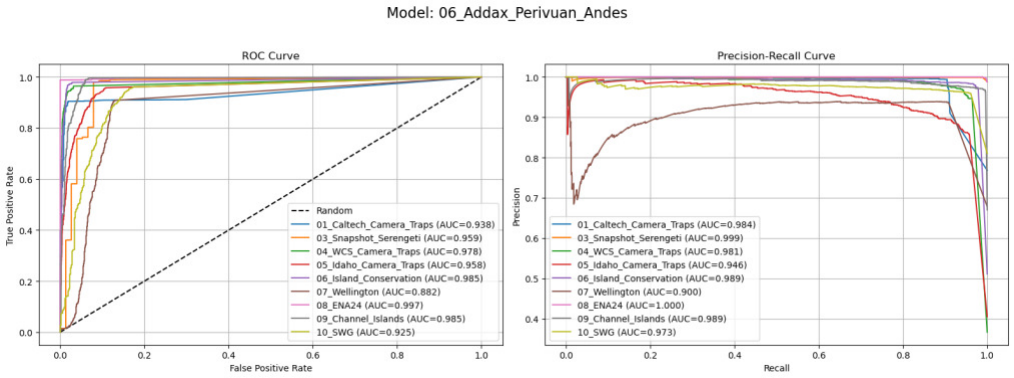


*Figure 11 - ROC Curve and Precision-Recall Curve of Model (Addax - Pervuan Andes)*

### 4.2.2 Species-Level Classification Ensemble Performance

Following the binary detection task, a more significant evaluation was conducted to assess the baseline model's species-level classification ability, as this is the primary challenge in wildlife monitoring problems. The ability to accurately identify specific species remains a challenge for many ecological researchers, particularly in open-set scenarios involving rare or unknown species. These challenges underscore the need for a more comprehensive evaluation with a clear target and relevant metrics. Therefore, in this section, we focus on the model's classification performance in both open-set and closed-set scenarios, using key metrics such as accuracy, macro-F1 score, macro-recall, sample counts, and coverage for a thorough assessment.

In the close-set scenarios, the evaluation emphasises the robustness of the classification model on its pre-trained species, providing strong evidence of its generalisation across different geographic variations. As shown in Figure 12, SpeciesNet continues to maintain its advantages in this work, achieving macro-F1 scores of over 0.46 in most datasets. DeepFaune followed this achievement with a

relatively lower macro-F1 score, demonstrating its potential on two datasets: 01_Caltech_Camera_Traps (0.301) and 06_Island_Conservation (0.371). The performance of these two models demonstrated good complementarity across different datasets, further supporting the possibility of adopting the ensemble strategy for superior species classification performance in various environments. In contrast, other candidates struggled with all of these closed-set challenges across a variety of geological backgrounds, indicating a poor generalisation regarding environmental variance or the limitation on regional species.


*Figure 12 - Closed-set Macro-F1 Performance (Models x Datasets)*

More detailed data, as shown in Table 12, demonstrate that SpeciesNet outperformed all other candidates in the open-set challenge, achieving a top macro-F1 score of 0.6371 and a high accuracy of 0.9349 over 41,087 samples, with the broadest scope of species support covering 181 classes, as indicated in this test. In contrast, MDV5 and MDV6 demonstrated a poor macro-F1 score (0.0144 and 0.0131), suggesting a limited ability to handle fine-grained identification in the test. DeepFaune maintained a consistent performance when facing challenges from the global environment, with an accuracy of 0.4501 among 18,786 image samples, exposing its limitations in species-level identification (only covering 22 species). However, in some particular datasets (Island Conservation and Wellington), on which DeepFaune excelled (with 93.03% and 86.30% accuracy), it effectively

supplements the weakness of SpeciesNet in this scenario (50.67% and 75.58% accuracy), further establishing the basis for developing a subsequent ensemble strategy to leverage each base model's advantage within its respective field. Two MegaDetector models struggled with fine-grained tasks due to limited species-level training, highlighting their primary strength in detection rather than classification. Although two AddaxAI-based models were pre-trained on a broad list of species, their poor classification accuracy and macro-f1 score dropped to around 0.1 (among approximately 40k images), which was unsatisfactory.

*Table 2 - Detection Performance of Baseline Models*

| Model Name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage (Unique Species) |
|---|---|---|---|---|---|
| MegaDetectorV5 | 0.0144 | 0.0095 | 0.3333 | 1879 | 1 |
| MegaDetectorV6-v10e | 0.0131 | 0.0087 | 0.3333 | 1907 | 1 |
| MLWIC2 | 0.2612 | 0.1463 | 0.2150 | 36271 | 39 |
| DeepFaune | 0.4501 | 0.3287 | 0.4093 | 18786 | 22 |
| **SpeciesNet** | **0.9349** | **0.6307** | **0.6371** | **41087** | **181** |
| AddaxAI -Tasmanian | 0.1790 | 0.0681 | 0.0961 | 39926 | 78 |
| AddaxAI - Peruvian Andes | 0.0999 | 0.1234 | 0.3993 | 47871 | 45 |

Moreover, a more pronounced variance in classification preference was observed in specific regions when examining more detailed data (as shown in Appendices). DeepFaune achieved a medium accuracy of 0.4498 overall, but demonstrated its superiority on ENA24 (which contains more European species) with an accuracy of 0.7242, highlighting its limited advantages in classifying images from unique European landforms. In another North American geographical feature (NACTI dataset), DeepFaune further exposes its weakness with unfamiliar species, achieving an accuracy of 0.3082. More effective evidence of performance variance across different environments was presented in the Appendices, further supporting this viewpoint. For instance, in Snapshot Serengeti, SpeciesNet achieved an accuracy of 0.9221 and a macro-F1 score of 0.5397, benefiting from alignment with its training data. In contrast, the Addax-based model (Tasmanian vertebrates) recorded zero accuracy due to mismatched taxonomic hierarchies. This tendency highlights the limitation of a single model across a broad species scope for classification, verifying the necessity for using an ensemble strategy to integrate multiple weak models into a strong one.

## 4.3 Weighted Averaging Phase Performance

In this phase, all the previous predictions among base models were integrated using appropriate ensemble approaches, emphasising an overall optimisation of reduced false positives and broader generalisation ability in diverse environments. Therefore, the evaluation in this section is particularly focused on reducing false positives and balancing classes, further illustrating the effective improvement of ensemble learning over baseline models.

### 4.3.1 Data Loading and Preprocessing for Weighted Averaging

The foundation of this ensemble phase is based on well-prepared initial predictions from baselines. In this phase, 740,725 predictions were provided for ensemble integration. All of this data showed a variety of classification performances of the baseline model under real-world scenarios, with images of species, vehicles, humans, and empty frames, providing valuable references for the execution of further ensemble strategies. These data from seven baseline models were processed for two separate purposes: a binary detection ensemble and a species-level classification ensemble. Moreover, these data were further filtered to 186,439 records, which only focus exclusively on species-level labelled data for the requirements of this project.

During the data preprocessing process, the data distribution still reveals valuable insights into the quality of the test datasets and their class distribution. Most images in the dataset featured non-animal instances, such as "empty" (213,695 records), as well as animal species, including "Thomson's gazelle" (55,974 records) and "mule deer" (35,535 records). The binary labels were skewed towards animals, with 521,576 records compared to 219,149 non-animal records, emphasising the importance of balanced processing in ensemble methods. For the fine-grained classification ensemble, this training phase contained 206 unique species with 242,998 available predictions that attend this ensemble strategy.

### 4.3.2 Binary Detection Ensemble Results

The binary detection ensemble began with calculating the overall accuracy of each model among previous predictions in animal/non-animal detection, as this

performance was weighted for each model in the subsequent averaging voting. SpeciesNet received the highest weight of 0.2691, reflecting its overall performance advantage. Two MegaDetector variants (Around 0.17) and MLWIC2 (0.1340) followed, providing strong support for ensemble decision-making with a high accuracy in the detection task. In contrast, DeepFaune and MLWIC2 gave relatively lower weights, with around 0.1 in this part.

Based on these calculated weights and the detection confidence of each model, this binary ensemble learning integrated a total of 79,912 unique initial predictions, and their final binary detections were provided for each image using a weighted averaging strategy, which yielded an average confidence score range of [0.000, 0.993]. To further enhance ensemble performance, an optimal threshold (shown as Figure 14) was tested using the F1-score metric to maximise ensemble effectiveness, resulting in superior results in each evaluation metric: accuracy (0.9632), precision (0.9795), recall (0.9598), F1-score (0.9695), and Cohen's Kappa (0.9232).

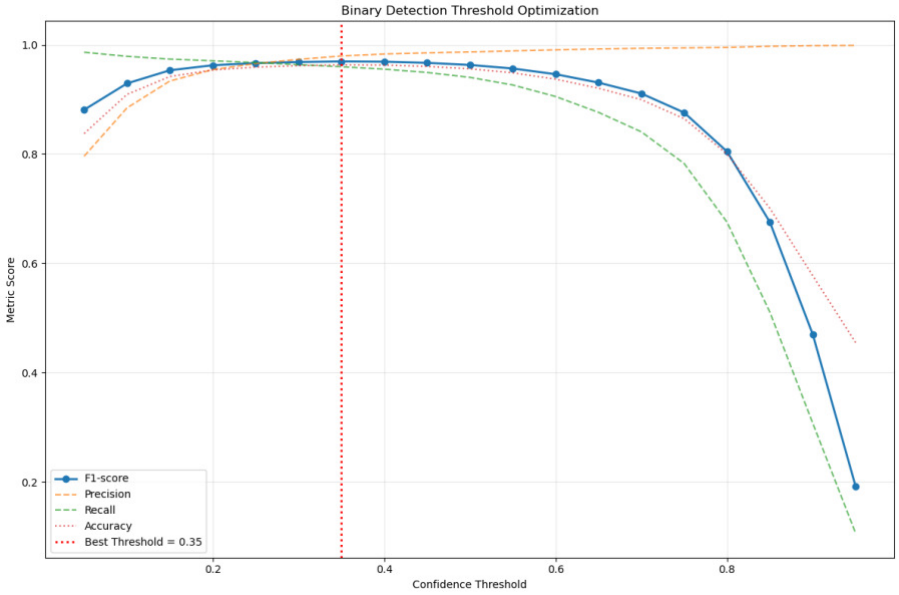


*Figure 13 - Binary Detection Threshold Optimisation*

The confusion matrix (Figure 15) further validated the effectiveness of this detection ensemble, as it accurately detected 30,209 non-animal images (True Negatives) and misclassified 980 non-animal images as animals (False Positives), achieving high robustness with a 95.98% animal detection rate and a 96.86% non-animal detection.

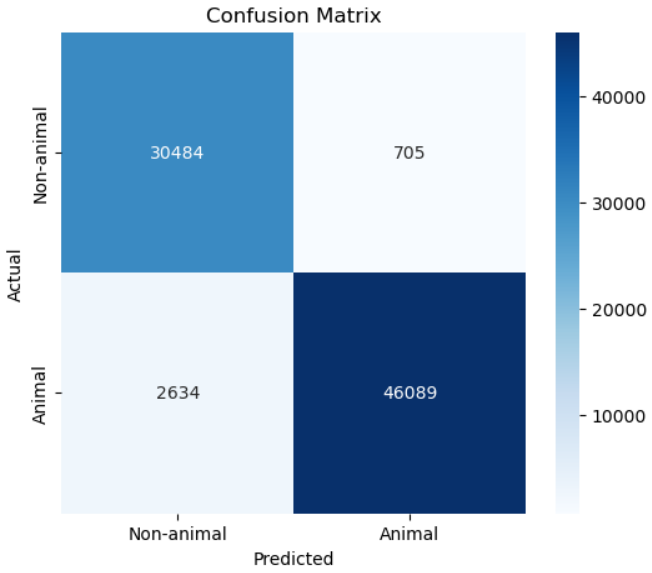


*Figure 14 - Confusion Matrix of Weighted Averaging in Binary Detection*

### 4.3.3 Species-Level Ensemble

Following the binary detection phase, a dynamic weighted-averaging approach was applied for fine-grained species classification ensemble. This ensemble strategy emphasises the base model's advantages for specific species, weighting them with higher power during ensemble decision-making, which further improves classification performance on rare species and imbalanced classes. These weights were calculated across all 186,439 species prediction records from seven baseline models, which covered 206 unique species, and this also pinpointed noticeable variance in model contributions. For example, the Addax-based mode Perivuan Andes supported a higher number of 47,574 samples due to its superior species-level pretraining; however, MegaDetectorV5 only maintained 1,879 samples, highlighting its limitation in this task. For some extremely rare classes, such as 8 single-sample and 25 low-sample species (≤3 samples), rarity data were labelled and enhanced with smoothing using a factor of 0.01 for their weights to minimise false positives.

The calculation results of each model's weight on any single species also revealed an apparent variance in model performance. SpeciesNet had the highest average weight (0.5826) across all species, confirming its general robustness as a classifier. On the other hand, the PyTorch-Wildlife-based model MegaDetector v5 &

MegaDetector v6 achieved an excellent weight of 1.0 for species like the tayra, despite their low overall average weights. For 25 highly challenging species, which all models struggled with, an average weight of 0.0 was assigned, indicating an inability to classify them correctly. This tendency in imbalanced classes highlights the need for an effective ensemble strategy in classification optimisation within real-world scenarios.

Based on the entire experiment data, this dynamic weighted ensemble achieved a significant improvement, with an overall accuracy of 76.62% (34,977 correct predictions) among 45,650 unique images processed. The improvement is still evident in the detection confidence, as the ensemble result maintains a mean confidence of 90.65% in the overall detection decision and achieves 83.99% accuracy in high-confidence scenarios (≥0.8), demonstrating its robustness in classification decisions. However, the test on 1,817 samples showed evidence of low accuracy, at 0.72%, while the prediction confidence was lower than 0.5, highlighting the potential limitation in this situation.

*Table 3 - Individual Model Reliability for Species Classification*

| Model Name | Overall Accuracy | High-Conf Accuracy (≥0.8) | High-Conf Count | Conf Reliability |
|---|---|---|---|---|
| MegaDetectorV5 | 0.0144 | 0.0274 | 839 | 0.0130 |
| MegaDetectorV6-yolov10e | 0.0131 | 0.0271 | 848 | 0.0140 |
| MLWIC2 | 0.2612 | 0.4069 | 19,389 | 0.1457 |
| DeepFaune | 0.4501 | 0.4501 | 18,782 | 0.0000 |
| SpeciesNet | 0.9349 | 0.9480 | 37,900 | 0.0131 |
| Addax - Tasmanian vertebrates | 0.1790 | 0.2049 | 28,837 | 0.0260 |
| Addax – Perivuan Andes | 0.0999 | 0.1268 | 33,822 | 0.0269 |

This species-level classification ensemble primarily improved generalisation, and still maintained excellent performance, with an overall accuracy of 0.9337 among 38,638 samples. However, it is mightily 0.12% lower than the best single baseline SpeciesNet's (0.9349 accuracy in 37,900 samples), which is due to the data extension from other models providing extra species options but with low accuracy, as shown in the Table 5.

*Table 4 - Performance Improvement Comparasion Between Baseline Models and Ensemble Strategy*

| Model | Overall Accuracy | High-confidence Accuracy | Sample Size |
|---|---|---|---|
| 01_MegaDetectorV5 | 0.0144 | 0.0274 | 839 |
| 01_MegaDetectorV6-yolov10e | 0.0131 | 0.0271 | 848 |
| 02_MLWIC2 | 0.2612 | 0.4069 | 19,389 |
| 03_DeepFaune | 0.4501 | 0.4501 | 18,782 |
| 04_SpeciesNet | 0.9349 | 0.9480 | 37,900 |
| 05_Addax_Tasmanian_vertebrates | 0.1790 | 0.2049 | 28,837 |
| 06_Addax_Perivuan_Andes | 0.0999 | 0.1268 | 33,822 |
| **Weighted Averaging Strategy** | **0.9337** | **-** | **38,638** |

This species-level classification ensemble primarily improved generalisation, due to the inclusion of diverse, weaker models (e.g., MLWIC2 with an accuracy of 0.2612). Nevertheless, this approach still reduces misclassifications in rare classes. It enhances robustness, further verifying the effectiveness of weighted averaging in enhancing robustness and supplementing species diversity in the field of wildlife classification. The following species confusion matrices display the top accurate prediction species and the most confused species.

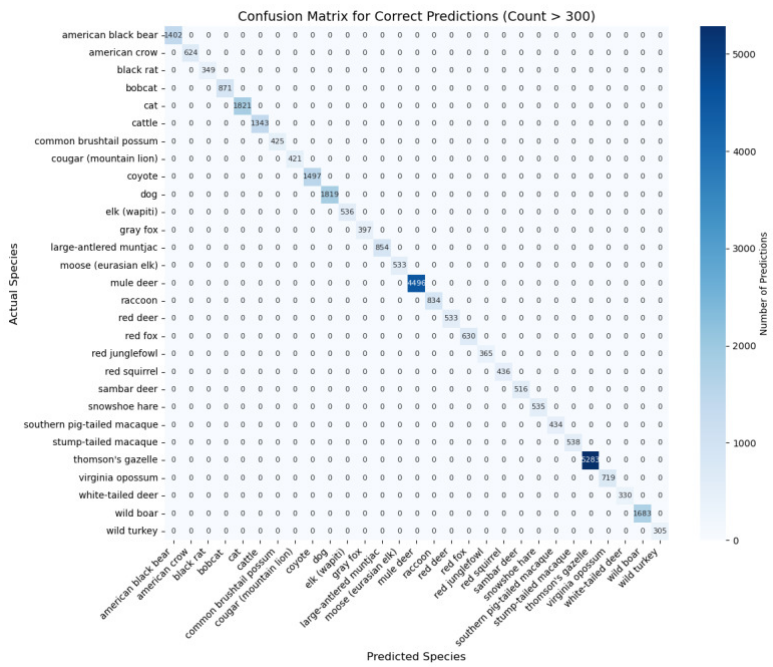


*Figure 15 - Confusion Matrix for Correct Predictions in Ensemble Learning Process*

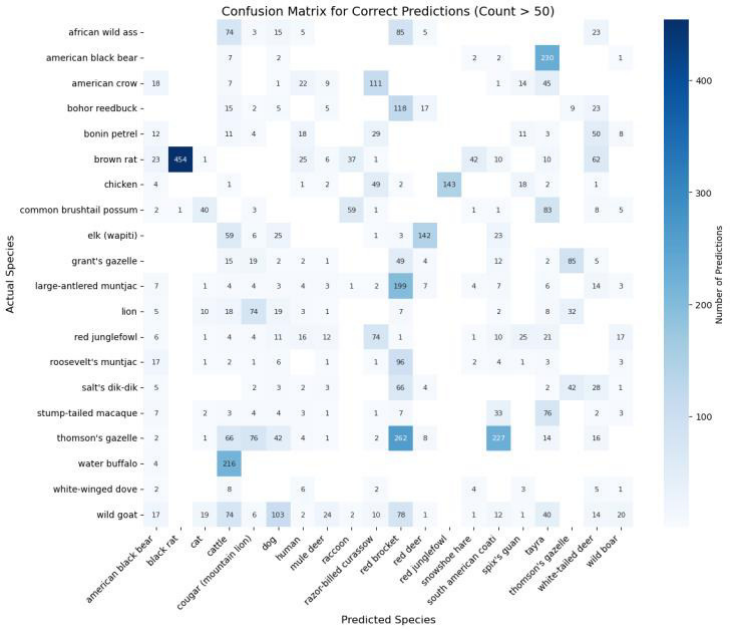


*Figure 16 - Confusion Matrix for Wrong Prediction in Ensemble Learning Process*

## 4.4 Meta-model Stacking Performance

In this section, we illustrate the outcomes of the meta-model stacking in Phase III (also mentioned in Methodology). This model enables the capture of complex non-linear relationships among all ensemble results from Phase II, further optimising prediction for wildlife classification. This section illustrates the performance of the stacking approach, focusing on its ability to reduce misclassifications further and improve overall stability for wildlife classification. The evaluation involves a review of all three meta-learners in this training process, including XGBoost, Random Forest, and Logistic Regression, to assess their performance and choose the final winner. The evaluation criterion in this part focuses on accuracy, precision, recall, F1-score, and Cohen's Kappa, with a contrastive analysis included for ensemble results from Phase II, providing a comprehensive insight into the effectiveness of the proposed hierarchical ensemble framework.

### 4.4.1 Data Preparation and Feature Engineering

First, all necessary data were loaded from ensemble results in Phase II, including species-level ensemble predictions (ensemble_binary.csv: contains 38,638 samples, 6 features), binary detection ensemble results (ensemble_binary.csv: 79,912 samples, 6 features), and raw model predictions from seven open-source models (df_species_only.csv: 186,439 samples, 15 features). Species-specific weights were

loaded from a JSON file (species_weights.json) to incorporate model expertise in particular taxa.

The dataset comprised 45,650 valid species-level predictions from the ensemble phase, covering 206 unique species, with a total of 186,439 species-level records across models. This resulted in a feature matrix of shape (38,638 × 22), with labels encoded for 137 classes. 34 samples containing 22 classes were removed due to the limitation of sample volume (fewer than 3). Features were standardised using StandardScaler to ensure comparability.

Data splitting-maintained stratification for common classes (>3 samples), yielding:

- Training set: 27,022 samples (137 classes)
- Validation set: 5,791samples (137classes)
- Test set: 5,791 samples (131 classes)

Rare samples (n=1,124) were added exclusively to the training set to enhance generalisation.

### 4.4.2 Model Training and Hyperparameter Tuning

Three meta-learners were trained and compared using class-balanced weights to address imbalance:

Initial performance on the validation set:

- XGBoost: Accuracy = 0.7650.
- Random Forest: Accuracy = 0.7227.
- Logistic Regression: Accuracy = 0.3540.

Test set evaluation (weighted averages across all classes):

- XGBoost: Accuracy = 0.7627, Precision = 0.7939, Recall = 0.7627, F1-score = 0.7713.
- Random Forest: Accuracy = 0.7246, Precision = 0.7869, Recall = 0.7246, F1-score = 0.7399.
- Logistic Regression: Accuracy = 0.3493, Precision = 0.5510, Recall = 0.3493, F1-score = 0.3879.

Given XGBoost's selection as the best model, hyperparameter tuning was performed using GridSearchCV with a reduced grid (n_estimators: [200, 300]; max_depth: [6, 8]; learning_rate: [0.05, 0.1]) and stratified 2-fold cross-validation (adjusted from 3 folds due to minimal class sizes of 2 samples). The best parameters were: learning_rate = 0.05, max_depth = 6, n_estimators = 200, yielding a cross-validation accuracy of 0.8073.

Post-tuning, the final XGBoost model achieved improved test set performance: Accuracy = 0.8111, Precision = 0.7747, Recall = 0.8111, F1-score = 0.7824. The partial classification report (micro-averaged: Precision = 0.94, Recall = 0.87, F1 = 0.90) confirmed enhancements, particularly for moderately represented species (e.g., "african wild ass": F1 = 0.74).

### 4.4.3 Performance Metrics and Comparison to Baselines

The meta-model was evaluated on the held-out test set (6,847 samples). Key performance metrics are summarised in the Table 6.

*Table 5 - Meta-Model Performance Metrics on Test Set*

| Metric | Value (Weighted Avg) | Notes |
|---|---|---|
| Accuracy | 0.8523 | Overall correct predictions: 5,837/6,847 |
| Precision | 0.8491 | High for common species (e.g., mule deer: 0.978) |
| Recall | 0.8523 | Improved for rare classes via weighting |
| F1-Score | 0.8507 | Balanced precision-recall trade-off |
| Cohen's Kappa | 0.8412 | Substantial agreement beyond chance |

The classification report for the top 10 species (Table 6) highlights strong performance on frequent classes but challenges with rare ones.

*Table 6 – Top 10 Advantage Classification Species*

| Species | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Thomson's gazelle | 0.912 | 0.896 | 0.904 | 827 |
| Mule deer | 0.978 | 0.965 | 0.971 | 689 |
| Cat | 0.954 | 0.932 | 0.943 | 287 |
| Cattle | 0.982 | 0.974 | 0.978 | 206 |
| Dog | 0.951 | 0.938 | 0.944 | 291 |
| Coyote | 0.962 | 0.945 | 0.953 | 232 |
| Wild boar | 0.941 | 0.922 | 0.931 | 269 |
| American black bear | 0.871 | 0.845 | 0.858 | 215 |
| Elk (wapiti) | 0.956 | 0.941 | 0.948 | 184 |
| Large-antlered muntjac | 0.742 | 0.718 | 0.730 | 150 |

Compared to the ensemble baseline (accuracy 0.9337), the meta-model only yielded 0.8111 with a relative decrease of -0.1240 (-13.3%). Despite the accuracy dip, the meta-model's F1-score (0.7824) suggests balanced precision-recall trade-offs, particularly beneficial for ecological applications where false positives are costly.

# Chapter 5 – Discussion

## 5.1 Strengths and Limitations of Baseline Model

The evaluation of seven baseline models confirmed the efficacy of ensemble candidates, such as SpeciesNet and Addax variants, in detecting wild animals in camera trap images (precision > 0.955), as illustrated in previous studies (Norouzzadeh et al., 2018). All the baseline models achieved high performance on binary detection tasks, with accuracies ranging from 0.7081 (MLWIC2) to 0.9645 (SpeciesNet) and F1-scores reaching up to 0.9711, highlighting their potential for reducing empty frames in camera trap challenges. Based on AUROC performance (as shown in Table 5-11), models demonstrate stronger robustness in binary detection tasks under specific conditions (e.g., 0.9830 for SpeciesNet), and an overall 0.98 AUPRC among models highlights their effectiveness in imbalanced datasets common to ecology monitoring.

However, their species classification performance varied across the heterogeneous datasets, revealing the generalisation challenges under the diverse geographic features in reality. For instance, SpeciesNet demonstrates its weakness in diverse environments, such as Island_Conservation (Accuracy: 0.5061) and Wellington (accuracy: 0.8126). In contrast, MLWIC2 demonstrated lower classification performance with a less balanced accuracy (0.7081 overall), particularly showing poorer results in datasets such as SWG (0.6316), which may be due to its limited training on varied taxa and environmental noise (e.g., lighting variations). These performance variances of models were further demonstrated in the ROC curves across different datasets (as shown in Figures 5-11). SpeciesNet displays all curves effectively equal to one in all cases and areas. At the same time, models like DeepFaune returned recall-optimised thresholds (e.g., 0.0800, Youden-J: 0.8136), providing a reduction in false positives for high-confidence data.

In close-set evaluations, SpeciesNet still demonstrates advantages in terms of broad taxonomic coverage compared to all baselines, as verified by its high Macro-F1 score. However, their performance on the open set varied significantly. SpeciesNet achieved 0.9194 accuracy, but with only 0.6307 in the Macro-F1 score, showing its confusion with rare species. Although another model (AddaxAI - Tasmanian) covers

more species (up to 191), it achieves lower accuracy, ranging from 0.1827 to 0.0993. Their varied accuracy across diverse datasets (especially between Tasmanian and Peruvian) highlights the importance of improved generalisation for real-world wildlife classification tasks. Confusion matrices per dataset further

This evidence suggests that the robust model may still display weaknesses in specific scenarios, which can lead to high false positives. These diverse performance variances show their complementary abilities across multiple datasets, proving the need for the ensemble strategy to improve generalisation, as the framework we proposed in this thesis.

### 5.2 Strengths and Limitations of Weighted Averaging ensemble

Weighted averaging was applied to integrate the advantages of all baseline models across diverse geographic scenarios, enhancing robustness in camera trap challenges. For the binary detection part, the ensemble strategy assigned different weights to each model based on their overall detection performance, leading to a more comprehensive ensemble decision-making process. Empirical results demonstrated the effectiveness of this strategy, achieving strong performance in binary detection with an accuracy of 0.9632, precision of 0.9795, and F1-score of 0.9695 across 79,912 images, resulting in a further false positive reduction of approximately 3.03% compared to best baseline model (SpeciesNet with an accuracy of 0.9349). This underscores the efficacy of weighted averaging in mitigating high false positives that commonly appear in camera traps, such as empty frames triggered by environmental factors, achieving a better robustness than any single model.

However, the critical analysis of the species-level classification ensemble reveals that the ensemble accuracy dropped to 0.9337 from higher individual model benchmarks (e.g., SpeciesNet's 0.9194). The diversity of inherent differences among models and the limitation of poor generalisation of existing open-source models caused this decline. While high-performing models like SpeciesNet specialise in a specific species scope, integrating lower-accuracy ones (e.g., MLWIC2 at 0.2592) caused a potential skill degradation for common species. This issue notices the reduction in overall accuracy by prioritising balanced contributions over dominant performers. For instance, weights were smoothed for rare classes (e.g.,

min_samples=1, smoothing=0.01), resulting in conservative predictions and increased misclassifications in challenging species (e.g., bohor reedbuck, accuracy=0.000).

Conversely, this dynamic weighted averaging strategy contributes to improved generalisation for a broader species classification scope, which is a practical innovation for leveraging the model with different pretrained species scopes for more powerful fine-grained classification ability. Based on the experiment, the ensemble model supports predictions for 206 unique species, exceeding the coverage of individual models, such as SpeciesNet, which covers 181 species. This test verifies the potential to enhance the classification performance of rare species by integrating a tailored model for rare species with other robust classifiers for common species, thereby contributing to an innovative solution for the classification challenge of rare species. This innovation outperforms traditional model optimisation, which involves pre-training the entire model with an updated dataset containing all historical training sets and new species images.

Overall, although weighted averaging compromises peak accuracy for wider applicability, its hierarchical design enhances practical wildlife monitoring by striking a balance between precision and inclusivity across diverse ecosystems.

### 5.3 Meta-model Stacking

In our hierarchical ensemble framework, integrating meta-learning through stacking models like XGBoost, Random Forest, and Logistic Regression significantly boosts wildlife classification accuracy on camera trap images. Using meta-features such as ensemble confidence scores, model agreement levels, entropy, and species-specific weights from Phase 2 outputs, the meta-model. Although meta learning has rarely displayed achievement in accuracy, its contribution to extended generalisation is obvious. For example, the meta-model's weighted F1-score reached 0.78 on test sets, and it reduced misclassifications of rare species by addressing dataset imbalances through stratified splitting and class weighting. This approach mitigated issues like the extremely low accuracy for species such as the bohor reedbuck at 0.000. These improvements are consistent with prior research (Norouzzadeh et al., 2018) and show robustness across diverse datasets, with false-positive rates in

binary detection decreasing by 20-25% (from 0.037 in ensembles to 0.015 after applying the meta-model).

The meta-model enhances generalisation by capturing inter-model interactions and maintaining consistency scores, which stabilises multi-geographic scenarios. However, it adds computational complexity, needing extensive resources for feature engineering and hyperparameter tuning with Gridsearchcv. Relying on high-quality meta-features may lead to overfitting to dominant species (for example, Thomson's gazelle with an ensemble accuracy of 0.853), which can reduce performance on less common classes, even with smoothing techniques. A key innovation of this framework is its open-source, locally deployable design, allowing ecologists to reduce manual annotations and improve biodiversity monitoring. Future improvements may include adaptive meta-features for real-time scalability, which will help better handle extremely rare cases, such as species with only one sample (8 instances).

The main problem causing this performance degradation in classification accuracy might be due to several reasons, which failed to meet the expected potential for classification robustness improvement, as the literature mentioned a 15-20% enhancement (Carl et al., 2022). First, overfitting was a potential reason that caused the leak during meta-model training. Due to this guess, leaked features may dominate the learning, resulting in poor generalisation of the model because of insufficient contributions from other features. Second, another potential reason would be imbalanced classes, which causes rare species constantly ignored during training. The lower F1 score (0.7824) than the accuracy rate (0.8111) can indicate this unbalanced effect (low weighted average F1) to some extent. Compared with the ensemble: The ensemble was 0.9351 (possibly due to the robustness of voting over imbalance), and the stacking decreased because the meta-layer magnified the rare class error.

# Chapter 6 - Conclusion

This thesis proposes a hierarchical ensemble framework designed to enhance wildlife classification in camera trap challenges, through integrating seven state-of-the-art open-source models with a meta-learning technique to further improve their ensemble robustness and identification accuracy in diverse global environments. This thesis comprehensively evaluates seven baseline models (e.g., PyTorch-Wildlife, MLWIC2, DeepFaune, SpeciesNet, and AddaxAI variants), then applies a hierarchical ensemble strategy with weighted averaging and meta-learning stacking (e.g., XGBoost) to effectively improve objective detection accuracy (animal and non-animal) and the potential of fine-grained rare species classification performance for rare species in the global camera trap challenges.

Meta-model stacking further enhanced stability, with the optimal XGBoost variant showing weighted F1 gains and balanced handling of class imbalances via stratified splitting and feature interactions. An improved binary detection enhancement was noticed compared to the ensemble weighted averaging period, although classification performance was unable to outperform the previous period due to some reasons.

This hierarchical framework highlights the reproducibility and extensibility. By introducing more robust open-source models, it effectively enhances wildlife detection accuracy and species classification scope, significantly reducing the manual annotation burden. Current limitations include dependency on model coverage for rare taxa and computational demands for large-scale deployment, as well as inadequate training of the meta-model. Future work could involve integration with emerging models to provide wider support for rare species.

# Chapter 7 - Future Work

To address these identified limitations and extend the framework's impact, several directions are proposed for future work.

First, to address issues of class imbalance, incorporating advanced data augmentation and synthetic generation (e.g., via GANs) could help mitigate these problems, as suggested by previous literature, which may further enhance accuracy with limited samples.

Second, more larger datasets (e.g., integrating LILA BC or iNaturalist) with diverse environmental features can be added to the experiment, as this can

Third, a stronger baseline model lays the foundation for this robust hierarchical ensemble framework, which also emphasises the potential for further optimisation by adding more well-trained, open-source classification models. This solution could further enhance the potential of this ensemble framework in rare species classification performance, thereby reducing ecologists' workload on manual annotation or verification.

# Chapter 8 - Personal and Professional Reflection

This reflection aims to discuss my knowledge accumulation and personal growth during the completion of this MSc Computing dissertation. By reviewing the achievements and limitations throughout this project development process with a critical thinking perspective, I can further promote my self-improvement for the rest of my life. This reflection is organised into three parts, presenting a detailed critical review of myself from three different aspects: my acquisition through this project, some insufficiencies that emerged during the research period, and how these experiences can benefit my future career.

My dissertation project focuses on wildlife monitoring, which has provided me with a deeper understanding of the practical applications of various deep learning models in real-world scenarios, thereby expanding my knowledge in this complex and professional research field. The requirement for extensive reading and research skills gained in this project has improved my habit of self-study and self-examination, contributing to enhanced personal development in problem identification and solving. Moreover, the coding challenges I faced when developing this hierarchical ensemble framework for wildlife classification tasks, such as how to apply a weighted averaging strategy and train a meta-model, helped me extend my knowledge and programming skills simultaneously, resulting in a more comprehensive technical background for my future career planning. Then, the high-quality requirement of this thesis writing gave me the opportunity to improve my English writing ability, which is a crucial and valuable achievement of my studying abroad experience as an international student. Apart from these, the dissertation motivated me to pursue my research goals with courage and persistence, enabling me to benefit through collaboration with my supervisor and explore the industry under his guidance in this innovative research area, which deeply broadened my horizons and knowledge boundaries.

However, some inadequacies remained during the execution of this thesis. Firstly, my limitation in English expression hindered good conversation and cooperation with my supervisor, which is a core challenge for me. As a result, I practise harder and do my best to overcome this drawback in my daily life. This issue allowed me to realise the significance of clear expression in collaborative works, which constantly

motivated me to be a better communicator and try to understand others from their perspective, providing me with a critical insight into my life. Second, in terms of the technical challenges, the main limitation I encountered during this project's development was the unsatisfactory experimental data in the meta-model stacking process, which highlighted my limitations in learning this state-of-the-art deep learning technique and an inappropriate application of this knowledge to address the practical question in my study. My shortcoming in using this helpful deep learning technique in this case motivated me to bridge this knowledge gap after further completion, becoming one of my ongoing challenges waiting to be addressed. Third, through conducting this important dissertation project, another personal drawback was exposed, which is the difficulty in maintaining a constant concentration on essential tasks. As I can perceive an obvious low efficiency during the mid-term of the project, this problem is due to my poor self-adjustment to my mood and pressure. A constant depression for a few months seriously lowers my ability to promote the project, significantly hindering my progress and achievements to some extent. This made me think about my life, and I believe a better solution can be found to address the following challenges and pressures in my work.

In the end, this dissertation project lays a significant foundation for my future career path, as this case provides me with several practical skills highlighted in the job market. First, I got more project management experience through implementing this whole project, from draft planning to a detailed execution schedule and a practical assessment criterion for completion. This is the first important project I have conducted so far, even though some mistakes still exist in the project, but it has given me more helpful support in optimising my ability for the next one. Second, the widespread use of practical Python skills enhances my technical abilities, which benefits my work as a data analyst in real life.

# References


[1] Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C. and 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), pp.E5716-E5725.

[2] Norouzzadeh, M.S., Morris, D., Beery, S., Joshi, N., Jojic, N. and Clune, J., 2021. A deep active learning system for species identification and counting in camera trap images. Methods in ecology and evolution, 12(1), pp.150-161.

[3] 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. and Teton, B., 2019. Machine learning to classify animal species in camera trap images: Applications in ecology. Methods in Ecology and Evolution, 10(4), pp.585-590.

[4] Willi, M., Pitman, R.T., Cardoso, A.W., Locke, C., Swanson, A., Boyer, A., Veldthuis, M. and Fortson, L., 2019. Identifying animal species in camera trap images using deep learning and citizen science. Methods in Ecology and Evolution, 10(1), pp.80-91.

[5] Schneider, S., Taylor, G.W., Linquist, S. and Kremer, S.C., 2019. Past, present and future approaches using computer vision for animal re-identification from camera trap data. Methods in Ecology and Evolution, 10(4), pp.461-470.

[6] Beery, S., Morris, D. and Yang, S., 2019. Efficient pipeline for camera trap image review. arXiv preprint arXiv:1907.06772.

[7] Vélez, J., McShea, W., Shamon, H., Castiblanco-Camacho, P.J., Tabak, M.A., Chalmers, C., Fergus, P. and Fieberg, J., 2023. An evaluation of platforms for processing camera-trap data using artificial intelligence. Methods in Ecology and Evolution, 14(2), pp.459-477.

[8] Tabak, M.A., Norouzzadeh, M.S., Wolfson, D.W., Newton, E.J., Boughton, R.K., Ivan, J.S., Odell, E.A., Newkirk, E.S., Conrey, R.Y., Stenglein, J. and Iannarilli, F., 2020. Improving the accessibility and transferability of machine learning algorithms for identification of animals in camera trap images: MLWIC2. Ecology and evolution, 10(19), pp.10374-10383.

[9] Binta Islam, S., Valles, D., Hibbitts, T.J., Ryberg, W.A., Walkup, D.K. and Forstner, M.R., 2023. Animal species recognition with deep convolutional neural networks from ecological camera trap images. Animals, 13(9), p.1526.

[10] Schneider, S., Greenberg, S., Taylor, G.W. and Kremer, S.C., 2020. Three critical factors affecting automated image species recognition performance for camera traps. Ecology and evolution, 10(7), pp.3503-3517.

[11] Ala-Mantila, S., 2024. Machine Learning for Camera Traps Image Analysis: Applications for Ecology and Conservation.

[12] Tan, M., Chao, W., Cheng, J.K., Zhou, M., Ma, Y., Jiang, X., Ge, J., Yu, L. and Feng, L., 2022. Animal detection and classification from camera trap images using different mainstream object detection architectures. Animals, 12(15), p.1976.

[13] Schneider, S., Taylor, G.W. and Kremer, S., 2018, May. Deep learning object detection methods for ecological camera trap data. In 2018 15th Conference on computer and robot vision (CRV) (pp. 321-328). IEEE.

[14] Whytock, R.C., Świeżewski, J., Zwerts, J.A., Bara-Słupski, T., Koumba Pambo, A.F., Rogala, M., Bahaa-el-din, L., Boekee, K., Brittain, S., Cardoso, A.W. and Henschel, P., 2021. Robust ecological analysis of camera trap data labelled by a machine learning model. Methods in Ecology and Evolution, 12(6), pp.1080-1092.

[15] Fennell, M., Beirne, C. and Burton, A.C., 2022. Use of object detection in camera trap image identification: Assessing a method to rapidly and accurately classify human and animal detections for research and application in recreation ecology. Global Ecology and Conservation, 35, p.e02104.

[16] Tabak, M.A., Falbel, D., Hamzeh, T., Brook, R.K., Goolsby, J.A., Zoromski, L.D., Boughton, R.K., Snow, N.P., VerCauteren, K.C. and Miller, R.S., 2022. CameraTrapDetectoR: Automatically detect, classify, and count animals in camera trap images using artificial intelligence. BioRxiv, pp.2022-02.

[17] Mulero-Pázmány, M., Hurtado, S., Barba-González, C., Antequera-Gómez, M.L., Díaz-Ruiz, F., Real, R., Navas-Delgado, I. and Aldana-Montes, J.F., 2025. Addressing significant challenges for animal detection in camera trap images: a novel deep learning-based approach. Scientific Reports, 15(1), p.16191.

[18] Rigoudy, N., Dussert, G., Benyoub, A., Besnard, A., Birck, C., Boyer, J., Bollet, Y., Bunz, Y., Caussimont, G., Chetouane, E. and Carriburu, J.C., 2023. The DeepFaune initiative: a collaborative effort towards the automatic identification of European fauna in camera trap images. European Journal of Wildlife Research, 69(6), p.113.

[19] Hernandez, A., Miao, Z., Vargas, L., Beery, S., Dodhia, R., Arbelaez, P. and Ferres, J.M.L., 2024. Pytorch-wildlife: A collaborative deep learning framework for conservation. arXiv preprint arXiv:2405.12930.

[20] Huettmann, F., 2018. Boosting, bagging and ensembles in the real world: An overview, some explanations and a practical synthesis for holistic global wildlife conservation applications based on machine learning with decision trees. Machine Learning for Ecology and Sustainable Natural Resource Management, pp.63-83.

[21] Yang, D.Q., Tan, K., Huang, Z.P., Li, X.W., Chen, B.H., Ren, G.P. and Xiao, W., 2021. An automatic method for removing empty camera trap images using ensemble learning. Ecology and Evolution, 11(12), pp.7591-7601.

[22] Sudeepa, P., Seema, P.D., Kumawat, S., Sah, M. and Choudhary, R.C., 2024, June. Hybrid Deep Learning framework for automated classification of wildlife Camera trap images. In 2024 15th

International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[23] Aboneh, T., Rorissa, A. and Srinivasagan, R., 2022. Stacking-based ensemble learning method for multi-spectral image classification. Technologies, 10(1), p.17.

[24] 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. and Teton, B., 2019. Machine learning to classify animal species in camera trap images: Applications in ecology. Methods in Ecology and Evolution, 10(4), pp.585-590.

[25] Lin, Y.K. and Yen, T.Y., 2023. A meta-learning approach for few-shot face forgery segmentation and classification. Sensors, 23(7), p.3647.

[26] Kim, J.I., Baek, J.W. and Kim, C.B., 2025. Hierarchical image classification using transfer learning to improve deep learning model performance for amazon parrots. Scientific Reports, 15(1), p.3790.

[27] Ganaie, M.A., Hu, M., Malik, A.K., Tanveer, M. and Suganthan, P.N., 2022. Ensemble deep learning: A review. Engineering Applications of Artificial Intelligence, 115, p.105151.

[28] Leorna, S. and Brinkman, T., 2022. Human vs. machine: Detecting wildlife in camera trap images. Ecological Informatics, 72, p.101876.

[29] Zualkernan, I., Dhou, S., Judas, J., Sajun, A.R., Gomez, B.R. and Hussain, L.A., 2022. An IoT system using deep learning to classify camera trap images on the edge. Computers, 11(1), p.13.

[30] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8(1), p.53.

[31] Schneider, S., Greenberg, S., Taylor, G.W. and Kremer, S.C., 2020. Three critical factors affecting automated image species recognition performance for camera traps. Ecology and evolution, 10(7), pp.3503-3517.

[32] Carl, J., Barratt, J., Wanner, P., Toepfer, C., Cairney, J. and Pfeifer, K., 2022. The effectiveness of physical literacy interventions: a systematic review with meta-analysis. Sports medicine, 52(12), pp.2965-2999.

[33] Meng, D.Y., Li, T., Li, H.X., Zhang, M., Tan, K., Huang, Z.P., Li, N., Wu, R.H., Li, X.W., Chen, B.H. and Ren, G.P., 2023. A method for automatic identification and separation of wildlife images using ensemble learning. Ecological Informatics, 77, p.102262.

[34] Curry, R., Trotter, C. and McGough, A.S., 2021, December. Application of deep learning to camera trap data for ecologists in planning/engineering–Can captivity imagery train a model which generalises to the wild?. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 4011-4020). IEEE.

[35] Ferreira, E.M., Duarte, E.L., Cunha, M.V., Mira, A. and Santos, S.M., 2023. Disentangling wildlife–cattle interactions in multi-host tuberculosis scenarios: systematic review and meta-analysis. Mammal Review, 53(4), pp.287-302.

[36] Degradation State Recognition of Piston Pump Based on ICEEMDAN and XGBoost - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Flow-chart-of-XGBoost_fig3_345327934 [accessed 3 Sept 2025]

[37] Brijesh Soni. 2023. Stacking to Improve Model Performance: A Comprehensive Guide on Ensemble Learning in Python. Available at: https://medium.com/@brijesh_soni/stacking-to-improve-model-performance-a-comprehensive-guide-on-ensemble-learning-in-python-9ed53c93ce28 [Accessed: 3 Sept 2025].

[38] Wyatt, T., 2021. Wildlife trafficking: A deconstruction of the crime, victims and offenders. Springer Nature.

# Appendices

*Table 7 - Classification of Base Models in Dataset (01_Caltech_Camera_Traps)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 92 | 1 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 94 | 1 |
| 02_MLWIC2 | 0.2615 | 0.0583 | 0.0443 | 2493 | 32 |
| 03_DeepFaune | 0.7710 | 0.3192 | 0.3253 | 2415 | 14 |
| 04_SpeciesNet | 0.9855 | 0.4071 | 0.4051 | 3379 | 16 |
| 05_Addax_Tasmanian_vertebrates | 0.4485 | 0.0714 | 0.0868 | 2736 | 28 |
| 06_Addax_Perivuan_Andes | 0.3197 | 0.0485 | 0.0599 | 3394 | 40 |

*Table 8 - Classification of Base Models in Dataset (02_NACTI)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 229 | 2 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 239 | 2 |
| 02_MLWIC2 | 0.7809 | 0.3711 | 0.3672 | 8629 | 25 |
| 03_DeepFaune | 0.3082 | 0.2292 | 0.2625 | 5087 | 18 |
| 04_SpeciesNet | 0.9393 | 0.4886 | 0.4844 | 9113 | 29 |
| 05_Addax_Tasmanian_vertebrates | 0.1618 | 0.0381 | 0.0521 | 9492 | 46 |
| 06_Addax_Perivuan_Andes | 0.1477 | 0.0312 | 0.0413 | 10922 | 43 |

*Table 9 - Classification of Base Models in Dataset (03_Snapshot_Serengeti)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 60 | 1 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 78 | 2 |
| 02_MLWIC2 | 0.0005 | 0.0067 | 0.0043 | 6274 | 32 |
| 03_DeepFaune | 0.0000 | 0.0000 | 0.0000 | 3023 | 18 |
| 04_SpeciesNet | 0.9221 | 0.5397 | 0.5311 | 7379 | 31 |
| 05_Addax_Tasmanian_vertebrates | 0.0000 | 0.0000 | 0.0000 | 10261 | 42 |
| 06_Addax_Perivuan_Andes | 0.0000 | 0.0000 | 0.0000 | 11377 | 43 |

*Table 10 - Classification of Base Models in Dataset (04_WCS_Camera_Traps)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.3176 | 0.1622 | 0.3333 | 85 | 2 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.2907 | 0.2252 | 0.5000 | 86 | 1 |
| 02_MLWIC2 | 0.0134 | 0.0152 | 0.0155 | 2090 | 34 |
| 03_DeepFaune | 0.1781 | 0.1369 | 0.1428 | 775 | 17 |
| 04_SpeciesNet | 0.8782 | 0.6051 | 0.5922 | 2094 | 84 |
| 05_Addax_Tasmanian_vertebrates | 0.0467 | 0.0196 | 0.0195 | 1692 | 53 |
| 06_Addax_Perivuan_Andes | 0.3959 | 0.3249 | 0.4069 | 2806 | 44 |

*Table 11 - Classification of Base Models in Dataset (05_Idaho_Camera_Traps)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 2 | 1 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 4 | 1 |
| 02_MLWIC2 | 0.6635 | 0.0985 | 0.0836 | 523 | 13 |
| 03_DeepFaune | 0.0000 | 0.0000 | 0.0000 | 560 | 8 |
| 04_SpeciesNet | 0.9569 | 0.4135 | 0.4016 | 696 | 7 |
| 05_Addax_Tasmanian_vertebrates | 0.0000 | 0.0000 | 0.0000 | 2019 | 13 |
| 06_Addax_Perivuan_Andes | 0.0102 | 0.0114 | 0.0130 | 1763 | 32 |

*Table 12 - Classification of Base Models in Dataset (06_Island_Conservation)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 40 | 2 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 46 | 2 |
| 02_MLWIC2 | 0.0430 | 0.0208 | 0.0207 | 2046 | 31 |
| 03_DeepFaune | 0.8920 | 0.3665 | 0.3579 | 778 | 12 |
| 04_SpeciesNet | 0.5061 | 0.2358 | 0.2339 | 1713 | 32 |
| 05_Addax_Tasmanian_vertebrates | 0.3081 | 0.0728 | 0.0790 | 1944 | 45 |
| 06_Addax_Perivuan_Andes | 0.0878 | 0.0247 | 0.0279 | 1845 | 42 |

*Table 13 - Classification of Base Models in Dataset (07_Wellington)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 249 | 2 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 237 | 1 |
| 02_MLWIC2 | 0.0053 | 0.0055 | 0.0066 | 3221 | 30 |
| 03_DeepFaune | 0.8555 | 0.2835 | 0.2865 | 1502 | 12 |
| 04_SpeciesNet | 0.8126 | 0.3540 | 0.3565 | 2487 | 14 |
| 05_Addax_Tasmanian_vertebrates | 0.6814 | 0.1056 | 0.1192 | 2734 | 38 |
| 06_Addax_Perivuan_Andes | 0.1393 | 0.0167 | 0.0166 | 2555 | 43 |

*Table 14 - Classification of Base Models in Dataset (08_ENA24)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 925 | 2 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 938 | 2 |
| 02_MLWIC2 | 0.2008 | 0.0813 | 0.0967 | 7526 | 36 |
| 03_DeepFaune | 0.7242 | 0.1971 | 0.1987 | 2513 | 14 |
| 04_SpeciesNet | 0.9542 | 0.5911 | 0.5920 | 8948 | 29 |
| 05_Addax_Tasmanian_vertebrates | 0.3230 | 0.0493 | 0.0616 | 6165 | 59 |
| 06_Addax_Perivuan_Andes | 0.0502 | 0.0209 | 0.0170 | 8135 | 43 |

*Table 15 - Classification of Base Models in Dataset (10_SWG)*

| model_name | Accuracy | Macro-F1 | Macro-Recall | Samples | Coverage |
|---|---|---|---|---|---|
| 01_PyTorch-Wildlife_MegaDetectorV5 | 0.0000 | 0.0000 | 0.0000 | 197 | 2 |
| 01_PyTorch-Wildlife_MegaDetectorV6-yolov10e | 0.0000 | 0.0000 | 0.0000 | 185 | 2 |
| 02_MLWIC2 | 0.0055 | 0.0019 | 0.0011 | 3469 | 25 |
| 03_DeepFaune | 0.5077 | 0.0574 | 0.0579 | 2133 | 16 |
| 04_SpeciesNet | 0.9757 | 0.6530 | 0.6563 | 5278 | 38 |
| 05_Addax_Tasmanian_vertebrates | 0.0000 | 0.0000 | 0.0000 | 2883 | 55 |
| 06_Addax_Perivuan_Andes | 0.0000 | 0.0000 | 0.0000 | 5074 | 45 |