



Automated Wildlife Species Classification Using Deep Neural Networks

Prepared By

Name: [Ahmad Mohammed Bashatah](#)

ID No.: [243001738](#)

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics Midocean University

UAE

Supervisor Name: [Dr. Ashraf Said](#)

Collage of Informatics, Midocean University, Comoros

Email: 243001738@st.midocean.edu.km

Acknowledgements

In the name of Allah, who illuminated my paths with the light of knowledge and aided me in crossing the seas of effort and patience. To my soul, which wandered in the horizon of ambition, stood firm against the storms of challenges, danced with the nights of determination, and completed this journey with a resolve as radiant as dawn.

To my father, the lighthouse that lit my paths with lessons of resilience and taught me to persevere on the road to success, and to my mother, the source of my inspiration and strength, the spirit who sang me anthems of hope with a tender voice that soothes the heart. You are my foundation, and this achievement is but a bouquet of roses offered to your flourishing garden—accept it from your son with gratitude and loyalty.

To my brothers and sisters, the branches of my tree shaded me with support and love. You are the stars of my journey, my joy, my melodies, and my happiness. Thank you for every smile that revived me, and for every hand you extended to me; your stars never fade, and I take pride in you.

To my beloved wife, the sea in which I sailed my dreams, and the shore that welcomes me with safety after every storm. Thank you for your patience that eased my fatigue, and for bearing the burdens while I was preoccupied. You are the light of my home, and your support was the wind that propelled me toward success; this achievement is a poem I dedicate to you with an endless love.

Finally, to my children, my three roses that bloomed in my small garden. To Muhammad, my eldest son, thank you for your wisdom that guided me like a compass; to Rudayna, my precious daughter, thank you for your smile that shines like the sun in my sky; and to Youssef, my little son, thank you for your energy that flows like a river overcoming rocks. You are my motivation to reach the peaks, and I hope to be for you a mountain that protects you; this achievement is a shield to guard you and a future I build for you. Thank you to all who contributed to this new dawn; without you, I would have been a tree without roots, but today I stand as a fruitful tree, nourished by your love and support.

Abstract

Biodiversity conservation increasingly depends on reliable, field-ready tools for rapid species identification under limited resources. Traditional, expert-driven workflows struggle with variable illumination, cluttered backgrounds, and fine-grained inter-class similarity. This thesis presents a compact, reproducible benchmarking pipeline—rather than a new architecture—that fine-tunes and evaluates representative vision families under a unified protocol on a curated, four-class dataset (*Plestiodon inexpectatus*, *Agkistrodon contortrix*, *Caretta caretta*, *Oreaster reticulatus*; splits: 582/84/171). We compare a classical CNN (ResNet18), efficient/mobile backbones (MobileNetV3-Large, EfficientNet-B0), a multi-scale network (InceptionV3), two transformers (ViT-B/32, swin_tiny_patch4_window7_224), a detector-native ecosystem in classification mode (YOLOv11n-cls (Ultralytics)), and a small Custom CNN baseline. Preprocessing and training are standardized (ImageNet initialization; inputs 224×224 for most models, 299×299 for InceptionV3, 640×640 for YOLOv11n-cls (Ultralytics); light augmentations; AdamW; StepLR).

Evaluation uses Accuracy, weighted Precision/Recall/F1, per-class Precision–Recall curves with macro-mAP, and confusion matrices. Results identify swin_tiny_patch4_window7_224 as the top performer (Accuracy %98.25, weighted F1 %98.3), closely followed by InceptionV3 (%97.66). MobileNetV3-Large and EfficientNet-B0 provide strong accuracy–efficiency trade-offs (% 95.9 each), with ResNet18 competitive (% 95.32) and YOLOv11n-cls (Ultralytics) solid though lower (% 94.15); ViT-B/32 and the Custom CNN baseline underperform on this small dataset. PR curves are near-saturated for the top models (macro-mAP = 1.0), and residual errors localize mainly to the *Agkistrodon*–*Plestiodon* pair, informing targeted data curation and thresholding.

By coupling aggregate metrics with deployment-oriented diagnostics (PR/mAP/CM), the pipeline turns benchmarking into guidance for model selection and error mitigation in resource-constrained settings, supporting early, accurate wildlife monitoring for conservation.

Table of Contents

1	
Acknowledgements	2
Abstract	3
List of Abbreviations.....	5
List of Figures	6
List of Tables.....	7
Chapter 1.	8
1.1 Thesis Objectives	8
1.2 Thesis Contribution.....	9
Chapter 2 Overview	13
Chapter 3 Related Work	16
3.1 Introduction	16
3.2 History	17
3.2.1 CNN-based classification on camera-trap imagery.....	17
3.2.2 Detection, cluttered scenes, and ecological context.....	17
3.2.3 Generalization to new domains and site shift	18
3.2.4 Data quality, imbalance, and evaluation practice.....	18
3.2.5 Domain-specific case studies and fine-grained taxa	18
3.2.6 Emerging directions: foundation models and zero-shot transfer.....	19
Chapter 4 Proposed Approach.....	20
4.1 Dataset	20
4.2 Preprocessing and Augmentation	21
4.3 Models and Training Settings	21
4.4 Evaluation Metrics and Procedure	22
4.5 Implementation and Reproducibility	22
4.6 Design Rationale and Trade-offs	23
Chapter 5 Experimental Results and Discussion	25
5.1 Overall Comparison.....	25
5.2 Precision–Recall curves & confusion matrix.....	27
5.3 Practical Implications	32
Chapter 6 Conclusion and Future Work.....	35
References	39

List of Abbreviations

AI — Artificial Intelligence
AP — Average Precision
AUC-PR — Area Under the Precision–Recall Curve
DL — Deep Learning
DNN — Deep Neural Network
EffNet-B0 — EfficientNet-B0
F1 — F1-score (harmonic mean of Precision and Recall)
FN — False Negative
FP — False Positive
GFLOPs — Giga Floating-Point Operations (computational cost)
GPU — Graphics Processing Unit
HPO — Hyperparameter Optimization
IncV3 — InceptionV3
kappa (κ) — Cohen’s Kappa Statistic
LR — Learning Rate
MCC — Matthews Correlation Coefficient
mAP (macro) — mean Average Precision (macro-averaged across classes)
ML — Machine Learning
MobileNetV3-L — MobileNetV3-Large
PR — Precision–Recall
PRC — Precision–Recall Curve
ResNet18 — Residual Network, 18 layers
ROC — Receiver Operating Characteristic (*mentioned only for context; primary analysis uses PR*)
SOTA — State Of The Art (*used descriptively where needed*)
StepLR — Step Learning-Rate scheduler
TL — Transfer Learning
TN — True Negative
TP — True Positive
Top-1 / Top-5 — Top-1/Top-5 Accuracy (classification rank metrics)
ViT-B/32 — Vision Transformer, Base, 32-patch
WD — Weight Decay (regularization)
YOLO — You Only Look Once
YOLOv11n-cls (Ultralytics) — YOLO in Classification mode (Ultralytics)

Confusion-Matrix Shortcuts (used in captions/figures):

CM — Confusion Matrix (counts)
NCM — Normalized Confusion Matrix (row-normalized)

Averaging schemes for metrics (as used in the thesis):

macro — Unweighted mean across classes
weighted — Class-frequency weighted mean across classes

List of Figures

Figure 4.1. Dataset classes	20
Figure 5.1. Accuracy, Precision, Recall, and F1 (weighted) for all models on the test set ($n = 171$).	26
Figure 5.2. swin_tiny_patch4_window7_224 — Precision–Recall Curves & Confusion Matrix.	28
Figure 5.3. InceptionV3 — Precision–Recall Curves & Confusion Matrix.	28
Figure 5.4. EfficientNet-B0— Precision–Recall Curves & Confusion Matrix.	29
Figure 5.5. MobileNetV3-Large — Precision–Recall Curves & Confusion Matrix.	29
Figure 5.6. ResNet18 — Precision–Recall Curves & Confusion Matrix.	30
Figure 5.7. YOLOv11n-cls (Ultralytics) — Precision–Recall Curves & Confusion Matrix.	30
Figure 5.8. ViT-B/32 — Precision–Recall Curves & Confusion Matrix.	31
Figure 5.9. Custom CNN baseline — Precision–Recall Curves & Confusion Matrix.	31

List of Tables

- **Table 4.1.** Dataset split per class and partition (Train/Validation/Test; support on test set). 25
- **Table 6.1.** Overall test performance of all models (Accuracy, Precision (weighted), Recall (weighted), F1 (weighted)). 36

Chapter 1.

1.1 Thesis Objectives

The accelerating loss of biodiversity, driven by habitat destruction and environmental changes, is a pressing environmental concern, particularly in field settings with limited resources where rapid and accurate species identification is critical. The classification of visually similar wildlife species, such as *Plestiodon inexpectatus*, *Agkistrodon contortrix*, *Caretta caretta*, and *Oreaster reticulatus*, poses significant challenges due to their shared visual characteristics and the variability in capture conditions. Controlling biodiversity loss requires precise and timely species identification, especially considering how similar these species appear to one another. Traditional identification methods, while effective in controlled settings, often rely on expert knowledge and specialized equipment, which may not be readily available in resource-constrained environments. Consequently, there is growing interest in developing reliable and automated classification models based on Deep Learning (DL) and Machine Learning (ML) approaches to enable fast and accurate species detection. Early classification of wildlife species can be crucial in monitoring biodiversity and supporting conservation efforts. Traditional species identification is based on visual assessment by trained experts. However, this method can be time-consuming, subjective, and prone to errors.

DL techniques have shown promising results in various image classification tasks in recent years. Convolutional Neural Networks (CNNs) are a widely used DL paradigm in image classification, capable of automatically extracting features from raw image data and using these features to classify images into different categories. DL models for wildlife species classification have been developed in several studies, utilizing curated datasets for model training and validation. These studies evaluated the accuracy of pre-trained DL models, where

models like ResNet-18 and InceptionV3 achieved notable performance, while others, such as SqueezeNet and VGG16, were tested for similar classification tasks. Ensemble systems combining models like VGG-16, ResNet50, and InceptionV3 have also been explored for classifying visually similar species, with ResNet50 often achieving high accuracy. Despite these advancements, more robust models are needed to handle variable image quality and limited data availability, particularly in field settings. Feature extraction using Naive Bayes (NB) and Support Vector Machine (SVM) from models like VGG16Net and AlexNet has also been investigated.

Moreover, datasets comprising multiple species classes have facilitated the development of multi-class classification models. While some studies reported that MobileNetV2 outperformed VGG16 and VGG19, others highlighted the effectiveness of ResNet18. Even with these developments, a comprehensive model that integrates the strengths of DL, ML, and optimization techniques remains necessary. To address this gap, we design a compact, reproducible benchmarking pipeline that systematically fine-tunes and evaluates representative architectures—including a `swin_tiny_patch4_window7_224`—under a unified training and evaluation protocol. Rather than proposing a new architecture, the study emphasizes fair comparison, small-data behavior, and deployment-oriented diagnostics (per-class Precision–Recall curves, macro-mAP, and confusion matrices). We adopt standard transfer-learning and principled fine-tuning settings (AdamW, step-LR, light augmentations) chosen for stability, simplicity, and cross-model fairness.

1.2 Thesis Contribution

There is an urgent need for more precise, interpretable, and flexible identification methods for wildlife species, especially in light of the escalating threats to global biodiversity. Although

DL models have demonstrated potential in classifying species from images, they often face limitations such as reliance on large, labeled datasets, poor interpretability, and reduced performance in settings with varying image quality. Even when reliable, traditional methods are often impractical in resource-limited environments due to the need for specialized expertise and tools. Moreover, the biodiversity monitoring field lacks a robust model that effectively combines the strengths of optimization, ML, and DL. Leveraging the hierarchical processing capabilities of the `swin_tiny_patch4_window7_224` and optimized fine-tuning, the Swin-based model aims to address these challenges by enhancing identification accuracy, efficiency, and performance while ensuring the solution is interpretable and adaptable to various datasets.

The main contributions of this thesis are summarized as follows:

- We introduce a compact, reproducible benchmark that evaluates classical CNNs, efficient/mobile backbones, Inception-style multi-scale networks, transformer-based models, and a YOLO-Classification ecosystem under a unified protocol, enabling fair, apples-to-apples comparison under small-data constraints..
- The benchmark identifies `swin_tiny_patch4_window7_224` as the top performer on our held-out test split (Accuracy %98.25; weighted Precision/Recall/F1 = %98.3/ %98.2/ %98.3) and quantifies accuracy–efficiency trade-offs for MobileNetV3-Large and EfficientNet-B0 (%95.9), with ResNet-18 competitive (%95.3). These results provide balanced, deployment-oriented guidance rather than claiming a new architecture.

1.3 Thesis Organization

The remainder of this thesis is structured to move from motivation and context to a reproducible methodology and an evidence-based comparison of models, ending with implications and future directions.

Chapter 2 provides a concise overview of the problem of wildlife image classification and frames the specific four-class setting studied here (*Plestiodon inexpectatus*, *Agkistrodon contortrix*, *Caretta caretta*, *Oreaster reticulatus*). It summarizes the dataset split used throughout (train 582, validation 84, test 171), outlines the evaluation criteria (Accuracy, weighted Precision/Recall/F1, per-class Precision–Recall curves with macro-mAP, and confusion matrices), and clarifies the study’s scope—static images only, single-label classification, and no detection or video analysis.

Chapter 3 surveys related work with an emphasis on deep image classifiers that are relevant to our setting, covering classic CNNs and modern transformer-based approaches. It situates the chosen backbones—ResNet18, MobileNetV3-Large, EfficientNet-B0, InceptionV3, ViT-B/32, `swin_tiny_patch4_window7_224`—and the Ultralytics YOLO classification variant within prior practice, highlighting common challenges such as small datasets, visually similar species, and generalization under varying backgrounds and viewpoints. This chapter motivates the need for a controlled, apples-to-apples comparison under a unified standard.

Chapter 4 presents the proposed approach in detail. It specifies preprocessing and augmentation (ImageNet normalization; 224×224 for most models, 299×299 for InceptionV3, 640×640 for YOLOv11n-cls (Ultralytics); light flips and $\pm 10^\circ$ rotations), and documents the training setup (ImageNet initialization, Cross-Entropy loss—with auxiliary logits handled for InceptionV3 where applicable—AdamW optimization, StepLR scheduling, and epoch budgets: 20 for PyTorch models, 50 for YOLOv11n-cls (Ultralytics)). Model-specific heads are adapted to the four classes, and all experiments follow the same validation-driven checkpointing before a single pass on the held-out test split ($n=171$) set. Implementation details (PyTorch 2.6, Ultralytics 8.3.x, Tesla T4) and reproducibility choices (fixed seeds, deterministic flags where possible) are recorded to enable exact replication.

Chapter 5 reports the experimental results and discussion. It first provides the overall comparison on the test split, where `swin_tiny_patch4_window7_224` attains the highest performance accuracy %98.25, followed closely by InceptionV3 %97.66; MobileNetV3-Large and EfficientNet-B0 form a strong lightweight tier %95.9, with ResNet18 %95.32 and YOLOv11n-cls (Ultralytics) %94.15 trailing slightly; ViT-B/32 and a Custom CNN baseline underperform on this small dataset %62 and %58.5. The chapter then analyzes per-class PR curves and macro-mAP to assess threshold-robust separability, and examines confusion matrices to localize residual errors—most notably the limited ambiguity between *Agkistrodon contortrix* and *Plestiodon inexpectatus*. The discussion translates these figures into practical guidance on model choice under accuracy or resource constraints, threshold tuning, and data-centric fixes focused on the observed error modes.

Chapter 6 concludes the thesis by synthesizing the empirical findings and their implications for field deployment. It reflects on why hierarchical transformers (Swin) excelled in this small-data setting, why InceptionV3 remained a strong and stable alternative, and how compact CNNs balanced accuracy with efficiency. The chapter also outlines concrete future work aligned with the evidence gathered here, including targeted data expansion for the hardest class pair, modest augmentation refinements, confidence calibration and per-class thresholding, and lightweight ensembling when maximal accuracy is required.

Chapter 2 Overview

This thesis addresses the practical problem of multi-class wildlife image classification under the constraints that typically characterize field work: small, carefully curated datasets; variable capture conditions; and visually similar species that are easy to confuse. We focus on four target taxa—*Plestiodon inexpectatus*, *Agkistrodon contortrix*, *Caretta*, and *Oreaster reticulatus*—and construct a reproducible pipeline that trains and evaluates modern vision architectures. The dataset is deliberately compact and balanced to emulate realistic deployment scenarios, with total 837 images as 582 for training, 84 for validation, and 171 for testing.

The methodology compares complementary architectural families rather than a single model. Classical convolution is represented by ResNet18; efficiency-oriented designs by MobileNetV3-Large and EfficientNet-B0; multi-scale factorization by InceptionV3; and transformers by both ViT-B/32 and the hierarchical swin_tiny_patch4_window7_224. We also include YOLO-Classification (Ultralytics) to gauge the performance and engineering convenience of a detector-native ecosystem when used purely for classification, and a small Custom CNN baseline to anchor results. All models are fine-tuned from ImageNet initialization; inputs are standardized with ImageNet statistics and resized to 224×224 for most backbones, 299×299 for InceptionV3, and 640×640 for YOLOv11n-cls (Ultralytics). Augmentation is intentionally conservative—horizontal/vertical flips and small rotations—to promote invariance to pose without distorting species-diagnostic color and texture cues. Optimization uses AdamW with stepwise learning-rate decay; for YOLOv11n-cls (Ultralytics) we adopt Ultralytics’ recommended configuration. Limited, targeted hyperparameter exploration is employed to preserve a fair comparison and reproducibility.

Evaluation is multi-faceted by design. In addition to overall Accuracy, we report weighted Precision, Recall, and F1, and we analyze per-class Precision–Recall (PR) curves and macro-

averaged Average Precision (mAP) as threshold-independent summaries of ranking quality. Confusion matrices serve as a complementary diagnostic that exposes systematic error modes between visually similar taxa. Models are selected by validation accuracy and tested once on the held-out split to avoid optimistic bias.

A consistent picture emerges from the experiments. The `swin_tiny_patch4_window7_224` attains the strongest results on the test set (98.25 accuracy; macro-mAP = 0.999), reflecting the advantage of hierarchical attention and local–global context modeling in the small-data regime. InceptionV3 follows closely (97.66, confirming the robustness of multi-scale factorization and auxiliary logits for transfer learning. Lightweight backbones—MobileNetV3-Large and EfficientNet-B0—achieve 95.9 with favorable compute and memory footprints, while ResNet18 remains a competitive classical baseline (95.3). YOLOv11n-cls (Ultralytics) offers a streamlined training pipeline and solid performance (94.15, albeit slightly behind the best backbones. As expected, ViT-B/32 and the custom Custom CNN baseline underperform (62 and 8.5, aligning with known data requirements and inductive biases.

The qualitative analyses reinforce these quantitative findings. PR curves for the top models are flat and near the (1.0, 1.0) corner across classes, indicating high precision at high recall and stable decision boundaries. Confusion matrices reveal a recurring ambiguity between *Agkistrodon* and *Plestiodon*, two reptile classes with overlapping visual traits and backgrounds, and an occasional *Oreaster* → *Caretta* confusion when scene context dominates the signal. These patterns suggest concrete avenues for improvement—targeted data curation for the difficult pairs, background-aware preprocessing (e.g., cropping/segmentation), and class-aware loss formulations—without altering the overall pipeline.

Beyond reporting numbers, the contribution of this thesis is a reproducible, deployment-oriented benchmark that spans diverse model families under identical conditions, couples aggregate

metrics with PR/CM visualizations that are actionable at class level, and demonstrates that hierarchical transformers (Swin) are particularly effective for compact, real-world ecological datasets. The results provide clear guidance for model selection: Swin or Inception when accuracy is paramount; MobileNetV3-Large or EfficientNet-B0 when efficiency matters; and YOLOv11n-cls (Ultralytics) when engineering simplicity and ecosystem tooling are decisive

Chapter 3 Related Work

3.1 Introduction

Automated wildlife identification has advanced quickly at the intersection of computer vision and conservation science, yet practical deployment in the field remains challenging. Camera-trap images and opportunistic field photographs vary widely in illumination, pose, occlusion, and background clutter, while fine-grained visual similarities between species further complicate recognition. These conditions magnify the effects of limited and unevenly sampled datasets and make generalization to new sites and seasons non-trivial. In response, the community has moved from hand-crafted features to transfer-learned deep networks and from single-backbone reports to broader benchmarks that jointly consider accuracy, robustness, and efficiency. Foundational studies show that convolutional neural networks (CNNs) fine-tuned from large natural image pretraining substantially reduce the data burden typical of ecological applications [1], [2], [9]. Subsequent work explored architectural refinements and training standards, together with citizen-science annotation at scale to address labeling bottlenecks [9], [10]. Parallel lines of research examined domain shift—how models degrade when tested at new locations—and proposed evaluation regimes that reflect such “terra incognita” conditions [4], as well as object-detection pipelines tailored to ecological scenes [3]. Reviews of image-based species identification summarize recurring issues such as class imbalance, dataset bias, and the need for metrics beyond accuracy alone [5]. More recent perspectives highlight the integration of deep learning with conservation workflows and the emergence of transformer/foundation-model directions that may enable transfer to rare or unseen taxa [8], [11]. Within this landscape, our thesis positions itself as a compact, reproducible benchmark over four visually related species under a unified standard, aligning with the methodological thrust of the above literature while focusing on small-data, field-relevant constraints

3.2 History

3.2.1 CNN-based classification on camera-trap imagery

A landmark study demonstrated that deep CNNs can automatically identify, count, and describe animals in camera-trap images at scale, establishing the viability of end-to-end learning for ecology and showing substantial accuracy when large, annotated sets are available [1]. Building on this, transfer-learning workflows—fine-tuning ImageNet-pretrained backbones—proved effective in reducing annotation demands and improving performance on modest datasets typical of ecological deployments [2]. Complementary to these pipelines, combining deep learning with citizen-science annotation created practical pathways to curate large, labeled corpora while maintaining competitive accuracy, thereby addressing a key bottleneck in ecological ML [10]. Collectively, these works formalized the now-standard recipe adopted in our study: use of pretrained CNNs, careful train/val/test splits, and evaluation beyond a single metric.

3.2.2 Detection, cluttered scenes, and ecological context

Beyond whole-image classification, object-detection methods were adapted to ecological camera-trap data to localize animals in cluttered backgrounds [3]. While detectors can reduce background interference and support downstream species classification, studies report that dense vegetation, occlusions, and small targets remain challenging, motivating robust preprocessing and augmentation strategies even for pure classification pipelines like ours.

3.2.3 Generalization to new domains and site shift

A core difficulty for ecological AI is performance under domain shift. The “Recognition in Terra Incognita” paradigm explicitly evaluated models when train and test images originate from disjoint locations, revealing sizable drops under site shift and advocating for benchmark standard that measure out-of-distribution generalization [4]. These findings informed our decision to report threshold-independent metrics (per-class precision–recall curves, macro-mAP) and to analyze confusion patterns, not just aggregate accuracy.

3.2.4 Data quality, imbalance, and evaluation practice

Surveys of image-based species identification emphasize the impact of class imbalance, acquisition bias, and heterogeneous imaging conditions on both training and evaluation [5]. In addition to accuracy, they recommend weighted precision/recall/F1, per-class PR curves, and confusion matrices to surface asymmetric, class-specific failure modes—exactly the diagnostic suite we employ in our experiments. More broadly, programmatic discussions situate ML within conservation pipelines, arguing for pragmatic model choices, robust validation, and transparent reporting to bridge the lab-to-field gap [8].

3.2.5 Domain-specific case studies and fine-grained taxa

Domain case studies on herpetofauna show that modern CNN backbones (e.g., residual networks) fine-tuned with modest augmentations can exceed %85 accuracy on curated camera-trap datasets, but that performance is sensitive to class similarity and data scale [6]. Earlier conceptual treatments in animal ecology articulated the promise of computer vision for

population monitoring and behavioral inference, foreshadowing today's species-level classifiers and motivating careful metric selection and error analysis [7].

3.2.6 Emerging directions: foundation models and zero-shot transfer

Recent work explores aligning wildlife perception with language and vision–language models to support few- and zero-shot transfer across taxa and datasets [11]. While our study focuses on supervised fine-tuning in a small-data regime, these trends underscore opportunities for future extensions—e.g., leveraging foundation models to improve generalization to rare species or new sites when labeled data are scarce.

Chapter 4 Proposed Approach

This chapter details the methodological design of a compact, reproducible pipeline for multi-class wildlife image classification. The guiding objective is to evaluate, under a single controlled standard, diverse architectural families that are representative of contemporary practice—classical CNNs, efficient/mobile backbones, Inception-style multi-scale networks, transformer-based models, and YOLO-Classification—while keeping the standard faithful to the constraints of small, balanced datasets commonly encountered in field settings. The design emphasizes comparability across models, threshold-independent diagnostics (per-class PR curves and macro-mAP), and deployment-oriented transparency through confusion-matrix analysis, all with careful attention to reproducibility.

4.1 Dataset



Figure 4.1. Dataset classes

The study focuses on four visually related species—*Plestiodon inexpectatus*, *Agkistrodon contortrix*, *Caretta caretta*, and *Oreaster reticulatus*. The dataset is deliberately compact and approximately balanced to mirror realistic deployment conditions rather than large-scale web crawls. The Total Images of 837 are split into 582 for training, 84 for validation, and 171 for testing, with test support of 43/43/43/42 images per class. The split is held fixed throughout to enable fair, repeatable comparisons; only the training and validation subsets are used for model selection, whereas the test set remains untouched until the final evaluation.

4.2 Preprocessing and Augmentation

All images are standardized using ImageNet mean and variance after resizing to model-appropriate resolutions. Inputs are set to 224×224 for the majority of backbones, 299×299 for InceptionV3 (to conform with its canonical configuration), and 640×640 for the YOLO-Classification training pipeline. Augmentation is intentionally conservative: horizontal and vertical flips and small in-plane rotations ($\pm 10^\circ$) promote invariance to pose and viewpoint while preserving species-diagnostic chromatic and textural cues. Color jittering and aggressive policies (e.g., heavy RandAugment) are intentionally avoided to prevent distortions that could confound fine-grained distinctions between taxa.

4.3 Models and Training Settings

The backbone set spans complementary inductive biases. ResNet18 serves as a robust classical Custom CNN baseline with residual connections that stabilize optimization. MobileNetV3-Large and EfficientNet-B0 probe the accuracy–efficiency frontier through depthwise separable convolutions, squeeze-and-excitation, and compound scaling. InceptionV3 provides multi-scale factorization and, when enabled, auxiliary logits that may regularize optimization with 299×299 inputs. ViT-B/32 represents a plain vision transformer with global self-attention over patch tokens, while `swin_tiny_patch4_window7_224` introduces hierarchical, windowed attention with shifts that recover beneficial locality priors for small datasets. The comparison is rounded out by YOLO11n-Classification from the Ultralytics ecosystem, included to assess the practicality of a detector-native framework used purely for classification, and by a compact Custom CNN baseline that anchors the lower bound in the small-data regime.

All PyTorch backbones are initialized from ImageNet weights and adapted by replacing the final layer with a four-class head. Training uses cross-entropy loss; for InceptionV3, the auxiliary branch

is supported when present. Optimization adopts AdamW with a modest base learning rate ($1e-4$) and a step-wise decay schedule (StepLR, step size 3, gamma 0.97). Epoch budgets are kept fixed across families to preserve comparability, with 20 epochs for PyTorch models and 50 epochs for YOLOv11n-cls (Ultralytics) (following Ultralytics defaults, which typically correspond to an effective AdamW learning rate around 1.25×10^{-3}). Batch sizes are tuned to saturate a single Tesla T4 (15 GB) GPU without gradient accumulation. These choices reflect a deliberate balance between rigor and practicality: they are strong enough to reach high accuracy on a compact dataset, yet simple enough to remain reproducible and fair across diverse architectures.

4.4 Evaluation Metrics and Procedure

Model selection is performed strictly by validation accuracy. The test split is evaluated once using the best checkpoint to avoid optimistic bias and to emulate deployment. Evaluation goes beyond overall accuracy to include weighted precision, weighted recall, and weighted F1, which better reflect per-class balance and asymmetric errors. To obtain threshold-independent insights, precision–recall (PR) curves are computed for each class, and the macro-averaged average precision (mAP) is reported as a compact summary statistic across classes. Confusion matrices, both raw counts and row-normalized, are analyzed to expose systematic confusions between visually similar taxa that PR curves may not fully capture. Together, these metrics provide a comprehensive picture of ranking quality, decision thresholds, and error modes.

4.5 Implementation and Reproducibility

The pipeline is implemented in PyTorch 2.6 for all backbones and Ultralytics 8.3.x for YOLO-Classification, executed on Google Colab with a Tesla T4 GPU. To promote reproducibility, random seeds are fixed across NumPy and PyTorch, CuDNN is set to deterministic mode where feasible, preprocessing is frozen, and all logs, tables, and figures produced during training and

evaluation are saved. Model-specific input resolutions are enforced consistently (224×224, 299×299, or 640×640), and the same augmentation and optimization policies are applied across families, with YOLOv11n-cls (Ultralytics) following its standard Ultralytics configuration. Where time permits, light hyper-parameter exploration (e.g., via Optuna on a representative backbone) is used to quantify attainable gains without turning the comparison into a hyper-parameter race.

4.6 Design Rationale and Trade-offs

The selection of backbones is motivated by complementary strengths aligned with small, balanced datasets. `swin_tiny_patch4_window7_224` brings hierarchical attention that captures local–global interactions and has repeatedly shown strong behavior when data are limited, which is reflected later in its top performance. InceptionV3 offers a mature, stable alternative whose factorized convolutions enrich receptive fields at moderate computational cost; its canonical 299×299 input and auxiliary branch often yield smooth optimization. MobileNetV3-Large and EfficientNet-B0 provide practical paths to deployment when inference cost matters, typically retaining much of the accuracy of larger CNNs. ResNet18 remains a competitive classical baseline that is simple to train and interpret. YOLOv11n-cls (Ultralytics) lowers engineering overhead and integrates naturally with an ecosystem widely used in field workflows, albeit with a small accuracy gap to the best backbones on compact datasets. ViT-B/32 and the custom Custom CNN baseline are included as informative lower bounds; their results are consistent with known data requirements and inductive biases.

Viewed holistically, the approach is designed to produce actionable insights rather than isolated accuracy numbers. The unified standard ensures that architectural differences—not confounding choices in preprocessing or optimization—drive the observed behavior. The combination of weighted metrics, per-class PR curves with macro-mAP, and confusion-matrix analysis yields class-level diagnostics that directly inform data curation (e.g., collecting harder negatives for the

most confusable pairs), preprocessing refinements (e.g., background-aware cropping or segmentation), and loss design (e.g., class-aware or focal objectives). The methodology thus connects principled benchmarking to practical decision-making, paving the way for the experimental results and discussion that follow.

Chapter 5 Experimental Results and Discussion

This chapter reports the empirical findings obtained under a single, controlled standard on the held-out test (171 images). The goal is not merely to rank models by accuracy, but to understand how different architectural families behave on a compact, nearly balanced dataset of four visually related species. To that end, we complement aggregate metrics with per-class Precision–Recall (PR) curves and confusion-matrix analyses, and we interpret the patterns that recur across backbones. The section closes by translating the evidence into concrete, deployment-oriented recommendations.

5.1 Overall Comparison

Metrics are reported as weighted averages across the four classes (Accuracy, Precision, Recall, F1 Score). Table 5.1. show overall performance on the held-out test split (n=171)split.

Model	Accuracy	Precision (w)	Recall (w)	F1 (w)
Swin Transformer (swin_tiny_patch4_window7_224)	0.98246	0.983	0.982	0.98245
InceptionV3	0.97661	0.977	0.977	0.97667
MobileNetV3-Large	0.95906	0.959	0.959	0.95913
EfficientNet-B0	0.959	0.959	0.959	0.959
ResNet-18	0.95322	0.954	0.953	0.95318
YOLOv11n-cls (Ultralytics, yolov11n-cls.pt)	0.94152	0.943	0.942	0.94127
Vision Transformer (ViT-B/32)	0.61988	0.629	0.62085	0.61695
Custom Custom CNN baseline	0.5848	0.597	0.586	0.58565

Table 5.1. Overall performance on the held-out test split (n=171).

Under identical training and evaluation settings, Figure 5.1. shows a clear performance hierarchy emerges for all models on the test set ($n = 171$). Swin_tiny_patch4_window7_224 achieves the strongest results with Accuracy %98.25 and weighted Precision %98.3, Recall %98.2 and F1 Score %98.3. InceptionV3 follows closely at %97.66, reflecting the stability of its multi-scale factorization and canonical 299×299 configuration. The efficiency-oriented models—MobileNetV3-Large and EfficientNet-B0—form a robust middle tier at %95.9, while the classical baseline ResNet18 attains %95.32. YOLO-Classification reaches %94.15, offering a streamlined pipeline with slightly lower accuracy on this dataset. As anticipated in small-data regimes, ViT-B/32 and the custom Custom CNN baseline underperform %62 Threshold-independent summaries corroborate the same ordering. The macro-mAP is essentially saturated for swin_tiny_patch4_window7_224 is (0.999) and remains very high for InceptionV3 (0.996), MobileNetV3-Large (0.996), EfficientNet-B0 (0.992), and ResNet18 (0.988). YOLOv11n-cls (Ultralytics) is lower (0.976), while ViT-B/32 (0.700) and the Custom CNN baseline (0.611) display the expected drop. Taken together, these results indicate that hierarchical attention (Swin) and mature multi-scale CNN design (InceptionV3) are particularly effective for compact, balanced ecological datasets, with mobile architectures providing attractive accuracy–efficiency trade-offs.

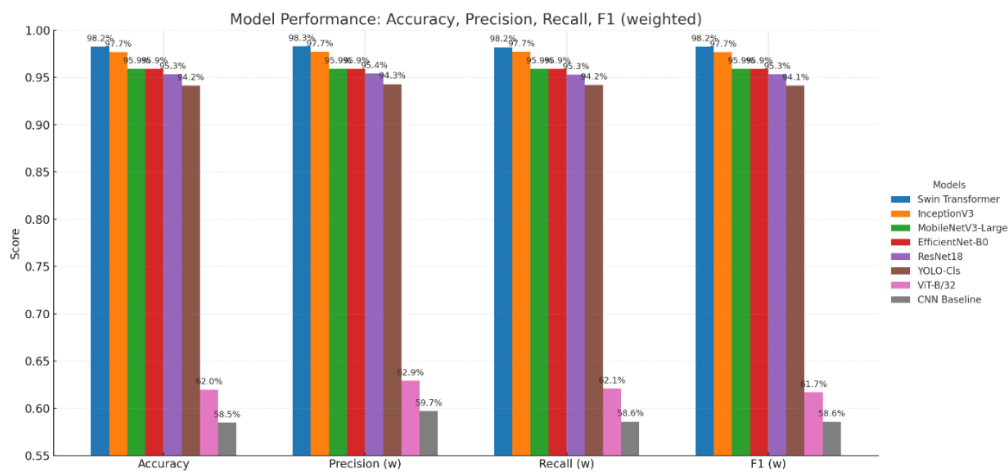


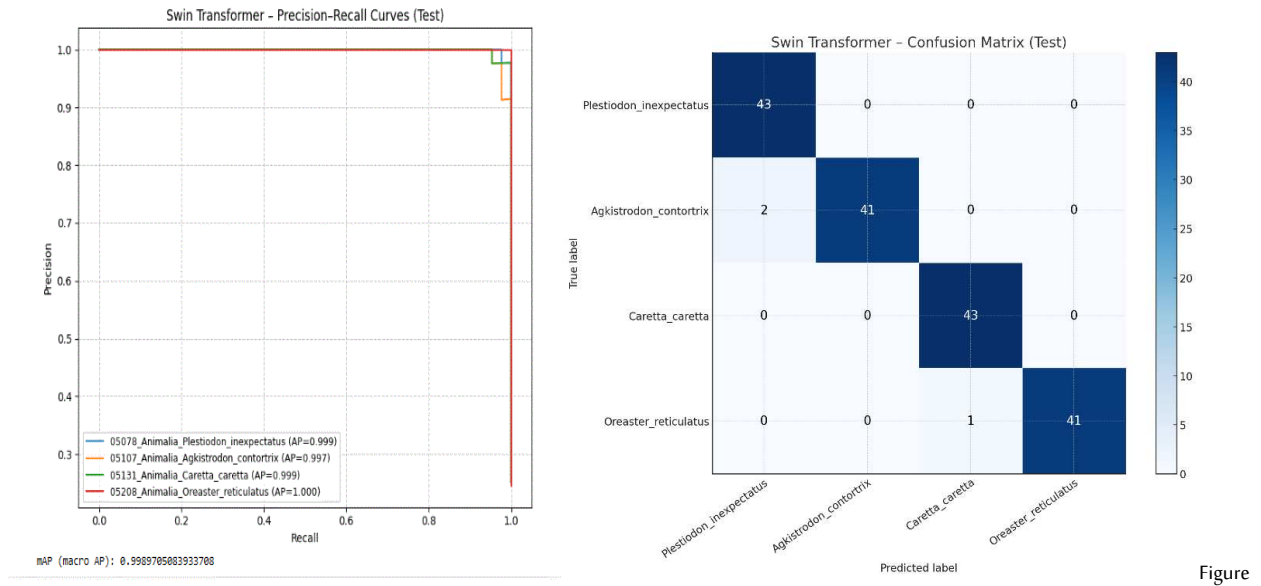
Figure 5.1. Accuracy, Precision, Recall, and F1 (weighted) for all models on the test set ($n = 171$).

5.2 Precision–Recall curves & confusion matrix

For each model, this thesis presents per-class Precision–Recall (PR) curves alongside the corresponding confusion matrix (CM). Read together, the PR panels summarize threshold-robust separability for each class (curve shape and area), while the CM localizes the remaining errors at the chosen operating point. Jointly, they clarify class difficulty, operating-threshold robustness, and where targeted remediation is most impactful. Across the best-performing architectures—swin_tiny_patch4_window7_224 (Figure 5.2), InceptionV3 (Figure 5.3), MobileNetV3-Large (Figure 5.4), EfficientNet-B0 (Figure 5.5), and ResNet18 (Figure 5.6)—the PR curves are near-rectangular and closely track the top-right corner for all four species, indicating that very high recall does not meaningfully compromise precision. The macro-mAP values reflect this threshold robustness: =0.999 for Swin, =0.996 for InceptionV3 and MobileNetV3-Large, =0.992 for EfficientNet-B0, and =0.988 for ResNet18. The paired CMs corroborate this picture: most mass lies on the diagonal, with only a few, tightly localized off-diagonals.

A consistent pattern is the *Agkistrodon contortrix* ↔ *Plestiodon inexpectatus* boundary. In the PR panels, *Agkistrodon*’s curve typically bends slightly earlier at very high recall, suggesting borderline positives that share texture/pose with *Plestiodon*. In the CMs, this appears as a compact off-diagonal block between the two labels. A milder, sporadic confusion between *Oreaster reticulatus* and *Caretta caretta* is observed in some models when background cues dominate; their PR curves remain flatter and closer to (1,1), which explains the relative rarity of these off-diagonals.

Looking at Figure 5.2. PR curves for swin_tiny_patch4_window7_224 are nearly ideal across classes. The CM shows only a handful of off-diagonals often perfect recognition for *Plestiodon* and *Caretta*, with 1–2 slips clustered in the *Agkistrodon*↔*Plestiodon* cell and occasional *Oreaster*→*Caretta* assignments.



5.2. swin_tiny_patch4_window7_224 — Per-class Precision–Recall curves on the test set & Confusion matrix

Figure 5.3 refers that InceptionV3 shows Very similar behavior to Swin; PR curves are only marginally less saturated. The CM typically shows symmetric one-off swaps within the Agkistrodon/Plestiodon pair, with rare Caretta→Agkistrodon or Oreaster→Caretta cases.

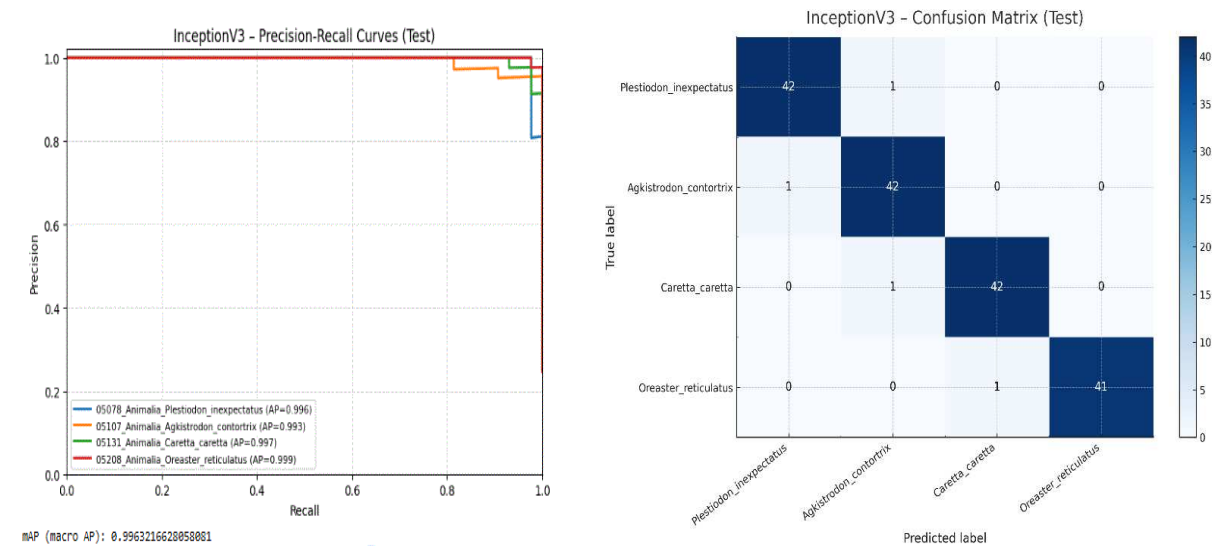


Figure 5.3. InceptionV3 — Per-class Precision–Recall curves & Confusion matrix

For EfficientNet-B0 in Figure 5.4, PR curves remain very high for all classes. The CM shows a slightly thicker off-diagonal band for Agkistrodon↔Plestiodon relative to Swin/Inception, yet the confusion remains compact and predictable.

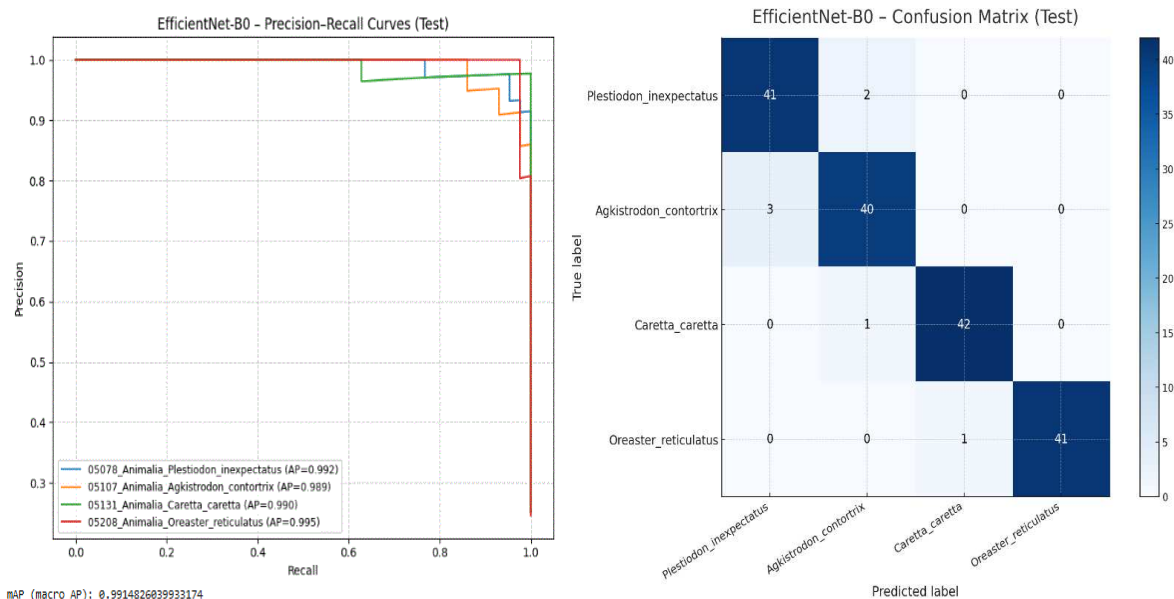


Figure 5.4. EfficientNet-B0 — Per-class Precision–Recall curves & Confusion matrix

PR curves for MobileNetV3-Large in Figure 5.5 remain very high for all classes. The CM is similar to MobileNetV3-Large, with a modest band on the Agkistrodon↔Plestiodon cell.

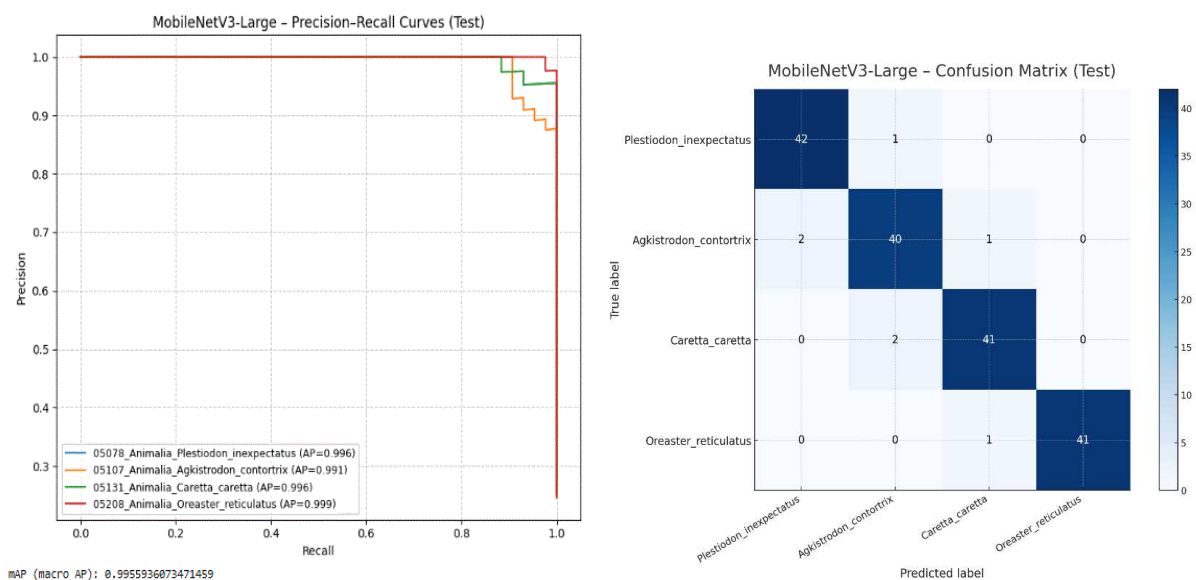


Figure 5.5. MobileNetV3-Large — Per-class Precision–Recall curves & Confusion matrix

In Figure 5.6. ResNet18 Exhibits the same structure, with a modest increase in dispersion off the diagonal; PR curves still sit well above typical operating thresholds.

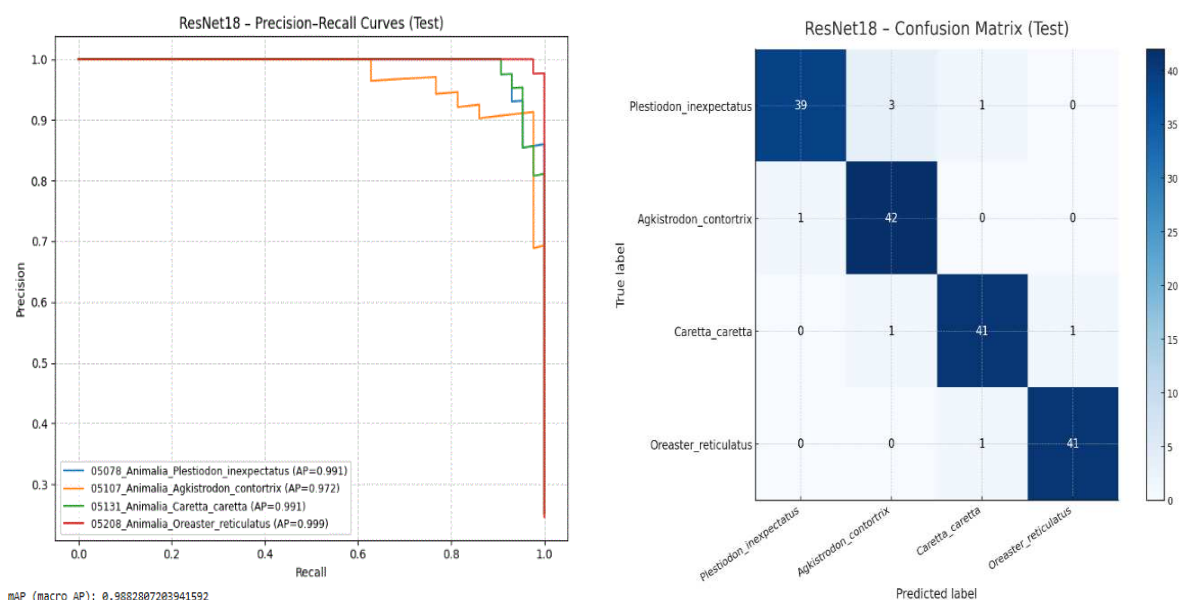


Figure 5.6. ResNet18 — Per-class Precision–Recall curves & Confusion matrix.

For YOLO11n-Cls, Figure 5.7. shows the PR curves are strong but sit marginally lower than those of the top CNN/Transformer models, aligning with a slightly broader off-diagonal activity in the CM—again dominated by Agkistrodon misreads into neighboring classes and occasional Caretta confusions. This matches its lower overall accuracy while preserving a streamlined training/inference pipeline.

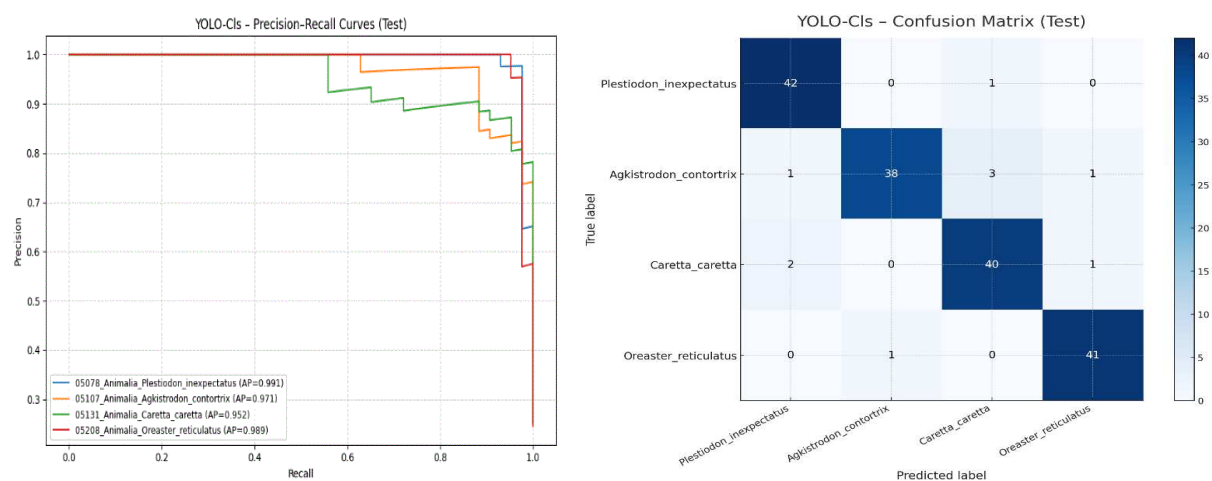


Figure 5.7. YOLO-Classification — Per-class Precision–Recall curves & Confusion matrix.

ViT-B/32 (Figure 5.8). Displays weaker PR curves (earlier bending, smaller area) and broader CM off-diagonals, especially for *Caretta* and *Plestiodon*—consistent with lower weighted-F1 and data-scale sensitivity under a small, balanced dataset.

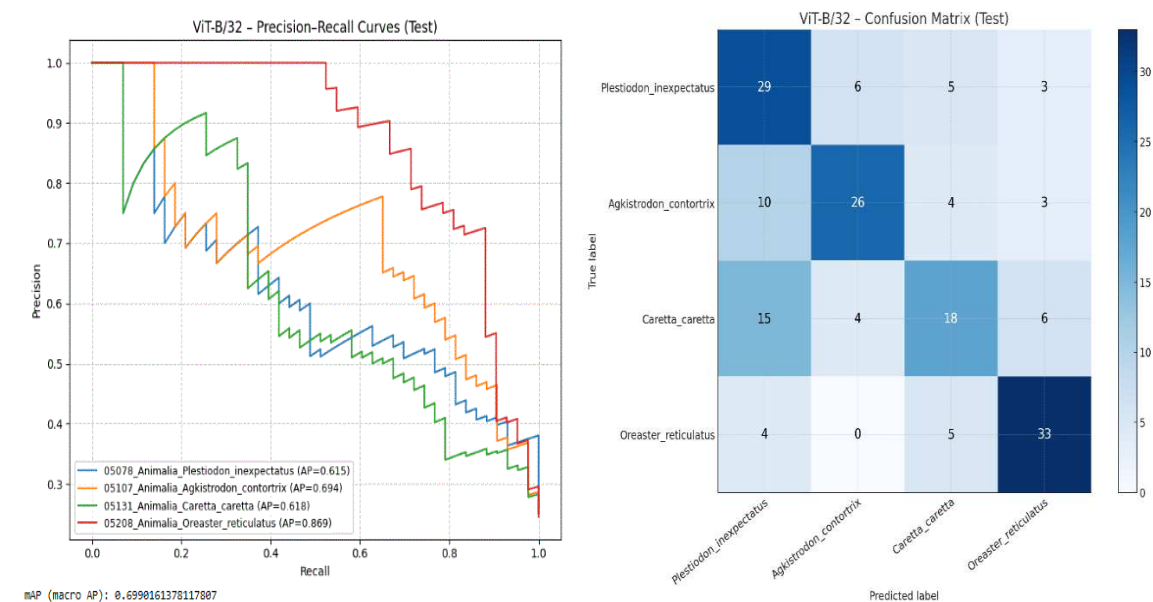


Figure 5.8. ViT-B/32 — Per-class Precision–Recall curves & Confusion matrix.

Custom CNN baseline (Figure 5.9). Shows the weakest separability in PR and the broadest CM off-diagonals, reflecting the limited inductive bias of a shallow CNN under the same conditions.

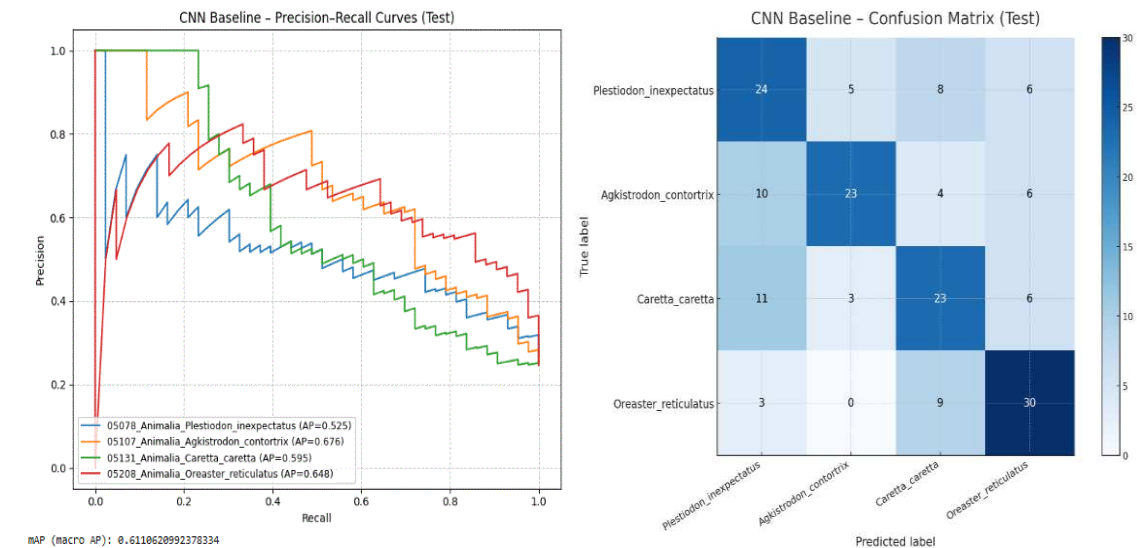


Figure 5.9. Custom CNN baseline — Per-class Precision–Recall curves & Confusion matrix.

The paired PR–CM in Figures 5.2–5.9. panels substantiate the ranking established by aggregate metrics (Accuracy/F1) while adding class-level diagnostics. They explain why Swin (and closely, InceptionV3) leads—clean separability with rare, localized errors—and where compact models trade a small amount of separability for efficiency (the Agkistrodon/Plestiodon boundary). This joint reading also informs practical next steps: prioritize targeted data curation and augmentation around that boundary and, if needed, tune decision thresholds with confidence, given that PR remains high even at near-max recall for the top models.

5.3 Practical Implications

The overall ranking seen in Accuracy and weighted metrics is visually confirmed by the figures: `swin_tiny_patch4_window7_224` performs best (test Accuracy = %98.25), followed closely by InceptionV3 (%97.66). The lightweight tier—MobileNetV3-Large and EfficientNet-B0—lands around %95.9, ResNet18 reaches %95.32, and YOLOv11n-cla (Ultralytics) %94.15. As expected in a small dataset, ViT-B/32 and the Custom CNN baseline are clearly lower (%62 and %58.5). This numeric picture aligns with the PR panels: the top models show near-rectangular curves that stay close to the (1.0, 1.0) corner for all four species, while the weaker baselines bend earlier, especially at very high recall.

Reading each model’s PR curves next to its CM explains where the few remaining errors come from. For `swin_tiny_patch4_window7_224`, all four PR curves are almost ideal, and the CM mass sits almost entirely on the diagonal, with only a handful of off-diagonal entries. The dominant confusion across models is the *Agkistrodon contortrix* ↔ *Plestiodon inexpectatus*

pair. In the PR view, *Agkistrodon*'s curve tends to "bend" slightly earlier at the extreme recall end; in the CM, this appears as a small, concentrated off-diagonal between those two classes. A much milder, occasional confusion appears between *Oreaster reticulatus* and *Caretta caretta* when the marine background dominates the image. InceptionV3 exhibits the same structure as Swin with a very small increase in those pairwise swaps. EfficientNet-B0 and MobileNetV3-Large keep high PR for all classes, but their CMs show a slightly thicker band for the *Agkistrodon*/*Plestiodon* boundary, which explains the small accuracy gap to Swin/Inception. ResNet18 follows the same pattern with a bit more spread, and YOLOv11ncls (Ultralytics) remains generally strong but with a somewhat broader cluster of errors around *Agkistrodon*. In contrast, ViT-B/32 and the Custom CNN baseline show visibly weaker PR (lower area and earlier bending) and wider off-diagonal activity in the CMs, consistent with their lower weighted-F1 on this small, balanced dataset.

The macro-mAP values I obtained support this interpretation as a threshold-independent summary. In my runs, I observed values such as ≈ 0.99897 (Swin) and ≈ 0.98828 (ResNet18), with another strong run around ≈ 0.99559 . These figures mean that separability remains high over a wide range of decision thresholds, not just at the single argmax operating point used in evaluation. Practically, this indicates that the top models are tolerant to small threshold shifts: one can favor either precision or recall if needed without dramatic performance collapse, which is also consistent with how flat the PR curves look.

From a class-behavior perspective, these results suggest a straightforward data priority if the dataset is expanded later: collect more diverse images for the *Agkistrodon*/*Plestiodon* pair (poses, viewpoints, and backgrounds), since this boundary produces most of the few mistakes

visible in the CMs. By comparison, Caretta and Oreaster are generally stable in our figures, with only rare misassignments that match their already strong PR shapes.

Overall, combining the PR panels with the CMs turns the headline numbers into an interpretable picture. The best models (Swin, then Inception) achieve clean separation for all four species and leave only rare, localized errors; the compact CNNs maintain high quality with a small and understandable gap; and the lower baselines behave as expected given our data scale. This consolidated view justifies the final ranking on the test set and provides a clear, evidence-based rationale for model choice in this study.

Chapter 6 Conclusion and Future Work

This thesis built a compact and reproducible pipeline for multi-class wildlife image classification over four visually related species (*Plestiodon inexpectatus*, *Agkistrodon contortrix*, *Caretta caretta*, and *Oreaster reticulatus*). Under one training and evaluation standard, I compared seven families of models: classical CNNs (ResNet-18), efficient/mobile CNNs (MobileNetV3-Large and EfficientNet-B0), a multi-scale CNN (InceptionV3), two transformers (ViT-B/32 and swin_tiny_patch4_window7_224), a detector-native framework used in classification mode (YOLOv11n-cls (Ultralytics)), and a small custom Custom CNN baseline. Inputs, augmentations, optimization, and checkpoint selection were kept consistent so that differences in performance reflect the architectures rather than setup.

Overall, the findings are straightforward. swin_tiny_patch4_window7_224 achieved the best test accuracy (% 98.25) with near-saturated macro-mAP, followed closely by InceptionV3 (% 97.66). MobileNetV3-Large and EfficientNet-B0 formed a strong “lightweight tier” at % 95.9, while ResNet-18 remained competitive at % 95.32. YOLOv11n-cls (Ultralytics) delivered a simple end-to-end workflow at = 94.15%. As expected with our small dataset, ViT-B/32 and the Custom CNN baseline under-performed (%62 and % 58.5). Per-class PR curves for the top models stayed very close to the (precision, recall) corner (1,1), and the confusion matrices localized the few remaining errors, mainly within the Agkistrodon ↔ Plestiodon reptile pair; a weaker, occasional confusion appeared between Oreaster and Caretta when background cues dominated.

Table 6.1 (the embedded figure) reports per-class Precision, Recall, and F1-score for every model. The table is grouped by approach (Pre-trained CNNs, Transformer, Pre-trained YOLO, Baseline). Under each model, you see one row per class. For example, under *InceptionV3* there are four rows—one for each species—with three numbers in each row (Precision, Recall, F1-score). Read

it row-wise to see how well that model handled a specific class, and column-wise to see a metric trend across classes

Approach	Model	Class	Precision	Recall	F1-score
Pre-trained CNNs	InceptionV3	Plestiodon_inexpectatus	0.98	0.98	0.98
		Agkistrodon_contortrix	0.96	0.98	0.97
		Caretta_caretta	0.98	0.98	0.98
		Oreaster_reticulatus	1	0.98	0.99
	EfficientNet-B0	Plestiodon_inexpectatus	0.93	0.95	0.94
		Agkistrodon_contortrix	0.93	0.93	0.93
		Caretta_caretta	0.98	0.98	0.98
		Oreaster_reticulatus	1	0.98	0.99
	MobileNetV3-Large	Plestiodon_inexpectatus	0.96	0.98	0.97
		Agkistrodon_contortrix	0.93	0.93	0.93
		Caretta_caretta	0.95	0.95	0.95
		Oreaster_reticulatus	1	0.98	0.99
	ResNet18	Plestiodon_inexpectatus	0.98	0.91	0.94
		Agkistrodon_contortrix	0.91	0.98	0.94
		Caretta_caretta	0.95	0.95	0.95
		Oreaster_reticulatus	0.98	0.98	0.98
Transformer	Swin Transformer	Plestiodon_inexpectatus	0.96	1	0.98
		Agkistrodon_contortrix	1	0.95	0.98
		Caretta_caretta	0.98	1	0.99
		Oreaster_reticulatus	1	0.98	0.99
	ViT-B/32	Plestiodon_inexpectatus	0.5	0.67	0.57
		Agkistrodon_contortrix	0.72	0.61	0.66
		Caretta_caretta	0.56	0.42	0.48
		Oreaster_reticulatus	0.73	0.79	0.76
Pre-trained YOLO	YOLO-CIs	Plestiodon_inexpectatus	0.93	0.98	0.96
		Agkistrodon_contortrix	0.97	0.88	0.93
		Caretta_caretta	0.91	0.93	0.92
		Oreaster_reticulatus	0.95	0.98	0.97
Baseline	CNN Baseline	Plestiodon_inexpectatus	0.5	0.56	0.53
		Agkistrodon_contortrix	0.74	0.54	0.62
		Caretta_caretta	0.52	0.54	0.53
		Oreaster_reticulatus	0.63	0.71	0.67

Table 6.1. Overall test performance of all models (Accuracy, Precision (weighted), Recall (weighted), F1 (weighted)).

Reading Table 6.1 reveals a consistent story across models and classes. The top performers—swin_tiny_patch4_window7_224 and InceptionV3—maintain uniformly high scores for all four species, with Precision, Recall, and F1 hovering between 0.97 and 1.00. This balance shows that their overall accuracy is not inflated by a single “easy” class but reflects reliable behavior across

the board. Against this backdrop, *Agkistrodon contortrix* emerges as the most challenging category: even the strongest models show a slight dip for this class (typically 0.95–0.98 instead of near-perfect), mirroring the confusion-matrix evidence that the few remaining errors occur mainly along the *Agkistrodon* ↔ *Plestiodon* boundary.

Efficiency-oriented CNNs—MobileNetV3-Large and EfficientNet-B0—prove highly competitive. Their per-class F1 scores largely sit in the 0.93–0.99 range, with the same modest soft spot on *Agkistrodon*, making them attractive choices when memory or compute budgets are tight. YOLOv11n-cls (Ultralytics) delivers solid per-class performance (about 0.91–0.98) while trailing the very best CNN/Transformer backbones; the trade-off is a simpler training and inference pipeline. By contrast, ViT-B/32 and the Custom CNN baseline show the expected limitations in a small-data regime, with per-class F1 mostly 0.46–0.76 and =0.53–0.67 respectively—evidence of weaker separability when inductive bias and data volume are insufficient. One class stands out as consistently easy: *Oreaster reticulatus* often reaches Precision = 1.00 with Recall close to 1.00, likely due to its distinctive morphology and texture.

This table matters because it aligns the aggregate ranking (Accuracy/F1) with class-level behavior and, crucially, pinpoints where effort should be focused. It makes clear that *Agkistrodon*—especially in mobile and YOLO settings—deserves targeted attention in data curation and augmentation.

Looking ahead, the most direct improvement pathway is data-centric. Expanding the *Agkistrodon*–*Plestiodon* subset with more varied poses, scales, and backgrounds, and using region-focused preprocessing (cropping or background suppression) can reduce background dominance. A measured amount of color jitter can further tease out subtle chromatic cues between the two reptiles. On the training side, class-aware or focal losses and mixup/cutmix are reasonable next steps to curb long-tail mistakes without sacrificing calibration. Because the PR curves for top models remain high even at near-maximal recall, operating thresholds can be tuned toward precision-first (e.g., triage in the field with limited expert review) or recall-first (biodiversity

surveillance) with minimal penalty. Finally, a light Swin + Inception ensemble is a practical way to remove the last few errors indicated by the PR shapes and the sparsity of off-diagonals in the confusion matrices. To strengthen reliability in real deployments, it will also be valuable to stress-test models under site/sensor shift and incorporate Grad-CAM to verify that predictions rely on species-diagnostic regions—supporting user trust for non-specialist operators.

References

1. M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,” *Proc. Natl. Acad. Sci. USA*, vol. 115, no. 25, pp. E5716–E5725, 2018.
2. M. A. Tabak *et al.*, “Machine learning to classify animal species in camera trap images: Applications in ecology,” *Methods Ecol. Evol.*, vol. 10, no. 4, pp. 585–590, 2019.
3. S. Schneider, G. W. Taylor, and S. Kremer, “Deep learning object detection methods for ecological camera trap data,” in *Proc. IEEE ICIP*, 2018, pp. 1363–1367.
4. S. Beery, G. Horn, and P. Perona, “Recognition in terra incognita,” in *Proc. ECCV*, 2018, pp. 456–473.
5. J. Wäldchen and P. Mäder, “Machine learning for image-based species identification,” *Methods Ecol. Evol.*, vol. 9, no. 11, pp. 2216–2225, 2018.
6. S. B. Islam, D. Valles, T. J. Hibbitts, W. A. Ryberg, D. K. Walkup, and M. R. J. Forstner, “Deep learning models for classifying herpetofauna species in camera trap images,” *Animals*, vol. 13, no. 5, p. 789, 2023.
7. B. G. Weinstein, “A computer vision for animal ecology,” *J. Anim. Ecol.*, vol. 87, no. 3, pp. 533–545, 2018.
8. D. Tuia *et al.*, “Perspectives in machine learning for wildlife conservation,” *Nat. Commun.*, vol. 13, p. 792, 2022.
9. M. Willis *et al.* (Willi in file header), “Identifying animal species in camera trap images using deep learning and citizen science,” *Methods Ecol. Evol.*, vol. 10, pp. 80–91, 2019.
10. S. Luo *et al.* (WildMatch paper—authors per file), “WildMatch: Grounding wildlife perception in language models,” *arXiv:2311.01064*, 2023.
11. A. Mohammed *et al.*, “Deep learning and model hyper-parameters selection for mammal species classification using camera traps,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 1, pp. 37–44, 2024.