UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE BIOLOGIA

Automatic Animal Recognition in Wildlife Conservation Programmes using Deep Convolutional Neural Networks

Susana Teixeira de Sousa

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:
Cátia Pesquita, Manuel J. Fonseca

2025

Agradecimentos

Deixo o meu agradecimento à Professora Margarida Santos-Reis, do Centro de Ecologia, Evolução e Alterações Ambientais da Faculdade de Ciências da Universidade de Lisboa, cuja criação e mentoria do estudo de campo constituíram a base desta dissertação.

Um agradecimento muito especial é devido à investigadora Ana Luísa Barros, cuja contribuição foi indispensável, nomeadamente pela disponibilização de todas as imagens que sustentaram este estudo, bem como pela generosa partilha do seu conhecimento e experiência de campo.

Acknowledgments

I would like to express my gratitude to Professor Margarida Santos-Reis, from the Centre for Ecology, Evolution and Environmental Changes at the Faculty of Sciences, University of Lisbon, whose establishment and mentorship of the field study formed the foundation of this dissertation.

A very special acknowledgment is extended to researcher Ana Luísa Barros, whose contribution was indispensable, particularly through the provision of all the images that supported this study, as well as through the generous sharing of her knowledge and field expertise.

Resumo

Enquadramento e Motivação

A conservação da biodiversidade é um dos maiores desafios do século XXI, sobretudo perante a intensificação das pressões antropogénicas e das alterações climáticas globais. Neste contexto, a monitorização da vida selvagem assume-se como essencial para compreender o estado dos ecossistemas, avaliar impactos das atividades humanas e orientar políticas de conservação.

Entre os métodos utilizados para o estudo das comunidades de fauna selvagem destaca-se a foto-armadilhagem, que, através da instalação estratégica de câmaras acionadas por movimento, possibilita a recolha de informação sem interferir no comportamento natural das espécies. Tradicionalmente, estes estudos dependiam da observação direta e da classificação manual dos indivíduos.

É neste enquadramento que os avanços recentes em Inteligência Artificial (IA), nomeadamente na visão computacional e nas Redes Neuronais Convolucionais (CNNs), oferecem oportunidades inéditas. Estas técnicas têm demonstrado desempenhos notáveis em tarefas de classificação de imagens em domínios como a medicina ou a condução autónoma, motivando a sua aplicação à ecologia.

No âmbito desta dissertação, o trabalho de campo foi realizado em habitats protegidos em Portugal. Ao contrário de outras regiões do globo, onde os habitats naturais permanecem relativamente livres de intervenção humana, os ecossistemas mediterrânicos estão profundamente interligados com a mesma, pelo que a monitorização de carnívoros selvagens constitui uma prioridade. Este estudo insere-se na investigação desenvolvida pela equipa coordenada pela Professora Margarida Santos-Reis e pela investigadora Ana Luísa Barros, do Centro de Ecologia, Evolução e Alterações Ambientais da Faculdade de Ciências da Universidade de Lisboa, com quem foi estabelecida parceria no âmbito deste projeto.

Objetivos

O presente trabalho tem como objetivo central avaliar a aplicação de redes neuronais convolucionais (CNNs) na classificação automática de imagens provenientes de armadilhas fotográficas, com particular incidência sobre espécies de carnívoros mediterrânicos. A investigação visa analisar a viabilidade de utilização destas técnicas como ferramenta de aceleração e apoio ao trabalho de campo, reduzindo o tempo necessário para a anotação e processamento manual de dados.

De forma mais específica, a dissertação procura:

- Avaliar a eficácia de diferentes arquiteturas de CNN na identificação automática de animais em habitats naturais, considerando métricas de desempenho e robustez em diferentes contextos ecológicos.
- Testar estratégias de transfer learning, de modo a explorar a capacidade de generalização de modelos pré-treinados em grandes bases de dados de imagens, quando aplicados a conjuntos de dados locais de menor dimensão.
- Explorar métodos de adaptação incremental, com o intuito de ajustar modelos previamente treinados a novos contextos ambientais através da utilização de pequenas amostras de dados anotados, avaliando o impacto na precisão e na eficiência do processo.

Assim, este trabalho pretende contribuir simultaneamente para o desenvolvimento metodológico no campo da ecologia computacional e para a criação de soluções práticas que potenciem a integração de inteligência artificial em programas de monitorização da biodiversidade.

Dados Utilizados

A investigação baseou-se em três grandes conjuntos de dados:

Companhia das Lezírias (2013–2016): ~300.000 imagens, das quais apenas uma fração incluía animais e menos ainda carnívoros.

Companhia das Lezírias (2020–2021): >322.000 imagens, com cerca de 3% de carnívoros.

Grândola (2022): 34.270 imagens, com maior proporção de carnívoros (8,3%).

No total, os dados ultrapassaram 650.000 imagens recolhidas entre 2013 e 2022. As espécies-alvo incluíram: raposa (Vulpes vulpes), texugo (Meles meles), gineta (Genetta genetta), fuinha (Martes foina), doninha (Mustela nivalis), lontra (Lutra lutra) e mangusto (Herpestes ichneumon).

Metodologia

A metodologia seguida neste trabalho foi estruturada para garantir não apenas a robustez estatística dos resultados, mas também a sua relevância prática em contextos ecológicos reais. O processo dividiu-se da seguinte forma.

1. Preparação e pré-processamento dos dados

As imagens recolhidas através das armadilhas fotográficas apresentavam elevada heterogeneidade, tanto em termos de qualidade visual (iluminação, resolução, presença de ruído) como em termos de conteúdo (imagens vazias, animais parcialmente visíveis, movimentos rápidos). Assim, o primeiro passo consistiu na normalização das imagens para uma resolução padrão, de forma a reduzir a variabilidade introduzida pelo equipamento e permitir a aplicação consistente das redes neuronais.

Adicionalmente, procedeu-se à categorização em classes (vazio, não-carnívoros, carnívoros), estabelecendo-se um enquadramento adequado ao objetivo central do estudo e aplicada uma seleção manual de amostras representativas, garantindo a diversidade de dados.

2. Modelos preliminares

Numa fase inicial, foram treinados modelos convolucionais simples, desenvolvidos de raiz, que apesar de demonstrarem um desempenho insuficiente, reforçaram a necessidade de recorrer a arquiteturas mais profundas e complexas, já validadas em tarefas de visão computacional.

3. Transfer learning

A estratégia central da investigação assentou no transfer learning, aproveitando arquiteturas pré-treinadas usando como base o banco de imagens ImageNet, uma das bases de dados mais utilizadas em visão computacional. As arquitecturas que mostraram melhores resultados foram:

- VGG16: reconhecida pela sua simplicidade e elevada capacidade de generalização em tarefas de classificação.
- ResNet50: baseada em blocos residuais, que mitigam o problema do desaparecimento do gradiente em redes profundas.

O uso destas arquiteturas permitiu não apenas comparar desempenhos, mas também identificar quais as características estruturais mais adequadas à classificação de fauna em armadilhas fotográficas.

4. Validação cruzada

A avaliação do desempenho dos modelos recorreu a validação cruzada em 10 ou 5 folds. Esta técnica estatística assegura estimativas robustas da capacidade preditiva, reduzindo o risco de sobreajuste a subconjuntos específicos dos dados. Este procedimento revelou-se essencial, dado o desequilíbrio intrínseco das classes (número reduzido de imagens com carnívoros face ao grande volume de imagens vazias).

5. Adaptação incremental

Um dos desafios centrais identificados foi a generalização para novos locais, fenómeno condicionado pelo chamado *domain shift*. Para lidar com este problema, implementou-se uma abordagem de treino incremental, na qual modelos previamente ajustados foram expostos a pequenas amostras (10–20%) de imagens anotadas provenientes de ambientes distintos. Esta estratégia permitiu avaliar a eficácia de uma adaptação com custos reduzidos em termos de anotação manual, mantendo a pertinência prática do método para investigadores com recursos limitados.

6. Ensemble learning

Finalmente, foram exploradas técnicas de ensemble learning, combinando previsões de múltiplos classificadores para reforçar a robustez global. Estratégias como votação majoritária e soft voting (ponderação por probabilidades) foram aplicadas, testando o equilíbrio entre ganhos de desempenho e aumento do custo computacional. Esta fase teve como objetivo principal investigar a estabilidade dos resultados face à variabilidade intrínseca dos dados.

Resultados Principais

Devido às condicionantes deste trabalho (capacidade computacional limitada), o modelo VGG16 destacou-se pela forma como permitiu combinar rapidez e resultados, atingindo valores de *precision* e *recall* superiores a 97% em tarefas binárias (animal vs. vazio; carnívoro vs. outros).

A validação cruzada confirmou a consistência dos modelos dentro de cada local de treino.

A generalização para novos contextos foi fraca: modelos treinados em Santarém mostraram degradação significativa quando aplicados a Grândola.

A adaptação incremental com 10–20% de novos dados anotados resultou em ganhos substanciais de desempenho, demonstrando a viabilidade da adaptação prática.

Os *ensembles* aumentaram a robustez global, embora com custo computacional acrescido.

Discussão

A investigação confirmou a eficácia das CNNs associadas a transfer learning como solução para a classificação automática de imagens de armadilhas fotográficas. No entanto, os resultados evidenciaram uma limitação relevante: a dependência do contexto. Este fenómeno, conhecido como *domain shift*, ocorre quando variações ambientais — iluminação, vegetação, ângulos de captura ou características específicas da fauna — reduzem o desempenho dos modelos treinados noutros locais.

A introdução de adaptação incremental mostrou-se crucial para mitigar este problema, revelando que mesmo pequenas quantidades de dados locais permitem recuperar grande parte da precisão perdida. Do ponto de vista prático, esta abordagem representa uma solução sustentável para ecólogos e gestores ambientais, dado que reduz o esforço de anotação manual sem comprometer a qualidade dos resultados.

Outro aspeto relevante foi a escolha do *recall* como métrica prioritária. Em ecologia, garantir que todos os potenciais registos de carnívoros sejam detetados é mais importante do que minimizar falsos positivos, uma vez que estes últimos podem ser facilmente filtrados por revisão humana. Esta decisão metodológica reflete uma adaptação da análise de desempenho às necessidades concretas da conservação da biodiversidade.

Relevância para os ODS

Este estudo contribui diretamente para três Objetivos de Desenvolvimento Sustentável (ODS) da ONU:

ODS 12 (Consumo e Produção Sustentáveis): ao otimizar o esforço humano e os custos de análise, promove-se a utilização eficiente de recursos.

ODS 13 (Ação Climática): ao melhorar a monitorização de espécies sensíveis, reforça-se a capacidade de compreender os impactos das alterações climáticas.

ODS 15 (Proteger a Vida Terrestre): ao aprofundar o conhecimento sobre carnívoros mediterrânicos, fornece-se informação crítica para a conservação da biodiversidade.

Conclusões

As CNNs, com recurso a transfer learning, são ferramentas poderosas para apoiar a monitorização da fauna selvagem.

A adaptação incremental com pequenas amostras é viável e eficaz, permitindo transferir modelos entre locais distintos com custos reduzidos.

O fluxo de trabalho desenvolvido é flexível e replicável, podendo ser utilizado por ecólogos para reduzir o esforço de classificação manual em novos projetos.

Perspetivas futuras

O trabalho abre caminho para:

- Exploração de arquiteturas mais recentes, como Vision Transformers, que prometem maior generalização.
- Integração de informação espaço-temporal (sequências de imagens, metadados ambientais).
- Desenvolvimento de interfaces acessíveis para utilização em campo por investigadores não especialistas em IA.

Palavras-chave:

Informática Ecológica; Monitorização da Vida Selvagem; Foto-armadilhagem; Redes neuronais convolucionais; Visão computacional.

Abstract

Biodiversity conservation increasingly depends on innovative approaches to monitor wildlife communities in the face of environmental change and anthropogenic pressures.

This dissertation evaluates the use of Convolutional Neural Networks (CNNs) and transfer learning to automate the classification of camera-trap images, with a specific focus on carnivorous species in Mediterranean ecosystems.

The study draws upon three datasets comprising more than 650,000 images collected between 2013 and 2022 in two protected areas: Companhia das Lezírias (Santarém) and Grândola (Setúbal). A combination of experimental approaches was employed, including models trained from scratch and transfer learning strategies leveraging pre-trained architectures. Cross-validation was implemented to ensure robustness, and ensemble methods were tested to further stabilize performance.

Results revealed that VGG16 consistently achieved a good level of performance, with precision and recall surpassing 97% in binary classification tasks (animal vs. empty; carnivore vs. other). However, a significant limitation emerged in the form of poor cross-site generalization, highlighting the effects of environmental variability and domain shift. To mitigate this, incremental retraining experiments using only 10–20% of annotated images from new environments demonstrated substantial performance improvements, confirming the viability of adaptation under resource-constrained conditions.

This dissertation concludes that CNN-based models with transfer learning constitute a powerful, adaptable tool for wildlife monitoring. Rather than proposing a fixed model, it advances a replicable workflow that can be tailored to local datasets. The findings reinforce the potential of artificial intelligence to complement ecological expertise, reducing manual workload and strengthening the capacity for long-term biodiversity monitoring.

Keywords

Ecological informatics; Wildlife monitoring; Camera-trap images; Convolutional Neural Networks (CNNs); Computer vision.

Acronyms

AI – Artificial Intelligence

API – Application Programming Interface

CNN – Convolutional Neural Network

CVPR – Conference on Computer Vision and Pattern Recognition

DNN – Deep Neural Network

ECCV – European Conference on Computer Vision

EXIF – Exchangeable Image File Format

FP – False positives

FN – False negatives

GB – Gigabyte

GPU – Graphics Processing Unit

ICLR – International Conference on Learning Representations

IEEE – Institute of Electrical and Electronics Engineers

ILSVRC – ImageNet Large Scale Visual Recognition Challenge

IQR – Interquartile Range

MIT – Massachusetts Institute of Technology

ML – Machine Learning

RAM – Random Access Memory

ROC – Receiver Operating Characteristic

SDG – Sustainable Development Goals

WILDS – (benchmark dataset for distribution shifts in ML)

TP – True positives

TN – True negatives

Ciências ULisboa

Table of Contents

Image Index

Formula Index

Tables Index

1. Introduction

Contemporary ecosystems are subject to unprecedented pressures arising from climate change and the intensified exploitation of natural resources. These factors contribute to the degradation of habitats whose persistence and renewal depend on the dynamic interactions between animal communities and their environments. Habitats that were once considered stable and largely "natural" are increasingly endangered, underscoring the need for rigorous strategies to safeguard biodiversity.

Traditional ecological monitoring techniques, including field identification and manual classification of photographic records, are essential but inherently resource intensive. Their reliance on human expertise constrains both scalability and accuracy. The widespread adoption of camera-trapping has mitigated some of these limitations, enabling non-invasive monitoring of wildlife communities through strategically positioned, motion-activated cameras. Nonetheless, the interpretation of camera-trap images has historically required manual observation, thereby limiting the efficiency of large-scale ecological studies.

Recent advances in artificial intelligence (AI), particularly within the domains of computer vision and deep learning, provide new opportunities to address these challenges. Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in tasks involving object detection and image classification, and their application to ecological datasets offers a pathway toward automating wildlife monitoring and enhancing conservation practices.

The present dissertation investigates the use of CNN-based approaches for the classification of camera-trap images collected in Mediterranean ecosystems in Portugal, where natural habitats are closely interwoven with human activity. In these contexts, carnivorous species fulfil a critical ecological role in maintaining ecosystem balance through predatory regulation of prey populations. Consequently, monitoring their presence and distribution is a priority for conservation science.

This study is based on research conducted by a team that includes Professor Margarida Santos-Reis and researcher Ana Luísa Barros from "Centro de Ecologia, Evolução e Alterações Ambientais" at Faculdade de Ciências da Universidade de Lisboa, with whom we have established a partnership for this project.

Their team has conducted extensive camera-trap monitoring in protected areas, including the Companhia das Lezírias (in the Tagus Estuary Natural Reserve, Santarém) between 2013–2016 and again from 2020 onwards, and in Grândola (Sado Estuary Natural Reserve) from 2022. These datasets serve as the empirical foundation for the current study.

Figure 1.1 - Installation of a camera trap in the protected area of Companhia das Lezírias (October 22nd, 2020)

Figure 1.1 shows researcher Ana Luísa Barros installing a camera trap during the deployment conducted in October 2020 at Companhia das Lezírias, within the Tagus Estuary Natural Reserve.

This dissertation contributes directly to several **United Nations Sustainable Development Goals (SDGs)** by addressing the challenges of biodiversity monitoring in Mediterranean ecosystems. In particular, it advances:

- **SDG 12 – Responsible Consumption and Production**, by demonstrating how artificial intelligence can optimize the use of large ecological datasets, reducing the time, human resources, and environmental costs associated with traditional manual species identification.
- **SDG 13 – Climate Action**, by providing automated tools for wildlife monitoring that facilitate the detection of ecological changes driven by climate variability, thereby supporting timely and evidence-based conservation responses.
- **SDG 15 – Life on Land**, by focusing on carnivorous species whose ecological roles are critical for maintaining biodiversity and ecosystem balance, and by showing how deep learning can enhance the effectiveness of habitat conservation strategies.

By integrating advanced computational methods into ecological research, this work not only improves the efficiency and accuracy of wildlife monitoring but also reinforces the contribution of science to the global sustainability agenda.

The Sustainable Development Goals addressed in this work are shown in Figure 1.2.

Figure 1.2 – Sustainable Development Goals related to this work (SDG 12, 13, and 15). Source: United Nations, https://www.un.org/sustainabledevelopment/

2. Objective

This chapter outlines the evolution of the project's objectives in response to practical challenges encountered during the research process. Initially conceived as an effort to achieve species-level classification of small carnivores using camera trap imagery, the project's scope was progressively adjusted to account for ecological, technical, and data-related limitations. These included the nocturnal habits and morphological similarity of target species, poor image quality under field conditions, and the scarcity of annotated training data. In response, the study shifted focus toward the development of a more generalisable and adaptable image classification pipeline, aimed at distinguishing small carnivores from non-target subjects across diverse field sites. This chapter details the rationale behind each refinement, situating the final objective within broader ecological and methodological contexts.

The initial aim was to construct a model capable of identifying individual carnivore species, either through models trained from scratch or via transfer learning based on the ImageNet [1][2] dataset. However, this objective proved difficult to achieve due to several limitations, including small sample sizes for certain species, poor image quality, and the predominantly nocturnal behaviour of the target animals, which constrained the acquisition of high-quality photographs.

Accurate classification of small, long-tailed carnivores in camera trap imagery remains a recognised challenge in ecological research. Morphological similarity, cryptic colouration, and the inherent technical constraints of remote sensing often hinder reliable identification, especially when only partial views of animals are captured [3][4]. While camera traps are invaluable non-invasive monitoring tools, they frequently capture individuals under suboptimal conditions—such as low light, motion blur, or oblique angles—which obscure diagnostic features and complicate species-level classification [5]. This issue is particularly pronounced in regions of sympatry, where ecologically similar species overlap in both space and time, rendering traits such as tail length, body proportions, or pelage banding insufficient for consistent discrimination [6].

In response to these challenges, the project's focus shifted from species-level classification to the development of a modelling pipeline designed to first distinguish images containing animals from empty frames, and subsequently to identify small carnivores within those animal-containing images.

However, this approach was also limited by data availability—particularly the absence of empty frames in recent field campaigns. Previously collected empty images proved unreliable as baselines when applied to novel environments, further constraining model performance. As a result, the research objective was refined to focus on developing a method capable of distinguishing small carnivores from other non-target subjects captured in camera-trap images, with an emphasis on achieving generalization across new locations.

In this context, the evaluation strategy was deliberately adjusted to prioritize recall over precision. This trade-off is especially relevant in ecological studies, where the primary concern is minimizing false negatives—that is, ensuring that no instances of the target species are overlooked—even at the cost of a higher rate of false positives. Detecting all potential occurrences of carnivores is therefore considered more critical than achieving absolute certainty in the classification of every individual image.

The overarching goal throughout has been to provide a practical tool that alleviates the burden of repetitive image classification for researchers. By adopting this revised approach, the present study seeks to establish a flexible methodological framework that enables field researchers—particularly those

operating with limited computational resources—to build, adapt, and improve image classification models locally, in a manner responsive to their specific ecological and operational constraints.

3. Fundamentals

In the field of computer vision, deep learning—particularly through Convolutional Neural Networks (CNNs)—has revolutionised tasks such as object detection, classification, and recognition. These advancements have increasingly been applied in ecological and conservation contexts, where camera traps generate large volumes of image data that require efficient and reliable analysis. In this regard, a solid theoretical foundation is essential to frame the methodological approach and contextualise the results of this study.

This chapter outlines the key concepts underpinning the research. It begins with an overview of artificial intelligence, machine learning, and related topics such as neural networks and deep learning. This is followed by a description of convolutional architectures and their role in solving computer vision problems. The chapter then introduces transfer learning and fine-tuning—two strategies particularly relevant when working with limited or imbalanced datasets. Finally, it reviews commonly used CNN models (e.g., VGG16, ResNet50), along with essential techniques such as cross-validation and evaluation metrics, which form the methodological backbone of the experiments conducted in this dissertation.

3.1. Artificial Inteligence

Artificial Intelligence (AI) has been a prominent research field since the 1950s, with its formal inception typically dated to 1956. As described by Russell and Norvig in Artificial Intelligence: A Modern Approach [7] — widely regarded as the most influential textbook in the field—AI refers broadly to the study and design of intelligent agents capable of perceiving their environment and acting autonomously to achieve goals [8].

Definitions of AI vary depending on whether the emphasis is on human-like behaviour or rational problem-solving. For instance, Haugeland (1985) described AI as “the exciting new effort to make computers think... machines with minds, in the full and literal sense,” reflecting a cognitive perspective. In contrast, Luger and Stubblefield (1993) defined it more functionally as “the branch of computer science concerned with the automation of intelligent behaviour.”

3.2. Machine Learning

Following the introduction of AI, a natural progression is to examine Machine Learning (ML)—a subfield of AI concerned with enabling machines to learn from data. According to Russell and Norvig [7], learning in this context refers to the machine’s ability not only to act appropriately in a given situation but also to improve its performance over time based on experience [8].

Machine learning involves the development of algorithms that can identify patterns in data and generalise these patterns to new, unseen inputs, without being explicitly programmed for each task. These algorithms rely on statistical methods to derive representations from data that support prediction, classification, or decision-making.

ML algorithms are typically classified according to the availability of labelled data. The two principal paradigms are:

- Supervised learning, where models are trained on datasets with known input–output pairs.
- Unsupervised learning, where models seek to discover patterns in unlabelled data [9][10].

In supervised learning, the goal is to learn a function that maps inputs to known outputs by minimising the difference between predicted and actual values. This includes tasks such as classification (e.g., assigning species labels to images) and regression (e.g., predicting population counts). Common algorithms include logistic regression, support vector machines, decision trees, and more recently, deep neural networks [11].

By contrast, unsupervised learning aims to uncover underlying structures in data without reference to labelled outcomes. Techniques such as clustering (e.g., k-means, Gaussian mixture models) and dimensionality reduction (e.g., PCA) are used to reveal latent groupings or simplify data representations [10].

In ecological applications, unsupervised learning has been used to group species by appearance, detect anomalies (e.g., rare or invasive species), or create compressed feature embeddings for downstream tasks [12]. Increasingly, semi-supervised and self-supervised methods are also used to overcome the challenges of label scarcity in large ecological datasets [13][14].

3.3. Deep Learning

Deep learning is a subfield of machine learning that focuses on algorithms inspired by the structure and function of the human brain—namely, artificial neural networks [15]. Unlike traditional machine learning models, which often rely on hand-crafted features, deep learning architectures are designed to learn hierarchical representations of data through multiple layers of abstraction. This makes them particularly effective for complex tasks such as image recognition, natural language processing, and speech analysis [11].

At the core of deep learning is the deep neural network (DNN), which consists of an input layer, one or more hidden layers, and an output layer. Each neuron in a hidden layer computes a weighted sum of its inputs, applies a non-linear activation function (e.g., ReLU or sigmoid), and passes the result to the next layer. During training, the model adjusts the weights of each connection using backpropagation and an optimisation algorithm (e.g., stochastic gradient descent) to minimise prediction error [16].

A key strength of deep learning lies in its ability to automatically extract features from unstructured data. For example, in image processing, early network layers may detect basic features such as edges or textures, while deeper layers capture more abstract representations like shapes, objects, or even semantic content (e.g., species identity) [15]. This capacity for end-to-end learning has led to significant performance improvements across a wide range of domains [11][17].

Today, deep learning is the foundation of many real-world AI systems, including:

- Virtual assistants (e.g., Siri, Alexa),
- Recommendation engines (e.g., Netflix, Amazon),
- Machine translation (e.g., Google Translate),
- Medical diagnostics (e.g., tumour detection in imaging),
- Autonomous vehicles (e.g., pedestrian and traffic sign recognition) [11].

However, deep learning also presents several challenges. It typically requires:

- **Large volumes of labelled data** for effective training.
- **High computational resources** (e.g., GPUs or TPUs).
- And often functions as a black box, offering limited interpretability regarding how decisions are made [18].

To address these limitations, ongoing research in explainable AI, transfer learning, and few-shot learning aims to improve the transparency, efficiency, and adaptability of deep models [17][18].

Basic Functioning of a Deep Neural Network

A deep neural network (DNN) operates through a sequence of layers, each performing a specific function:

- **Input layer**: Accepts raw data (e.g., image pixels or audio signals).
- **Hidden layers**: Consist of artificial neurons that compute weighted sums of inputs, apply activation functions, and pass outputs to the next layer.
- **Activation**: Non-linear activation functions (e.g., ReLU, sigmoid) determine whether a neuron "fires," enabling the model to capture complex patterns.
- **Output layer**: Produces final predictions, classifications, or regressions based on the transformed data.

This layered structure allows the model to learn increasingly complex representations, making deep networks well-suited for tasks involving pattern recognition, classification, and prediction [11][15].

3.4. Computer vision

One of the most prominent branches of deep learning is computer vision, a scientific and technological discipline that studies methods to enable computers to perform tasks traditionally carried out by human vision. Its central objective is to achieve high-level understanding of visual data, including static images, image sequences, and videos [19].

This understanding involves the extraction, analysis, and interpretation of information embedded in visual inputs, with the aim of approximating the results that human observers would achieve. In practice, computer vision develops models and algorithms capable of recognizing patterns, identifying objects, segmenting scenes, and even interpreting complex spatiotemporal dynamics [20].

Historically, early computer vision systems relied on hand-crafted features (such as edges, textures, or gradients) and statistical methods for classification [21]. However, these approaches often struggled with variability in real-world data. The emergence of deep learning, particularly convolutional neural networks (CNNs), transformed the field by allowing models to learn hierarchical feature representations directly from raw pixels [11][15].

Today, computer vision underpins a wide range of applications: in ecology, deep models analyse camera-trap images for automatic species recognition [22]; in medicine, they assist in diagnostic imaging [23]; in autonomous vehicles, they enable perception of pedestrians and road signs [24].

Ongoing research continues to push the field toward greater robustness, interpretability, and generalization through advances such as transformer-based vision models and transfer learning frameworks [25].

3.5. CNN (Convolutional Neural Networks)

Convolutional Neural Networks (CNNs) are a class of deep learning architectures specifically designed to process data with a grid-like topology, such as images. CNNs are now the predominant approach for computer vision tasks, due to their ability to automatically extract hierarchical visual features [15].

A CNN is typically composed of three types of layers:

- **Convolutional layers**: Apply filters to input images, detecting low-level features such as edges and textures in the early layers, and progressively more complex patterns in deeper layers.
- **Pooling layers**: Reduce the spatial dimensions of feature maps using statistical operations such as max pooling or average pooling. This reduces computational cost, minimizes overfitting, and enables the network to capture invariant features.
- **Fully connected layers**: Connect every neuron in one layer to every neuron in the next. These layers integrate extracted features and produce the final classification or prediction.

Convolutional and pooling layers appear multiple times throughout the learning model, with the initial layer (data input) always being a convolutional layer, and the final layer (output) being a fully connected layer.

With each layer, the CNN increases in complexity by identifying progressively larger portions of the image, moving from simple recognition of colours and edges to identifying larger and more complex shapes.

CNNs thus learn hierarchical representations: from basic edges and colours to shapes and eventually entire objects or species.

Figure 3.1 shows a schematic illustration of the Convolutional Neural Networks workflow.

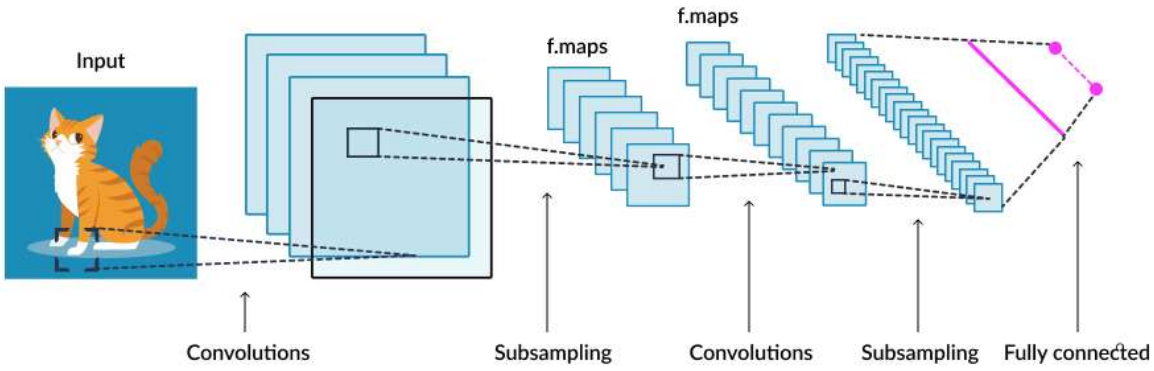


Figure 3.1 - Convolutional Neural Networks schematic illustration.

**Adapted from "The general block diagram for transfer learning implementation" in Zohaib Mushtaq et al. (2020), as presented on ResearchGate *.*

**Source: adapted from Qamar, S. (2019). "The Rise of Convolutional Neural Networks in Computer Vision". Medium. Available at: https://medium.com/*

3.5.1. Convolutional Networks used in this study

VGG16

VGG16 is a deep convolutional neural network architecture developed by the Visual Geometry Group at the University of Oxford by Karen Simonyan and Andrew Zisserman and is first referred to in their work published in 2015 ("Very Deep Convolutional Networks for Large-Scale Image Recognition") [26].

It consists of **16 weight layers** (13 convolutional layers and 3 fully connected layers) and was designed to explore the impact of network depth on classification accuracy. The architecture is characterized by its simplicity and uniformity: it employs small **3 × 3 convolutional filters** throughout the network, stacked in increasing depth, followed by max-pooling layers for dimensionality reduction.

One of the key contributions of VGG16 was to demonstrate that increasing the depth of a convolutional neural network, while maintaining small filter sizes, can significantly improve performance in large-scale image recognition tasks. Trained on the **ImageNet dataset** (containing over 1.2 million images and 1,000 object categories), VGG16 achieved top 5 error rates among the best-performing models of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014. Despite its relatively high computational and memory demands, VGG16 remains widely used for **transfer learning** and **feature extraction**, particularly in computer vision applications where pretrained models are adapted to new domains.

ResNet50

ResNet50 is a convolutional neural network architecture introduced by He et al. (2016) [27] as part of the **Residual Network (ResNet)** family. It contains **50 layers**, including convolutional, batch normalization, pooling, and fully connected layers. The defining feature of ResNet architectures is the

introduction of **residual learning blocks**, which include **skip connections** (identity shortcuts) that allow the gradient to bypass one or more layers during backpropagation.

This innovation addressed the problem of **vanishing gradients**, which had previously limited the training of very deep networks. By enabling the successful training of networks with more than 100 layers, ResNet demonstrated that increasing depth could substantially improve performance without degradation. ResNet50, trained on ImageNet, achieved state-of-the-art accuracy in the **ILSVRC 2015 competition**, outperforming previous models such as VGG16. It has since become one of the most widely used backbones for computer vision tasks, particularly for **fine-tuning** and **transfer learning** in domains with limited labelled data.

EfficientNetB7

EfficientNet, proposed by Tan and Le (2019) [28], is a family of CNN architectures designed to balance accuracy and computational efficiency through a compound scaling method that jointly adjusts network depth, width, and input resolution. **EfficientNetB7**, the largest and most accurate variant, achieved state-of-the-art ImageNet performance with fewer parameters and FLOPs than comparably accurate models. It employs MBConv blocks and squeeze-and-excitation modules to enhance parameter efficiency, though its high computational demands typically require advanced GPUs. Despite this, its accuracy makes it well suited for transfer learning in ecological and medical imaging.

Inception (GoogLeNet and successors)

The Inception architecture (Szegedy et al., 2015) [29] introduced the **Inception module**, which applies multiple convolutional and pooling operations in parallel, enabling efficient multi-scale feature extraction at reduced computational cost. Later versions (Inception-v2, v3, v4) incorporated factorized convolutions and batch normalization to further stabilize and improve training (Szegedy et al., 2016) [30]. In ecological computer vision, Inception models are valuable when datasets contain heterogeneous patterns, such as multiple species or diverse habitats.

DenseNet201

DenseNet (Huang et al., 2017) [31] is distinguished by dense connectivity, where each layer receives inputs from all preceding layers and passes its outputs to all subsequent ones. This structure promotes feature reuse, improves gradient flow, and reduces parameter requirements compared to equally deep traditional CNNs.

.

3.6. Transfer Learning and Fine-Tunning

Transfer learning is a powerful strategy in deep learning that involves leveraging knowledge acquired from one task to improve performance on a different, but related, task. This approach is especially valuable in scenarios where labelled data are limited or costly to obtain—common conditions in fields such as ecological monitoring, medical imaging, or remote sensing [32].

Figure 3.2 illustrates the schematic representation of the transfer learning workflow.

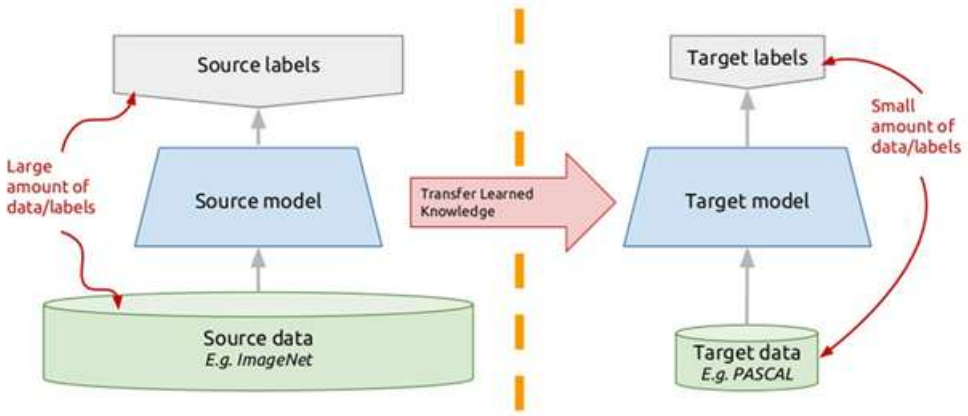


Figure 3.2 – Schematic representation of the transfer learning workflow: a pre-trained model (top) is adapted to a new target task (bottom) by replacing the final layer and fine-tuning on the new data

**Source: Adapted from "Transfer Learning Explained" by Integrate.ai (2018), Medium; Available at https://miro.medium.com/v2/resize:fit:1100/format:webp/1*Z11P-CjNYWBofEbmGQrptA.png.*

There are two main paradigms for training deep learning models:

- Training from scratch: The model's weights are randomly initialized and updated exclusively based on the new dataset. While this approach allows complete task-specific customisation, it requires large volumes of annotated data and substantial computational resources to avoid overfitting and ensure generalisation [11].
- Transfer learning: The model begins with weights pretrained on a large, generic dataset (such as ImageNet), and these weights are reused—either fully or partially—for a new task. This significantly reduces training time and data requirements, while often improving generalisation to the target domain [33][34].

In computer vision applications, **CNNs pretrained on ImageNet**—a dataset containing over 1.2 million images and 1,000 object categories—are commonly used as feature extractors or as a foundation for fine-tuning in domain-specific tasks, such as identifying animal species from camera trap images or detecting disease markers in medical scans.

Transfer Learning vs Fine-Tuning

Transfer learning is often implemented using one of two strategies: feature extraction or fine-tuning. Though closely related, these approaches differ in scope, computational cost, and adaptability to the target domain [32][35]. Table 3.1 shows a comparison between the two approaches.

Table 3.1 – Comparative table between Transfer Learning with Fixed Feature Extraction and Fine-Tuning

	Transfer Learning	
Aspect	**Fixed Feature Extraction**	**Fine-Tuning**
Use of Pre-Trained Model	Yes	Yes
Layers Trained	Only newly added classifier layers	Classifier + selected pre-trained layers
Computational Cost	Lower	Higher
Risk of Overfitting	Lower (fewer parameters updated)	Higher (more parameters updated)
Adaptability to New Domain	Limited	Greater
Typical Learning Rate	Standard	Smaller (to preserve pre-trained knowledge)

As referred, both feature extraction and fine-tuning are strategies in transfer learning, where a model trained on a large benchmark dataset (e.g., ImageNet) is adapted to a new, usually smaller, domain-specific dataset. While related, they differ in scope, flexibility, and computational demand.

- **Feature Extraction**: In this approach, the pretrained model's convolutional base is kept frozen, and only new fully connected layers (usually the classification head) are trained on the target dataset. This is suitable when the new dataset is small or similar in distribution to the original training data.

- **Fine-Tuning**: This strategy involves unfreezing some or all of the pretrained layers and retraining them on the new dataset. Fine-tuning enables the model to adjust its internal representations more precisely to the target domain but requires careful tuning of hyperparameters (especially the learning rate) and carries a higher risk of overfitting if the dataset is small.

Relevance to Ecological Applications

In ecological image analysis, where datasets are often imbalanced, small, or collected in diverse environments, transfer learning has proven highly effective. It allows researchers to repurpose general visual knowledge—such as the ability to detect animal shapes or textures—towards specialised tasks like species classification, presence detection, or behavioural annotation, even when domain-specific images are limited in number or quality [36].

By reducing reliance on large-scale annotations and expensive computational resources, transfer learning democratizes the use of deep learning models, enabling ecologists, conservationists, and field researchers to build and adapt powerful image recognition tools for local use.

Recently, large-scale pre-trained models such as Big Transfer (BiT) have demonstrated strong generalization across diverse vision tasks, further reinforcing the relevance of transfer learning in ecological applications [37].

3.7. Metrics

Accuracy – indicates how often the model was correct within the total set of predictions made. This metric can sometimes be misleading when the model is trained on datasets with imbalanced classes in terms of the number of available examples. In the present study, efforts were made to balance all datasets—both training and validation—therefore this indicator is considered important.

Accuracy can be computed using alternative formulations, as presented in Formulas 3.1 and 3.2.

$$\textbf{Accuracy} = \textbf{Correct predictions} / \textbf{All predictions}$$

Formula 3.1 – Formula for Accuracy

Or

$$\textbf{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Formula 3.2 – Alternative Formula for Accuracy
TP = True positives; TN = True negatives;
FP = False positives; FN =False negatives

Precision – indicates how often the model was correct when predicting a given class. Formula 3.3 defines this metric:

$$\textbf{Precision} = \textbf{True positives} / (\textbf{True positives} + \textbf{False Positives})$$

Formula 3.3 – Formula for Precision

Recall – indicates how often the model was able to identify all elements belonging to a given class. This metric is particularly relevant in the present study, as the objective is to minimize the loss of carnivore images; any image successfully identified may be relevant for field research, potentially even corresponding to a previously unrecorded species in the study area. The formula to calculate this metric is defined in Formula 3.4.

$$\textbf{Recall} = \textbf{True Positives} / (\textbf{True Positives} + \textbf{False Negatives})$$

Formula 3.4 – Formula for Recall

Median, Interquartile Range (IQR), and Mean)

The interquartile range (IQR) quantifies the spread of the middle 50% of a dataset and is calculated as the difference between the 75th percentile (Q3) and the 25th percentile (Q1):

By focusing on the central portion of the distribution, the IQR is robust to outliers [28]. Formula 3.5 presents this metric:

$$\textbf{IQR} = Q3 - Q1$$

Formula 3.5 – Formula for the interquartile range

The median is the value separating an ordered dataset into two equal halves, providing a robust measure of central tendency, particularly for skewed or non-normal distributions [38].

In contrast, the arithmetic mean is the sum of all values divided by the number of observations. While widely used, the mean is sensitive to extreme values, which can bias its representation of the dataset when distributions are skewed [39].

3.8. The Recall–Precision Trade-Off in Ecological Application

The balance between recall and precision represents a fundamental trade-off in supervised classification. Increasing recall reduces the number of false negatives (i.e., missed detections), but often comes at the expense of precision, as the model is more likely to classify uncertain cases as positives [40]. In ecological applications, particularly those involving rare or cryptic species, maximizing recall is often preferable because missing an individual observation may result in significant loss of ecological information [22].

In this context, the approach taken in the present study—adjusting thresholds and class weights to privilege recall—follows best practices in conservation-oriented machine learning. By ensuring that animal detections are rarely missed, even at the cost of accepting more false positives, the models align with the broader ecological priority of preserving valuable data for downstream validation and species identification.

3.9. The Role of the F1-Score

While precision and recall provide complementary perspectives on model performance, their trade-off can make it difficult to assess overall effectiveness. The **F1-score**, defined as the harmonic mean of precision and recall, provides a single metric that balances the two [41]. It is particularly useful in imbalanced datasets such as those derived from camera traps, where high accuracy can be misleading. The F1-score was therefore considered in the evaluation of models, as it provides a more holistic measure of predictive quality, especially in the context of carnivore detection, where both avoiding false negatives (recall) and limiting false positives (precision) are relevant to ecological decision-making.

F1-Score formula is presented in Formula 3.6.

$$\textbf{F1-Score} = (\ 2 * \textbf{Precision} * \textbf{Recall}\ ) / (\ \textbf{Precision} + \textbf{Recall}\ )$$

Formula 3.6 – Formula for the F1-Score

3.10. CNN Parameter Explanation and Tuning

To optimize performance, several parameters were adjusted and their effects evaluated:

- **Epochs**: The number of complete passes through the training dataset. Higher epoch counts typically allow models to refine their learning but increase the risk of overfitting. In this study, 500 epochs were found to balance performance and generalization.

- **Activation Function**: Determines how weighted inputs are transformed into outputs for each neuron. The sigmoid function was used, mapping inputs to a [42][43] range suitable for binary classification tasks.

- **Learning Rate**: Controls the step size during optimization. Smaller values (e.g., 0.0000001) resulted in slower but more stable convergence, particularly important when training on relatively small datasets.

- **Dropout Rate**: A regularization parameter that randomly disables a fraction of neurons during training to prevent overfitting. A dropout rate of 0.4 produced the best balance in this study.

- **Class Weights**: By increasing the relative weight of the "animal" class, the model was encouraged to prioritize recall over precision, which was particularly important given the ecological aim of minimizing false negatives for carnivore detection.

- **Decision Threshold**: The probability cutoff for assigning class labels. Adjusting this threshold above the default 0.5 (e.g., to 0.7) allowed recall to be favoured at the expense of precision, reflecting the project's emphasis on retaining as many animal detections as possible.

3.11. Other techniques used

Flattening Convolutional Layers

In convolutional neural networks (CNNs), the convolutional layers and pooling layers are responsible for extracting spatial features from input data, such as edges, textures, and increasingly abstract patterns across different layers [15]. The output of these layers is typically a multi-dimensional tensor (height × width × number of feature maps). While this representation is effective for spatial processing, it is not directly compatible with the fully connected layers (dense layers) that usually perform the final classification task in a CNN.

The operation of flattening addresses this mismatch. To flatten a convolutional layer means to reshape its multidimensional output tensor into a one-dimensional vector, preserving the learned feature activations while discarding explicit spatial structure [11]. This vector can then be fed into one or more dense layers, which interpret the features holistically and map them to the output classes.

For example, if the final convolutional layer produces a feature map of size 7 × 7 × 512, flattening will transform this into a 1D vector of length **25,088** (= 7 × 7 × 512). This vector is then used as input for fully connected layers, where weights can be optimized for classification tasks such as object recognition or species identification in ecological camera trap images.

Importance of Flattening

- **Bridges convolutional and dense layers:** Flattening provides a structural link between localized feature extraction and global decision-making.

- **Preserves learned features:** Although spatial relationships are discarded at this stage, the activations still encode meaningful patterns relevant to classification.

- Common in classical CNNs: Architectures such as VGG16 [26] rely heavily on flattening before their fully connected layers.

- Alternatives exist: Modern architectures like ResNet [27] often replace flattening with global average pooling, which reduces each feature map to a single scalar, improving generalization and reducing overfitting.

K-Fold Cross-Validation

K-fold cross-validation is a widely used statistical resampling technique designed to evaluate the performance and generalizability of machine learning models [44]. Its main purpose is to mitigate the risk of overfitting and provide a more reliable estimate of model performance than a single train–test split.

The procedure involves partitioning the available dataset into **k equally (or nearly equally) sized subsets, or "folds."** The model is trained on **k–1 folds** and evaluated on the remaining fold. This process is repeated **k times**, with each fold serving once as the validation set and the remaining folds as the training set. The performance results (e.g., accuracy, precision, recall, F1-score) are then averaged across all k iterations to produce a robust estimate of the model's predictive ability.

Ensemble learning technique

Ensemble learning is a machine learning strategy that combines multiple base learners into a single predictive model, with the aim of improving accuracy, generalization, and robustness compared to individual models. The rationale is that while single models may introduce biases or errors, their aggregation can balance weaknesses and exploit complementary strengths [45].

Diversity among models is essential and can be achieved by training on different subsets of data, employing distinct algorithms, or varying hyperparameters [46]. Aggregation methods include:

- **Hard voting**, where the final class label is chosen by majority rule across classifiers.
- **Soft voting**, where predicted class probabilities are averaged (or weighted), allowing models with higher confidence or better calibration to exert greater influence.
- **Averaging**, commonly applied in regression tasks by taking the mean of outputs.

The principal ensemble paradigms are: bagging (Bootstrap Aggregating), which trains models independently on resampled datasets and aggregates their predictions (e.g., Random Forest) [47]; boosting, which builds models sequentially so that each new learner corrects errors from its predecessors (e.g., AdaBoost, XGBoost) [48]; and stacking, which employs a meta-model to learn how best to combine the predictions of multiple base learners [49].

Ensembles are advantageous in reducing overfitting, enhancing predictive accuracy, and improving robustness to noisy data. Their main drawbacks are the computational cost of training multiple models and the reduced interpretability compared to single-model approaches [46].

3.12. Python Programming Language

Python is a high-level, general-purpose programming language created by Guido van Rossum and first released in 1991.

It is dynamically typed and interpreted, supporting multiple paradigms such as procedural, object-oriented, and functional programming. Designed for readability and simplicity, Python has become widely adopted in scientific research due to its extensive standard library and ecosystem of third-party packages. In computational biology and related fields, libraries such as NumPy, SciPy, TensorFlow, and Biopython make it particularly suitable for data analysis, modelling, and machine learning applications [50][51][52].

3.13. State of the art

To develop a comprehensive understanding of computer vision and to inform the methodological choices of this study, a review of key developments in animal identification was conducted. This review traced the evolution of the field—from early manual identification methods to contemporary deep learning-based approaches—and directly shaped the design and implementation of a robust, ecologically relevant workflow.

The use of computer vision for animal identification in ecology dates back to the early 1990s. Two pioneering studies by Whitehead [53] and Mizroch et al. [54] explored the identification of individual whales using digitized slides and photographs. Whitehead [53] analysed 1,015 sperm whales (*Physeter macrocephalus*), identifying individuals through scars and scratches, while Mizroch et al. [54] worked with 9,051 photographs of 790 humpback whales (*Megaptera novaeangliae*), using pigmentation patterns and other distinctive traits. Both approaches relied entirely on manual annotation, and recognition accuracy remained below 45%, underscoring the difficulty of scaling such methods.

A significant step forward was made in 2004, when Ravela and Gamble [55] applied camera-trap imagery to salamander recognition. Their approach, which extracted features such as colour intensities, brightness histograms, and curvature/orientation descriptors, achieved a higher accuracy of 72%. This marked a shift toward automated, though still hand-crafted, feature extraction.

The first attempt to generalize individual recognition across multiple species followed in 2007, when Burghardt and Campbell [56] developed a system based on 3D representations derived from synchronized multi-camera setups. Features were extracted using pixel-difference descriptors and classified with AdaBoost. Despite being computationally intensive, the approach achieved strong results, including 92–97% accuracy in penguins [57] and 95% in tigers [58].

The introduction of convolutional neural networks (CNNs) [59] profoundly altered the trajectory of computer vision. While CNNs had existed conceptually since the 1980s, their major breakthrough came in 2012, when Krizhevsky, Sutskever, and Hinton [1] dramatically reduced the ImageNet classification error rate from 26% to 15%. This performance leap highlighted the superiority of end-to-end feature learning over traditional, manually engineered descriptors. The success of deep CNNs quickly translated into ecological research. Chen et al. [60] were among the first to test CNNs on camera-trap imagery in 2014, although their system achieved only 38% accuracy, showing that significant methodological adaptation was still required. By 2016, traditional feature-based approaches were still common. Manohar, Sharath Kumar, and Hemantha Kumar [61] compared supervised and unsupervised classifiers across 2,000 images from 20 species. Their results showed that supervised methods, enhanced by dimensionality reduction, outperformed unsupervised clustering (79.54% vs. 75.46% accuracy).

From 2017 onwards, however, CNN-based methods became dominant, aided by the availability of very large, labelled datasets. One of the most influential was the Snapshot Serengeti dataset, comprising approximately 3.2 million camera-trap images of 48 species in Tanzania. Gomez, Salazar, and Vargas were among the first to explore this dataset, achieving around 57% accuracy [22]. Subsequent studies

demonstrated rapid improvement. Nguyen et al. [62], working with 80,000 images in Australia, first implemented a binary animal/no-animal classifier with 95.65% accuracy, followed by a multi-class species classifier reaching 88.23%. In 2018, Norouzzadeh et al. [22] applied deep CNNs to a subset of about 1 million Serengeti images, achieving over 94% accuracy for species classification, comparable to volunteer performance but with vastly greater efficiency, reducing annotation effort by an estimated 17,000 hours. Their system further extended functionality by performing multi-animal counts, identifying juveniles, and providing top 5 ranked predictions to manage uncertainty.

That same year, Tabak et al. [63] applied CNNs to more than 3 million camera-trap images collected across five U.S. states. Their model achieved 98% accuracy when validated against the training dataset and 94% accuracy on out-of-sample Serengeti images, demonstrating cross-continental generalizability. Together, these works showed that CNNs were not only accurate but also scalable across ecological contexts. More targeted studies also emerged. For example, Chen, Little, Mihaylova, Delahay, and Cox [64] used 8,368 images from rural settings to monitor badgers, achieving 98.05% accuracy in binary classification tasks, illustrating how CNNs could be applied effectively even with smaller datasets.

In parallel, methodological reviews began consolidating these advances and highlighting best practices for ecological applications of deep learning. Beery et al. [63], for instance, provided a systematic assessment of machine learning approaches for camera-trap images, discussing dataset characteristics, model architectures, and evaluation strategies. Their review emphasized the importance of standardized benchmarks, robust handling of class imbalance, and transparent reporting, thereby shaping the methodological foundations for subsequent large-scale ecological studies.

In addition to classification-focused studies, recent work has emphasized the importance of accessibility and efficiency in ecological image analysis through the development of modular, user-friendly workflows. A notable example is the AddaxAI framework, which offers an open-source, scalable, and cross-platform pipeline for camera-trap image analysis. Deployed via Docker containers, AddaxAI facilitates seamless deployment of detection, counting, and species identification tasks using deep learning. Importantly, it integrates with the widely used Camelot software—automating the embedding of AI-derived labels into image metadata and CSV output for expert validation or analysis. A case study in Tasmanian wildlife monitoring highlights its utility, demonstrating how such tools can bridge technical gaps and enhance reproducibility in ecologically applied machine learning workflows [65].

Building on this groundwork, large community datasets such as Snapshot Serengeti [66] further transformed the field by generating millions of labelled images through citizen-science support, creating a foundation for machine-learning research and ecological inference at scale. These datasets have enabled CNNs to move beyond species identification toward ecological applications. For example, Beery and Van Horn [67] conducted a 2024 study across two contrasting ecosystems—the Maasai Mara savannah in Kenya and the Terai dry forest in Nepal—evaluating how CNNs perform when training data quality, dataset size, and architecture vary. Their results showed that ecological metrics such as species richness, occupancy, and activity patterns derived from neural network predictions deviated by less than 5% from expert labels and remained robust even when 20% of the training data was noisy or removed. Compared to earlier studies [22][63], which established CNN feasibility for classification, Beery and Van Horn demonstrated that deep learning could generate reliable ecological metrics, reinforcing its utility for large-scale biodiversity monitoring.

Despite these advances, one of the most persistent challenges is domain shift: models trained in one geographic or ecological context often underperform when applied to new sites with different illumination, backgrounds, or species assemblages. Beery, Van Horn, and Perona [68] formalized this problem in their *Recognition in Terra Incognita* benchmark, showing that cross-location performance

is consistently weaker than within-location results. Follow-up benchmarks such as iWildCam and the WILDS suite [69] have reinforced this issue, providing standardized protocols for evaluating robustness to distribution shift. Research has since focused on fine-tuning, domain adaptation, and transfer learning strategies to narrow this gap [68][71].

Modern ecological pipelines increasingly decouple detection from classification. MegaDetector [70], an open-source model for detecting animals, humans, and vehicles in camera-trap images, has become a widely used preprocessing tool, enabling the efficient removal of empty frames. Its recent updates (2024–2025) have improved detection accuracy, expanded interoperability, and streamlined integration into conservation workflows [72]. Complementary tools such as the Modular End-to-End Wildlife Classifier (MEWC) provide both command-line and graphical interfaces, making deep learning accessible to ecologists with varying technical expertise. Integrated with systems like Camelot for data management, MEWC has already been deployed successfully in case studies of Tasmanian wildlife [72].

Taken together, these developments highlight the rapid evolution of AI-assisted wildlife monitoring. From early manual identification with low accuracy, the field has progressed through handcrafted descriptors and feature-based methods to CNN-driven classification pipelines that achieve human-level accuracy and, increasingly, reliable ecological inference. Current research priorities include overcoming domain shift, addressing ecological challenges such as extreme class imbalance and low-quality nocturnal imagery, and developing integrated workflows that combine automated predictions with expert validation.

These priorities underscore the field's transition from proof-of-concept studies to deployment-ready systems capable of supporting biodiversity monitoring at scale.

4. Problem Description and Data Sources

Understanding the scope and quality of the data is fundamental to developing and evaluating machine learning models for wildlife image classification.

In ecological studies, the heterogeneity of camera trap datasets—stemming from differences in equipment, sampling strategies, and annotation practices—directly influences the robustness and generalisability of model performance.

This project relied on three distinct groups of images; each collected under different conditions and using different methods of classification. Together, these datasets illustrate the challenges posed by variation in image availability, quality, and annotation consistency, while also providing a diverse foundation for training and validation.

4.1. Dataset Groups and Characteristics

The images used in this study were distributed into **three groups**, each characterized by different methods of classification and image acquisition.

The **first group** consisted of images obtained between 2013 and 2016 at *Companhia das Lezírias*. This dataset contained approximately **300,000 images**, which had been classified using an Excel file. Within this group, the dataset included empty images, images containing animals not directly relevant to the study (e.g., domestic animals and wild herbivores), as well as wild carnivores. Empty images triggered by wind movement, along with those involving human interference, were excluded from classification. Out of the 300,000 total images, only **4,380** were identified as containing animals, and of these, **1,014** depicted carnivores, representing approximately **0.34% of the total dataset**.

The **second and third groups** consisted of photographs collected between 2020 and 2022 by the research team supporting this project. The number of photographs in these groups was substantially larger than in the first dataset. In these more recent groups, image classification was embedded in the metadata (EXIF) of the images themselves. Only images classified as containing an identifiable element beyond the landscape were available for analysis, as all empty images had been discarded by the research team. This exclusion was primarily due to constraints associated with the **volume of information** being processed.

The **second group** comprised photographs taken between 2020 and 2021. This dataset contained **322,027 images** from *Companhia das Lezírias*, of which **9,672** were identified as carnivores (**~3%**). The images were organized chronologically according to the periods in which they were collected:

- Companhia das Lezírias, February–July 2020 (*Browning Dark Ops Pro X* camera)
- Companhia das Lezírias, October–December 2020 (*Browning Dark Ops Pro X* camera)
- Companhia das Lezírias, January–February 2021 (*Cuddeback* camera)
- Companhia das Lezírias, June–September 2021 (*Cuddeback* camera)
- Companhia das Lezírias, September–October 2021 (*Cuddeback* camera)
- Companhia das Lezírias, October–December 2021 (*Browning Dark Ops Pro X* camera)

The **third group** consisted of images obtained from camera traps installed in *Grândola*. This dataset included **34,270 images**, of which **2,843** corresponded to carnivores, representing a much higher concentration of **8.3%**. The dataset was organized as follows:

- Grândola, January–March 2022 (*Browning Dark Ops Pro X* camera)
- Herdade da Ribeira Abaixo, Grândola, March–April 2022 (*Cuddeback* camera)

The **discrepancy in the proportion of carnivore images** between the two sites (*Santarém* and *Grândola*) is not addressed in detail in this study. Possible explanations may include more effective camera positioning, differences in the relative abundance of target species, or the selective exclusion of images considered less relevant for the doctoral research project that originally produced the dataset used in this dissertation.

Finally, the availability of the three groups of photographs for the present project was **staggered**. Initially, only the images from the first group were accessible. Over time, however, access to the additional groups was progressively granted, enabling the use of the full set of data for the final models.

Table 4.1 summarizes these figures to provide a clearer overview of the issue.

Table 4.1- Table illustrating the imbalance problem in each class in all 3 datasets studied

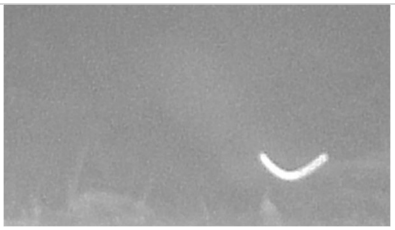
	Comp. Lezírias 2013/16		Comp Lezírias 2020/21		Grândola 2022	
	Num. Images	%	Num. Images	%	Num. Images	%
Carnivores	3,366	1.12%	9,672	3.00%	2843	8.30%
Other content	1,014	0.34%	312,355	97.00%	31,427	91.70%
Empty	295,620	98.54%	-	-	-	-
Total	300,000		322,027		34,270	

4.2. Species to be identified

The species to be identified are carnivores known to have activity in the study regions, namely:

- Vulpes vulpes (red fox) – Figures 4.1.1, 4.1.2 and 4.1.3.
- Martes foina (stone marten) – Figures 4.2.1, 4.2.2 and 4.2.3.
- Lutra lutra (European otter) – Figures 4.3.1, 4.3.2 and 4.3.3.
- Mustela nivalis (least weasel) – Figures 4.4.1, 4.4.2 and 4.4.3.
- Meles meles (European badger) – Figures 4.5.1, 4.5.2 and 4.5.3.
- Genetta genetta (Common genet) – Figures 4.6.1, 4.6.2 and 4.6.3.
- Herpestes ichneumon (Egyptian mongoose) – Figures 4.7.1, 4.7.2 and 4.7.3.

Of the species present, the most frequently observed are the red fox and the European badger (about 60% of samples), followed by common genets and Egyptian mongooses. Stone martens, otters, and weasels appear in smaller quantities, making their identification more prone to errors.

Animal photo	Good animal exposure caught by camera trap	Photo with anotation identifying the animal
V. Vulpes – Red Fox.		
 Figure 4.1.1 - Available at https://mundo-animal.fandom.com/pt/wiki/Raposa	 Figure 4.1.2	 Figure 4.1.3
M. Foina – Stone marten.		
 Figure 4.2.1 - Available at https://pt.wikipedia.org/wiki/Fuinha	 Figure 4.2.2	 Figure 4.2.3
L. Lutra – Otter		
 Figure 4.3.1 - Available at https://pt.wikipedia.org/wiki/Lontra-europeia	 Figure 4.3.2	 Figure 4.3.3
M. Nivalis – Weasel.		
 Figure 4.4.1 - Available at https://mundo-animal.fandom.com/pt/wiki/Doninha	 Figure 4.4.2	 Figure 4.4.3

M. Meles – Badger.

Figure 4.5.1 - Available at https://parquebiologico.pt/animais-plantas/fauna/mamiferos/item/texugo

Figure 4.5.2

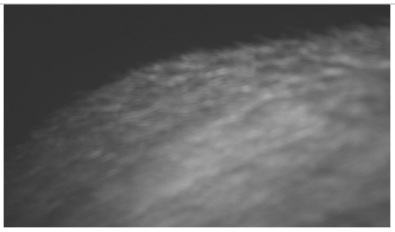


Figure 4.5.3

G. Geneta - Common genet

Figure 4.6.1 - Available at https://en.wikipedia.org/wiki/Genet_%28animal%29

Figure 4.6.2

Figure 4.6.3

H. ichneumon – mongoose.

Figure 4.7.1 - Available at https://www.museubiodiversidade.uevora.pt/elenco-de-especies/biodiversidade-actual/herpestes-ichneumon/

Figure 4.7.2

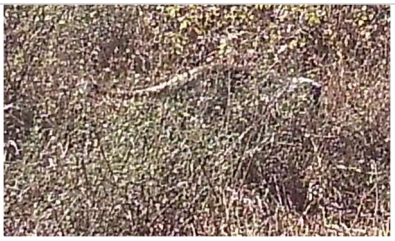


Figure 4.7.3

4.3. Challenges

The challenges inherent in the overall photographs and their classification are:

1. **Inconsistent classification:** Some photographs are labelled when they show other animals or humans, but in others, these elements are discarded.
2. **Nocturnal nature of the species:** The species to be identified are primarily nocturnal (all except the Egyptian mongoose).
3. **Blurred images:** Photographs often capture animals in motion, which tends to result in blurry images or partial captures of the animals' bodies.

4. **Context dependency:** In manual classification, humans often consider the context in which the photograph is taken (e.g., a slight change in vegetation might suggest the presence of an animal).
5. **Unintended camera activation:** Elements such as wind, moving vegetation, or the passage of humans or other animals can trigger the camera without capturing the intended targets.
6. **Camera placement:** In many cases, cameras were positioned ~1 m above ground level. Given the small size of most carnivores, this setup limited capture angles. In 2022, some cameras were lowered to ~30 cm, but this increased the likelihood of vegetation interference.
7. **Lack of empty images:** The more recent datasets (Groups 2 and 3) excluded empty photographs, complicating binary "animal vs. empty" classification tasks.

Together, these challenges highlight the complexity of applying deep learning methods to ecological datasets, underscoring the need for careful model design and evaluation.

To illustrate some of the referred to challenges in image classification. Images 4.8 through 11 show some more example demonstrating the problem:

Figure 4.8 - Twilight image classified as having a fox (V. Vulpes).

Figure 4.9 - Twillight image classified as having a badger (M. meles).

Figure 4.10 - Twillight image classified as having a boar (S. Scrofa) – animal not part of this study.

Figure 4.11 - Image classified as having a human.

5. Methodology

This chapter presents the methodological framework adopted in this dissertation. The section is structured into four parts: computational environment, data preparation, modelling approaches, and evaluation strategy.

5.1. Computational Environment

All models were implemented using Python and executed within a Jupyter Notebook environment managed via Anaconda, with GPU acceleration enabled to enhance computational efficiency. The development framework primarily relied on two libraries: TensorFlow, which served as the low-level numerical computation backend [73], and Keras, which provided a high-level API for streamlined model prototyping and experimentation.
Model training and evaluation were conducted on two different machines:

1. An **ASUS ROG Strix G** laptop equipped with an **Intel Core i7-9750H processor** and **16 GB of RAM**, used until May 2022.
2. A more advanced **ASUS ROG Strix** system featuring an **AMD Ryzen 9 5900HX processor**, **32 GB of RAM**, and an **NVIDIA RTX 3070 GPU**, employed from that point onward to support more computationally intensive experiments.

5.2. Data Preparation

The first step involved normalizing the image set so that each image's classification could be read by an automated process.

The goal was to use the classification of each image to separate them into folders that could feed the learning models.

The available photographs (from Group 1) lacked proper identification, requiring data extraction from an Excel document.

The main issues in this task were related to discrepancies between the data in the Excel file and the image file names, making accurate mapping difficult.

After this initial processing, it was possible to distribute the images into three different folders:

1. Empty images or those with human presence,
2. Images containing non-carnivorous animals,
3. Images containing carnivores.

The first step consisted of normalizing the image dataset in order to enable automated processing of the classifications associated with each image. The goal was to use these classifications to separate the images into distinct folders, which could then serve as inputs for the learning models.

The photographs initially available (Group 1) did not include intrinsic identifiers, requiring the extraction of metadata from an accompanying Excel file. The main challenges during this stage were

related to discrepancies between the information recorded in the Excel document and the corresponding image files, which in some cases prevented the correct mapping and classification of certain images.

After this processing, the images were distributed into three categories: (i) empty images or those containing humans, (ii) images containing non-carnivore animals, and (iii) images containing carnivores.

Subsequently, photographs from Group 2 became available. Unlike the first set, these images did not include empty photographs. Their classification was carried out using the EXIF metadata embedded in each file. Although individual images could reference multiple species present within the same frame—posing certain challenges for classification—the separation of this dataset into folders was comparatively more straightforward.

The next phase involved selecting the images to be fed into the models. Inspection of the folders revealed a characteristic feature of the data collection process: each camera, once triggered, captured a rapid sequence of photographs, often producing dozens of nearly identical images. In order to obtain a reasonably balanced sample across different locations and time intervals, the solution adopted was manual selection. This consisted of browsing the source folders using sequential thumbnails and choosing representative subsets of images from each sequence, with the aim of covering all bursts of photographs.

Throughout the course of the project, this technique of separating and organizing images into folders remained the standard procedure for dataset preparation.

In any case, the relationship between **model complexity** and the **computational capacity of the machine** used for training was clearly a determining factor in the final choice of the model applied in the results of this study.

In this study, Keras models require square images of **224 × 224 pixels** as input. However, the available source images are typically rectangular, with widths ranging between 2,000 and 4,000 pixels. Consequently, it was necessary to normalize each image to a square format. To avoid the risk of cropping parts of the animal under analysis, this was achieved by adding black margins, thereby preserving the integrity of the visual information relevant for classification.

Keras models are deep learning models developed using *Keras*, a high-level neural network API written in Python that provides an accessible interface for designing, training, and evaluating neural architectures [74]. Keras is typically used in combination with TensorFlow, an open-source deep learning framework that serves as its computational backend [75]. Pre-trained Keras models are widely employed in computer vision tasks, as they are generally trained on large-scale datasets such as ImageNet and require input images of a standardized format.

The computer used for the initial training experiments was relatively limited in processing capacity and therefore only permitted the use of images at substantially reduced resolutions in order to avoid exhausting system resources. On this machine (Strix G), the models tested produced only modest results, primarily due to the inability to process input images at resolutions greater than 64 × 64 pixels.

Once the training procedures were transferred to the Scar system, less complex models were able to employ images at 224 × 224 pixels, consistent with the requirements of the original pre-trained architectures.

5.3. Preliminary models

As a preliminary approach, a series of sequential models were developed from scratch, incorporating image pre-processing procedures and employing the Adam optimiser with a learning rate of 0.00001.

The Adam optimizer was selected for training the convolutional neural networks in this study because it adaptively adjusts learning rates for each parameter by combining momentum with gradient scaling, thereby ensuring efficient convergence in high-dimensional models [76].

Compared to stochastic gradient descent (SGD), which often requires extensive manual tuning of the learning rate and momentum, and RMSProp, which only normalizes updates based on past squared gradients, Adam offers a more robust and efficient optimization strategy for complex tasks. A very small learning rate of 0.00001 was employed to stabilize training on the relatively limited and heterogeneous ecological datasets, reducing the risk of overshooting and overfitting while supporting more reliable generalization to new camera-trap images

The models were trained using a variable number of epochs, with the dataset partitioned into 75% (3,429 images) for training and 25% (1,143 images) for testing. The initial objective of these experiments was to establish a binary classification capable of distinguishing between images containing animals and those without detectable fauna or machinery.

Given the primary objective of this project—evaluate how feasible would be to provide a tool for field researchers that would make their work more efficient—various weighting combinations between "animal" and "non-animal" classes were tested. The goal was to minimize the number of animal images mistakenly discarded, while maintaining acceptable levels of overall accuracy.

Class balancing was a critical consideration throughout the modelling process. Since camera-trap datasets are inherently imbalanced—often containing a large proportion of empty images compared to animal images—balancing techniques were implemented to ensure that the models did not become biased towards the majority class. Approaches included

(i) adjusting class weights during training, thereby penalizing misclassifications of underrepresented classes more heavily,
(ii) oversampling minority categories to artificially increase their representation, and
(iii) ensuring proportional representation of each class in both the training and validation datasets.

Such strategies are widely recognized in the literature as effective means of reducing bias in ecological machine learning applications [77][78]. These procedures were particularly important for improving recall in carnivore detection, as the study prioritized minimizing false negatives in this class.

Class creation, validation, and testing followed a structured approach aligned with the objectives of the study. Four key steps were established:

1. Create and validate a model to identify images containing animals, distinguishing them from empty images.
2. Create and validate a model based exclusively on animal images, with the objective of distinguishing carnivores from all other animals.
3. Create and validate a model capable of distinguishing carnivore images from all other cases, including empty images, humans, and herbivores.

4. As a final step, develop a pipeline tested on a new dataset obtained from a separate camera-trapping location (Grândola), which had not been included in training. This pipeline first identified animal presence and subsequently classified carnivores, enabling a comparison with previous results. While this final iteration was acknowledged to have a "not fully realistic" component—since the non-animal images were sourced only from Companhia das Lezírias—it nonetheless provided an important validation test using previously unseen data.

In general, the class prediction produced by each model for a given image is expressed as a real number between **0 and 1**, where the extremes of the interval indicate the highest certainty regarding the class membership of the image. The decision to classify an image as "animal" or "empty" was based on whether the predicted probability was below (animal) or above (empty) a threshold of **0.5**.

After twelve training iterations, during which small modifications were introduced to parameters such as the number of epochs and the optimizer learning rate, it was observed that the classification results (in well-structured models) typically followed the trend illustrated in the graph presented in Figure 5.1.

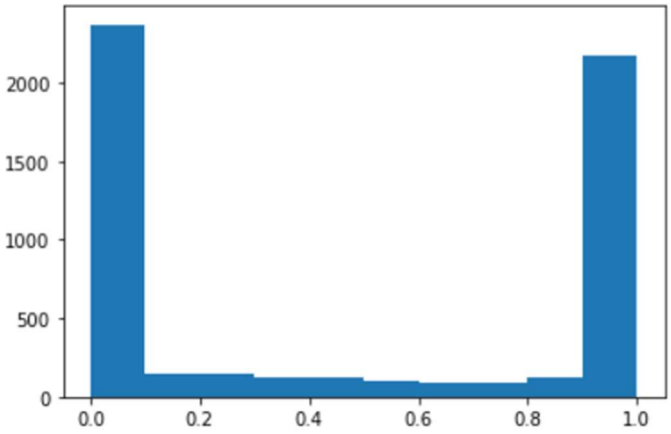


Figure 5.1 – Prediction distribution for classes animal (leaning towards 0.0) or other for threshold 0.7

This observation led to the conclusion that higher levels of **recall**—the metric prioritized in this study—could be achieved by increasing the classification threshold. For instance, raising the threshold from 0.5 to **0.7** shifted the model's sensitivity in favour of detecting animal images, thereby reducing the risk of false negatives. Unless otherwise specified, the results presented in this document correspond to the default threshold value of **0.5**.

The most balanced performance results were achieved with the following configuration: **500 epochs, sigmoid activation function, learning rate = 0.0000001, dropout rate = 0.4**, yielding **precision of 88%** and **recall of 82%** for the "animal" class. Furthermore, by assigning a greater weight to the "animal" class, it was possible to reach **precision of 67%** while improving recall to **93%**, underscoring the trade-off between these two metrics.

The model configuration is described as follows:

Layer (type)		Output Shape	Param #
conv2d_1	(Conv2D)	(None, 128, 128, 32)	896
max_pooling2d_1	(MaxPooling 2D)	(None, 64, 64, 32)	0
conv2d_2	(Conv2D)	(None, 64, 64, 32)	9248
max_pooling2d_2	(MaxPooling 2D)	(None, 32, 32, 32)	0

conv2d_3	(Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_3	(MaxPooling 2D)	(None, 16, 16, 32)	0
conv2d_4	(Conv2D)	(None, 16, 16, 64)	18496
max_pooling2d_4	(MaxPooling	(None, 8, 8, 64)	0
conv2d_5	(Conv2D)	(None, 8, 8, 128)	73856
max_pooling2d_5	(MaxPooling 2D)	(None, 4, 4, 128)	0
dropout_1	(Dropout)	(None, 4, 4, 128)	0
flatten_1	(Flatten)	(None, 2048)	0
dense_3	(Dense)	(None, 128)	262272
dense_4	(Dense)	(None, 1)	129

==========

Total params: 374,145
Trainable params: 374,145
Non-trainable params: 0

The Model is structured into successive convolution–pooling blocks that progressively reduce the spatial resolution while expanding the representational depth of the feature maps. Beginning with low-level feature extraction through 3×3 convolutions, the network doubles the number of filters at deeper stages (from 32 up to 128), enabling the capture of increasingly complex visual patterns. A **dropout layer** is introduced before the transition to the fully connected layers, mitigating overfitting by encouraging regularization. The flattened features are then passed through a dense layer with 128 units, which consolidates high-level abstractions. The architecture concludes with a dense output layer configured for binary classification, producing the final prediction.

Subsequently, a model well known for its strong performance—**ResNet50**—was tested. This model was implemented from scratch and trained for **500 epochs** using images resized to **128 × 128 pixels**. The results obtained for the *animal* class were highly promising, with **96% precision** and **94% recall**.

Although these results were encouraging (particularly with ResNet50), the **large number of epochs required,** and the significant **computational time** involved highlighted the need to explore alternative approaches. Consequently, experiments were extended to **simpler and faster models** (given the limitations of the available hardware), such as **VGG16**, leveraging **pre-trained models** through **Transfer Learning** and **Fine-Tuning**.

The preference for VGG16 in contexts where computational resources are limited is justified by its simpler architecture and reduced training complexity compared to ResNet50. While VGG16 contains more parameters overall (~138 million) than ResNet50 (~25.6 million), its architecture is more sequential and uniform, consisting primarily of stacked convolutional layers with small (3×3) filters followed by max pooling. This design makes VGG16 easier to implement and optimize on machines without high-performance GPUs [26]. In contrast, ResNet50 relies on residual connections that, although highly effective at preventing vanishing gradients and enabling deeper training [27], introduce additional computational overhead and memory demands during training.

For this reason, on **personal computers with limited processing power or memory, like the one used in this study**, VGG16 often proves to be more practical:

- It requires **fewer computational operations per forward/backward pass** when smaller input image sizes are used.
- Pre-trained VGG16 models are widely available and can be adapted with minimal training effort through **transfer learning**, reducing both time and resource consumption.
- Its simplicity makes it suitable for rapid prototyping and for environments where training efficiency is prioritized over state-of-the-art accuracy.

Thus, while ResNet50 remains the superior model in terms of **overall accuracy and generalization**, **VGG16 represents a computationally feasible alternative** in scenarios where hardware limitations restrict the viability of training more complex architectures.

5.4. Transfer Learning with Fine-tuning

The next step involved applying Transfer Learning, using as a basis pre-trained models originally trained on the ImageNet database—a large-scale, publicly available dataset comprising millions of images manually classified into over a thousand categories [79]. ImageNet has been instrumental in the development of deep learning models, serving as a standard pre-training source for architectures such as VGG, ResNet, and Inception.

In Transfer Learning, the decision of which layers to retrain typically follows a decision matrix [34] illustrated in Figure 5.2. This matrix considers both (i) the size of the available dataset and (ii) the degree of similarity between the new data and the original ImageNet training data. For instance, very large datasets (on the order of >100 million labelled samples) may justify full retraining of all model layers, whereas small or moderately sized datasets often benefit from retaining low-level feature extractors (e.g., edge or texture detectors in early convolutional layers) and retraining only the higher layers.

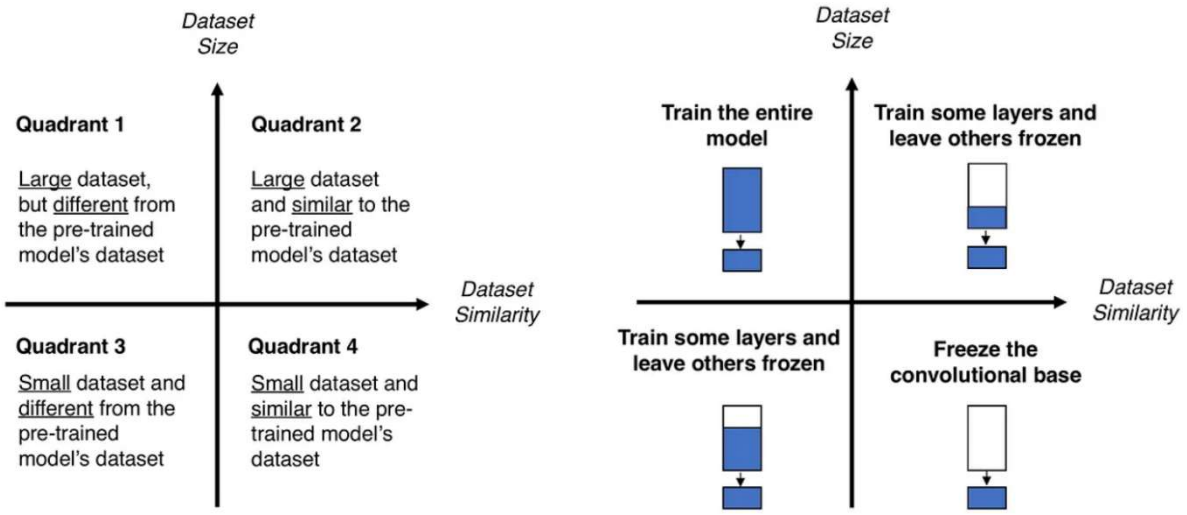


Figure 5.2 (i) On the left: Data volume similarity matrix.; (ii) On the right: Decision map for fine tuning of pre-trained models.

*Source: Adapted from Medium, "Transfer learning and fine-tuning", Available at https://miro.medium.com/v2/resize:fit:640/format:webp/1*heOde2iTazjgrF7YzvOFyO.png and https://miro.medium.com/v2/resize:fit:640/format:webp/1*7ZD-u-h8hFPuN2PYJvLMBw.png*

A key question is therefore: *what qualifies as a "large" dataset*? While in principle this refers to datasets containing tens of millions of images, some studies in ecological and resource-constrained domains have

shown that much smaller datasets can still achieve strong results. For example, Chen et al. (2019) [64] demonstrated that approximately **10,000 images** were sufficient to achieve **98% accuracy in the identification of badgers**, while Besser et al. (2020) [80] suggested that datasets of **15,000 samples** may be adequate for effective training. On this basis, the dataset used in the present study—although smaller than ImageNet—can reasonably be classified as a **medium-sized dataset**.

In terms of similarity to ImageNet, two situations must be distinguished:

1. **Animal vs. empty images:** In the initial training stages, the classification problem of distinguishing animal presence from empty backgrounds was considered broadly similar to ImageNet, which contains numerous landscape images.
2. **Carnivores vs. non-carnivores:** In subsequent experiments, however, it became clear that differentiating carnivores from other animals (often in challenging, low-visibility conditions) represented a **distinct classification task**, less related to ImageNet categories. According to the decision matrix, this scenario falls into the quadrant of **small dataset with dissimilar data**, where best practice involves freezing the initial layers of the pre-trained model (which capture general low-level features) and retraining only the upper layers (which capture task-specific features).

Model Comparison and Results

Several pre-trained models were tested using Transfer Learning, with the following outcomes:

- **EfficientNetB7:** 94% precision / 93% recall (50 epochs)
- **ResNet50:** 98% precision / 97% recall (50 epochs)
- **Inception:** 79% precision / 94% recall (50 epochs)
- **DenseNet201:** 99% precision / 88% recall (30 epochs)
- **VGG19:** 97% precision / 97% recall (30 epochs)
- **VGG16:** In 10-fold cross-validation, obtained an average of **99% precision** and **97% recall**, requiring between 31 and 50 epochs per fold. Importantly, these results were achieved with higher-resolution inputs (224×224 pixels) and with a larger volume of training examples, though still not the complete dataset.

Based on these results, VGG16 was selected for continuation. Despite being simpler in design than the more recent architectures, VGG16 proved highly effective in this context, particularly due to the following factors:

- **Computational efficiency:** Its straightforward architecture demanded fewer hardware resources, an important consideration given the project's computational constraint.
- **Ease of implementation:** VGG16 is widely available in pre-trained form and straightforward to adapt for Transfer Learning.
- **Minimal preprocessing requirements:** Unlike other models, where custom preprocessing was necessary, VGG16 achieved optimal results with only the default settings of the ImageDataGenerator, which applies basic data augmentation across epochs.

This combination of **strong performance reduced computational overhead, and ease of adaptation** justified the selection of VGG16 as the model of choice for subsequent stages of this study.

5.5. Validation of VGG16 with 10-Fold Cross-Validation

To ensure that the results obtained in the first iteration of **VGG16** (precision = **99.24%**, recall = **96.86%**, after 31 epochs in animal identification) were not due to chance, a **10-fold cross-validation procedure** was implemented.

At this stage of the study, the field research team had already provided additional annotated images, increasing the dataset size. However, the larger dataset significantly slowed training on the **Strix-G** machine. Access to a more powerful computer (**Scar**) then became available, allowing more computationally intensive training to be performed.

From this point forward, all models were trained with the following configuration:

- **Input resolution**: 224 × 224 pixels
- **Optimizer**: Adam with variable learning rate between 0.0001 and 0.000001
- **Batch size**: 32
- **Loss function**: Binary Cross-Entropy

The training set comprised **15,783 images (70%)** distributed across two classes, while the validation set contained **6,750 images (30%)**. Since empty images had been discarded from the newer datasets, additional unused images from Group 1 were reintroduced to maintain **balanced class distributions**, a fundamental step to avoid classifier bias [81].

It should also be noted that the two most recent datasets from Companhia das Lezírias—where camera positioning relative to the ground had been altered—were not included at this stage. These datasets will be incorporated in later validation steps to increase the **diversity of the training data** and consequently improve model robustness.

Model Architecture

The models in this phase followed the fourth quadrant of the data similarity matrix [34]. The convolutional layers of the original pre-trained VGG16 were frozen, while new layers were added on top to adapt the model to the specific classification task.

The architecture can be summarized as follows:

- Input layer: 224 × 224 × 3 images
- Convolutional blocks from VGG16 (frozen)
- Flatten layer to convert 3D feature maps into 1D vectors
- Dense layer with 128 neurons
- Dropout layer with a rate of 0.2 (to reduce overfitting)
- Output layer with sigmoid activation for binary classification

The Model is thereby structured in five convolutional blocks, each composed of multiple stacked 3×3 convolutional layers followed by max-pooling, progressively abstracting spatial details while enriching feature representations. The earlier layers (up to block 5) are retained as **non-trainable feature extractors**, leveraging pre-trained weights to capture low- and mid-level image patterns. On top of these, a task-specific classification head is added, consisting of a flattening operation, a fully connected dense layer with 128 neurons, and a dropout regularization step to reduce overfitting. The final dense layer outputs a **single prediction node**, consistent with binary classification tasks.

With over 17 million parameters—of which only a fraction is trainable—this configuration should strike a balance between reusing robust pre-trained visual features and fine-tuning higher-level abstractions for the target ecological dataset.

A complete summary of the model architecture is provided below:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0

dense (Dense)	(None, 128)	3211392
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129

===

Total params: 17,926,209

Trainable params: 3,211,521

Non-trainable params: 14,714,688

The choice of the sigmoid activation function is appropriate for binary classification, since it maps outputs to the interval [32, 33], representing class probabilities [11].

The sigmoid activation function is presented in Formula 5.1:

$$\textbf{sigmoid(x) = 1 / (1 + exp(-x))}$$

Formula 5.1 – Formula for Sigmoid activation function

During training, **callbacks** were used to save the best-performing model based on validation precision, ensuring that the final evaluation was conducted with the most effective weights.

The graph presented by Figure 5.3 reflects the learning progression, a general overview of the evolution of all models. Even with the introduction of the dropout layer, some overfitting can still be seen.

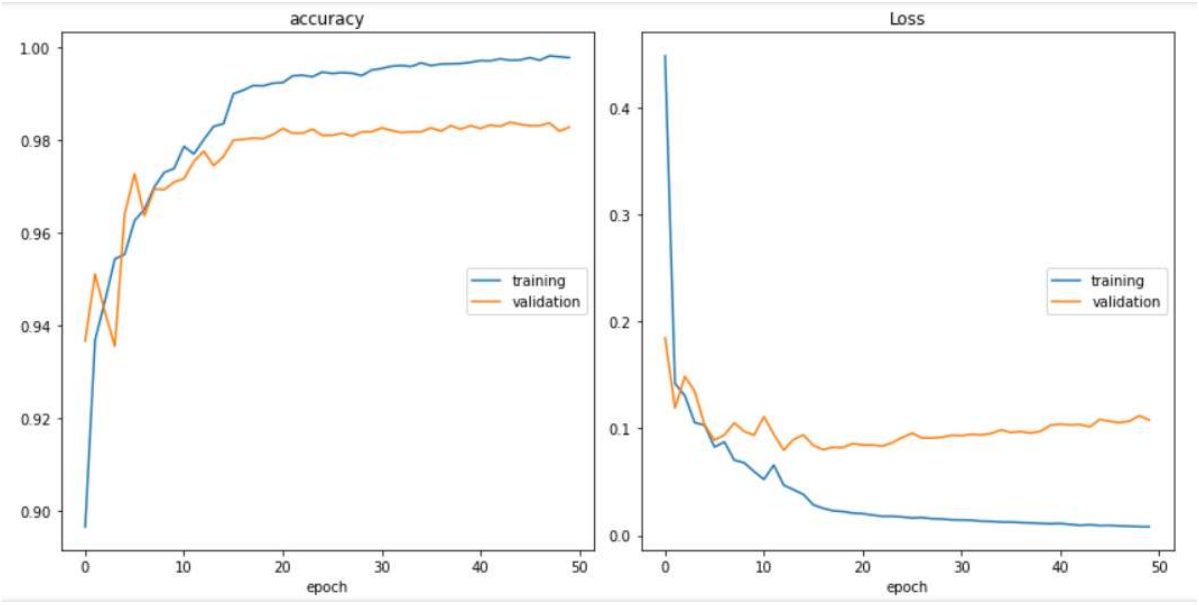


Figure 5.3 – General learning progression after the introduction of the dropout layer

Animal vs. empty Results

The validation results for the *animal vs. empty* classification task are presented in quartile distributions (Q1, median, Q3), with variability measured by the **interquartile range (IQR)**. The consistently low

IQR values demonstrate low variability between folds, indicating strong **stability and reliability** of the results.

Median precision for the *animal* class reached **0.9898**, while median recall was **0.9721**. Conversely, the *empty* class achieved **0.9726 precision** and **0.9899 recall**.

Table 5.1 summarizes the results of the results of the validation of the model obtained from the detection of animals vs. empty images.

This table is similar to others presented throughout this presentation.

Here we can see the Precision and Recall results.

The cells that start with a blue indicator have the value obtained in each iteration (0.9835 equals 98.35% and so on). The first and third Quarter are written in bold, and the Median is expressed by two values separated by a vertical dash.

Lines below each blue indicator are the aggregate measures of all 10 iterations:

- **IQR** → shows how consistent results are across folds. A low IQR indicates that model performance is stable and consistent.
- **Median** → shows the "typical" value in the distribution, less affected by extremes.
- **Average (mean)** → shows the global tendency but is more sensitive to outliers.

Table 5.1 – Animal vs. Empty results with 10-fold validation

Precision			Q1		Median			Q3		
Animal	0.9835	0.9848	**0.9870**	0.9891	0.9895	0.9900	0.9903	**0.9906**	0.9907	0.9924
IQR	0.0037									
Median	0.9898									
Average	0.9888									
Empty	0.9666	0.9693	**0.9718**	0.9718	0.9718	0.9735	0.9739	**0.9755**	0.9783	0.9794
IQR	0.0037									
Median	0.9726									
Average	0.9732									
Recall										
Animal	0.9659	0.9686	**0.9713**	0.9713	0.9713	0.9730	0.9736	**0.9751**	0.9781	0.9793
IQR	0.0039									
Median	0.9721									
Average	0.9727									
Empty	0.9837	0.9849	**0.9873**	0.9893	0.9896	0.9902	0.9905	**0.9908**	0.9908	0.9926
IQR	0.0036									
Median	0.9899									

Carnivore vs. Empty or Other Results

In this experiment, **9,295 training images** and **3,900 validation images** were used, divided between the *carnivore* and *empty/other* classes.

- Median precision: **0.9645** (carnivore), **0.9593** (empty/other)

- Median recall: **0.9590** (carnivore), **0.9646** (empty/other)

The results indicate robust discriminative performance, although slightly lower than the animal vs. empty classification.

Table 5.2 summarizes these results:

Table 5.2 - Animal vs. Empty or Other results with 10-fold validation

Precision			Q1		Median			Q3		
Carnivore	0.9534	0.9559	**0.9595**	0.9638	0.9641	0.9649	0.9654	**0.9657**	0.9671	0.9709
IQR	0.0061									
Median	0.9645									
Average	0.9631									
Empty/Other	0.9534	0.9558	**0.9564**	0.9583	0.9592	0.9595	0.9610	**0.9636**	0.9657	0.9657
IQR	0.0072									
Median	0.9593									
Average	0.9599									
Recall										
Carnivore	0.9528	0.9554	**0.9564**	0.9579	0.9590	0.9590	0.9610	**0.9636**	0.9656	0.9662
IQR	0.0072									
Median	0.9590									
Average	0.9597									
Empty/Other	0.9528	0.9559	**0.9595**	0.9641	0.9641	0.9651	0.9656	**0.9662**	0.9672	0.9713
IQR	0.0067									
Median	0.9646									
Average	0.9632									

Carnivore vs. Other Results

Since empty images were unavailable for most datasets, an additional experiment was conducted distinguishing *carnivores* from *other animals or machinery*. The results were similar to the previous experiment, suggesting that the models had already achieved strong generalization for detecting empty images.

- Median precision: **0.9657** (carnivore), **0.9620** (other)
- Median recall: **0.9721** (carnivore), **0.9899** (other)

This reinforced the hypothesis that empty image detection was already effectively learned. However, this hypothesis could be conclusively confirmed due to the absence of empty images in the later datasets.

The summary is presented in Table 5.3.

Table 5.3 - Carnivore vs. Other results with 10-fold validation

Precision			Q1		Median			Q3		
Carnivore	0.9608	0.9620	0.9631	0.9632	0.9652	0.9661	0.9666	0.9686	0.9689	0.9694
IQR	0.0055									
Median	0.9657									
Average	0.9654									
Other	0.9539	0.9561	0.9596	0.9599	0.9604	0.9636	0.9643	0.9647	0.9651	0.9657
IQR	0.0051									
Median	0.9620									
Average	0.9613									
Recall										
Carnivore	0.9659	0.9686	0.9713	0.9713	0.9713	0.9730	0.9736	0.9751	0.9781	0.9793
IQR	0.0039									
Median	0.9721									
Average	0.9727									
Other	0.9837	0.9849	0.9873	0.9893	0.9896	0.9902	0.9905	0.9908	0.9908	0.9926
IQR	0.0036									
Median	0.9899									
Average	0.9890									

All tables summarizing gathered data during these trainings are stored with public access here: https://zenodo.org/records/17076927

The corresponding Zenodo repository also provides access to the complete set of model results evaluated in this study.

5.6. A failed experiment: model application pipeline from Animal vs. Empty to Carnivore vs. Other

Following the development of preliminary models, a sequential pipeline was designed to evaluate their practical applicability in real-world scenarios. The goal was to assess whether a two-step classification process could improve performance or at least behave as well as earlier results using discovered models: first identifying images containing animals and subsequently distinguishing carnivores from other species. This approach was intended to approximate the actual workflow required in ecological monitoring, where large volumes of images include both empty frames and diverse non-target species

Pipeline Description and Results

The experimental pipeline was applied to datasets not previously included in training but still originating from the Companhia das Lezírias but captured at different time periods. The methodology consisted of two steps:

1. **Step 1:** Classify images as *animal* vs. *empty*.

2. **Step 2:** Re-classify images identified as animals into *carnivore* vs. *other*.

While Step 1 performed satisfactorily, the results in Step 2 degraded substantially, with precision and recall dropping to values close to random performance (≈50%).

The primary hypothesis for this performance decline was the **distributional shift** between datasets, particularly differences in camera positioning, lighting, and environmental conditions. Furthermore, it was conjectured that the sequential nature of field data (e.g., bursts of consecutive frames showing the same individual) created contextual cues that the models were unable to exploit. Since Convolutional Neural Networks (CNNs) process each image in isolation, this mismatch likely hindered generalisation across temporally correlated image sequences. This interpretation aligns with observations reported in the ecological machine learning literature, such as Beery et al. (2018), who emphasised the challenges of domain adaptation in wildlife recognition tasks ("Recognition in Terra Incognita").

CNN-based models often struggle to generalize across ecological sites due to **domain shift**. Background environments differ in vegetation, soil, lighting, and camera settings, leading models to learn spurious correlations that fail in new contexts. Species may also appear in different poses or activity patterns depending on terrain layout and local behaviour, which further reduces transferability.

Data limitations exacerbate this issue: camera-trap datasets are imbalanced, and rare species often lack sufficient training samples to produce generalizable features. In addition, CNNs may overfit to non-biological cues (e.g., sensor artifacts, timestamps) present in one site but absent in another. Finally, community composition varies between reserves, so the "non-target" class differs across locations, complicating binary or group-level classification.

Together, these factors explain why high performance within a site does not translate to equally strong results across sites, highlighting the need for incremental retraining, domain adaptation, and broader ecological representation in training datasets.

In hopes of mitigating this issue, the models were evaluated on small, well-annotated subsets containing clearer animal images before extending predictions to the full dataset. However, the results remained consistent with the initial findings, showing no significant improvement in Step 2 performance.

Conclusion

This experiment demonstrated the limitations of applying sequential pipelines without explicitly accounting for contextual and environmental variability. While the approach proved effective for distinguishing animal vs. empty images, its inability to generalise in carnivore classification highlights the need for models capable of handling **domain shifts** and incorporating **temporal context**.

This issue highlights the need for more consistent approaches when integrating new images into the modelling process. In particular, hybrid architectures and ensemble strategies offer promising avenues, as they can incorporate spatiotemporal information and thereby enhance robustness across heterogeneous datasets [70].

6. Training through sampling in new locations

6.1. Objective

The objective of this chapter is to evaluate the effectiveness of training strategies that incorporate small subsets of manually classified images from new locations.

In line with the overarching aim of enabling field researchers to obtain reliable results with minimal annotation effort, this chapter investigates whether limited site-specific sampling is sufficient to adapt existing models to novel environments.

Given the strong influence of environmental context on model performance, the experiments are restricted to the carnivore vs. other classification task.

While this focus ensures methodological realism, the framework developed here may, in future work, be extended to the more challenging animal vs. empty classification problem—an application of particular value to researchers, as it directly addresses the need to filter out the majority of camera-trap images.

6.2. Method

The methodology followed comprised the following steps:

1. **Creation of two independent datasets**, each including the classes *carnivore* and *other*:
   a. The first dataset contained 21,334 images from Companhia das Lezírias (Groups 1 and 2), subdivided into 16,002 for training and 5,332 for validation.
   b. The second dataset contained 5,686 images from Grândola, also distributed across the two classes.
2. **Training of the *carnivore vs. other* model** using VGG16 with transfer learning, under multiple parameter variations:
   - Image preprocessing (with or without explicit centring).
   - Retraining depth (all layers, retaining a fixed number of base layers, or only retraining the final adaptation layers).
   - Dropout rates (to mitigate overfitting).
   - Size of intermediate dense layers.
   - Dimensionality reduction strategies: Flatten() vs. feature extraction with GlobalAveragePooling2D.
   - Early stopping vs. model checkpointing.
   - Data splits: 2-way (training/validation) vs. 3-way (training/validation/test).
   - Fixed vs. variable learning rates (range: 0.0001–0.000001).
3. **Validation of the trained model** against the Grândola dataset, without further modifications.

4. **Incremental retraining using 10% of Grândola samples**, to adapt the model to the new location. This comprised 400 training images and 168 validation images (70/30 split). Validation was then conducted on the remaining 90%.
5. **Extension of retraining to 20% of Grândola samples**, with results compared against predictions on the remaining 80%.
6. **Direct transfer learning on Grândola data**, using VGG16 pretrained weights but without leveraging the Companhia das Lezírias training phase.

All promising results were validated using 5-fold cross validation.

6.3. Transfer Learning with Flatten-Based Adaptation

In this section, we examine transfer learning approaches in which adaptation from the pretrained base network was achieved through the use of a Flatten() layer.

Flatten-based adaptation is a classical technique in convolutional neural networks, whereby the multidimensional tensor produced by the final convolutional layers is reshaped into a one-dimensional vector. This transformation discards explicit spatial structure while preserving the learned activations, thereby enabling their integration into fully connected layers for classification.

While more recent architectures often replace flattening with global average pooling—a method that reduces each feature map to a single scalar by averaging activations across spatial dimensions, and which is also employed later in this study—the decision was made to examine flattening first due to its potential advantages in the present context.

Global pooling is known to enhance generalization and reduce overfitting, particularly in large-scale datasets, but it also compresses spatial information that may be critical for distinguishing subtle morphological cues in ecological images. Flattening, by contrast, retains the full set of feature activations, thereby providing a richer input to the classification layers. For these reasons and given the moderate size and complexity of the datasets analysed here, flatten-based adaptation was selected as a pragmatic and effective strategy.

6.3.1. Training carnivore vs. other on Companhia das Lezírias Data

Initial tests compared preprocessing strategies (with vs. without centring). Default preprocessing performed slightly better.

Following the transfer learning decision matrix (small dataset, different problem), the first 8 base layers were frozen while the remaining layers were retrained.

Training ran up to 220 epochs, using the Adam optimizer with a fixed learning rate of 0.000001.

Variable learning rates were tested but did not yield improvements.

Dropout was increased to 0.8 to reduce overfitting, although some overfitting remained visible, as it is shown on Figure 6.1:

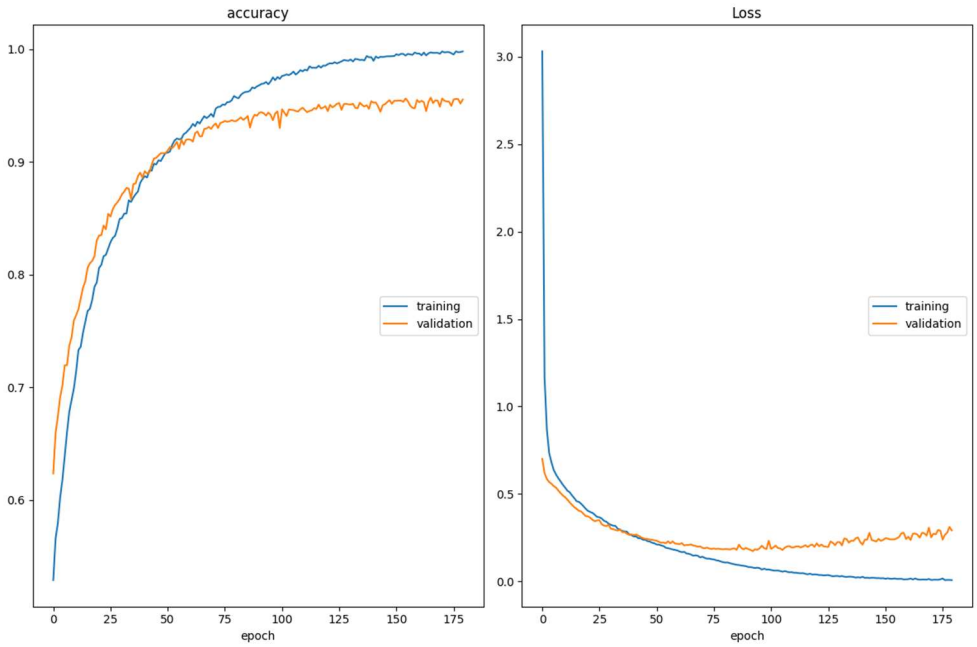


Figure 6.1. - learning progression with dropout at 0.8

Summary of Results (median values):

- *Carnivore*: Precision 94.56%, Recall 96.10%.
- *Other*: Precision 96.03%, Recall 94.45%.

These results are described by table 6.1.

Table 6.1 - Carnivore vs. other on Companhia das Lezírias Data results with 5-fold validation

	Q1		Median	Q2	
Loss	0.2447	0.2504	0.3571	0.4074	0.5101
Precision					
Carnivore	0.9345	0.9443	**0.9456**	0.9600	0.9603
	0.9394			0.9602	
IQR	0.0208				
Median	0.9456				
Average	0.9489				
Other	0.9538	0.9542	**0.9603**	0.9613	0.9644
	0.9540			0.9629	
IQR	0.0089				
Median	0.9603				
Average	0.9588				
Recall					
Carnivore	0.9535	0.9539	**0.9610**	0.9825	0.9651
	0.9537			0.9738	
IQR	0.0201				

Median	0.9610				
Average	0.9632				
Other	0.9325	0.9434	**0.9445**	0.9602	0.9606
	0.9380			0.9604	
IQR	0.0225				
Median	0.9445				
Average	0.9482				

Compared to prior 10-fold cross validation, results described on table 6.1 were slightly lower, confirming the pattern that performance disperses as more diverse locations are added.

Importantly, prediction confidence was consistently near 0 or 1, indicating reliable outputs, further supported by low binary cross-entropy loss (0.24–0.51).

6.3.2. Validation on Grândola Data

The underlying idea of conducting training with 10% and 20% of the sample was to determine whether substantial differences would emerge between a smaller dataset (which is therefore considerably easier to obtain) and a larger one.

Using the same model as the starting point for transfer learning (specifically, the model obtained in fold 3), in order to ensure direct comparability across the different folds, and despite retraining only the adaptation layers corresponding to the upper sections of the model, the best results were achieved by maintaining the model exactly as it had been trained previously—namely, with the same initial layers frozen and the remaining layers open for training. In this case, the first eight layers were kept frozen.

This indicates that, for all practical purposes, these new datasets are treated by the network as representing a problem distinct from the previous one, even though the images appear to be quite similar.

The training was conducted with a variable learning rate (0.00001 for epochs 1 to 60, and 0.000001 for the remaining epochs), for a maximum of 100 epochs. However, only one of the folds benefited from the lower learning rate, since all others reached their best results within 50 epochs or fewer.

When applied directly to the Grândola dataset, model performance, as detailed in Table 6.2, decreased substantially:

- *Carnivore*: Precision median 71.9%, Recall median 85.26%.
- *Other*: Precision median 81.07%, Recall median 61.13%.

Table 6.2 - Carnivore vs. other on Grândola Data results with 5-fold validation

Precision	Q1		Median	Q2	
Carnivore	0.6858	0.6980	**0.7190**	0.7247	0.7353
	0.6919			0.7300	
IQR	0.0381				
Median	0.7190				
Average	0.7126				
Other	0.8013	0.8088	**0.8107**	0.8252	0.8304

	0.8051			0.8278	
IQR	0.0228				
Median	0.8107				
Average	0.8153				
Recall					
Carnivore	0.8389	0.8484	**0.8526**	0.8690	0.8593
	0.8437			0.8642	
IQR	0.0205				
Median	0.8526				
Average	0.8536				
Other	0.6310	0.6113	**0.6113**	0.6641	0.6908
	0.6212			0.6775	
IQR	0.0563				
Median	0.6113				
Average	0.6417				

This finding reinforced the necessity of implementing context-specific adaptation strategies.

6.3.3. Retraining with 10% of Grândola Samples

Using 10% of the Grândola dataset:

- Training with frozen first 8 layers provided the best performance, indicating the network treats Grândola as a distinct problem.
- Training ran for up to 100 epochs with variable learning rate scheduling.

Results

- *Carnivore*: Precision 86.01%, Recall 87.57%.
- *Other*: Precision 87.32%, Recall 85.93%.

Table 6.3.- Retraining with 10% of Grândola Samples results with 5-fold validation

	Q1		Median	Q2	
Max Train accuracy	0.815	0.8150	**0.8210**	0.8210	0.8210
Train Loss	0.6781	0.7570	0.8603	0.9308	0.9501
Precision					
Carnivore	0.8586	0.8592	**0.8601**	0.8615	0.8616
	0.8589			0.8616	
IQR	0.0027				
Median	0.8601				
Average	0.8602				
Other	0.8586	0.8686	**0.8732**	0.8732	0.8768
	0.8636			0.8750	

IQR	0.0114				
Median	0.8732				
Average	0.8701				
Recall					
Carnivore	0.8585	0.8699	**0.8757**	0.8787	0.8796
	0.8642			0.8792	
IQR	0.0150				
Median	0.8757				
Average	0.8725				
Other	0.8558	0.8570	**0.8593**	0.8691	0.8691
	0.8564			0.8691	
IQR	0.0127				
Median	0.8593				
Average	0.8621				
Validation Accuracy	0.8589	0.865	**0.8654**	0.8658	0.8683

Table 6.3 shows that validation accuracy on the 90% unseen data exceeded that of the training subset (86.54% vs. 82.10%), suggesting effective generalization.

Threshold adjustments showed trade-offs:

- Recall increased from 87.57% (threshold 0.5) to 91.17% (threshold 0.9), at the cost of Precision decreasing from 86.01% to 82.23%.

It is also noteworthy that the learning curves in these experiments are considerably more irregular than those observed thus far, even though the same model parameters were applied. This phenomenon can be explained by the substantially smaller amount of data available, which makes the "path" toward achieving satisfactory results considerably more erratic.

Figure 6.2 presents one of the graphs obtained from the training conducted with the 10% sample.

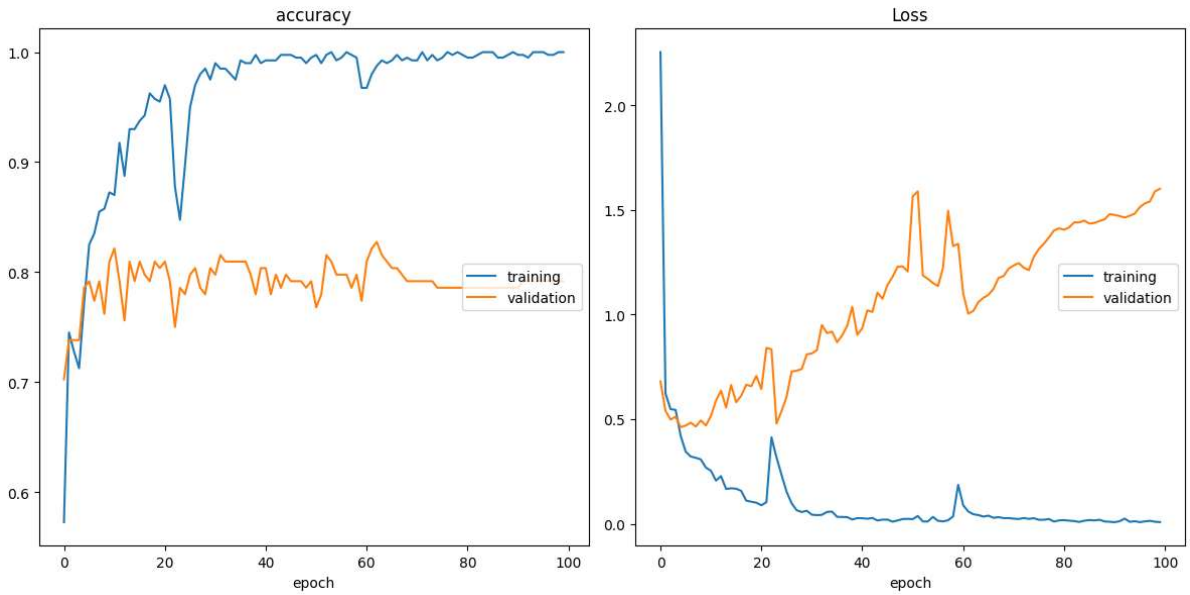


Figure 6.2 - learning progression with 10% sampling

Figure 6.2 also shows evidence that the best result was achieved at epoch 63, where a decrease in the loss function was also observed, although at a value considerably higher than those recorded when precision was lower. This phenomenon may occur when the model, despite producing correct predictions, exhibits lower confidence in its outputs, which in this case implies a slightly greater number of predictions falling away from the extreme values of the probability interval.

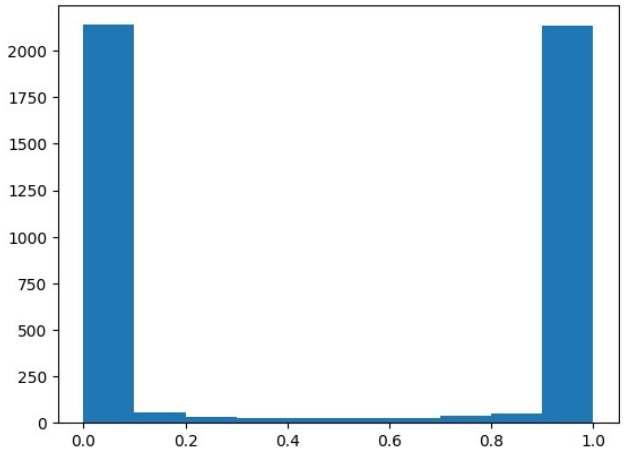


Figure 6.3 – learning predictions on 10% sampling

This deviation suggests that, in pursuit of the goal of minimizing the loss of carnivore detections without incurring a substantial reduction in precision, it may be appropriate to increase the classification threshold.

Figure 6.3 indicates a slight margin for improvement in recall for the carnivore's class with an increased decision threshold.

Table 6.4. presents the comparative results for the carnivore class when applying thresholds of 0.7 and 0.9 against the standard threshold of 0.5.

Table 6.4 - Retraining with 10% of Grândola Samples results with variable threshold

	0.5 threshold		0.7 threshold		0.9 threshold	
	Median	Average	Median	Average	Median	Average
Precision	86.01%	86.02%	84.29%	84.32%	82.23%	82.22%
Recall	87.57%	0.8725	89.14%	89.05%	91.17%	0.9127
F1-Score	86.78%	86.63%	86.65%	86.62%	86.47%	86.51%

The results presented in table 6.4 demonstrate that increasing the classification threshold to 0.9 leads to an improvement in recall of over 3%. However, this gain occurs at the expense of nearly a 4% reduction in precision. This trade-off highlights the inherent balance between maximizing the detection of true positives (recall) and minimizing false positives (precision).

F1-score remains remarkably stable (86.78% at threshold 0.5, 86.65% at 0.7, and 86.47% at 0.9). This indicates that the gains in recall at higher thresholds are nearly balanced by the losses in precision, resulting in only marginal differences (<0.4 percentage points) across thresholds.

Furthermore, the similarity observed between the mean and median values suggests a high degree of consistency across folds, indicating that the variability of the models is minimal and that their performance can be considered robust.

Ultimately, the decision regarding the most appropriate threshold should be guided by the specific research objectives. In contexts where the primary concern is to minimize false negatives (i.e., avoiding the omission of carnivore detections), a higher threshold may be justified. Conversely, in studies where maintaining precision is paramount, a lower threshold might be more advantageous.

6.3.4. Retraining with 20% of Grândola Samples

Using the same technique, training was repeated with 20% of the total images available as examples. In this case, the number of epochs required to reach the optimal model was, on average, slightly higher, with two folds benefiting from the lower learning rate.

A summary of the results obtained when the model was applied to the remaining 80% of the images is presented in Table 6.5:

Table 6.5 - Retraining with 20% of Grândola Samples results with 5-fold validatiom

	Q1		Median	Q2	
Max Train accuracy	0.882	0.8850	**0.8850**	0.8880	0.8880
Train Loss	0.6030	0.8719	0.9228	0.9347	0.9369
Precision					
Carnivore	0.8727	0.8752	**0.8807**	0.8816	0.8842
	0.8740			0.8829	
IQR	0.0090				
Median	0.8807				
Average	0.8789				
Other	0.8855	0.8898	**0.8918**	0.8969	0.8995
	0.8877			0.8982	
IQR	0.0105				
Median	0.8918				
Average	0.8927				
Recall					
Carnivore	0.8857	0.8923	**0.8932**	0.8998	0.9020
	0.8890			0.9009	
IQR	0.0119				
Median	0.8932				
Average	0.8946				
Other	0.8699	0.8716	**0.8778**	0.8800	0.8840
	0.8708			0.8820	
IQR	0.0113				
Median	0.8778				
Average	0.8767				
Validation Accuracy	0.8811	0.8848	**0.8857**	0.8866	0.8899

Using 20% of the dataset:

- Precision median improved to 88.07%.
- Recall median improved to 89.32%.
- Validation accuracy reached 88.57%.

Threshold 0.9 again yielded higher recall (93.27%) but reduced precision (82.55%).

As can be observed from table 6.6, and as expected, overall gains from 20% samples were incremental, but not dramatic compared to 10%.

In adition, table 6.6 presents a comparison of results for the different thresholds. The difference in recall for the model trained with 20% of the samples is similar to that observed with 10%—approximately four percentage points.

Table 6.6. Retraining with 20% of Grândola Samples results with variable thresholds

	0.5 threshold		0.7 threshold		0.9 threshold	
	Median	Average	Median	Average	Median	Average
Precision	88.07%	87.89%	84.98%	85.00%	82.55%	82.26%
Recall	89.32%	89.46%	90.73%	91.21%	93.27%	93.47%
F1-Score	88.69%	88.67%	87.76%	88.00%	87.58%	87.51%

Table 6.7 compares the medians (which end up having values very close to the mean) of precision and recall for the carnivore class for samples of 10% and 20% in training and thresholds of 0.5 and 0.9:

Table 6.7. – Comparing 0.5 and 0.9 thresholds on 10% and 20% sampling training

	10%				20%			
	0.5 threshold		0.9 threshold		0.5 threshold		0.9 threshold	
	Median	Average	Median	Average	Median	Average	Median	Average
Precision	86.01%	86.02%	82.23%	82.22%	88.07%	87.89%	82.55%	82.26%
Recall	87.57%	0.8725	91.17%	0.9127	89.32%	89.46%	93.27%	93.47%
F1-Score	86.78%	86.63%	86.47%	86.51%	88.69%	88.67%	87.58%	87.51%

As with the 10% sample model, the graphs from the 20% learning curve are also quite winding, but they eventually stabilize in the final epochs.

The F1-score exhibits slightly greater variability across thresholds yet remains consistently within a high and satisfactory range. This indicates that the observed gains in recall are achieved without a substantial reduction in precision.

Figures 6.4 and 6.5 present two examples of graphs obtained from this learning curve in different folds:

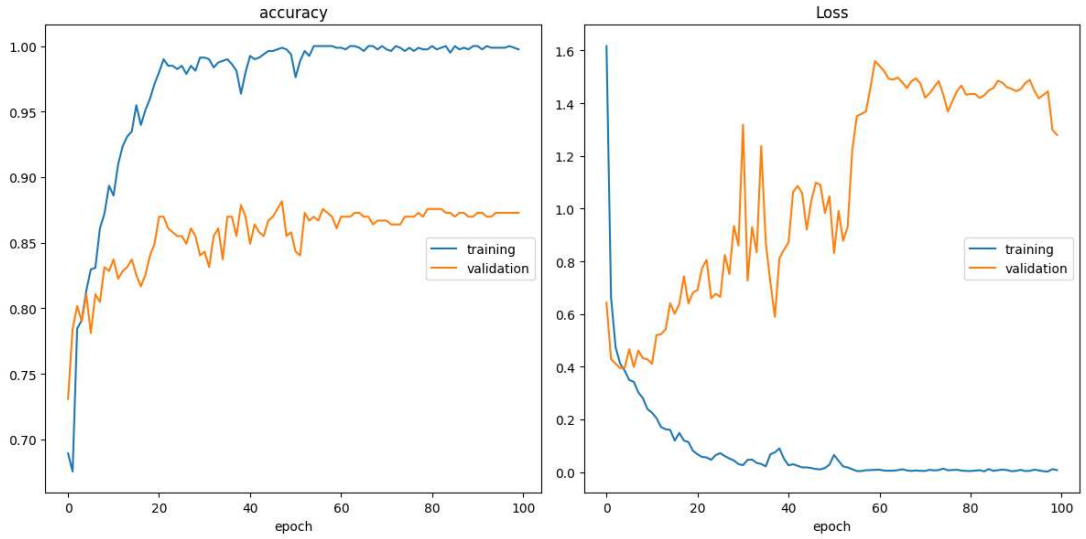


Figure 6.4 - learning progression with 20% sampling showing a regular learning curve

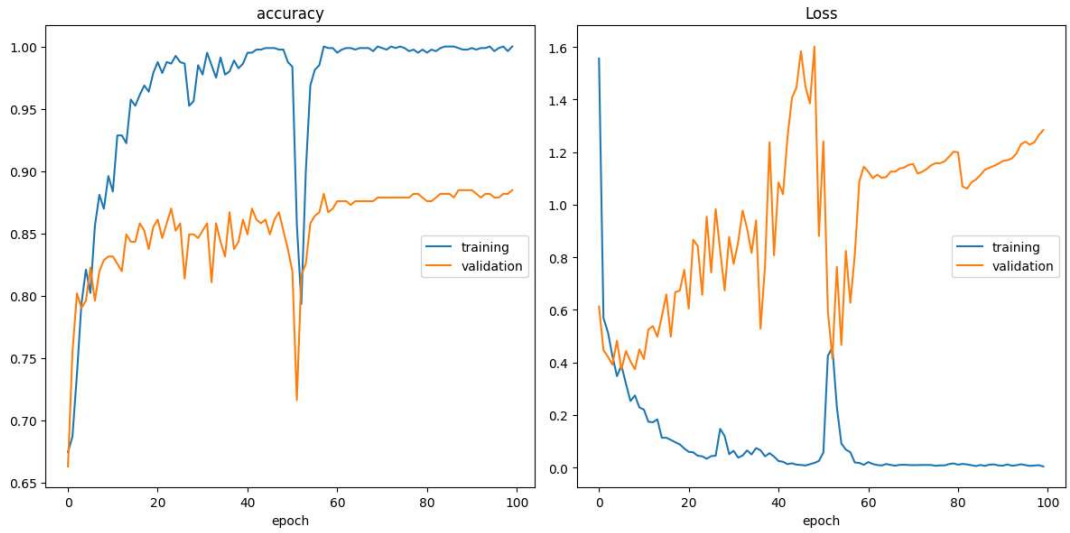


Figure 6.5 – learning progression with 20% sampling showing a very large dip during training but with similar best result in terms of accuracy

In the specific case of epoch 4, reading the graph revealed the question of whether the best epoch was the one automatically identified. To clarify, we tested the results of the models obtained in epochs 28 and 58. Although the loss was lower and the recall was slightly higher (90.59% and 90.81% vs. 90.20% for epoch 88), accuracy was much lower (87.89% and 86.05% vs. 88.07% for epoch 88).

All graphs show accuracy and loss because these were the only metrics available for automatic graphic creation. As such, definitions of good results were based in these metrics, and the aimed goal of maximizing recall was refined through changes in prediction threshold.

6.3.5. Direct Transfer Learning from ImageNet

At this stage of the project, it had become evident that each new location required a distinct learning process—an observation primarily drawn from the fact that the inclusion of additional locations into the main model consistently resulted in lower accuracy. This raised a critical question: is it possible to

construct, from scratch, a carnivore prediction model relying solely on transfer learning with the original VGG16 weights?

To address this, subsets corresponding to 10% and 20% of the images collected in Grândola were used, and the same training procedure previously applied to the Companhia das Lezírias datasets was replicated. The validation was performed using the portion of images not included in training.

The results obtained were significantly lower than those achieved through transfer learning based on the models trained with the Companhia das Lezírias data. Tables 6.8 and 6.9 provide a summary of these outcomes.

Table 6.8 – Results for direct transfer-learning from ImageNte on Grândola across 5-fold validation

Precision	Q1		Median	Q2	
Carnivore	0.7797	0.8007	**0.8028**	0.8176	0.8360
	0.7902			0.8268	
IQR	0.0366				
Median	0.8028				
Average	0.8074				
Other	0.7967	0.7974	**0.8013**	0.8054	0.8251
	0.7971			0.8153	
IQR	0.0182				
Median	0.8013				
Average	0.8052				
Recall					
Carnivore	0.7898	0.7905	**0.7956**	0.8070	0.8382
	0.7902			0.8226	
IQR	0.0325				
Median	0.7956				
Average	0.8042				
Other	0.7632	0.7991	**0.8046**	0.8238	0.8449
	0.7812			0.8344	
IQR	0.0532				
Median	0.8046				
Average	0.8071				

Table 6.9 – Results for using models obtained in C. Lezírias with 10% sampling from Grândola across 5-fold validation

Precision	Q1		Median	Q2	
Carnivore	0.8398	0.8451	**0.8478**	0.8536	0.8668
	0.8425			0.8602	
IQR	0.0178				
Median	0.8478				
Average	0.8506				
Other	0.8274	0.8475	**0.8569**	0.8579	0.8731
	0.8375			0.8655	
IQR	0.0280				
Median	0.8569				
Average	0.8526				
Recall					
Carnivore	0.8229	0.8475	**0.8549**	0.8615	0.8765
	0.8352			0.8690	
IQR	0.0338				
Median	0.8549				
Average	0.8527				
Other	0.8356	0.8479	**0.8492**	0.8497	0.8686
	0.8418			0.8592	
IQR	0.0174				
Median	0.8492				
Average	0.8502				

In comparative terms, the use of direct transfer resulted in substantially lower outcomes, both in precision and recall, with a decrease of up to 8 percentage points in recall for the 10% model. This finding indicates that, whenever possible, each new location should build upon a model previously trained on data from other sites.

Table 6.10 provides a comparative summary.

Table 6.10 – Results when using 10% and 20% sampling from Grândola using as base the model obtained from C. lezírias vs. Direct Transfer Learning with ImageNet data

	10% sampling		20% sampling	
	Base model	Direct transfer	Base model	Direct transfer
Precision	86.01%	80.28%	88.07%	84.78%
Recall	87.57%	79.56%	89.32%	85.49%
F1-Score	86.78%	79.92%	88.69%	85.13%

In summary, applying VGG16 pretrained weights directly to the Grândola dataset resulted in lower performance compared to models pretrained on Companhia das Lezírias.

With 10% sampling, the direct transfer model achieved a precision of 80.28% and a recall of 79.56%. Performance improved with 20% sampling, yielding 84.78% precision and 85.49% recall.

When comparing model performance in terms of the F1-score, the base model consistently outperformed the direct transfer approach across both sampling levels. With 10% sampling, the base model achieved an F1-score of 86.78%, whereas direct transfer lagged behind at 79.92%, a difference of nearly seven percentage points. Increasing the sampling proportion to 20% improved performance in both cases, with the base model reaching 88.69% and direct transfer 85.13%. Although the performance gap narrowed to approximately 3.5 percentage points, these results indicate that direct transfer remains less effective than training a base model under the given conditions.

This suggests that, in this ecological context, fine-tuning on locally sampled data is more advantageous than relying solely on direct transfer from ImageNet.

6.3.6. Reverse Validation: Grândola Models on Companhia das Lezírias

Another question that emerged was whether, after training and refining the model on new locations, it would be necessary to retain the earlier models for use in previous locations, or whether it would be reasonable to rely on the updated, transformed model.

In the case of direct transfer learning, this issue does not arise—the model trained exclusively on the Grândola dataset, when applied to the Companhia das Lezírias data, yielded inferior results due to its more limited training base, particularly when compared to the reverse scenario.

However, for models trained using 10% and 20% of the Grândola data through transfer learning from previously established models, the performance on Companhia das Lezírias data (similar to the original training samples but excluded from the base model training) was as follows. This comparison is particularly relevant to evaluate the extent to which transfer learning can generalize across distinct ecological contexts while maintaining robustness, accuracy and more importantly: recall.

Results of the Carnivore vs. Other reverse validation training

Table 6.11 – Results from validating 10% and 20% sampling models obtained in Grândola back in Companhia das Lezírias data

10% sampling						20% sampling					
Precision	Q1		Median	Q2		**Precision**	Q1		Median	Q2	
Carnivore	0.9552	0.9570	**0.9580**	0.9604	0.9643	Carnivore	0.9551	0.9561	**0.9687**	0.9604	0.9687
	0.9561			0.9624			0.9556			0.9646	
IQR	0.0063					IQR	0.0090				
Median	0.9580					Median	0.9687				
Average	0.9590					Average	0.9618				
Other	0.9107	0.9297	**0.9314**	0.9347	0.9351	Other	0.9118	0.9179	**0.9199**	0.9210	0.9297
	0.9202			0.9349			0.9149			0.9254	
IQR	0.0147					IQR	0.0105				
Median	0.9314					Median	0.9199				
Average	0.9283					Average	0.9201				
Recall						**Recall**					
Carnivore	0.9059	0.9272	**0.9314**	0.9347	0.9351	Carnivore	0.9074	0.9141	**0.9167**	0.9167	0.9272
	0.9166			0.9349			0.9108			0.9220	
IQR	0.0184					IQR	0.0112				
Median	0.9314					Median	0.9167				
Average	0.9269					Average	0.9164				
Other	0.9561	0.9580	**0.9602**	0.9617	0.9655	Other	0.9569	0.9584	**0.9606**	0.9617	0.9704
	0.9571			0.9636			0.9577			0.9661	
IQR	0.0066					IQR	0.0084				
Median	0.9602					Median	0.9606				
Average	0.9603					Average	0.9616				

Table 6.11 presents a full comparison of the Reverse Validation results obtained by applying models trained on 10% and 20% of the Grândola dataset back to Companhia das Lezírias. Table 6.12 shows these results in comparison with the original outcomes of the Companhia das Lezírias models.

Table 6.12 – Validation of the models obtainen in Grândola when comparing results from the ones obtained initially in C. Lezírias

	Companhia Lezírias	C. Lez. + Grândola	
		10% sampling	20% sampling
Precision	94.56%	95.80%	96.87%
Recall	96.10%	93.14%	96.06%

As previously verified, models trained exclusively on data from Companhia Lezírias achieved high performance. When a model trained on the combined dataset from Companhia Lezírias and Grândola was reapplied to the Companhia Lezírias data, results showed some variation depending on the sampling proportion.

With 10% sampling, precision increased slightly to 95.80%, but recall decreased to 93.14%, yielding a lower F1-score of 94.45%. With 20% sampling, precision rose further to 96.87% and recall recovered to 96.06%; however, the F1-score remained stable at 94.45%. These results suggest that while integrating heterogeneous data sources can improve precision and general robustness, it may initially reduce recall when applied back to a single site, particularly under limited sampling. With greater sampling, the model adapts better to inter-site variability, recovering recall while maintaining high precision. The stable F1-scores across conditions highlight a trade-off between precision gains and recall fluctuations in cross-site model transferability. The outcome, particularly with the model trained on 20% of the Grândola samples, was highly encouraging, as its re-application to the Companhia das Lezírias dataset essentially showed no appreciable loss in performance and suggests that, as models incorporate more diverse locations, they may gradually converge toward generalization, potentially reducing the need for retraining per location in the future.

6.4. Transfer Learning with Feature Extraction and Pooling

As previously noted, two distinct strategies were explored for implementing transfer learning in this study. The approach examined in this section is **feature extraction**, which leverages the representational capacity of a pretrained base model to generate informative features for adaptation to the target datasets [32][33][36].

The architecture is based on a transfer learning framework employing **VGG16** as a fixed feature extractor. Input images are resized to **224 × 224 × 3** and passed through the convolutional layers of the pretrained backbone, producing high-level feature representations of size **7 × 7 × 512** that capture both spatial and semantic information. To reduce dimensionality, a **GlobalAveragePooling2D** layer aggregates each feature map into a compact vector of 512 descriptors. A **dropout layer** with a rate of 0.8 is then applied to mitigate overfitting and enhance generalization. The classification stage consists of a fully connected dense layer with a single output neuron, enabling binary predictions.

This approach differs from the **flatten-based strategy** described in the previous section, where all activations are retained, resulting in a much larger feature space. By contrast, global average pooling compresses each feature map into a single scalar, thereby improving robustness and reducing overfitting—particularly valuable in ecological datasets with limited training samples—albeit at the cost of discarding some fine-grained spatial detail. The adoption of this method was therefore motivated by the need to balance representational richness with model simplicity, offering a complementary perspective to the flatten-based adaptation strategy.

The summary of the final model architecture is presented as follows:

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 56, 56, 3)]	0
tf.__operators__.getitem_1	(None, 224, 224, 3)	0

(SlicingOpLambda)		
tf.nn.bias_add_1 (TFOpLambda)	(None, 224, 224, 3)	0
vgg16 (Functional)	(None, 7, 7, 512)	14714688
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

==========

Total params: 14,715,201

Trainable params: 11,799,553

Non-trainable params: 2,915,648

_____

Overall, this design combines the representational strength of VGG16 with lightweight classification layers, balancing expressive feature extraction with computational efficiency.

It is worth noting that this technique of compacting the layers of the original model renders the training process considerably more efficient. While one epoch in the models studied in the previous chapter (using the original datasets) required nearly three minutes to execute, with this technique the execution time was reduced to approximately 1.5 minutes. These results were obtained using the same hardware and applying the **5-Fold Cross Validation** technique.

Overall, the results were very similar to those obtained with the previous method, showing slight improvements in **recall**, but with an equivalent degradation in **precision**. For this reason, the analysis will not be as detailed. Nevertheless, the main results are presented below.

6.4.1. Training carnivore vs. other

As previously mentioned, the results did not differ substantially from those obtained earlier with the flat base model; however, some minor improvements were observed.

Table 6.13 summarizes the results.

Table 6.13 – Results from training using feature extraction

Precision	Q1		Median	Q2	
Carnivore	0.9474	0.9499	**0.9514**	0.9561	0.9586
	0.9487			0.9574	
IQR	0.0087				
Median	0.9514				
Average	0.9527				
Other	0.9593	0.9634	**0.9675**	0.9694	0.9722
	0.9614			0.9708	

IQR	0.0094				
Median	0.9675				
Average	0.9664				
Recall					
Carnivore	0.9599	0.9636	**0.9681**	0.9700	0.9722
	0.9618			0.9711	
IQR	0.0093				
Median	0.9681				
Average	0.9668				
Other	0.9467	0.9490	**0.9505**	0.9554	0.9584
	0.9479			0.9569	
IQR	0.0091				
Median	0.9505				
Average	0.9520				

Table 6.14 – Median comparison on results obtained with Fine Tuning models done before and with Feature extraction

		Fine Tuning	Feature extraction
Precision	Carnivore	94.56%	95.14%
	Other	96.03%	96.75%
Recall	Carnivore	96.10%	96.81%
	Other	94.45%	95.05%
F1-Score	Carnivore	95.32%	95.97%
	Other	95.23%	95.89%

As shown in table 6.14, carnivores exhibited an increase in precision from 94.56% to 95.14% (+0.58 percentage points), an improvement in recall from 96.10% to 96.81% (+0.71 points), and a rise in F1-score from 95.32% to 95.97% (+0.65 points). A comparable pattern was observed for the 'other' class.

Although modest in magnitude, these improvements were consistent across both classes, indicating that feature extraction enhances precision and recall simultaneously, leading to more balanced and reliable classification outcomes than the Flatten-Base approach.

In an ecological monitoring context, even small but systematic gains in predictive performance are meaningful, as they help reduce misclassification bias across functional groups and contribute to more accurate estimates of species presence, abundance, and activity patterns.

6.4.2. Training extension using 10% and 20% of the Grândola samples

Similarly to the main model, the results obtained for the 10% and 20% sample trainings from the Grândola dataset differed only marginally from those achieved with the previously developed models.

The major advantage of this technique is undoubtedly its efficiency. Whereas, with the previous method, the best results were typically reached after 50 to 60 epochs, when using feature extraction comparable results can be achieved after only 20 to 30 epochs.

The summary table for these two training experiments is presented on table 6.15:

Table 6.15 – Results from using the new model with the Grândola dataset

10% Sampling						20% Sampling					
Precision	Q1		Median	Q2		**Precision**	Q1		Median	Q2	
Carnivore	0.8467	0.8535	**0.8551**	0.8586	0.8814	Carnivore	0.8582	0.8702	**0.8716**	0.8790	0.8807
	0.8501			0.8700			0.8642			0.8799	
IQR	0.0199					IQR	0.0157				
Median	0.9551					Median	0.8716				
Average	0.8791					Average	0.8719				
Other	0.8464	0.8511	**0.8751**	0.8832	0.8870	Other	0.8639	0.8819	**0.8851**	0.8868	0.8979
	0.8488			0.8851			0.8729			0.8924	
IQR	0.0364					IQR	0.0195				
Median	0.8751					Median	0.8851				
Average	0.8686					Average	0.8831				
Recall						**Recall**					
Carnivore	0.8390	0.8496	**0.8785**	0.8890	0.8891	Carnivore	0.8624	0.8857	**0.8857**	0.8879	0.9015
	0.8443			0.8891			0.8741			0.8947	
IQR	0.0448					IQR	0.0207				
Median	0.8785					Median	0.8857				
Average	0.8690					Average	0.8846				
Other	0.8390	0.8496	**0.8785**	0.8890	0.8921	Other	0.8536	0.8655	**0.8730**	0.8778	0.8800
	0.8443			0.8906			0.8596			0.8789	
IQR	0.0462					IQR	0.0194				
Median	0.8785					Median	0.8730				
Average	0.8696					Average	0.8700				

What can be inferred from the results for the carnivore class—the primary focus of this study—is that there are in fact very few differences compared with the previously obtained model, both in the case of training with 10% and with 20% of the data.

Table 6.16 presents a comparison for carnivore findings with the model analysed in the previous chapter:

Table 6.16 – Results obtained in Grândola through sampling in comparison with the ones obtained by the flatten-base model

	10% Sample training		20% Sample training	
	Flatten-Base	Feature extraction	Flatten-base	Feature extraction
Precision	86.01%	85.51%	88.07%	87.16%
Recall	87.57%	87.85%	89.32%	88.57%
F1-Score	86.78%	86.66%	88.69%	87.86%

When applying the models to a new site with limited labelled data for fine-tuning, the feature extraction approach did not lead to performance improvements over the Flatten-Base models. With 10% sample

training, precision slightly decreased (85.51% vs. 86.01%), recall showed only a marginal increase (87.85% vs. 87.57%), and the F1-score remained nearly unchanged (86.66% vs. 86.78%). Similarly, with 20% sample training, both precision (87.16% vs. 88.07%) and recall (88.57% vs. 89.32%) were lower for feature extraction, resulting in a reduced F1-score (87.86% vs. 88.69%).

These results suggest that, in the context of transfer to a new site, freezing pretrained feature layers (as in feature extraction) may limit the model's capacity to adapt to local data distributions, such as variations in habitat background, lighting, or species appearance. By contrast, the Flatten-Base approach allows the network to retrain end-to-end, enabling greater flexibility in capturing site-specific patterns. Consequently, the feature extraction technique was unsuccessful in improving metrics because it restricted adaptation to the new domain, whereas fine-tuning through the Flatten-Base approach better accommodated cross-site variability.

The primary benefit of this approach is, therefore, its efficiency, particularly in reducing computational demands and processing time.

6.5. Ensemble Models

Following the full cycle of model training and validation across different locations, distinct prediction patterns emerged among the various models. Although several candidate solutions were ultimately not selected as optimal, their performance plots revealed diverse behaviours: some models produced predictions skewed toward the extremes (closer to 0 or 1), while others tended to output probabilities clustered around one end of the spectrum.

Figures 6.6. and 6.7. show evidence of these two patterns.

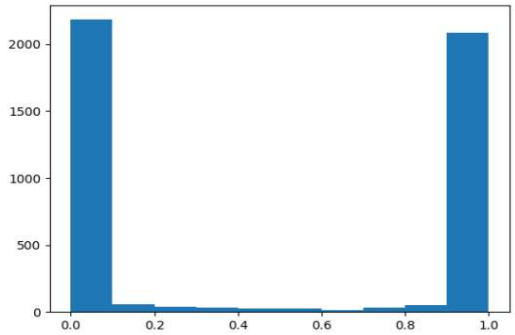


Figure 6.6 - Graph showing results on fold 0 with 20% of the data using transfer learning from the Grândola model

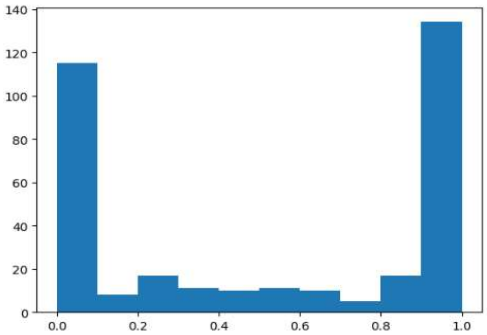


Figure 6.7 - Graph of results on fold 0 with 20% of the data – direct model

This variation in model behaviour suggests the potential utility of ensemble learning—a strategy that combines predictions from multiple individual models to improve overall performance. Techniques such as bagging, boosting, and stacking are well-documented in the literature for their ability to reduce variance, mitigate overfitting, and enhance generalization, especially in noisy or heterogeneous datasets [45].

In the present study, preliminary experiments with ensemble strategies yielded encouraging results. By aggregating predictions from models trained under different configurations and on varying datasets, we were able to achieve a more balanced trade-off between precision and recall. The ensemble method proved particularly effective in stabilizing performance across folds, capitalizing on the complementary strengths of individual models while compensating for their individual weaknesses.

Motivation

Given these insights, a natural question arose: Could combining several independently trained models improve performance? Concretely, for each image, we retrieved predicted probabilities from multiple models, averaged these values, and used the result as the final prediction. This corresponds to a soft-voting ensemble, where the final output is the mean of individual model probabilities.

This approach leverages the diversity among constituent models, aiming to reduce variance and produce more calibrated probability estimates. When the errors made by individual models are at least partially uncorrelated, ensemble averaging can increase recall without severely compromising precision.

In order to test this hypothesis, the following models were combined:

1. A model trained on the combined samples from *Companhia das Lezírias* together with 10% or 20% of the *Grândola* dataset.

2. A model trained on part of the *Companhia das Lezírias* data, followed by a second training phase with 10% or 20% of the *Grândola* data.

3. A model trained on the entirety of the *Companhia das Lezírias* data, followed by a second training phase with 10% or 20% of the *Grândola* data.

4. A model trained exclusively with 10% or 20% of the *Grândola* dataset.

The final formula for each prediction is represented on Formula 6.1:

(Model 1 + Model 2 + Model 3 + Model 4) / 4

Formula 6.1 – Formula for soft voting ensemble model tested

Initial experiments with this ensemble strategy showed promising results. The resulting sequence of performance graphs is presented in Figures 6.8 through 11:

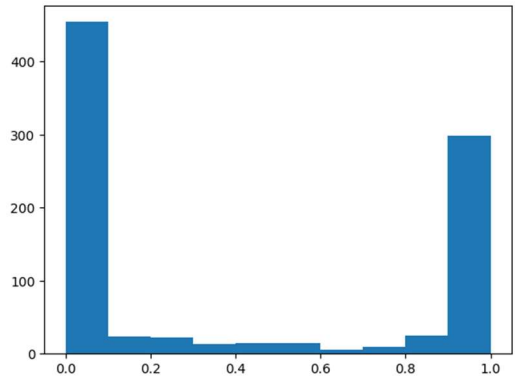


Figure 6.8 – Step 1 prediction distribution

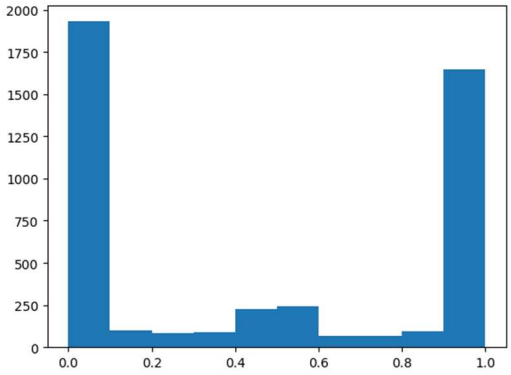


Figure 6.9 – Step 2 prediction distribution

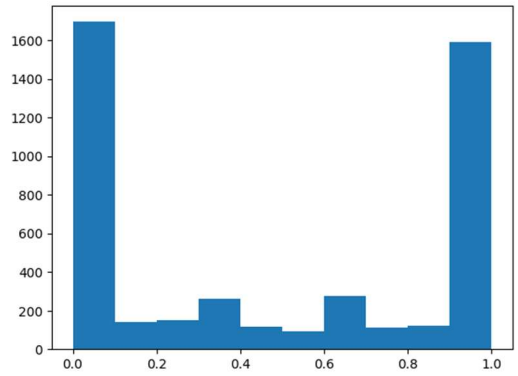


Figure 6.10 – Step 3 prediction distribution

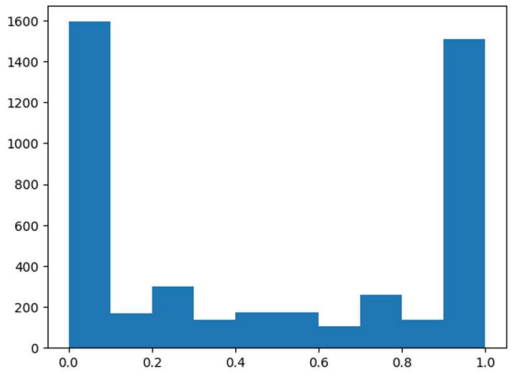


Figure 6.11 – Step 4 prediction distribution

Although the sequence of performance graphs indicates a progressive reduction in the model's prediction confidence, the overall recall for the carnivore class demonstrated a notable improvement. This statement may be verifiable in the following table.

The summary of the result obtained with a sample of 20% of the data from Grândola is represented on table 6.17:

Table 6.17 – Results obtained by using the Ensemble Model with Grândola sampling with 5-fold validation

Precision	Q1		Median	Q2	
Carnivore	0.8703	0.8707	**0.8707**	0.8761	0.8847
	0.8705			0.8804	
IQR	0.0099				
Median	0.8707				
Average	0.8745				
Other	0.9097	0.9160	**0.9175**	0.9191	0.9211
	0.9129			0.9201	
IQR	0.0073				
Median	0.9175				
Average	0.9167				
Recall					
Carnivore	0.9143	0.9209	**0.9209**	0.9235	0.9262

	0.9176			0.9249	
IQR	0.0072				
Median	0.9209				
Average	0.9212				
Other	0.8624	0.8633	**0.8637**	0.8695	0.8800
	0.8629			0.8748	
IQR	0.0119				
Median	0.8637				
Average	0.8678				

Table 6.18 - Comparative table between the Ensemble model and the previous best result for the carnivore class

	Simple Model	Ensemble Model
Precision	88.07%	87.07%
Recall	89.32%	92.09%
F1-Score	88.69%	89.51%

Although the ensemble approach resulted in a slight decrease in precision compared to the previous best-performing individual model, it achieved a **notable improvement in recall—nearly three percentage points higher** for the carnivore class. As previously stated, this trade-off is particularly relevant in the ecological monitoring contexts explored in this study, where **maximizing recall is often more critical than precision**, especially when the objective is to reduce false negatives (i.e., missed detections of target species).

Also, in terms of F1-score, the ensemble model (89.51%) outperformed the simple model (88.69%). This improvement (+0.82 points) reflects the ensemble's higher recall (92.09%) compensating for its slight loss in precision, resulting in a more balanced overall performance.

The ability of ensemble methods to capture a broader range of true positives suggests that **combining multiple models—each trained with different data subsets or configurations—can enhance sensitivity and robustness** in species classification tasks.

These results support the underlying hypothesis that this technique contributes positively to model generalization and performance consistency across varying data conditions. However, a comprehensive investigation into ensemble strategies—such as optimizing model selection, exploring weighting schemes, or comparing ensemble types (e.g., stacking versus soft voting)—would require a dedicated study with broader experimental scope and systematic evaluation. As such, while the findings are promising, **a full exploration of ensemble learning methods lies beyond the scope of the present work and is recommended as a direction for future research**.

7. Conclusion

This work set out to explore and evaluate the application of transfer learning for the automatic classification of wildlife camera-trap images, with a particular focus on distinguishing carnivores from other categories.

The analysis of camera-trap datasets revealed several specific challenges that affected classification performance. These included inconsistent labelling of non-target animals and humans, blurred or partial captures due to the nocturnal behaviour and motion of the species, and noise from unintended camera triggers. Camera placement and adjustments to reduce interference, as well as the exclusion of empty images in more recent datasets, further complicated automated classification. Collectively, these dataset-specific factors underscore the complexity of applying deep learning to ecological imagery and the need for careful model design, preprocessing, and evaluation, thereby motivating experiments with several models and parameter configurations.

Leveraging pre-trained convolutional neural network architectures, especially VGG16, demonstrated that transfer learning provides a robust framework for adapting models trained on large-scale datasets such as ImageNet to domain-specific ecological contexts with limited annotated data.

The choice to adopt VGG16, motivated by constraints on computational resources and the necessity to explore various architectural parameters, is supported by prior literature. As previously mentioned in the Preliminary Models section, He et al. (Deep Residual Learning for Image Recognition [27]) provide foundational evidence, while more recent studies from domains outside camera-trap applications also corroborate this selection. For instance, Deb et al. [82] demonstrate that VGG16-based models achieve superior performance across multiple evaluation metrics, surpassing baseline architectures including ResNet50. Similarly, Papoutsis et al. [83] report that VGG16 exhibits the fastest training times among the architectures considered.

The experimental results confirmed that VGG16, due to its relative simplicity and reduced computational demands, outperformed deeper and more complex architectures in resource-constrained environments while still achieving high levels of precision and recall. Furthermore, the use of k-fold cross-validation validated the robustness of the proposed approach, reducing the likelihood of overfitting and confirming the generalizability of the results. Complementary tests with different sampling strategies (10% and 20% of new data from Grândola) further revealed that even with relatively small subsets of data, the model was capable of adapting effectively to new environments, though with varying degrees of performance degradation depending on contextual differences in the datasets.

An important finding was the confirmation that each new geographical context represents a distinct learning challenge. While direct transfer learning yielded comparatively weaker results, incremental fine-tuning with previously trained models produced stronger outcomes, highlighting the value of cumulative learning across sites. This observation underscores the necessity of progressively enriching the dataset with diverse examples in order to improve generalization across ecological conditions.

In addition, the exploration of ensemble strategies, where predictions from multiple models were aggregated, showed promising results in increasing robustness and reducing the variability of outcomes. These findings suggest that ensembles represent a valuable avenue for further research, particularly in contexts where single-model predictions remain unstable or ambiguous.

In practical terms, the outcomes of this study provide a foundation for supporting field researchers by reducing the manual effort associated with image classification. By employing limited manually labelled

samples, it is possible to build models that extend classification to larger datasets with reasonable accuracy. This not only accelerates ecological research but also enhances scalability in wildlife monitoring initiatives.

Future work should address the systematic integration of ensemble approaches and voting mechanisms to further improve reliability, particularly in cases of high visual ambiguity. Additionally, the incorporation of temporal and contextual information—such as sequential image frames or environmental metadata—could enhance the capacity of models to disambiguate challenging cases. Expanding the diversity of datasets across different habitats and camera settings will also be essential to achieving more universally applicable models.

Overall, this study demonstrates that transfer learning, when carefully adapted and validated, is a powerful tool for ecological applications of deep learning. The insights gained not only advance methodological approaches in wildlife monitoring but also pave the way for practical deployment in real-world conservation contexts.

7.1. Future Work

The findings of this thesis demonstrate the feasibility of applying transfer learning approaches to wildlife image classification under conditions of limited and heterogeneous datasets. However, several avenues remain open for future research. First, increasing the diversity and volume of training data across different geographical locations would likely reduce the observed loss in performance when models are applied outside their original training domain. As shown in this work, the inclusion of even small percentages of new data from different environments improved generalization, suggesting that larger and more representative datasets could further enhance robustness.

Second, future efforts could explore the systematic optimization of hyperparameters and architectural adaptations, including the use of alternative feature extraction techniques such as Global Average Pooling. These strategies may help mitigate overfitting and improve the trade-off between precision and recall, particularly in challenging cases such as the identification of carnivores under varying environmental conditions.

Third, ensemble learning approaches represent a promising direction. Preliminary experiments conducted in this study indicated that combining predictions from multiple models can enhance classification stability and robustness. This aligns with the broader literature, where ensemble strategies have consistently demonstrated improved generalization in deep learning tasks. Thus, further research into voting strategies, stacking, or model averaging could provide valuable performance gains in wildlife monitoring applications.

Finally, expanding the scope of analysis to include semi-supervised or active learning methods would be beneficial. Given the high cost of manual annotation, techniques that leverage partially labelled or unlabelled datasets could significantly accelerate the development of reliable models while reducing dependence on extensive ground-truth data.

Taken together, these directions suggest that the integration of larger and more diverse datasets, architectural innovations, and ensemble-based strategies may yield increasingly accurate and practical solutions for supporting field researchers in biodiversity monitoring and conservation.

8. References

[1] Gomez Villa, A., Salazar, A., & Vargas, Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks"

[2] ImageNet. (n.d.). About ImageNet. Retrieved September 06, 2025, from https://www.image-net.org/about.php

[3] King, C. M., & Powell, R. A. (2007). The Natural History of Weasels and Stoats: Ecology, Behavior, and Management. Oxford University Press.

[4] Sato, J. J., Wolsan, M., Prevosti, F. J., D'Elía, G., Begg, C., Begg, K., Hosoda, T., Campbell, K. L., & Suzuki, H. (2012). Evolutionary and biogeographic history of weasel-like carnivorans (Musteloidea). Molecular Phylogenetics and Evolution

[5] Rovero, F., Zimmermann, F., Berzi, D., & Meek, P. (2013). "Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications. Hystrix, the Italian Journal of Mammalogy

[6] Gese, E. M. (2001). Monitoring of terrestrial carnivore populations. In J. L. Gittleman, S. M. Funk, D. Macdonald, & R. K. Wayne (Eds.), Carnivore Conservation

[7] Russell, Stuart; Norvig, Peter (1995). Artificial Intelligence: A Modern Approach. ISBN 0-13-103805-2.

[8] Saturn Cloud (July 10th 2023) "Understanding the Difference Between Flatten() and GlobalAveragePooling2D() in Keras"

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning (2nd ed.). Springer.

[10] Murphy, K. P. (2012). Machine learning: A probabilistic perspective. MIT Press.

[11] Ian Goodfellow, Yoshua Bengio, Aaron Courville (2016) "Deep Learning"

[12] Xia, C., Chen, Z., Li, B., & Luo, J. (2022). A survey of unsupervised learning in computer vision. ACM Computing Surveys, 55(6), 1–38. https://doi.org/10.1145/3501295

[13] Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2010). Semi-supervised learning. MIT Press.

[14] Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. Morgan & Claypool.

[15] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature. https://doi.org/10.1038/nature14539

[16] Aggarwal, C. C. (2018). Neural networks and deep learning: A textbook. Springer.

[17] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

[18] Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2021). A survey on deep learning for big data. Information Fusion, 42, 146–157. https://doi.org/10.1016/j.inffus.2017.10.006

[19] Szeliski, R. (2022). Computer vision: Algorithms and applications (2nd ed.). Springer.

[20] Forsyth, D., & Ponce, J. (2012). Computer vision: A modern approach. Pearson.

[21] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[22] Norouzzadeh, M. S., et al. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, 115(25), E5716–E5725. https://doi.org/10.1073/pnas.1719367115

[23] Litjens, G., et al. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

[24] Chen, C., Seff, A., Kornhauser, A., & Xiao, J. (2015). DeepDriving: Learning affordance for direct perception in autonomous driving. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2722–2730.

[25] Dosovitskiy, A., et al. (2021). An image is worth 16×16 words: Transformers for image recognition at scale. International Conference on Learning Representations (ICLR).

[26] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR). arXiv:1409.1556.

[27] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[28] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML).

[29] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[30] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[31] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[32] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[33] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(9), 1–40. https://doi.org/10.1186/s40537-016-0043-6

[34] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? Advances in Neural Information Processing Systems (NeurIPS), 27, 3320–3328.

[35] Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

[36] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. https://doi.org/10.1109/JPROC.2020.3004555

[37] Kolesnikov, A., et al. (2020). Big Transfer (BiT): General visual representation learning. European Conference on Computer Vision (ECCV).

[38] Rob J. Hyndman, Yanan Fan, Sample Quantiles in Statistical Packages, The American Statistician, Vol. 50, No. 4 (Nov., 1996)

[39] DeGroot, M. H., & Schervish, M. J. (2012). Probability and Statistics. Addison-Wesley.

[40] Saito, T., & Rehmsmeier, M. (2015). The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE, 10(3), e0118432.

[41] Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies.

[42] IBM. (n.d.). Convolutional Neural Networks (CNNs). IBM. Retrieved from https://www.ibm.com/topics/convolutional-neural-networks

[43] Gonçalo Curveira-Santosa, Tiago A. Marquesc, Mats Björklunda, Margarida Santos-Reis (2017) "Mediterranean mesocarnivores in spatially structured managed landscapes: community organisation in time and space"

[44] Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics Surveys.

[45] Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In International Workshop on Multiple Classifier Systems (pp. 1–15). Springer.

[46] Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms

[47] Breiman, L. (1996). Bagging Predictors. Machine Learning, 24(2), 123–140 .

[48] Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119–139.

[49] Wolpert, D. H. (1992). Stacked Generalization. Neural Networks, 5(2), 241–259 .

[50] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

[51] Oliphant, T. E. (2007). Python for scientific computing. Computing in Science & Engineering, 9(3), 10–20.

[52] Millman, K. J., & Aivazis, M. (2011). Python for scientists and engineers. Computing in Science & Engineering, 13(2), 9–12.

[53] H. Whitehead (1990) "Computer assisted individual identification of sperm whale flukes"

[54] S. A. Mizroch, J. A. Beard, and M. Lynde (1990) "Computer assisted photo-identification of humpback whales"

[55] Ravela, S., & Gamble, L. (2004). On Recognizing Individual Salamanders. Proceedings of the Asian Conference on Computer Vision (ACCV 2004), Part 2, pages 741–747. Springer.

[56] Burghardt, T., & Campbell, N. W. (2007). Individual animal identification using visual biometrics on deformable coat patterns. In *Proceedings of the 5th International Conference on Computer Vision Systems* (ICVS 2007). Springer. https://doi.org/10.1007/978-3-540-77296-5_21

[57] R. B. Sherley, T. Burghardt, P. J. Barham, N. Campbell, and I. C. Cuthill (2010) "Spotting the diference: towards fully-automated population monitoring of african penguins spheniscus demersus"

[58] L. Hiby, P. Lovell, N. Patil, N. S. Kumar, A. M. Gopalaswamy, and K. U. Karanth (2009) "A tiger cannot change its stripes: using a three-dimensional model to match images of living tigers and tiger skins"

[59] Fukushima, K., & Miyake, S. (1982) "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition"

[60] Chen, G., Han, T. X., He, Z., Kays, R., & Forrester, T. (2014). "Deep convolutional neural network based species recognition for wild animal monitoring"

[61] Manohar, Sharath Kumar, and Hemantha Kumar (2016) "Supervised and Unsupervised Learning in Animal Classification"

[62] Hung Nguyen, Sarah J. Maclagan, Tu Dinh Nguyen, Thin Nguyen, Paul Flemons, Kylie Andrews, Euan G. Ritchie and Dinh Phung (2017) "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring"

[63] Michael A. Tabak, Mohammad S. Norouzzadeh, David W. Wolfson, Steven J. Sweeney, Kurt C. Vercauteren, Nathan P. Snow, Joseph M. Halseth, Paul A. Di Salvo, Jesse S. Lewis, Michael D. White, Ben Teton, James C. Beasley, Peter E. Schlichting, Raoul K. Boughton, Bethany Wight, Eric S. Newkirk, Jacob S. Ivan, Eric A. Odell, Ryan K. Brook, Paul M. Lukacs, Anna K. Moeller, Elizabeth G. Mandeville, Jeff Clune, Ryan S. Miller (2018) "Machine learning to classify animal species in camera trap images: Applications in ecology"

[64] Ruilong Chen, Ruth Little, Lyudmila Mihaylova, Richard Delahay, Ruth Cox (2019) "Wildlife surveillance using deep learning methods"

[65] Brook, B. W., Buettel, J. C., van Lunteren, P., Rajmohan, P. P., & Aandahl, R. Z. (2025). *MEWC: A user-friendly AI workflow for customised wildlife-image classification*. **Peer Community Journal, 5**, article e57. https://doi.org/10.24072/pcjournal.565

[66] Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., & Packer, C. (2015). *Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna*. **Scientific Data**, 2, Article 150026. https://doi.org/10.1038/sdata.2015.26

[67] Beery, S., Van Horn, G., & Perona, P. (2024). *Evaluating the robustness of deep learning models for ecological metrics in camera trap images*. arXiv. https://arxiv.org/abs/2408.14348

[68] Sara Beery, Grant van Horn, Pietro Perona (2018) *Recognition in Terra Incognita* https://arxiv.org/abs/1807.04975

[69] Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., & Liang, P. (2021). *WILDS: A benchmark of in-the-wild distribution shifts*. In *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 5637–5664). PMLR. https://doi.org/10.5555/3495724.3496101 / https://wilds.stanford.edu/datasets/ "Datasets - WILDS"

[70] https://github.com/agentmorris/MegaDetector? "agentmorris/MegaDetector - GitHub".

[71] https://www.sciencedirect.com/science/article/abs/pii/S0925231225004989 "Enhancing generalization in camera trap image recognition"

[72] Ahumada, J. A., Fegraus, E., Birch, T., Flores, N., Kays, R., O'Brien, T. G., … Dancer, A. (2019). Wildlife Insights: A platform to maximize the potential of camera trap and other passive sensor wildlife data for the planet. *Environmental Conservation*. https://doi.org/10.1017/S0376892919000298

[73] TensorFlow. (n.d.). *TensorFlow official site*. Retrieved from https://www.tensorflow.org/

[74] Chollet, F. (2015). Keras. GitHub repository.

[75] Abadi, M., Barham, P., Chen, J., et al. (2016). TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 265–283.

[76] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[77] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks.

[78] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. Journal of Big Data.

[79] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. Proceedings of the IEEE Conference on Computer Vision and P

[80] Karl-Ludwig Besser, Bho Matthiesen, Alessio Zappone, Eduard A. Jorswieck (26th May 2020) "Deep Learning Based Resource Allocation: How Much Training Data is Needed?"

[81] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

[82] Nibedita Deb, Tawfikur Rahman (2025) *An efficient VGG16-based deep learning model for automated potato pest detection.* https://www.sciencedirect.com/science/article/pii/S2772375525006409

[83] Ioannis Papoutsis, Nikolaos Ioannis Bountos, Angelos Zavras, Dimitrios Michail, Christos Tryfonopoulos (2023) *Benchmarking and scaling of deep learning models for land cover image classification.* https://www.sciencedirect.com/science/article/pii/S0924271622003057

8.2. Further Reading

- Evidently AI. (2025, August 20). Accuracy vs. precision vs. recall in machine learning: What's the difference? In Classification metrics guide. Retrieved [Date of Access], from Evidently AI: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall
- Gonçalo Curveira-Santos, Nuno M. Pedroso, Ana Luísa Barros, Margarida Santos-Reis (2019) "Mesocarnivore community structure under predator control: Unintended patterns in a conservation context"
- Gyanendra K. Verma and Pragya Gupta (2018) "Wild Animal Detection Using Deep Convolutional Neural Network"
- IBM. (n.d.). *Machine Learning*. Retrieved from https://www.ibm.com/think/topics/machine-learning
- Keras.io. (n.d.). *Keras API Documentation*. Retrieved from https://keras.io/
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI).
- Kuntal Kumar Pal, Sudeep K. S. (2016) "Preprocessing for Image Classification by Convolutional Neural Networks"
- N. Banupriya, S. Saranya, Rashmi Swaminathan, Sanchithaa Harikumar, Sukitha Palanisamy (2020). "Animal detection using deep learning algorithm."
- Pedro Marcelino (October 23rd 2018) "Transfer learning from pre-trained models"
- Stefan Schneider, Graham W. Taylor, Stefan S. Linquist, Stefan C. Kremer (2018) "Past, Present, and Future Approaches Using Computer Vision for Animal Re-Identification from Camera Trap Data"