Advancing Wildlife Image Classification: A Comparative Study of Supervised and Self-Supervised Learning in Tiputini, Ecuador

Diego Villacreses, *Student Member IEEE,* Felipe Grijalva, *Senior Member IEEE,*

***Abstract*—The loss of biodiversity poses a major global challenge that requires advanced tools for effective wildlife monitoring. Camera traps generate large datasets, but the high cost and time required for manual annotation reduces their utility. This thesis investigates the application of supervised and self-supervised learning approaches to the classification of wildlife images using data from the Tiputini Biodiversity Station in Ecuador, a hotspot of global biodiversity. It evaluates state-of-the-art architectures, including Vision Transformers (ViT), ResNet, EfficientNet, and the self-supervised SimCLR framework, addressing challenges such as limited labeled data, imbalanced classes, environmental variability, and challenging image quality.**

A systematic comparison of results allows to assert a superior performance of ViT, achieving a classification accuracy of 90.1% on highly challenging datasets. SimCLR, while underperforming compared to supervised approaches, underscores the potential to take advantage of unlabeled data in resource-constrained ecological contexts. Key contributions include a tailored machine learning pipeline for biodiversity data, exhaustive hyperparameter optimization, and analyzing the trade-offs between computational efficiency and accuracy.

This research emphasizes the importance of fine-grained feature detection in ecological AI applications and suggests future exploration of self-supervised learning methods.

***Index Terms*—Supervised Learning, Self-Supervised Learning, Vision Transformers (ViT), SimCLR, Wildlife Image Classification.**

I. Introduction

BIODIVERSITY loss is an increasing global concern given the current rate of species extinction, estimated to be tens of times higher than the historical average over the past ten million years [1]. Effective biodiversity monitoring is essential for assessing human impact on habitats and species [2]. Camera traps have emerged as a widely used tool for non-invasive wildlife monitoring, generating vast amounts of visual data from biodiversity hotspots like the Tiputini Biodiversity Station (TBS) in Ecuador [3], [4].

However, the sheer volume of collected data, combined with the high cost and time required for manual annotation, represents a significant challenge. Deep learning models have shown promising results in automating wildlife image classification, offering scalable solutions to these challenges [5], [6], [7]. Although the final goal is to monitor population trends and species dynamics, much of the current research remains focused on improving classification accuracy [8].

The limited availability of high-quality labeled datasets has constrained progress, particularly in biodiversity-rich regions like TBS, where existing models have struggled to achieve satisfactory results [4]. This study aims to address these limitations by leveraging transfer learning with state-of-the-art architectures and exploring self-supervised learning techniques to improve classification performance on the TBS dataset.

A. Context and Problem Statement

Biodiversity monitoring is the cornerstone to understand ecosystems' health, to assess the impact of environmental changes, and inform conservation strategies, especially in the face of climate change and wide animal species extinction [2], [9]. Advances in camera trap technology have revolutionized wildlife monitoring, enabling large-scale, non-invasive data collection in remote ecosystems [3], [10]. However, the scale of these datasets poses significant challenges, particularly, given that data annotation is costly, time-consuming, and prone to error [7], [11], [2].

Deep learning has emerged as a powerful framework for automating image classification reducing dependence on manual annotation [12], [10]. Supervised learning approaches, particularly, Convolutional Neural Networks (CNNs) like ResNet [13] and EfficientNet [14], have demonstrated strong performance in general image classification tasks. Vision Transformers (ViT), which are based on attention mechanisms, have shown superior performance to CNNs in capturing fine-grained features for specific tasks [15].

Despite their success, applying these models to wildlife classification remains challenging due to three primary factors:

- **Data scarcity:** Supervised learning models require extensive labeled datasets, which are often unavailable in biodiversity monitoring due to the high cost of annotation. This reliance on annotated data represents a significant bottleneck in scaling Deep Learning based

D. Villacreses and F. Grijalva are with Universidad San Francisco de Quito USFQ

solutions for real-world biodiversity monitoring tasks [11], [7]. [12] states that addressing the scarcity of labeled data requires developing methods to leverage large amounts of unlabeled data such as unsupervised or self-supervised learning.
- **Domain generalization:** Models trained on datasets such as ImageNet often fail to generalize to ecological data due to domain shifts in species and environmental contexts, limiting their applicability across diverse ecosystems [10], [11].
- **Environmental factors:** Wildlife images are frequently low resolution, poorly lit, and occluded, further complicating classification tasks and requiring models robust to noisy inputs [3], [4].

B. Research Context: Tiputini Biodiversity Station

The Tiputini Biodiversity Station (TBS), located in Yasuní National Park, Ecuador, is a globally recognized hotspot of biodiversity, with hundreds of species recorded in a single square kilometer [16]. Over 97,000 images have been captured through camera traps at TBS, providing a rich dataset for ecological research. However, only 5,214 images have been labeled, focusing on two primary species: *Taypec* (65.1%) and *Taytaj* (34.9%). The classification of species in Tiputini presents unique challenges [4]:

- **Visual similarity:** The two species exhibit subtle morphological differences, making it difficult to differentiate them even for human annotators (Image 1).
- **Imbalanced data:** The dataset is skewed, with fewer images of *Taytaj* compared to *Taypec*, which can bias model training and evaluation. To mitigate this, undersampling of the majority class was employed, though alternative techniques such as oversampling or weighted loss functions could be explored in future work.
- **Environmental variability:** Images often include occlusions, poor focus, and diverse lighting conditions, reflecting the complexities of field data collection. **Non-independence of data:** Each image may contain multiple animals, which introduces the risk of having related instances (e.g., two animals from the same image) appearing in both training and testing sets during cross-validation. This could lead to data leakage. To address this, cross-validation is carefully designed to ensure that all animals from the same image are kept exclusively within either training, validation or testing set.

C. Research Objective and Scope

To address these challenges, the general objective of this study is to assess the effectiveness of supervised and self-supervised learning models for wildlife classification in TBS. The specific objectives are:

- **Specific Objective 1:** To compare the classification performance of supervised models (ResNet, MobileNet, EfficientNet and ViT) using transfer learning with different percentages of frozen layers and hyperparameter configurations, and a self-supervised approach (SimCLR) on a small labeled dataset.
- **Specific Objective 2:** To assess the trade-offs between computational efficiency and classification accuracy for these models in a resource-constrained ecological context.

D. Contributions

This research contribution is threefold:

1) **Comparative Analysis:** A systematic evaluation of supervised models using transfer learning with varying degrees of layer freezing and hyperparameters, and self-supervised models for wildlife classification.
2) **Tailored Pipeline:** Development of a robust machine learning pipeline, including data augmentation, transfer learning configurations, and hyperparameter optimization, designed to address the specific challenges of the TBS dataset.
3) **Hyperparameter Insights:** Exploration of the impact of key hyperparameters (e.g., learning rate, batch size, and number of epochs) on model performance, contributing to a deeper understanding of how to fine-tune deep learning models for biodiversity monitoring tasks.

E. Structure of the Document

The rest of this thesis is organized as follows:

- The **State of the Art** section states a review of the relevant literature and highlights how this document contributes to the current discussion in wildlife image classification.
- The **Materials and Methods** section provides an overview the TBS dataset, preprocessing steps, and experimental setup, including model configurations, hyperparameter tuning strategies, and evaluation metrics.
- The **Results and Discussion** section presents and analyzes the performance of the supervised and self-supervised models, emphasizing the role of transfer learning and hyperparameter tuning.
- The **Conclusion** summarizes the key findings, discusses limitations, and proposes directions for future work.

II. State of the Art

A. Deep Learning

Supervised Deep learning methods remain the predominant paradigm in computer vision, excelling in image classification when sufficient labeled data is available [17], such as Imagenet [18]. When labeled data is limited, alternative approaches tend to show better results, after a careful literature review, we want to highlight the following approaches:

i) transfer learning, ii) semi-supervised learning, iii) self-supervised learning and iv) active learning [19], [20], [21], [22]. A brief discussion about these approaches is included:

- **Transfer learning** uses models pretrained on large datasets and fine-tunes them for specific tasks with limited labeled data [23]. This approach reduces the need for extensive domain-specific data, making it particularly effective in scenarios like wildlife monitoring, where annotated datasets are small. Its advantages include reduced training time and improved performance on small datasets (compared against pure supervised learning). However, transfer learning assumes that the pretrained features are transferable to the target domain, which may not always hold true, particularly when the source and target domains differ significantly [24]. Recent research shows that the excessive usage of ImageNet is leading to overfitting and poor generalization when test images are *harder* than train images [25].
- **Semi-Supervised learning** leverages a small amount of labeled data from a large unlabeled dataset [26]. Two common approaches are: i) *Self-Training*: trains a model on labeled data, pseudo-labels unlabeled samples, and iteratively retrains with the expanded dataset; ii) *Label Propagation*: models data as a graph, spreading labels to unlabeled nodes based on similarity. However, this group of models heavily depends on the assumption that the unlabeled data follows a similar distribution as the labeled data [27].
- **Self-Supervised learning** uses a pretext tasks to learn from unlabeled data, enabling the model to acquire meaningful representations independently of human labeling [28]. Frameworks such as SimCLR have shown impressive results on benchmark datasets, narrowing the performance gap with supervised methods [28]. The main advantage of self-supervised learning is its ability to exploit large-scale unlabeled datasets, which are more readily available. However, the choice of pretext tasks and computational requirements can be significant challenges.
- **Active Learning** aims to minimize labeling effort by identifying the most informative samples for annotation. By iteratively asking a human annotator to label the most uncertain samples, active learning reduces the overall data labeling cost [29]. This framework has shown promising results in domains like medical diagnostics [30] or biodiversity monitoring [20]. However, the iterative nature of the process can be restrictive when the labeling team is no longer available.

Considering that we no longer have access to a team that could correctly label more images Active Learning is not a viable option. According to [28] Self-Supervised Learning tends to outperform Semi-Supervised Learning, statement confirmed by [31] for wildlife images. Additionally, in this study, the primary distinction between Supervised Learning and Transfer Learning lies in the percentage of frozen layers. Therefore, throughout this document, the term "Supervised Learning" includes both approaches unless otherwise specified.

B. Challenges of Camera-Trap Data

Based on the methodologies reviewed in the previous subsection, we now proceed to discuss the applications of deep learning in wildlife classification. While supervised and self-supervised learning offer powerful frameworks for image classification, their deployment in real-world contexts like wildlife monitoring must address the challenges presented by camera-trap data. These difficulties impact data quality and a tailored literature review must be performed to select relevant supervised architectures. Lets start with the challenges of camera trap data.

- **Environmental Variability:** Illumination issues, such as poor lighting at night, and weather conditions like fog or rain, often result in low-quality images [3], [5], [32]. Temporal changes, including seasonal variations, exacerbate inconsistencies within the dataset [3].
- **Motion Blur** Fast-moving animals often appear blurred, and some animals occupy only a small fraction of the frame, making feature extraction and classification more difficult [3], [5].
- **Occlusions and Perspective Issues:** Vegetation, environmental obstructions, or other animals frequently occlude the subject. Additionally, different proximity to the camera also increase the difficulty of computer vision tasks [3], [32].
- **False Triggers and Dataset Noise:** Non-animal triggers caused by wind, vegetation, or human activity often dominate datasets, requiring extensive preprocessing to remove irrelevant images [3], [5].
- **Dataset Imbalance:** Wildlife datasets are typically imbalanced, with rare species underrepresented and non-animal images disproportionately frequent [5], [32].
- **Camera Malfunctions:** Equipment issues, such as lens discoloration or hardware failures, introduce additional artifacts and noise into the dataset [3].

To address these challenges, specialized preprocessing techniques or human-driven image selection are often required. Also, a rigorous approach to result analysis is required to ensure reliable results.

1) Supervised Learning Applications in Wildlife Classification: Supervised Learning is the dominant paradigm for computer vision tasks, including wildlife classification, thanks to its ability to extract meaningful features from image data [12]. In the context of wildlife images obtained from camera traps, Supervised Learning address the complex challenges mentioned earlier when paired with Data Augmentation techniques. The use of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) has become prevalent, showing promising performance across diverse datasets.

Convolutional Neural Networks (CNNs). CNNs has been extensively employed in wildlife classification, the following architectures have shown promising results:

- **ResNet:** [11] reported an 83% accuracy using ResNet-50 (with a similar result using VGG-16) on a dataset compiled from camera traps in Gorongosa National Park, Mozambique. This dataset contains over 111 thousand images from 20 species and did not suffer from occlusion or heavy blurring, although it faced some challenges related to lighting conditions. Additionally, [10] demonstrated that ResNet-101 achieved an accuracy of 91.5% on the Snapshot Serengeti dataset for an image classification task. This dataset consists of 3.2 million camera trap images, including 48 labeled species. However, the data suffered from issues such as heavy imbalanced classes, blurring due to animal movement, varying lighting conditions, and differences in the distances of the targets.
- **VGG:** [32] leveraged VGG-16 with transfer learning to achieve an accuracy of 89.12%. The dataset comprised over 33 thousand labeled images from 19 species captured via camera traps in the Ladakh region of India. The authors do not discuss the complete extent of image-related challenges but mention that, as in other studies, issues as changing weather, improper illumination, and obstructed camera angles were present. They addressed the most problematic images by manually deleting outliers. Similarly, [22] employed the Missouri Camera Traps dataset, which consists of approximately 20,000 images distributed across six animal classes. Class imbalance and environmental variability, including differences in lighting, weather conditions, and backgrounds, made the classification task challenging. The authors found that VGG-16 achieved the best classification accuracy at 69.5%. As previously mentioned, [11] observed nearly identical results when comparing ResNet-50 and VGG-16.
- **EfficientNet:** Although EfficientNet is less frequently used than other architectures, it demonstrates strengths in training speed and achieves superior results on certain datasets. [5] applied EfficientNet to rank 7th in the iWildCam 2019 competition. The corresponding dataset contains over 243,000 training images representing 14 animal species. Key challenges included class imbalance, poor image quality, and regional variations.

Vision Transformers (ViTs): The introduction of Vision Transformers (ViTs) has marked a important shift in wildlife classification, moving from traditional CNNs to Transformer based architectures. Recent research demonstrating their superior performance on certain datasets. For example, [33] highlights the application of ViTs in marine animal classification, achieving higher accuracy (90%) than ResNet-50 on a dataset of marine wildlife containing 600 training samples across eight species. The dataset exhibits significant class imbalance, with images often displaying blurring and cluttered backgrounds. Similarly, [34] utilized the OpenAnimalTracks dataset, a challenging classification dataset designed to predict animal species based solely on their tracks. This dataset comprises 3579 images from 18 different species and suffers from severe class imbalance. ViT outperformed ResNet-50, VGG-16, and EfficientNet, achieving an accuracy of 68.01%.

C. Self-Supervised Learning

Although Supervised Learning is the dominant paradigm for computer vision tasks, when facing label scarcity Self-Supervised Learning could help to significantly improve accuracy [12], [28]. A recent literature review of 350 papers concluded that self-supervised learning plays a critical role in the development of ChatGPT-3 and ChatGPT-4, given its ability to effectively leverage unlabeled data, which constitutes the most prevalent dataset type for current GPT models [35]. For camera trap images, [36] shows an improvement using SimCLR over Supervised Learning in the Serengeti dataset, the same used by [10], reaching an accuracy of 94.4%, almost 3 percentage points improvement compared against ResNet-101. Other studies, like [37] evaluate various Self-Supervised Learning methods applied to camera trap datasets. The authors compare three methods: SimCLR, MoCo v2, and a novel temporal positive selection (TPS) method. On the Snapshot Serengeti dataset, TPS achieves the best results, with an accuracy of 64.2% using only 1% of labeled data, outperforming SimCLR (61.8%) and MoCo v2 (62.1%). At 10% labeled data, TPS reaches 76.2%, surpassing SimCLR (74.8%) and MoCo v2 (75.4%). This study concludes that Self-Supervised Learning hold great promise for biodiversity monitoring as unlabeled datasets expand. Although SimCLR is not always ranked as the top classification algorithm, its performance difference compared to the best performer is usually minimal. Moreover, its implementation is relatively straightforward compared to MoCo or other tailored algorithms. Therefore, it is the algorithm we will use in this document.

D. Supervised and Self-Supervised Algorithms

Considering the literature review from the last subsection, we are going to train ResNet, VGG, EfficientNet, Vision Transformers (ViT), and SimCLR. This subsection provides a technical overview of those algorithms.

- **ResNet (Residual Networks)**: introduced by [13], focus on reducing the vanishing gradient problem by introducing residual connections. These connections facilitate the training of deep architectures by enabling the learning of identity mappings, which allows a stable gradient movement during backpropagation. The backbone of ResNet's architecture consists of: i) a residual block, which bypass the input to an intermediate layer; ii) a convolutional layer; iii) batch normalization; iv) ReLU activation functions. ResNet-152 achieves a top-1 accuracy of **77.0%** on the ImageNet dataset [18].

- **VGG (Visual Geometry Group Networks)**: developed by [38], is characterized by its use of small convolutional filters (3x3) throughout the network, combined with max-pooling layers to gradually reduce spatial dimensions and increase the number of feature channels, enabling the extraction of more abstract and complex features at each stage. This uniform configuration enhances feature hierarchy learning and simplifies the design compared to earlier architectures. VGG16 achieves a top-1 accuracy of **71.5%**, while VGG19 reaches **71.6%** on ImageNet.
- **EfficientNet**: proposed by [39], introduces a compound scaling method to systematically scale network dimensions—depth, width, and input resolution—using a fixed scaling coefficient. The architecture is based on mobile inverted bottleneck convolutions (MBConv) and incorporates squeeze-and-excitation blocks for improved channel-wise feature recalibration. The EfficientNet-B7 variant achieves a top-1 accuracy of **84.4%** on ImageNet, while being significantly more computationally efficient than traditional architectures like ResNet and VGG.
- **Vision Transformers (ViT)**: Vision Transformers (ViT), introduced [15], uses self-attention instead of convolutional layers. Self-attention consist in dividing input images into fixed-size subsets, which are linearly represented into a sequence and then processed by a transformer decoder. The self-attention mechanism enables ViT to model to learn dependencies between all subsets within the input. Pretrained on ImageNet-21k and fine-tuned on ImageNet, ViT achieves a top-1 accuracy of **84.0%** with the ViT-B/16 configuration.
- **SimCLR (Simple Framework for Contrastive Learning)**: introduced by [28], is a self-supervised model that learns by maximizing the similitude between augmented frames (heavily distorted views of the original image) of the same image while separating frames of different images in a latent space. SimCLR achieve this objective by optimizing a contrastive loss. When fine-tuned on ImageNet, SimCLR achieves a top-1 accuracy of **76.5%** using a ResNet-50 encoder. SimCLR showcases the ability of self-supervised learning to rival or even outperform supervised algorithms in computer vision classification tasks.

E. Agnostic Approach to Training

Given the uncertainty about how pretrained features align with the unique characteristics of biodiversity datasets, this study adopts an agnostic approach to model training:

1) **Fully Retraining All Layers**: When labeled data is sufficiently available, retraining models like ResNet, MobileNet, EfficientNet, or ViT from scratch allows them to learn task-specific representations. However, this approach requires significant computational resources and risks overfitting in scenarios with limited data [12].
2) **Transfer Learning with Partial Fine-Tuning**: Transfer learning is particularly useful when labeled data is scarce or computational resources are limited to re-train the full architecture. Given the pretrained weights, it keeps knowledge from low-level features, such as edges and textures, while fine-tuning the last layers. When fine-tuning, the model learn high-level features of the data that usually represent data-specific information [17].
3) **Self-Supervised Learning with SimCLR**: offers an alternative by leveraging unlabeled data [28], we are going to use ResNet as encoder, retraining the full model without experimenting with percentage of frozen layers for computational reasons.

F. Key Insights

By evaluating these architectures under fully retrained, partially fine-tuned, and SimCLR-based self-supervised settings, this study represents a comprehensive analysis of current Computer Vision classifiers' performance for wildlife image classification.

III. Materials and Methods

A. Materials

1) Dataset Description: The datasets utilized in this study were collected at the Tiputini Biodiversity Station (TBS) in Yasuní National Park, Ecuador, one of the most biodiverse regions in the world [16]. These datasets are categorized into labeled and unlabeled subsets:

Labeled Dataset:: This dataset is the same as the one used on [4], it comprises of 5,214 images manually annotated and selected by biology experts. The dataset include two species: *Taypec* and *Taytaj*, species selected for their physical similitude, posing a particularly difficult task for computer vision. These images were captured using camera traps. All labeled images were resized to a fixed resolution of 416x416 pixels, given that our objective is pure classification we decide to use bounding box as given, hence, for each image we only use the information within bounding box.

Figure 1: Representative images of labeled data

Key characteristics of the labeled dataset:

- **Species Distribution:** The dataset is imbalanced, with 65.1% of images classified as *Taypec* and 34.9% as *Taytaj*.
- **Animal Count per Image:** *Taypec* images contain an average of 2.88 animals, with a maximum of 12, whereas *Taytaj* images feature an average of 1.71 animals, with a maximum of 7.
- **Diversity in Capture Conditions:** Images include a range of lighting conditions, times of day, and varying environmental backgrounds. Characteristics handpicked in order to provide the most diverse and challenging dataset.

Unlabeled Dataset:: The unlabeled dataset is significantly larger, containing approximately 97,000 high-resolution images captured through the same camera traps. Unlike the labeled dataset, it includes images of around 70 different, unidentified species, which remain unlabeled [4]. These images were processed using the MegaDetector-v5, model applied for this particular dataset by [40], which identified 45,000 frames containing wildlife and provided a bounding box, we only use the information within the bounding box. We used an arbitrary cut-point of 85% of confidence in order to select images with animals. This cut-point could be used as a hyperparameter in future studies.

Details about the unlabeled dataset:

- **Resolution Variability:** The images retain their original high-resolution format, which varies across the dataset. This higher resolution aids in detecting fine-grained details but adds computational time to the analysis.
- **Species Diversity:** The dataset features a diverse array of species and environmental conditions, which enhances its value for self-supervised learning approaches like SimCLR, where feature representation from a wide variety of contexts increases the probability of the model to improve accuracy in validation and test sets.
- **Role in Study:** This dataset complements the labeled subset by providing a rich source for unsupervised training.

2) Hardware and Software Specifications: The computational experiments were conducted using advanced hardware and software configurations to allow for the dataset size and complexity:

- **Hardware:**
  - NVIDIA A100 GPU with 80 GB of VRAM.
- **Software:**
  - **Frameworks and Libraries:**
    - Python: 3.10.15
    - PyTorch: 2.5.1+cu12.1
    - Torchvision: 0.20.1+cu12.1
    - PyTorch Lightning: 2.4.0
    - Transformers: 4.46.3
    - PIL (Pillow): 11.0.0
    - Pandas: 2.2.2
    - Numpy: 1.26.4
  - **Operating System:**
    - Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-124-generic x86_64)

3) Preprocessing Techniques: To optimize the datasets for training and evaluation, distinct preprocessing pipelines were applied to the labeled and unlabeled datasets, tailored to their respective roles in the study.

Unlabeled Dataset:

For the unlabeled dataset, a preprocessing pipeline was designed to maximize the effectiveness of SimCLR based on the recommendations from [41] a follow up paper from the original where states a possible best preprocessing for SimCLR.

- **Random Horizontal Flip:** Flips the image horizontally with a 50% probability.
- **Random Resized Crop:** Crops a random portion of the image and resizes it to 224×224 pixels, introducing positional variation.
- **Random Color Jitter:** Applies brightness, contrast, saturation, and hue adjustments with a probability of 0.8. Particularly:
  - Brightness, contrast, and saturation: randomly modified within $\pm 50\%$.
  - Hue: Randomly modified within ± 0.1.
- **Random Grayscale:** Introduces grayscale with a probability of 0.2, simulating images with reduced color information.
- **Gaussian Blur:** Introduces a Gaussian blur with a kernel size of 9, which applies a smoothing effect, simulating lower image clarity.
- **Normalization:** Normalizes the image pixel values to have a mean of 0.5 and a standard deviation of 0.5 for each channel, improving model convergence.

Labeled Dataset:

Due to the high degree of visual similarity between the *Taypec* and *Taytaj* classes, we adopted a comprehensive data augmentation pipeline to increase intra-class variance and improve generalization performance. Similar approaches have been suggested in prior research, which highlights the effectiveness of augmentation strategies such as random cropping, flipping, color jittering, and Gaussian blurring in handling visually challenging datasets [42]. The following steps were applied:

- **Training Preprocessing:**
  - **Random Resized Crop:** Crops a random region of the image and resizes it to the target size. The scale of the crop is randomly selected between 80% and 100% of the original size, with an aspect ratio chosen between 0.75 and 1.33.
  - **Random Horizontal and Vertical Flip:** Flips the image horizontally and vertically, each with a probability of 0.5.
  - **Random Color Jitter:** Introduces random variations in brightness, contrast, saturation, and hue,

with maximum changes of 0.4, 0.4, 0.4, and 0.2, respectively, applied with a probability of 0.8.
  - **Random Grayscale:** Converts the image to grayscale with a probability of 0.2.
  - **Gaussian Blur:** Applies a Gaussian blur with a kernel size of 5 and a sigma range of 0.1 to 2.0, applied with a probability of 0.5.
  - **Random Affine Transformations:** Applies geometric transformations simulating variations in viewpoint and orientation:
    * **Rotation:** Randomly rotates the image to a maximum of 15 degrees.
    * **Translation:** Shifts the image horizontally and vertically to a maximum of 10% of its size.
    * **Scaling:** Scales the image by a random factor between 0.9 and 1.1.
    * **Shearing:** Applies a shift in the x-axis while keeping y-axis fixed, with a maximum angle of 10 degrees.
  - **Random Erasing:** Randomly erases a subsection of the image with a probability of 0.5. The region encompasses between 2% to 33% of the image area. This region is replaced with random values.
  - **Normalization:** Pixel values were normalized to a mean of $[0.485, 0.456, 0.406]$ and a standard deviation of $[0.229, 0.224, 0.225]$, based on ImageNet parameters.
- **Validation and Test Preprocessing:** Validation and test does not uses any data augmentation, only formatting for easier computations:
  - **Resizing:** All images were resized to the target size.
  - **Normalization:** mean of $[0.485, 0.456, 0.406]$ and a standard deviation of $[0.229, 0.224, 0.225]$.

B. Methodology

The methodology for this study is structured into four distinct blocks: preprocessing, supervised training, self-supervised training, and performance comparison. Each block is tailored to address the challenging aspects of this particular dataset, using both labeled and unlabeled images.

1) Block 1: Preprocessing:

- **Unlabeled Dataset:** The unlabeled dataset, comprising approximately 97,000 images captured through camera traps at the Tiputini Biodiversity Station (TBS), was processed using *MegaDetector-v5*. Images with a detection confidence of 85% or higher were extracted, resulting in a filtered subset of 45,000 images. These images, featuring a wide variety of species and environmental conditions, were cropped into the suggested bounding box and then resized to 224×224 pixels, for compatibility with ResNet-50. Data augmentation, as described in the Preprocessing Techniques subsection, was applied, to comply with contrastive learning data requirements.
- **Labeled Dataset:** The suggested bounding box was used, then, the dataset was downsampled to ensure balanced classes. The balanced dataset was then divided into:
  - **Training set:** 70% of the images.
  - **Validation set:** 20% of the images.
  - **Testing set:** 10% of the images.

  Special care was taken to avoid data contamination, as many images contain multiple animals, which could appear in different subsets if not handled properly.
- **Contrastive Task Splitting:** For the self-supervised contrastive task, the unlabeled dataset was split into:
  - **Training set:** 80% of the images.
  - **Validation set:** 20% of the images.

  The labeled dataset was retained exclusively for supervised tasks, following the same train-validation-test split as described before.

2) Block 2: Supervised Training:

- **Models:** We use four select architectures based on the state-of-the-art review: i) ResNet-50, ii) VGG-16, iii) EfficientNet-B7, iv) ViT
- **Hyperparameter Optimization:** Hyperparameters were optimized via grid search:
  - **Learning rate:** {0.001, 0.0001}.
  - **Batch size:** {32, 64}.
  - **Frozen layers:** {0%, 15%, 50%, 80%, 95%}, allowing exploration of transfer learning with total and partial fine-tuning.

  Models were trained for up to 200 epochs, with early stopping applied after 10 epochs of no improvement in validation binary cross-entropy loss.

3) Block 3: Self-Supervised Training with SimCLR:

- **Encoder:** ResNet-50.
- **Data Augmentation:** as described in the Preprocessing Techniques subsection.
- **Contrastive Loss Optimization:** SimCLR's contrastive loss was optimized with the following parameters:
  - **Temperature parameter:** 0.1.
  - **Batch size:** 256.
  - **Learning rate:** 1e-4.

  Training was conducted for 500 epochs and model patience of 20 epochs to ensure convergence of the learned representations.
- **Fine-Tuning:** The learned feature representations from SimCLR were fine-tuned on the labeled training dataset. Performance was evaluated using the validation set to keep consistency with supervised models. This step also involves hyperparameter optimization of the classifier with the following values: i) batch size: 16, 32, 64; ii) learningn rate: 1e-5, 1e-4, 1e-3; iii) weight decay: 0.1, 0.01, 0.001, 0.

4) Block 4: Performance Comparison:

- **Initial Evaluation:** All models were first evaluated using a single train-validation split, using the accuracy

as the only metric given the balanced information we get after undersampling.

- **Cross-Validation:** The top two models from the initial evaluation were subjected to repeated k-fold cross-validation:
  - **Folds:** 5 folds.
  - **Repetitions:** 5 repetitions.

  Folds were sampled from the combined training and validation sets of the original labeled dataset and then undersampled.
- **Embedding Visualization:** To compare the learned feature representations, t-distributed Stochastic Neighbor Embedding (t-SNE) was applied. t-SNE provides a visual representation of the embeddings, helping understand class separability and clustering behavior.
- **Final Testing:** The best-performing model from the cross-validation phase was evaluated on the test set, providing a definitive measure of its generalization performance.

In order to better illustrate the pipeline of our work we present a graph summarizing a supervised learning example:

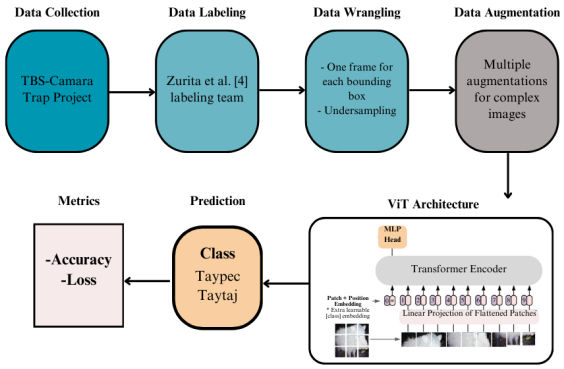

Figure 2: Flow Chart for ViT Training

The full implementation of this work can be found in the GitHub repository.

IV. Results and Discussion

A. Quantitative Results

- **Supervised Models:** Within the supervised models, Vision Transformer (ViT) achieved the highest accuracy, reaching **90.1%**. EfficientNet-B7 and ResNet-50 followed with accuracies of **88.0%** and **86.4%**, respectively. VGG-16 showed the weakest performance with 86.0% (Figure 4). These results correspond to the best accuracy for each architecture after grid-search, to see all results refer to Table I and II. Since ViT is the best result overall we compute its test accuracy getting a robust value **89.38**.
  Training time shows interesting results, ResNet-50 is the fastest with 4.12 minutes, followed by ViT with 9.6 minutes, VGG-16 with 10.5 minutes and EfficientNet with 12.3 minutes. Showing that the best results are achieved without the most expensive computational models, but if training speed is the principal concerning ResNet-50 could be used.
  Also, we can see that ViT and EfficientNet-B7 show similar training accuracy, but ViT have better generalization on unseen images. Additionally, VGG-16 shows high instability in validation accuracy (Figure 6).
  Broadly, these results demonstrate the effectiveness of ViT's attention-based architecture in distinguishing between the highly similar classes, despite the limited size of the labeled dataset. Statistical analysis after repeated k-fold cross-validation confirmed that the performance of ViT against EfficientNet is superior ($p < 0.05$).
- **SimCLR Performance:** The self-supervised SimCLR model underperformed all supervised approaches, achieving an accuracy of **84.0%** under the best hyperparameters of its classifier (Table III). Statistical analysis confirmed that the performance difference against ViT were significant ($p < 0.05$). Although for this particular application SimCLR did not showed stellar performance, its near the top accuracy suggest it could be valuable for future research during the labeling process, potentially reducing the need for expensive labeling efforts.
- **Frozen Layers Analysis:**
  The best validation accuracy for ViT is achieved when all its layers are retrained (0% frozen layers), suggesting that the features learned from ImageNet are not informative for these camera traps images. Observation which is further supported by the reduction in accuracy across all four supervised models as more layers are frozen. In contrast, SimCLR retrains the full Resnet-50 architecture achieving relative poor results. This performance gap suggests that this SimCLR configuration is less effective than supervised models for tasks that require fine-grained classification. Together, these findings emphasize the importance of full retraining for models when ImageNet features are insufficiently transferable, while also demonstrating the limitations of self-supervised approaches like SimCLR with ResNet-50 as encoder in certain domains.

B. Qualitative Results

- **Embedding Visualization (t-SNE):** The t-SNE visualization of learned embeddings (computed on validation dataset) shows differences between ViT and SimCLR. ViT's embeddings form an almost linearly separable space, underscoring its ability to learn features to differentiate between the two species. ViT's embeddings exhibit a relatively smooth dispersion of points, suggesting the absence of well-defined clusters beyond the separation of the target classes. In contrast, SimCLR's embeddings produce a more complex and less separable space, showing limitations when

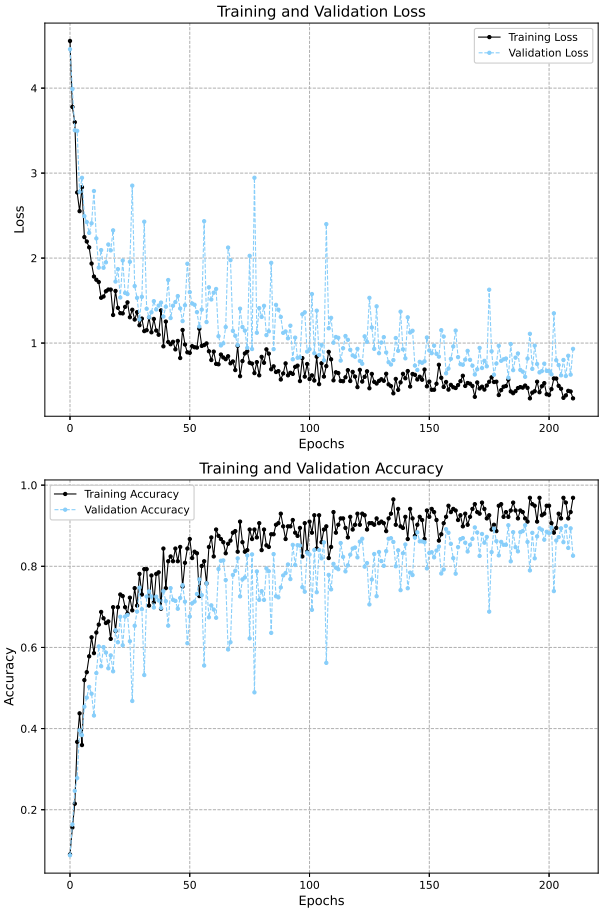

Figure 3: SimCLR Contrastive Training and Validation Results per Epoch.

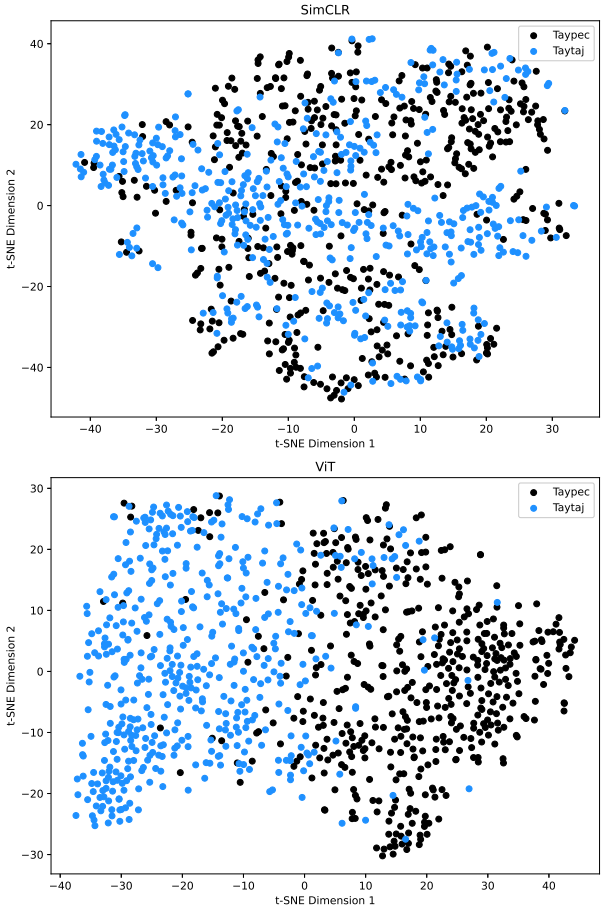

Figure 5: Embedding Visualization (t-SNE). Comparing SimCLR against best Supervised Model (ViT).

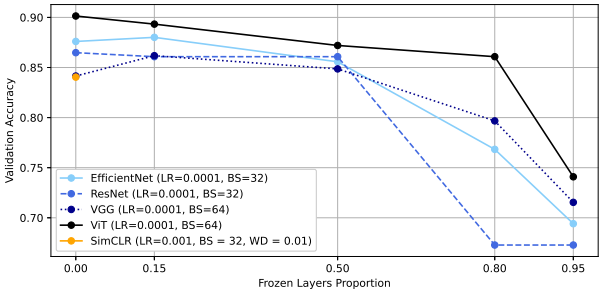

Figure 4: Best Accuracy per architecture and percentage of frozen layer: SimCLR vs. Supervised Learning Models.

learning a representation that captures the species differences.(Figure 5).

- **Error Analysis:** In order to perform an analysis of misclassifications we introduced a k-means clustering to the t-SNE space from ViT's embeddings (Figure 7). Despite the smooth structure of ViT's t-SNE, we identified 15 clusters. Insights were found from five randomly selected images per cluster (Figure 8): i) Clusters 0 and 5 are the more distant to the apparent linear separation of classes, those images are clear with no occlusion or blurring; ii) Cluster 2 contains most of the non separable samples, which correspond to very low quality frames; iii) Cluster 6 and 12 are also challenging regions, its images shows blurring, occlusion and poor lightning. Despite the poor quality of the images the classifier only shows 9.9% of error, highlighting again the robustness of ViT in the presence of suboptimal input conditions.

C. Discussion

- **Insights from Results:** The results clearly shows the advantages of ViT ability to detected small details and separate images based on them within a complex ecological dataset, demonstrating strong performance against CNN despite its reliance on a small labeled dataset. The results from SimCLR, although relative worse, could still be considered useful when labeled data is typically scarce and expensive to obtain.
- **Limitations:** The applicability of our results to other biodiversity datasets may face challenges due to domain-specific factors such as differing species characteristics and environmental conditions. Further experimentation is needed to address these challenges.

V. Conclusions

A. Summary of Findings

The study demonstrated the efficacy of transformer-based learning for challenging wildlife image classification. ViT outperformed by 2 percentage points the second-best model, with a validation accuracy of **90.1%** and a robust test accuracy of **89.38**. This highlights the importance of fine-grained feature detection in challenging environments. Furthermore, the comparatively strong performance of self-supervised learning (accuracy of 84%) suggests potential to reduce the need for extensive labeled datasets.

B. Broader Implications

Our findings emphasize the importance of fine-grained feature detection and the potential to leverage unlabeled data in ecological AI applications. Transformer-based image classifiers offer promising results in complex real-life datasets. Self-supervised learning approaches, such as SimCLR, might reduce the requirement of large labeled datasets, lowering the price of biodiversity monitoring. These techniques facilitate the development of more robust classification models, even in scenarios of poor-quality images and limited labeled data.

C. Future Work

Potential directions for future research can be explored to advance this:

Future research directions in this domain can focus on the following aspects:

- **Self-Supervised Algorithm Exploration:** Investigate and implement alternative self-supervised learning algorithms beyond SimCLR, given the rapid advancements in Deep Learning since its introduction in 2020. At the time of writing, the best self-supervised model for ImageNet is DINOv2 (with a transformer based architecture) with an accuracy of 82.7% against 76.5% of the original SimCLR [43].
- **Enhaced SimCLR Hyperparameter Optimization:** Perform further optimization of SimCLR hyperparameters and use additional encoder architectures like ViT instead of ResNet-50. This approach leads to an improvement of 3.5 percentage points in accuracy in ImageNet against standard SimCLR according to [44].
- **Active Learning for Labeling:** Incorporate active learning strategies to improve labeling efficiency if possible during the labeling process with biology experts.

Acknowledgment

The authors extend their gratitude to the TBS-Camara Trap Project at the Tiputini Biodiversity Station for providing the labeled and unlabeled datasets used in this thesis. We also sincerely thank *Universidad San Francisco de Quito* for supplying the computational resources.

References

[1] T. Jeff, "Humans are driving one million species to extinction," *Nature*, vol. 569, p. 171, 2019.

[2] F. Urbano, R. Viterbi, L. Pedrotti, E. Vettorazzo, C. Movalli, and L. Corlatti, "Enhancing biodiversity conservation and monitoring in protected areas through efficient data management," *Environmental Monitoring and Assessment*, vol. 196, no. 1, p. 12, 2024.

[3] F. Trolliet, C. Vermeulen, M.-C. Huynen, and A. Hambuckers, "Use of camera traps for wildlife studies: a review," *Biotechnologie, Agronomie, Société et Environnement*, vol. 18, no. 3, 2014.

[4] M.-J. Zurita, D. Riofrío, N. Pérez, D. Romo, D. S. Benítez, R. F. Moyano, F. Grijalva, and M. Baldeon-Calisto, "Towards automatic animal classification in wildlife environments for native species monitoring in the amazon," in *2023 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI)*. IEEE, 2023, pp. 1–6.

[5] A. Abuduweili, X. Wu, and X. Tao, "Efficient method for categorize animals in the wild," *arXiv preprint arXiv:1907.13037*, 2019.

[6] B. H. Curtin and S. J. Matthews, "Deep learning for inexpensive image classification of wildlife on the raspberry pi," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 0082–0087.

[7] L. Bothmann, L. Wimmer, O. Charrakh, T. Weber, H. Edelhoff, W. Peters, H. Nguyen, C. Benjamin, and A. Menzel, "Automated wildlife image classification: An active learning tool for ecological applications," *Ecological Informatics*, vol. 77, p. 102231, 2023.

[8] G. S. Ferrante, F. M. Rodrigues, F. R. Andrade, R. Goularte, and R. I. Meneguette, "Understanding the state of the art in animal detection and classification using computer vision technologies," in *2021 IEEE International Conference on Big Data (Big Data)*. Ieee, 2021, pp. 3056–3065.

[9] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," in *2014 IEEE international conference on image processing (ICIP)*. IEEE, 2014, pp. 858–862.

[10] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, 2018.

[11] Z. Miao, K. M. Gaynor, J. Wang, Z. Liu, O. Muellerklein, M. S. Norouzzadeh, A. McInturff, R. C. Bowie, R. Nathan, S. X. Yu *et al.*, "Insights and approaches using deep learning to classify wildlife," *Scientific reports*, vol. 9, no. 1, p. 8137, 2019.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[15] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] The Swiftest, "Biodiversity index," https://theswiftest.com/biodiversity-index/, accessed April 13, 2024.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[19] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham, and L. A. Gabralla, "A review on machine learning styles in computer vision—techniques and future directions," *IEEE Access*, vol. 10, pp. 107 293–107 329, 2022.

[20] M. S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic, and J. Clune, "A deep active learning system for species identification and counting in camera trap images," *Methods in ecology and evolution*, vol. 12, no. 1, pp. 150–161, 2021.

[21] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, "A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations," *Expert Systems with Applications*, vol. 242, p. 122807, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423033092

[22] N. Sheikh, "Identification and classification of wildlife from camera-trap images using machine learning and computer vision," Ph.D. dissertation, Dublin, National College of Ireland, 2020.

[23] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[24] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, pp. 1–40, 2016.

[25] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?" in *International conference on machine learning*. PMLR, 2019, pp. 5389–5400.

[26] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[27] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *International conference on machine learning*. PMLR, 2020, pp. 3897–3906.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[29] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[30] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical image analysis*, vol. 71, p. 102062, 2021.

[31] X. Zheng, "Rare wildlife recognition with self-supervised representation learning," *arXiv preprint arXiv:2211.05636*, 2022.

[32] S. S. Kathait, A. Kumar, P. Dhuliya, and I. Chauhan, "Deep learning-based model for wildlife species classification," Valiance Analytics Pvt. Ltd. and Doon University, Noida, Uttar Pradesh, Tech. Rep., 2024, accessed: 2024-07-21. [Online]. Available: https://valiancesolutions.com/wp-content/uploads/2024/02/Deep-Learning-based-Model-for-Wildlife-Species-Classification.pdf

[33] Y. Kesani and D. Mahato, "Advancements in marine animal classification through computer vision and machine learning," *ResearchGate*, 2023, accessed: 2024-11-24. [Online]. Available: https://www.researchgate.net/profile/Deepshikha-Mahato/publication/376683947_Advancements_in_Marine_Animal_Classification_through_Computer_Vision_and_Machine_Learning/links/65839da16f6e450f198d14a4/Advancements-in-Marine-Animal-Classification-through-Computer-Vision-and-Machine-Learning.pdf

[34] R. Shinoda and K. Shiohara, "Openanimaltracks: A dataset for animal track recognition," *arXiv preprint arXiv:2406.09647*, 2024.

[35] K. S. Kalyan, "A survey of gpt-3 family large language models including chatgpt and gpt-4," *Natural Language Processing Journal*, vol. 6, p. 100048, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949719123000456

[36] A. Tendle, ""the revolution will not be supervised": An investigation of the efficacy and reasoning process of self-supervised representations," *ResearchGate*, 2021, accessed: 2024-11-24. [Online]. Available: https://www.researchgate.net/publication/351625872_The_Revolution_Will_Not_Be_Supervised_An_Investigation_of_the_Efficacy_and_Reasoning_Process_of_Self-Supervised_Representations

[37] O. Pantazis, G. J. Brostow, K. E. Jones, and O. Mac Aodha, "Focus on the positives: Self-supervised learning for biodiversity monitoring," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 10 583–10 592.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[40] O. Cajamarca and F. Grijalva, "Clustering wildlife species in the amazon: Using vision transformers to analyze unlabeled images from the tiputini biodiversity station," 2024, unpublished Master's thesis, Universidad San Francisco de Quito - Master's Degree in Data Science.

[41] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in neural information processing systems*, vol. 33, pp. 22 243–22 255, 2020.

[42] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, pp. 1–48, 2019. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[43] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[44] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9640–9649.

Appendix A
Additional Results

Table I: Grid Search Results for Supervised Architectures (Part 1)

Model	Learning Rate	Batch Size	Frozen Layers Proportion	Validation Loss	Validation Accuracy	Training Time in Minutes
EfficientNet	0.0001	32	0.00	0.2871	0.8760	9.77
EfficientNet	0.0001	32	0.15	0.2888	0.8801	12.33
EfficientNet	0.0001	32	0.50	0.3147	0.8557	6.50
EfficientNet	0.0001	32	0.80	0.4773	0.7683	25.26
EfficientNet	0.0001	32	0.95	0.6031	0.6941	21.56
EfficientNet	0.0001	64	0.00	0.3290	0.8689	9.41
EfficientNet	0.0001	64	0.15	0.3251	0.8659	9.30
EfficientNet	0.0001	64	0.50	0.3307	0.8608	6.67
EfficientNet	0.0001	64	0.80	0.4944	0.7520	17.89
EfficientNet	0.0001	64	0.95	0.5896	0.7063	42.30
EfficientNet	0.0010	32	0.00	0.3753	0.8333	9.75
EfficientNet	0.0010	32	0.15	0.3305	0.8486	12.67
EfficientNet	0.0010	32	0.50	0.3608	0.8384	6.43
EfficientNet	0.0010	32	0.80	0.4969	0.7490	4.85
EfficientNet	0.0010	32	0.95	0.5862	0.6951	4.92
EfficientNet	0.0010	64	0.00	0.3560	0.8496	11.55
EfficientNet	0.0010	64	0.15	0.3492	0.8557	11.30
EfficientNet	0.0010	64	0.50	0.3695	0.8516	10.02
EfficientNet	0.0010	64	0.80	0.4891	0.7663	4.88
EfficientNet	0.0010	64	0.95	0.5759	0.7002	9.05
ResNet	0.0001	32	0.00	0.3181	0.8648	4.12
ResNet	0.0001	32	0.15	0.3422	0.8608	4.27
ResNet	0.0001	32	0.50	0.3594	0.8608	3.71
ResNet	0.0001	32	0.80	0.6090	0.6728	5.76
ResNet	0.0001	32	0.95	0.6090	0.6728	5.80
ResNet	0.0001	64	0.00	0.3542	0.8547	4.74
ResNet	0.0001	64	0.15	0.3709	0.8404	4.25
ResNet	0.0001	64	0.50	0.3558	0.8496	3.07
ResNet	0.0001	64	0.80	0.5956	0.6707	12.84
ResNet	0.0001	64	0.95	0.5956	0.6707	13.83
ResNet	0.0010	32	0.00	0.4398	0.7917	5.51
ResNet	0.0010	32	0.15	0.4125	0.8079	4.81
ResNet	0.0010	32	0.50	0.3743	0.8272	6.96
ResNet	0.0010	32	0.80	0.5570	0.7215	5.27
ResNet	0.0010	32	0.95	0.5570	0.7215	4.97
ResNet	0.0010	64	0.00	0.4150	0.8232	5.83
ResNet	0.0010	64	0.15	0.4117	0.8201	6.29
ResNet	0.0010	64	0.50	0.4080	0.8283	3.34
ResNet	0.0010	64	0.80	0.5497	0.7307	6.46
ResNet	0.0010	64	0.95	0.5497	0.7307	6.22

Table II: Grid Search Results for Supervised Architectures (Part 2)

Model	Learning Rate	Batch Size	Frozen Layers Proportion	Validation Loss	Validation Accuracy	Training Time in Minutes
VGG	0.0001	32	0.00	0.3826	0.8323	6.74
VGG	0.0001	32	0.15	0.3590	0.8476	6.69
VGG	0.0001	32	0.50	0.3702	0.8293	3.23
VGG	0.0001	32	0.80	0.4297	0.8028	8.82
VGG	0.0001	32	0.95	0.5539	0.7104	6.68
VGG	0.0001	64	0.00	0.3590	0.8415	9.38
VGG	0.0001	64	0.15	0.3305	0.8618	10.51
VGG	0.0001	64	0.50	0.3478	0.8486	4.75
VGG	0.0001	64	0.80	0.4386	0.7967	3.80
VGG	0.0001	64	0.95	0.5574	0.7154	7.53
VGG	0.0010	32	0.00	0.6913	0.5102	2.27
VGG	0.0010	32	0.15	0.6917	0.4837	2.72
VGG	0.0010	32	0.50	0.6932	0.5000	2.03
VGG	0.0010	32	0.80	0.4904	0.7673	2.73
VGG	0.0010	32	0.95	0.5276	0.7470	3.64
VGG	0.0010	64	0.00	0.6931	0.5000	4.60
VGG	0.0010	64	0.15	0.6932	0.5000	2.81
VGG	0.0010	64	0.50	0.6932	0.5000	2.76
VGG	0.0010	64	0.80	0.4536	0.7846	5.51
VGG	0.0010	64	0.95	0.5065	0.7612	5.85
ViT	0.0001	32	0.00	0.2498	0.8933	10.69
ViT	0.0001	32	0.15	0.2508	0.8984	9.01
ViT	0.0001	32	0.50	0.2860	0.8780	7.89
ViT	0.0001	32	0.80	0.3460	0.8384	6.09
ViT	0.0001	32	0.95	0.5243	0.7480	5.47
ViT	0.0001	64	0.00	0.2551	0.9014	9.63
ViT	0.0001	64	0.15	0.2662	0.8933	9.59
ViT	0.0001	64	0.50	0.2801	0.8720	8.38
ViT	0.0001	64	0.80	0.3353	0.8608	7.82
ViT	0.0001	64	0.95	0.5048	0.7409	10.58
ViT	0.0010	32	0.00	0.5560	0.7124	39.75
ViT	0.0010	32	0.15	0.3770	0.8404	19.57
ViT	0.0010	32	0.50	0.3292	0.8506	7.35
ViT	0.0010	32	0.80	0.3199	0.8699	4.62
ViT	0.0010	32	0.95	0.4690	0.7724	4.14
ViT	0.0010	64	0.00	0.6763	0.5772	9.54
ViT	0.0010	64	0.15	0.3133	0.8608	31.11
ViT	0.0010	64	0.50	0.3042	0.8689	8.35
ViT	0.0010	64	0.80	0.3132	0.8659	5.83
ViT	0.0010	64	0.95	0.4738	0.7774	3.65

Figure 6: SimCLR Contrastive Training and Validation Results per Epoch.

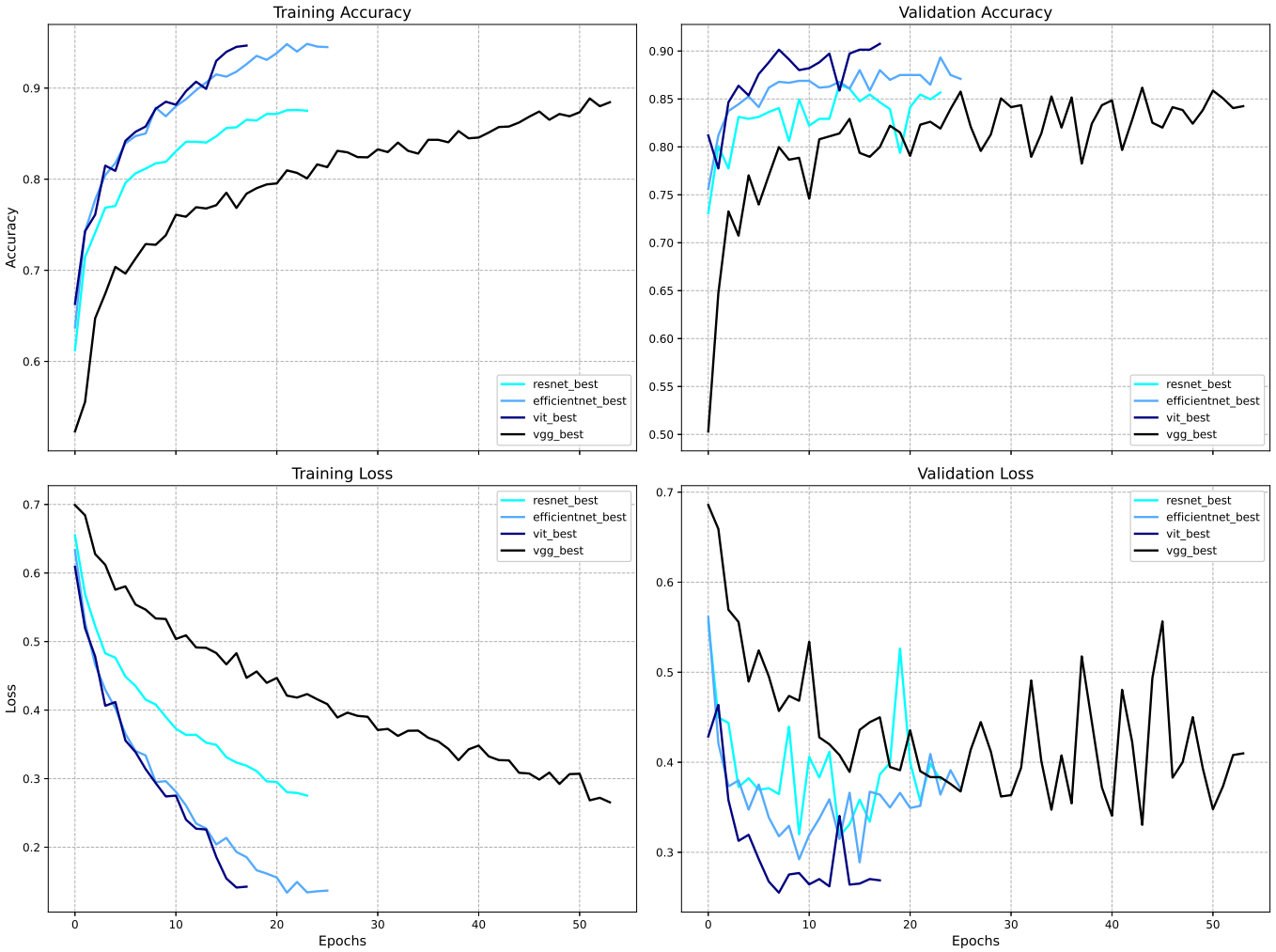

Table III: Grid Search Results for SimCLR Logistic Classifier

Batch Size	**Learning Rate**	**Weight Decay**	**Validation Accuracy**
16	0.00001	0.000	0.8130
16	0.00001	0.001	0.8130
16	0.00001	0.010	0.8130
16	0.00001	0.100	0.8130
16	0.00010	0.000	0.8343
16	0.00010	0.001	0.8354
16	0.00010	0.010	0.8333
16	0.00010	0.100	0.8343
16	0.00100	0.000	0.8323
16	0.00100	0.001	0.8354
16	0.00100	0.010	0.8354
16	0.00100	0.100	0.8333
32	0.00001	0.000	0.7988
32	0.00001	0.001	0.7988
32	0.00001	0.010	0.7988
32	0.00001	0.100	0.7978
32	0.00010	0.000	0.8333
32	0.00010	0.001	0.8293
32	0.00010	0.010	0.8303
32	0.00010	0.100	0.8333
32	0.00100	0.000	0.8364
32	0.00100	0.001	0.8364
32	0.00100	0.010	0.8404
32	0.00100	0.100	0.8384
64	0.00001	0.000	0.7815
64	0.00001	0.001	0.7805
64	0.00001	0.010	0.7815
64	0.00001	0.100	0.7815
64	0.00010	0.000	0.8262
64	0.00010	0.001	0.8272
64	0.00010	0.010	0.8252
64	0.00010	0.100	0.8262
64	0.00100	0.000	0.8374
64	0.00100	0.001	0.8364
64	0.00100	0.010	0.8394
64	0.00100	0.100	0.8404

Figure 7: Elbow Method Graph and t-SNE Colored by Cluster.

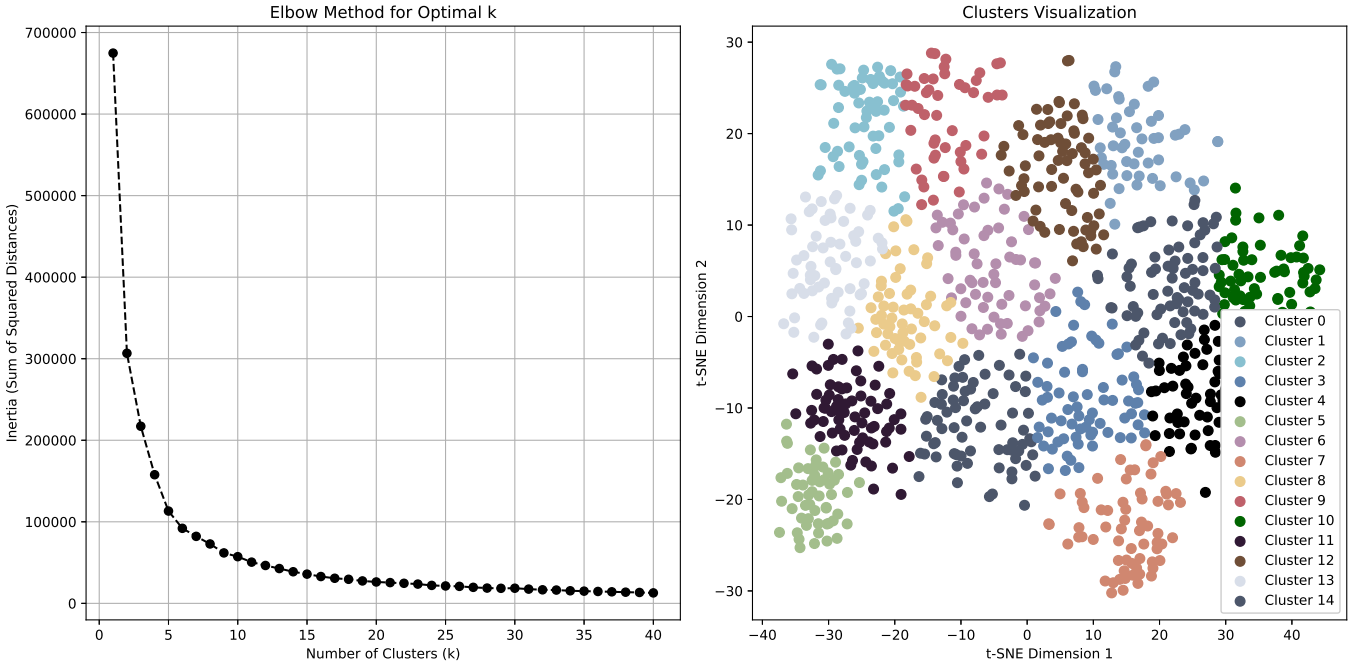

Figure 8: Sample Images from Selected Clusters.