Multi-Species Individual Animal Reidentification

Ava Gordon, A.J. Varchetti, Briana Basile
gordon.1269@osu.edu, varchetti.3@osu.edu, basile.48@osu.edu
Department of Computer Science and Engineering
CSE 5524 (Computer Vision): Final Project Report
The Ohio State University
April 2025

1 Abstract

The CVPR AnimalCLEF25 competition focuses on multi-species individual animal re-identification through recognition of unique features of certain species [1]. Specifically, the aim is to outperform the baseline competition site test accuracy of ˜30% which uses the MegaDiscriptor-L-384 [2]. Using Res-Net 50 encoders, a Batch Hard Triplet Loss model, and embedding with Python libraries and the Ohio SuperComputer, we achieve a test accuracy of $\sim 33\%$. With more time, less class imbalanced data, and more threshold tuning, we would hope to improve accuracy further.

2 Introduction

For each ˜2000 test images, the goal is to either identify if a) the animal is not present in the training dataset (i.e. it is "new") and predictably identify it as a specific loggerhead sea turtle species, a specific salamander species, or a specific Eurasian lynxe species from features learned in training or b) correctly identify that the image is present in the training dataset (i.e. it is not "new") and which *exact* image/species it matches in the training dataset [1].

3 Motivation

While we perform this modeling and analysis to strengthen our computer vision skills, the completion of this project has applications that are useful for studying wildlife behaviors. For example, animal re-identification can be helpful for tracking populations, migration routes, and habitat uses of species [1]. Furthermore, it can be helpful for identifying biodiversity threats and crafting evidence-based conservation strategies [1].

4 Approach

The training dataset contains 13,000 images representing 1,000 unique animal species and the test dataset contains 2,000 images featuring a variety of different animals. Our model outputs embeddings where if the embeddings between the query and the nearest match are similar, then the model will predict the nearest match. On the other hand, if they are far, then the model will predict new animals.

4.1 Baseline Approach

The baseline model has roughly 280 million parameters and is pre-trained on animal reidentification datasets. The MegaDiscriptor model provides a query image and then checks the closest cosine similarity between all images in the training dataset [2].

4.2 Advanced Approach

Our advanced approach aims to outperform the baseline approach test accuracy of around 30% [2]. We trained four separate ResNet-50 models where one of these models is used for species level identification. This specific model identifies whether a given animal is a turtle, lynxe, or salamander. The remaining ResNet-50 encoders will each be trained on one of the species with Triplet Loss where the goal is to identify individual animals by their unique IDs. During training, Batch Hard Triplet Loss was used where each batch contained 16 different animals with four images per animal. Batch Hard Loss performs where each anchor has a triplet pair adaptively constructed to be the one with the most error. This allows the model to focus on struggling comparisons and the batches are generated at random. All chosen images were augmented from a random choice of a horizontal flip, rotation, color jitter, crop, and affine. If there were not four images available for the corresponding chosen individual, then data augmentation listed was applied to generate new images for that individual to obtain four images. These models were trained until each reached a validation accuracy of 99%.

To utilize the model after training, all of the training data was first encoded and stored in respective species specific datasets. Thus, when an unknown query image is sent to the model, it first runs through the classifier to determine the species of the animal. Then, based on the determined species, the image is run through the corresponding encoder. The returned feature vector is then compared to the species dataset, and the euclidean distance nearest neighbor is returned. If the euclidean distance between the nearest neighbor and the unknown embedded query vector is less than a threshold, the unknown image is classified with the same individual ID of the nearest neighbor. If it was larger than the threshold, the model predicts the unknown image as a new animal. A visual representation of the model can be seen below:

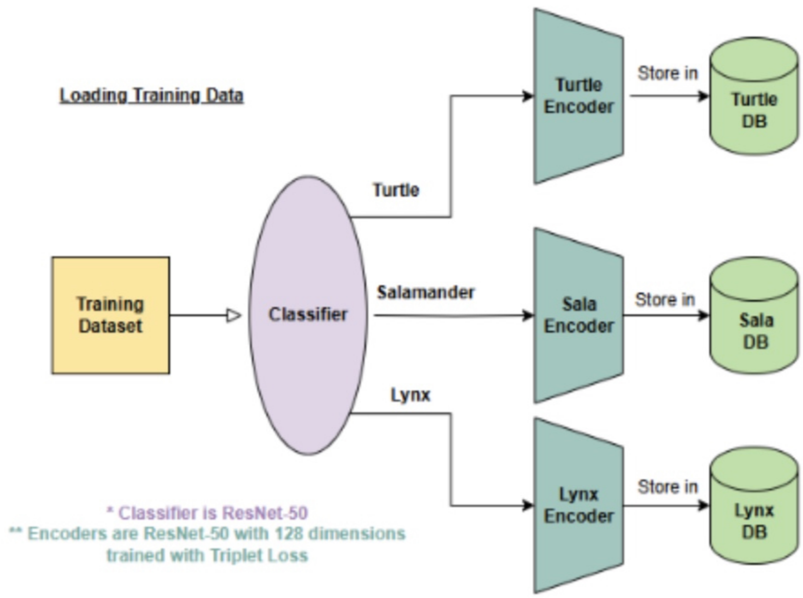

Figure 1: Process of storing training data embeddings.

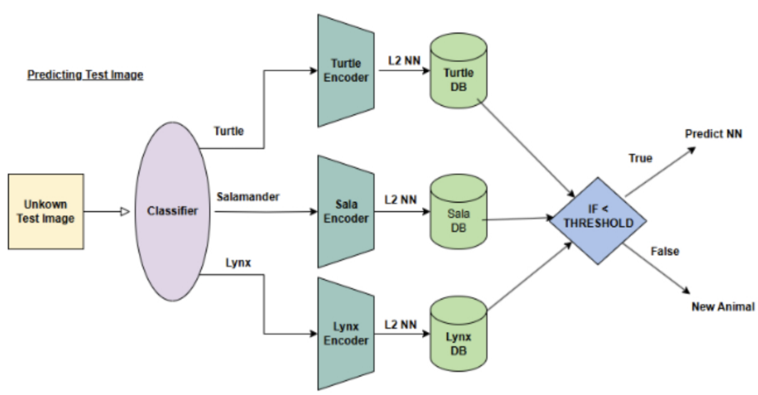

Figure 2: Process of classifying unknown test image

4.3 Libraries and Computational Resources

- PyTorch: for development of models, training, data augmentation, and other utility functions
- Pandas: for reading in the dataset metadata files
- Numpy: for matrix operations
- Pillow (PIL): for opening images
- Sklearn: for encoding the image ID strings into numerical values
- TensorFlow: the Batch Hard Triplet Selector was adapted from TensorFlow to PyTorch

The implementation structure and flow was modeled off of the paper *In Defense of the Triplet Loss for Person Re-Identification* and their corresponding GitHub implementation [3,4].

For implementation, the Ohio SuperComputer (OSC) was used. The models were trained on the Ascend Cluster with 1 node and 1 GPU. Testing was performed on our individual machines, which took around 20 minutes to create a test submission for the entire dataset to be tested directly on Kaggle's website.

5 Experiments, Evaluation, and Validation

The evaluation method utilized is accuracy which in this case, is the geometric mean between the correctly predicted known animals accuracy and the correctly predicted unknown animals accuracy. The geometric mean is chosen as it prevents a trivial solution of simply predicting a new animal for all test images as that model would not be useful, but would still score well on a normal average accuracy evaluation metric.

5.1 Data, Metrics and Results

Our Triplet Loss model achieved an accuracy of $\sim 33\%$, representing a 3% improvement over the baseline model. The employment of this Triplet Loss approach allowed for better distinguish between species with subtle visual differences.

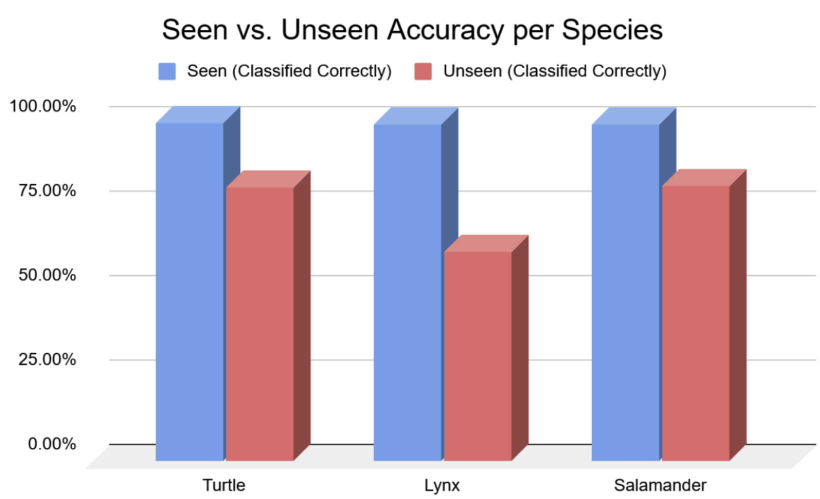

Figure 3: Accuracy broken down by species

6 Conclusions and Discussion

6.1 Insights

Our increase in accuracy relative to the baseline model highlights the potential of Triplet Loss in enhancing performance of image classification tasks in complex, biodiversity focused datasets.

6.2 Workload

- Data Collection and Pre-processing (research and acquiring data): $\sim 20\%$ (Gordon, Basile, Varchetti)
- Exploratory Data Analysis (analysis of class distributions and label imbalances): $\sim 5\%$ (Varchetti)
- Model Development and Design (choose and implement model architecture): $\sim 25\%$ (Varchetti)
- Training and Evaluation (train models on a subset of the data-set): $\sim 15\%$ (Gordon)
- Error Analysis (analyze misclassifications and adjust models accordingly): $\sim 5\%$ (Gordon, Basile, Varchetti)
- Result Visualizations: $\sim 5\%$ (Basile)
- Presentation, Documentation, Report, and Code Release: $\sim 30\%$ (Gordon, Basile, Varchetti)

6.3 Key Challenges

Class imbalance proved to be a major challenge in this project. Although the dataset included 13,000 images spanning 1,000 distinct animal species, the distribution of images across the three classes was uneven. This was likely a reason the model struggled to generalize to less frequent species.

Threshold tuning was also particularly challenging given the subtle visual differences between and within species, where even minor changes in threshold values affected classification outcomes. If the value is too low, the model is more likely to misclassify "new" classes as "known", and if the value is too high, the model is more likely to misclassify "known" classes as "new".

While our advanced model outperformed the baseline model, with more time, access to better-balanced training data, and additional augmentation techniques, we might anticipate a larger increase in test accuracy than just 3% and may see more substantial gains in model performance.

7 References

[1]. Picek, L. and Adam, L. and Papafitsoros, K. (2025). *AnimalClef25 @ CVPR-FGVC & LifeCLEF*. https://www.kaggle.com/competitions/animal-clef-2025/.

[2]. Picek, L. (2025). *AnimalCLEF2025: Starter notebook.* https://www.kaggle.com/code/picekl/animalclef2025-starter-notebook.

[3]. Hermans, A. and Beyer, L. and Leibe, B. (2017). *In Defense of the Triplet Loss for Person Re-Identification*. arxiv.org/abs/1703.07737.

[4]. @CoinCheung. (2018). *Github*. [Python] https://github.com/CoinCheung/triplet-reid-pytorch.