Preserving the Apex: Enhancing Tiger Conservation to Safeguard Ecosystems with "PyTorch-Wildlife"

Primary source (paperswithcode.com): **Pytorch-Wildlife**: A Collaborative Deep Learning Framework for Conservation

1. Abstract:

This project aims to develop a solution for detecting, classifying, and re-identifying tigers in wildlife imagery to support conservation efforts and monitor tiger populations. By leveraging the PyTorch-Wildlife library [1], we first detect animals within images, and then extend its functionality with a Convolutional Neural Network (CNN)-based classifier to verify whether the detected animal is a tiger. To achieve individual tiger identification (re-id), we implement a Siamese Network architecture designed to distinguish between similar and distinct tiger individuals by learning unique visual patterns, such as stripe arrangements, for accurate re-identification. This allows the system to match detected tigers with known individuals in a reference database or flag new, previously unobserved tigers. By using state-of-the-art deep learning techniques, such as CNNs and Siamese Networks, the project addresses the challenge of automatically identifying and tracking tigers, thereby providing a valuable tool for wildlife researchers and conservationists.

2. Introduction:

The global population of wild tigers has dramatically declined over the past century due to habitat loss, poaching, and human-wildlife conflict. Consequently, tigers are now classified as an endangered species, making their conservation a critical priority for wildlife organizations and governments. This issue is particularly significant in India, which is home to over 70% of the world's wild tiger population. As the largest habitat for tigers, India plays a pivotal role in their global conservation, with initiatives like Project Tiger and the National Tiger Conservation Authority (NTCA) established to protect and manage these populations.

Effective monitoring and management of the tiger population are crucial for maintaining biodiversity and ecological balance, as tigers are a keystone species that influence the health of entire ecosystems. However, traditional monitoring methods – such as pugmark analysis, camera traps, and direct sightings – are labor-intensive, prone to human error, and often yield incomplete or biased data, especially in India's dense forests and diverse terrains. Accurately identifying individual tigers is essential for population estimation, movement tracking, and studying social interactions, which are critical for developing effective conservation strategies. This project addresses these challenges by proposing an end-to-end solution that integrates object detection, classification, and re-identification of tigers using state-of-the-art deep learning architectures.

3. Literature Review:

The **PyTorch-Wildlife framework** [1] forms the foundation for our project by offering models for animal detection and a classification fine-tuning module. For detection, we use YOLO MegaDetector V5 (MDv5), while ResNet-50, fine-tuned on ImageNet weights, classifies detected animals as tigers. In extending this framework, we employ a Siamese network for individual tiger identification, aiding in population monitoring and census efforts for tigers. Amur Tiger Re-identification in the Wild (**ATRW**) dataset [5] provides annotated tiger images (with bounding box, pose keypoint, and identity annotations), which inform our data processing pipeline. Other studies explore various methodologies for re-identification and these collectively inform our project's methodology in recognizing individual tigers and contributing to conservation efforts:

A Hybrid Approach for Tiger Re-Identification: This study emphasizes the importance of visual data analytics in wildlife monitoring and conservation strategies. It proposes a hybrid approach that combines deep learning with traditional SIFT descriptor-based matching [6]. Additionally, several data transformations are applied to enhance the model's generalization capability across different views and image quality variations.

Individual Tiger Identification using Transfer Learning: This paper presents methods for classifying images of tigers into their respective individual classes using deep learning models. The proposed method employs a pipeline that integrates the YOLOv8 model for object detection and the EfficientNetB3 model with transfer learning for image classification, focusing on 98 unique tigers [7]. To achieve reliable accuracy, the model necessitates a minimum of 15 images per tiger, which may not always be practical in real-world situations. Additionally, the application of the SIFT descriptor-based algorithm for feature extraction and image matching against the ground truth exposed some inconsistencies.

Siamese Network-Based Pelage Pattern Matching for Ringed Seal Re-identification: In this paper, a method for re-identifying Saimaa ringed seals by matching pelage patterns was proposed [3]. The pelage pattern was extracted using a Sato tubeness filter, and similarities of the patches were computed with a Siamese network trained on a large dataset using a triplet loss function. These similarities were used to identify the best matches from a database of known individuals.

Animal Identification using Deep Transfer Learning for Data Fusion in Siamese Networks: This paper addresses automated animal identification across various species, exploring the application of established human visual biometrics techniques [4]. Specifically, it investigates the use of region proposal networks and deep transfer learning in Siamese networks for individual animal identification.

Similarity Learning Networks for Animal Individual Re-Identification - Beyond the Capabilities of a Human Observer: This research highlights the application of deep learning in one-shot learning tasks [2], particularly in animal re-identification, which can automate species population estimates from camera trap images. It demonstrates the use of similarity comparison networks and the generalization capabilities of deep convolutional neural networks across different domains [8].

Animal Recognition Using Siamese Network with Two Kinds of Backbone Networks: This study focuses on wildlife recognition, specifically the task of matching and outputting similarity probabilities between two images using twin networks, with the Siamese architecture as the primary framework (built using backbone CNNs – VGGNet and ResNet) [9].

4. Materials and Method:

In this project, we consolidated .jpg images from multiple sources to create a comprehensive dataset for training a ResNet-50 CNN to classify tigers and for training a Siamese network to identify individual tigers. To classify tigers as a species after the animal detection stage with YOLO MDv5, we selected .jpg images of animals with similar traits – specifically from the Felidae family and the Panthera genus within that family. We hypothesize that this binary classification approach encourages the neural network to learn subtle features that are beneficial in subsequent stages as we intend to modify and use the same classifier for re-identification of tigers. For identifying individual tigers, we utilized the 'Amur Tiger Re-identification' dataset, which contains annotated images that distinguish between different tiger individuals, as labeled by human observers. This dataset provided valuable ground truth data for training the model to differentiate between known tiger individuals and identify previously unobserved ones.

Dataset Name	Animal Category	No. of Images taken	Source
Amur Tiger Re-identification	Tiger	- 4413 (for tiger classification) - 1887 (for re-identification of 107 tiger individuals)	https://lila.science/datasets/atrw
Felidae | Cat family	Puma	32	https://www.kaggle.com/datasets/juliencalenge/felidae-tiger-lion-cheetah-leopard-puma/data
	Leopard	50	
	Cheetah	56	
	Lion	49	
Animal Image Dataset	Lion	717	https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals
Big Cats Image Classification Dataset	Cheetah	186	https://www.kaggle.com/datasets/patriciabrezeanu/big-cats-image-classification-dataset?select=animals
	Lion	182	
	Leopard	180	
Images of Deer	Deer	853	https://www.kaggle.com/datasets/faisalakhtar/images-of-deer-for-svm-classifier

Table 1: Summary of Datasets Used for Tiger Classification and Re-Identification

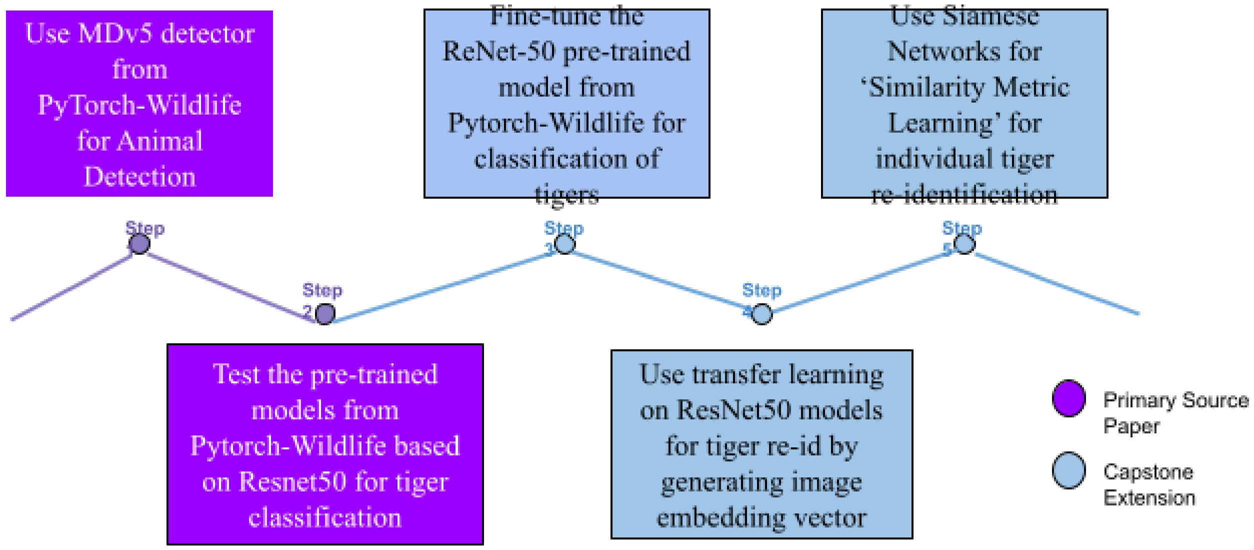

Figure 1: Extension workflow on "Pytorch-Wildlife" paper

Base Paper:

- The MDv5 (or alternatively MDv6-c) model from PyTorch-Wildlife, pre-trained on 3 million animal images from various ecosystems, will be used for animal detection to generate bounding boxes, ensuring effective object detection.
- The ResNet50 fine-tuning module from PyTorch-Wildlife will be used to specifically fine-tune the model for classifying tigers as a species.

Extension:

- We will fine-tune the ResNet50 classifier specifically to identify tigers, serving as a crucial step before implementing a Siamese network, as it will act as our feature extractor to provide embeddings.
- We will implement a Siamese architecture using backbone CNNs from the PyTorch-Wildlife framework (previous step) to learn a similarity function for recognizing unique individuals. Unlike models that classify individual tigers directly with predefined architecture, the Siamese network excels in distinguishing between pairs of inputs by focusing on their similarities. It is particularly effective in few-shot learning scenarios [2], enabling accurate predictions with minimal examples. This makes it especially useful for tiger re-identification, where limited images per identity are available.

Pipeline of tasks:

I. Data Preparation for fine-tuning the Binary Classifier: For fine-tuning the binary classifier, we used the Amur Tiger dataset from LILA, along with images of related species (e.g., pumas, cheetahs, leopards, jaguars) and other animals (e.g., lynxes, deer) sourced from Kaggle. The dataset was manually curated to remove irrelevant images and then converted to the required .JPG format. To increase the dataset size and improve model robustness, we applied data augmentation techniques such as horizontal and vertical flips, which expanded the total number of images from over 6,000 to approximately 9,800. The augmented dataset was then divided into three subsets: 60% for training, 20% for validation, and 20% for testing.

All images were resized to a standard dimension of 1280x1280 at the beginning of the pipeline, particularly for the detection phase. Annotations were created to label the images, with "Tiger" as 1 and "Non-Tiger" as 0. Including images of animals that have similar features to tigers ensures the model learns finer feature distinctions, improving its classification capabilities and setting a solid foundation for the subsequent stages of the project (re-id of individual tigers). This approach enhances the classifier's ability to distinguish tigers from other visually similar species, contributing to more accurate and reliable results in real-world scenarios.

II. CNN-based Binary Classifier Training: For the training of our CNN-based binary classifier, we utilize MegaDetectorV5 (MDv5), an extension of YOLOv5 designed specifically for animal detection in images. MDv5 identifies bounding boxes around various objects, including animals and non-animal entities. The MDv5 detector is pretrained on a vast dataset that enables it to recognize common entities found in camera trap images, such as humans, vehicles, and animals. It specifically detects animals based on the training data provided and discards any identified humans or vehicles. The cropped images of animals are then fed into the training process for the ResNet50 classifier, along with the corresponding labeled annotations.

Figure 2: Tiger (Top Left: Original image, Top Right: Bbox Cropped image). Non-Tiger (Bottom Left:Original image, Bottom Right: Bbox Cropped image). MDv5 crops the detected animal according to the dimensions of the predicted bounding box, removing much of the background noise and ensuring the model focuses on the object of interest.

The training process begins with loading images in batches of 32 using an annotation file. Each image is resized to a square format of 1280x1280 before being processed by the model, which operates with a stride of 64. The output of this step consists of cropped and resized images, which are stored in a folder labeled "Cropped_resized". This folder also contains an updated annotation file, which is subsequently used to train the Tiger/Non-Tiger classifier (roughly balanced dataset between the two classes – tiger and non-tiger). After detecting the animals, the cropped animal images were used to train a classifier based on the ResNet50 architecture to establish a decision boundary that differentiates between tigers and non-tigers. We trained both the ResNet-18 and ResNet-50 models using a dataset of 9,800 images, conducting the training over 30 epochs with a batch size of 32.

III. Data preparation for Siamese and Triplet network: The preparation of training data is influenced by the choice of loss function, specifically Contrastive Loss and Triplet Loss. Both approaches are designed to improve the model's capability to differentiate between similar and dissimilar image pairs. We utilized the Amur Tiger dataset, which includes cropped images of 107 individual tigers, with each tiger ID consisting of approximately 10 to 30 images (and more in some cases), resulting in a total of 1,887 labeled images available for exploration.

- Contrastive Loss minimizes the distance between embeddings of similar pairs and maximizes the distance between embeddings of dissimilar pairs.
- Triplet Loss ensures that the distance between an anchor image and a positive image (same tiger) is smaller than the distance to a negative image (different tiger).

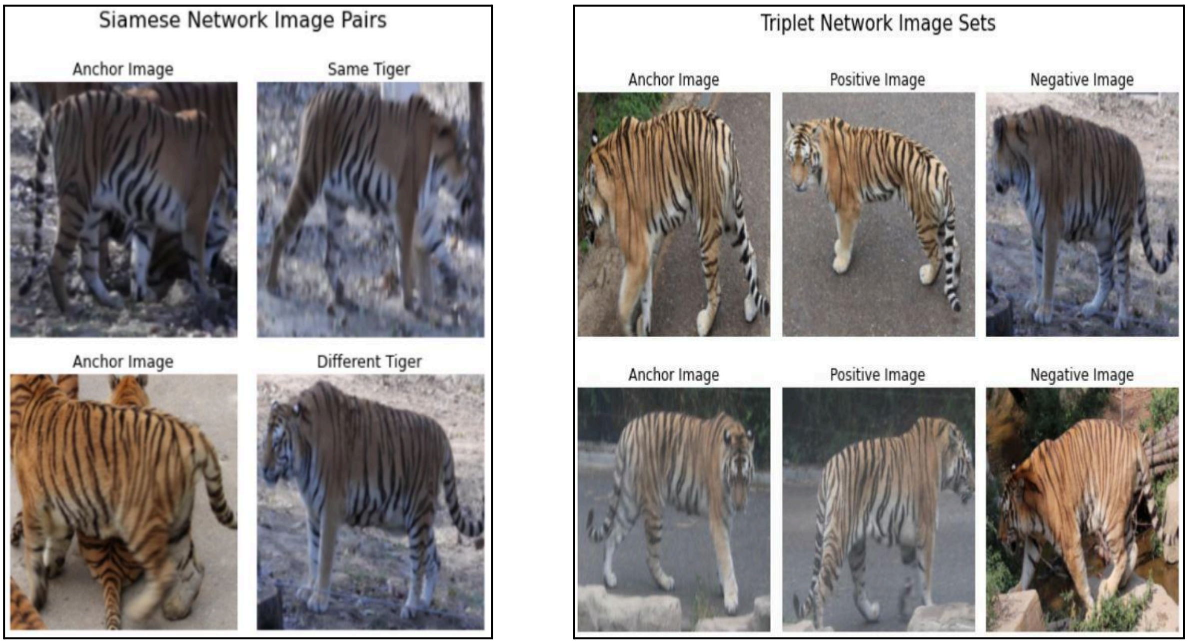

Figure 3: Left - Siamese network image pairs illustrating the anchor image of a tiger, a positive pair of the same tiger, and a negative pair of a different tiger. Right - Triplet network image pairs showing the anchor image, positive image of same tiger, and a negative image of a different tiger for comparison

IV. **Tiger re-id:**

Training a Siamese Network for Tiger Re-Identification: The goal of the re-identification task is to match a query image of a tiger with its corresponding images from a database of previously identified individuals. To achieve this, we modified and adapted the Tiger/Non-Tiger classifier developed in the previous stage to serve as the backbone CNN for a Siamese network architecture, allowing it to extract 512-dimensional feature embeddings from input images. The Siamese network was trained using contrastive loss, a method that penalizes positive pairs (images belonging to the same class) when their distances are large and negative pairs (images from different classes) when their distances are small. This contrastive learning approach enables the model to effectively differentiate individual tigers through pairwise comparisons.

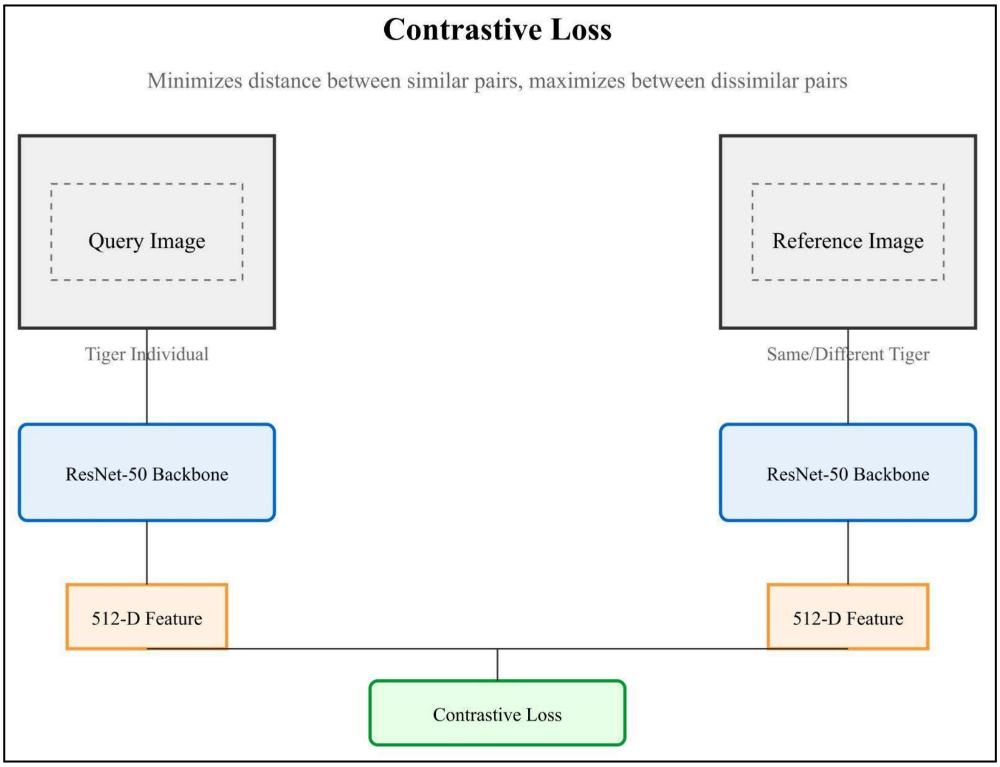

Figure 4: A query tiger image and a reference image from the database are processed through shared ResNet-50 backbones to extract 512-D feature embeddings. The contrastive loss minimizes the Euclidean distance between embeddings of similar pairs while maximizing it for dissimilar pairs, allowing the network to learn a feature space that clusters images of the same tiger together and separates those of different tigers.

Once trained, the model compares a query image against the reference database by calculating Euclidean distances between embeddings. Smaller distances indicate higher similarity, allowing the system to rank and match images based on proximity in the embedding space. This capability is essential for accurately identifying individual tigers and matching newly captured images with those of previously identified individuals. To identify a new tiger that does not belong to any in the database, a threshold for similarity needs to be established, which will be an area for further research. The model's performance was evaluated using Mean Average Precision (MAP) at various ranks (MAP@1, MAP@3, MAP@5, and MAP@7), providing insight into the effectiveness of the re-identification process.

5. Results:

ResNet 50 Tiger classifier results: Performance Metrics: We are able to achieve accuracy of,

Macro Accuracy: 94.40% (Average accuracy for each class without class imbalances)

Micro Accuracy: 95.23% (Global accuracy influenced by larger classes)

Tiger re-id results:

Mean Average Precision (MAP@K) is a metric that assesses how well the system ranks relevant images at the top of the list. It considers both the order of the rankings and the relevance of the retrieved images and is defined as the mean of Average Precision (AP) scores across all test images. The performance of our Siamese network for tiger re-identification was evaluated using Mean Average Precision (MAP) at different ranks. These metrics help quantify how effectively the model can identify a tiger by calculating the similarity between a query image and a set of reference images from the database.

To create a robust test dataset, we selected a minimum of two random images for each of the 107 individual tigers from the total of 1,887 labeled images. This selection constituted approximately 20% of the entire dataset, ensuring a diverse and representative sample for evaluating the model's performance. We also established a reference database containing up to eight random images for each known tiger ID, which is essential for assessing the model's effectiveness in matching a query image to previously identified tigers. Ideally, this database features a variety of poses for each tiger individual, including both left and right flanks of the tigers, as well as diverse lighting conditions and backgrounds.

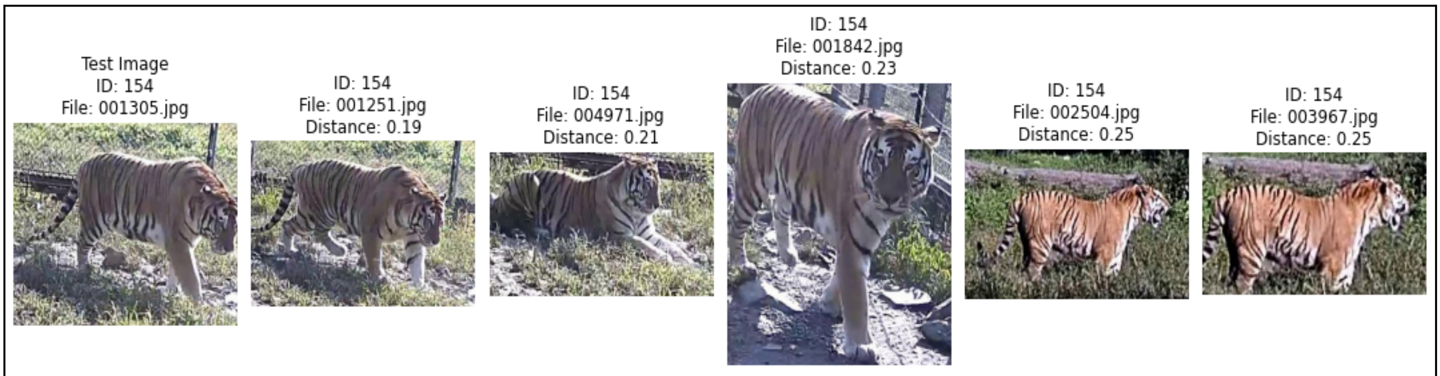

Figure 7: On the left, a test image of a tiger is shown, with the Euclidean distances to cropped images in the reference database arranged in increasing order from left to right. In this particular example, the top five retrieved images correspond to perfect matches for the test image's tiger ID.

The results achieved by our model demonstrate its ability to accurately match images of tigers:

- MAP@1: 0.7290: It correctly identified the query tiger in the top-1 retrieved image 72.90% of the time.
- MAP@3: 0.7952: The accuracy increased for the top-3 retrieved images, reaching 79.52%.
- MAP@5: 0.7983: When considering the top-5 images, the accuracy improved slightly to 79.83%.
- MAP@7: 0.7941: The performance remained consistent, achieving 79.41%.

We achieved a decent MAP@3 and MAP@5 scores indicating that the model effectively ranks relevant images in higher positions, demonstrating its effectiveness in top-k retrieval tasks. The Siamese network model's capability to differentiate between similar and dissimilar images is reflected in the consistently high MAP scores. This implies that the Siamese network effectively distinguishes individual tigers based on their unique stripe patterns, making it well-suited for re-identification tasks in wildlife conservation.

6. Implementation and User Benefit:

The primary goal of this project is to protect apex predators, such as tigers, which serve as indicators of ecosystem health. However, the approach can also be adapted for use with other species that can be identified by distinct stripe patterns or markings on their fur. The implementation of a Siamese network facilitates one-shot learning, enabling the identification of individual animals beyond the training dataset. This means there is no need to modify the architecture or retrain the neural network each time new tiger images are captured, thus streamlining the process of wildlife monitoring. Additionally, this technology can be invaluable for the Ministry of Environment, Forest and Climate Change in India, assisting the forest department in conducting tiger censuses every four years. By equipping this advanced image recognition system, they can more accurately count and monitor tiger populations, leading to better-informed conservation strategies.

7. Limitations and Further Improvements:

- The MDv6-c lightweight detector from the ‘PyTorch Wildlife’ is in beta testing. Once fully developed, it can enhance animal detection and make the entire system viable for edge computing applications.
- Appropriate data augmentation techniques can be adopted for individual re-identification (Re-ID) to make the model more robust and increase its generalizability.
- A drawback of the training dataset is that it is primarily based on zoo images, which limits the model's generalizability to wild tiger populations, diverse environmental conditions, and various camera types.
- The individual tiger images in the training dataset consist mainly of different frames from the same video, resulting in similar backgrounds and poses for images of the same tiger. This presents a challenge in effectively testing the model's performance in real-world applications.
- Including images of the left and right flanks of tigers is essential for individual re-identification, as tiger stripes are not symmetrical. This aspect is not considered in the current implementation.
- A distance threshold needs to be established to confidently determine whether a tiger is the best match in the database or if it is a newly captured, unknown tiger image.

8. References:

[1] A Collaborative Deep Learning Framework for Conservation Andres Hernandez* 1,2 Zhongqi Miao* 1 Luisa Vargas 1,2 Rahul Dodhia 1 Pablo Arbelaez 2 Juan M. Lavista Ferres 1.1 Microsoft AI for Good, Redmond. 2 Center for Research and Formation in Artificial Intelligence. Colombia.

[2] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” Cmu.edu. [Online].

[3] E. N. T. E. Kalviainen, “Siamese network based pelage pattern matching for ringed seal re-identification,” Thecvf.com. [Online].

[4] T. L. Van Zyl, M. Woolway, and B. Engelbrecht, “Unique animal identification using deep transfer learning for data fusion in Siamese networks,” in 2020 IEEE 23rd International Conference on Information Fusion (FUSION), 2020, vol. abs 1707 6642, pp. 1–6.

[5] L. Shuyuan, L. Jianguo, T. Hanlin, Q. Rui, and L. Weiyao, “ATRW: A benchmark for Amur Tiger Re-identification in the Wild,” arXiv [cs.CV], 2019

[6] A. Shukla, C. Anderson, G. S. Cheema, P. Guo, S. Onda, D. Anshumaan, S. Anand, and R. Farrell, "A Hybrid Approach for Tiger Re-Identification," *2019 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Seoul, Korea (South), Oct. 2019, pp. 4185-4194.

[7] S. S. Kathait, V. Singh, and A. Kumar, “Individual Tiger Identification using Transfer Learning,” Valiancesolutions.com. [Online].

[8] S. Schneider, G. W. Taylor, S. Linquist, and S. C. Kremer, “Similarity learning networks for animal individual re-identification -- beyond the capabilities of a human observer,” arXiv [cs.CV], 2019.

[9] Li, Sixu, Boyang and S. Xie, “Animal Recognition Using Siamese Network with Two Kinds of Backbone Networks”.

[10] S. Sharma, S. Timilsina, B. P. Gautam, S. Watanabe, S. Kondo, and K. Sato, “Enhancing Sika deer identification: Integrating CNN-based Siamese networks with SVM classification,” Electronics (Basel), vol. 13, no. 11, p. 2067, 2024

9. Appendix:

Hardware Configurations:

- CPU: 2 to 4 cores (16 GB memory per core)
- GPU: 2 with 16GB/core RAM with CUDA support (NVIDIA drivers for CUDA Toolkit 12.1)
- Storage: 100 GB (size may vary based on training data)

Software Requirements:

- OS: Linux
- Python: Version 3.8
- Tools/Libraries: Jupyter Notebook, Pytorch-Wildlife, PyTorch, Torchvision, Pillow, OpenCV, Ultralytics, PyTorch Lightning, ResNet, YOLOv5.

Model training details:

- Siamese Network for Tiger re-id:

  Architecture: Siamese Network with a pre-trained ResNet-50 backbone for feature extraction, Euclidean distance for Contrastive Loss function (margin = 1.0) with anchor and pair images.

  Hyperparameters: Adam optimizer (learning rate = 0.001), Learning Rate Scheduler: StepLR (step size = 10, gamma = 0.5), Training Epochs: 55.