Minimizing Human Labor for In-the-Wild Camera Trap Processing Pipeline

Haoyu Chen, Amy R. Reibman
Elmore Family School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN
{chen1562,reibman}@purdue.edu

2024 IEEE 26th International Workshop on Multimedia Signal Processing (MMSP) | 979-8-3503-8725-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/MMSP61759.2024.10743995

***Abstract*—Camera traps are an important tool in ecological studies for non-intrusive monitoring of various animals. However, annotating camera trap data usually requires large amount of human labor. Therefore, we propose a solution for practitioners with limited human resources: an automated pipeline for curating in-the-wild camera trap data that mimics human annotators and significantly reduces the amount of human labor needed. We also propose evaluation protocols for estimating system performance on unlabeled data, and present experiments that demonstrate our pipeline's strengths, weaknesses, and flexibility to accommodate users' requirements.**

***Index Terms*—In-the-wild data processing, animal detection, animal species classification, few-shot learning**

I. Introduction

This paper addresses the critical gap between research advancements and their real-world applications using camera trap images. In what follows, we will discuss the effect of data curation, challenges of unlabeled/in-the-wild data, and an end-to-end pipeline where detection performance and classification performance are intertwined.

Camera traps have been a valuable tools in ecological research, enabling the non-intrusive monitoring of wildlife across diverse environments. Over the recent years, the advancement of computer vision tools shows a bright future in automating the detection and classification of animal species for camera trap data. However, most current research relies heavily on human labor for data curation, which may not be available for every practitioner.

Therefore, in this paper, we propose a data curation pipeline that leverages an existing tool, temporal information, and few-shot learning to mimic the benefit from having humans involved in the data curation, and we demonstrate the pipeline on in-the-wild camera trap data collected from Senegal. Comparing to curated datasets such as Snapshot Serengeti and World Conservation Society (WCS) camera trap datasets, our raw data is "in-the-wild" due to 1) we have not used any human annotation to remove non-animal false triggers, and 2) many of the species present in our data are unique to the geographical location, and cannot be found in existing species classification solutions[1].

Our main contributions are: 1) proposing an automated data curation pipeline that mimics human annotators, 2) allowing the user to customize the pipeline considering the trade-off between accuracy and amount of data processed, and 3) proposed end-to-end evaluation strategies to estimate performance on unlabeled data.

II. Related Work

In recent years, as computer vision technology advances, many researchers have also adapted deep learning methods to assists camera trap studies in terms of classification [1], detection [2], or even re-identification [3]. A brief description of full processing pipeline is proposed in [4]; however, the paper provided few technical details and also requires additional labeling and training for the classifier. There has been very little research that considers the end-to-end processing pipeline and discusses relationship between detection results and classification results. This problem also exists in other communities; for example, [5] discusses the lack of joint consideration between face detection and recognition, as most recognition evaluation datasets assume perfect input data.

However, despite exciting news about technical advancements, there is still a gap between research and real world application. Much published research depends implicitly on extensive human annotations for both training data, and notably, evaluation data. In particular, the human annotation process acts as implicit quality filtering, and eliminates empty or un-recognizable objects during annotation, leading to overly optimistic evaluation performance that may not transfer to real-world data.

Take Snapshot Serengeti [6] as an example. Its web page[2] states that a total of 7.1M images has been collected. However, only 78,000 images are annotated with bounding box and species labels — about 1% of total images. We can identify at least two major data curation steps described in [6] and its tech blogs. Through a guided interface on Zooniverse, "citizen scientists" can toggle options like build, horn shape, or tail shape, to assist the identification of the species present in a picture. At least 10 users have to classify an image before a consensus rule is applied to consolidate user inputs. Other citizen scientist are assigned the task of identifying/confirming

This work is funded by NSF grant number 2022314, under the HUNTRESS (HUnting, Nutrition, Tool-use, Reproductive Ecology, and meat Sharing in Savanna chimpanzees) project.

[1] github.com/microsoft/SpeciesClassification

[2] lila.science/datasets/snapshot-serengeti

whether an image is empty[3]. Instead of needing consensus from 10 people, only 2 people per image is needed for this task.

Gomez Villa et al. [7] used Snapshot Serengeti to illustrate the point that well-curated data (balanced between species, no empty images, etc.) is better for training CNNs than less-curated data (unbalanced training dataset containing empty images). Beery et al. [8] explores the generalizability of classification and detection models, and emphasized the challenge of processing data from unseen locations. Also, using the well-known ImageNet as an example, the authors of [9] showed that selecting harder images rather than easy images could causes more than 10% drop in testing accuracy, which hints at the benefit of having curated/easy evaluation data. Similarly, we observe much worse performance on our in-the-wild data than published camera trap dataset such as WCS[4] or Snapshot Serengeti [6] when we applied existing detection tools.

III. Motivations from In-the-Wild Data

Under a broad project that investigates ecological basis of hunting and meat sharing in female savanna chimpanzees [10,11], our goal is to use camera traps to estimate the population density of several species that are hunted by chimpanzees. So far, we have received 8 shipments of data, totaling more than 67,000 videos from 341 camera locations. Details of these shipments can be found in Table II. We will refer to our data as “Senegal” data.

As the camera traps are automatically triggered by motion, many recorded videos are in fact empty and are likely triggered by background objects like branches or leaves rather than animals. These empty videos, when processed through an animal detection system, yield many false detections.

In our previous work [12], videos from 40 camera trap locations between shipment 2 and 3 are processed automatically without any human intervention. We then randomly sampled 8,000 bounding boxes out of all detection outputs; out of the 8,000 bounding boxes, 2,586 are background objects mis-detected as animals. That is to say, 32.3% detected animals are in fact false detections.

To provide a comparison, we ran the same detector (Microsoft MegaDetector v4 [4]) on a randomly sampled subset of the annotated portion of Snapshot Serengeti, and the false detection rate is only 3.16%.

Our initial assumption was that the model is prone to false detections; however, when evaluated on a dataset that is selected to not have empty images (e.g., labeled part of Snapshot Serengeti), false detections do not pose as an issue. To confirm that, we ran a short experiment. First, we randomly selected 200 images from our collection that we confirmed to contain animals, and labeled them with bounding box annotations — we call it “Senegal (curated)”. We then take two publicly available camera trap datasets — Snapshot Serengeti and WCS, and sample a subset of each to have similar bounding box size and count statistics as the labeled 200 images from our data. We then applied the same detector on these three sub-datasets, and the results are presented in Table I.

TABLE I: Detection Performance on Various Camera Trap Datasets

Dataset	Miss rate (%)	False detections rate (%)
Senegal (un-curated)	N/A	32.3
Senegal (curated)	1.18	2.91
Snapshot Serengeti	2.21	1.57
WCS	3.09	1.91

As we observe with “**Senegal (curated)**”, provided we manually remove empty images and only include images with animals, the false detection rate goes down drastically; notably, it is now in the same range as the two public datasets.

We believe that this small experiment also helps to show the difference between curated and un-curated data, as well as difficulties caused by having in-the-wild, un-curated data. Such difficulties can be solved by creating a massive amount of annotations using human verification, but not everyone has the manpower to do that.

Besides the benefit of curated data, we also need to consider the trade-off between human labor and the amount of useful data produced. For example, in a study for zebra re-identification [3], a pipeline is introduced to require less human labor; it is achieved by only having one species to consider and by incorporating many filtering stages to reject images that may not have high enough quality. This method results in a very strict pipeline and only yields 685 images out of 8.9 million images.

In this paper, we propose a processing pipeline that yields curated results while utilizing minimal human labor. We also show that we can alter components of our pipeline to reach a desired trade-off point, depending on the user’s need.

IV. Methodology

In this section, we first present our pipeline to curate in-the-wild camera trap data. We will then discuss the evaluation protocols for unlabeled data, in order to demonstrate the effectiveness of the pipeline.

A. *Processing Pipeline*

Our pipeline will focus on: 1) detecting potential animal sightings, 2) eliminating false detections, and 3) providing a species classification result. We will also show that the pipeline can be adjusted to meet the user’s need. A block diagram of the pipeline can be seen in Figure 1. We apply a heatmap-based detection filter to mimic the human effort of removing empty images (Section IV-A2), and a few-shot learning classifier with adjustable decision rules to mimic human annotations of species (Section IV-A3).

The majority of our data are in video format, and we design our full processing pipeline to obtain potential animal species classifications from videos. To reduce the number of potentially repetitive images, we only consider 1 frame from every 30 frames, which correspond to one second as our videos

[3] blog.snapshotserengeti.org/2019/05/28/machine-learning-and-citizen-sciencea-winning-combination/

[4] lila.science/datasets/wcscameratraps

TABLE II: Data shipments we have received.

Shipment	Videos	Images	Size (GB)	Cameras
1	2,998	106,283	187	48
2	25,606	10,433	1,304	110
3	3,489	2,392	132	31
4	6,894	1,410	284	37
5	11,523	356	642	39
6	4,759	1,354	167	22
7	6,391	1,548	275	19
8	5,939	879	252	35
Total	67,599	124,655	3,242	341

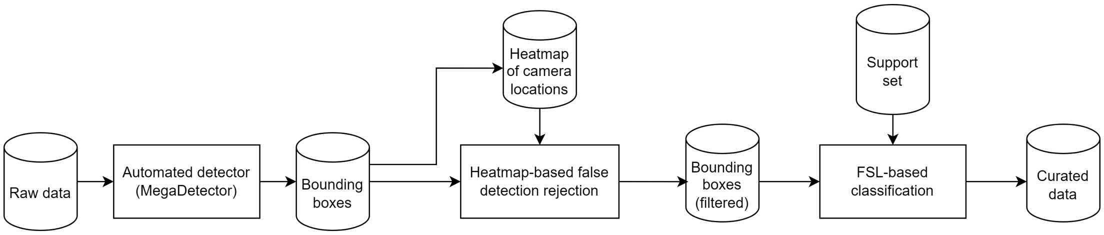

Raw data
Automated detector (MegaDetector)
Bounding boxes
Heatmap of camera locations
Heatmap-based false detection rejection
Bounding boxes (filtered)
Support set
FSL-based classification
Curated data

Fig. 1: Block Diagram of the Processing Pipeline

have 30 frames per second. This reduces processing time and redundant appearances.

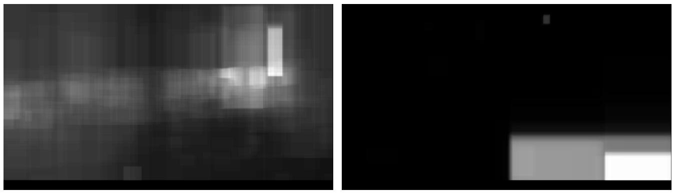
1) Detection: We then run detection on these sampled frames to creating bounding boxes of potential animals. Over the course of the project, we have used both Microsoft's MegaDetector v4 and v5 [4]. In this paper, we will only use the newer v5.

MegaDetector has been trained on millions of images from many public camera trap datasets plus additional private data[5], and has been used by more than 50 organizations. It is arguably the most popular detector specialized for detecting animals in camera traps, and it would be impossible for us to either train a model from scratch or fine-tune an existing model to match the performance of this model, using our 200 labeled images.

However, despite the massive amount of training data and many iterations of adjustments, the MegaDetector model is still not perfect. As we mentioned in Section III, it yields about 30% false detections when applied to shipments 2 and 3 of our data.

2) Detection result filtering: First, we propose to directly utilize the detection results to eliminate some of the false detections. The idea is to identify objects that appears repetitively in the same position, and reject them as background objects. Because we have observed that many cameras gradually shift positions, we choose not to use a single heatmap to represent a camera location, but instead compute a heatmap for every 100 sampled frames; the heatmap is then normalized to [0,1]. Then, for each detected bounding box, we compute its overlap with the normalized heatmap. Two sample heatmaps are shown in Figure 2: (a) is the heatmap resulting from many animal detections, which are distributed spatially, and (b) is the heatmap resulting from constant detection of the same background object.

A threshold is set to determine when a bounding box is classified as a background object should rejected. For simplicity here, we will outright reject the entire frame when we reject a background object. We refer to this heatmap-based detection result filtering as "heatmap" for short.



(a) Animal detections

(b) False detections

Fig. 2: Comparison between heatmaps from two locations, one featuring mostly animal detections and the other featuring mostly false detections.

3) Classification using few-shot learning: The core of few-shot learning (FSL) is to use a small number of labels for classification; FSL has been a popular topic recently, and many methods have been proposed. In our case, one could consider that instead of training a network to give explicit classification, the network is trained to extract features that separate classes. In FSL, the classification of unknown images are done by comparing their features to a small group of labeled images, known as "support set". The major benefit of FSL method is that the training can be done with publicly available data, and we only need to label a few images when we switch to our data with unseen species and environment. In contrast, traditional deep learning requires a much larger amount of training data to be able to adapt to new classes and environments.

When comparing several networks for our task [12], we see that PMF [13] performs the best so we use it here. We train the network purely on public data, namely Snapshot Serengeti.

[5] github.com/microsoft/CameraTraps/blob/main/archive/megadetector.md

Then, we labeled a few images per species using shipment 2 and 3 of our data; the number of labeled images is shown in Table III. Since the number of images for each species are different, we choose to randomly sample 10 images per species to construct the support set. Note that this also mimics a scenario where much less human labor is needed. More implementation details can be found in [12].

TABLE III: Total labeled images for each species

Species	Images	Species	Images
Baboon	108	Hartebeest	25
Buffalo	74	Oribi	22
Bushbuck	126	Patas monkey	30
Duiker	53	Roan antelope	97
Green monkey	99	Warthog	83
Guineafowl	86		

Typical FSL classification considers only the top-1 result. In other words, an unknown image's class is assigned to be the closest support image's class in the feature space. However, as we are observe many false detections in our data, we alter the FSL classification decision rule to try to reject false detections. Here, we apply a K-Nearest-Neighbor decision rule — for example, with a 3/5 decision rule, at least 3 images from the top-5 matches must be from the same species for it to be classified.

As mentioned above, we sample 10 images per species to create a balanced support set. Nevertheless, we are also interested in the effect of having more support images with an unbalanced distribution among species. So in this paper, we also perform additional experiments with all images we labeled (Table III) as the support set.

B. Evaluation Methods

As we have emphasized throughout this paper, we conduct our experiments on fully un-labeled, un-curated data. We apply two stages of evaluation. The first evaluates the detection performance to show the effectiveness of the heatmap detection filter. The second evaluates the end-to-end pipeline results for animal classification, to show the effect of individual modifications on the whole system.

To begin, we investigate the effectiveness of the heatmap detection filter by counting how many images with detections contain mistakenly detected background objects. First, we randomly sample 200 images from all images with at least one detection, and count the number of images with false detections; this step is repeated 5 times and an average with confidence interval is reported. This process is repeated to images after applying the heatmap filter.

Evaluating classification results without ground truth labels is more difficult. In addition, we face a more complex scenario of "classification in the presence of false detections" which is not considered by most classification-oriented research, where it is assumed that all inputs are relevant and an overall classification accuracy is sufficient. However, in a real application, it is almost impossible to have a perfect detection system that only returns relevant objects. Due to the presence of irrelevant objects, we propose an evaluation scheme similar to the precision-recall curve.

Our final task is to classify cropped bounding boxes into one of 11 species: baboon, buffalo, bushbuck, duiker, green monkey, guineafowl, roan antelope, warthog, hartebeest, oribi, and patas monkey. To alleviate human labor, we choose 5 species to verify the results: baboon, green monkey, bushbuck, duiker, and warthog. These 5 species are common enough to yield meaningful statistics, and are affected by two major error causes we observed with our data: 1) confusion between similar looking species (baboon vs. green monkey, bushbuck vs. duiker) and 2) false detections (branches, rocks) misclassified as animals.

First, for each species, we sample up to 50 images and record the number of correctly classified images. We then compute a **sampled precision**, which is the total number of correctly classified images divided by the total number of sampled images. As we add rejection schemes, we may inevitably reject actual animal images. To quantify that, we compute a projected number of correct classifications for each species. For example, in one experiment, the classifier indicates 2,731 images are baboons. We sampled 50 images and found that 44 of them are indeed baboons. Hence, we calculate that $44/50 * 2731 = 2403$ images are projected to be correctly classified as baboons. We then sum all species to get a **projected total number of correctly classified images**. This estimation saves us from viewing thousands of images. To provide another aspect of classification accuracy, we also compute the **projected precision**, which equals the projected total number of correctly classified images divided by the total number of classified images.

In the results section, we will show the relationship between precision measures and projected total number of correctly classified images.

V. Experiments and Results

Using our evaluation strategy for unlabeled data, we conduct experiments to examine the strength and weakness of the proposed pipeline on unlabeled, in-the-wild data. The source data of the experiments are from shipment 2 and shipment 5. These are from the same general region of Mont Assirik, but different camera locations. As mentioned earlier in Section IV-A3, our support set images are from shipment 2 and 3; hence, the addition of shipment 5 allows us to investigate the pipeline's performance on completely unseen/out-of-distribution data.

In this section, we perform two sets of experiments: Section V-A focuses only on detection performance and how the heatmap filter helps reduce false detections; Section V-B then examines end-to-end classification results of the pipeline and discusses the effect of various modifications.

A. Detection

To illustrate the effectiveness of heatmap, we perform three sets of experiments. All three experiments compare detection performance with or without the heatmap-based rejection that aims to reject false detections. The difference among them is the set of data on which the experiment is performed. We used the evaluation protocol described in Section IV-B: sample 200 images, count false detections images, and repeat.

First, Table IV shows detection results on shipment 2 data, which is all from ground-based cameras. It is obvious that applying the heatmap-based detection filter drastically decreases the amount of images with false detections.

TABLE IV: Detection Performance for Shipment 2 Data

Heatmap	False detections rate (%)
Without rejection	35.3 ± 3.98
With rejection	8.2 ± 2.18

We then performed experiment 2 and 3 on data shipment 5, which contains both ground-based and arboreal cameras, where the camera is mounted near tree crowns, around 3 meters above the ground. Figure 2-b shows an example of arboreal cameras. False detections happen much more frequently on arboreal cameras because 1) they are close to leaves and branches that moves, and 2) the camera itself is more likely to move compared to ground-based cameras. Therefore, here, we perform two sets of experiments, one with ground-based cameras only, and one with both ground-based and arboreal cameras.

The ground-only results are shown in Table V, and the ground-and-arboreal results are shown in Table VI. In both cases, we see that the heatmap filter helps drastically decrease false detection rate.

It it worth noting that we also see a significant increase in the numerical value of false detection rate with arboreal cameras included. This confirms our speculation that arboreal cameras produce more false detections, but it also shows that our filter eliminates false detections even with arboreal cameras.

TABLE V: Detection Performance for Shipment 5 Data (Ground)

Heatmap	False detections rate (%)
Without rejection	57.4 ± 4.92
With rejection	12.5 ± 2.30

TABLE VI: Detection Performance for Shipment 5 Data (Ground and Arboreal)

Heatmap	False detections rate (%)
Without rejection	72.6 ± 2.11
With rejection	22.9 ± 1.43

B. *Classification*

Next, we examine the end-to-end classification results, and explore how our modifications affect the performance.

Again, we begin our experiment with data shipment 2, and follow the evaluation protocol for unlabeled data specified in Section IV-B. The results are shown in Table VII. We created 7 different tests by varying the modifications, as indicated on the left part of the table. Except for the last row, which uses extra support images, all other use the exact same support set (same 10 images for each species) for consistent comparison.

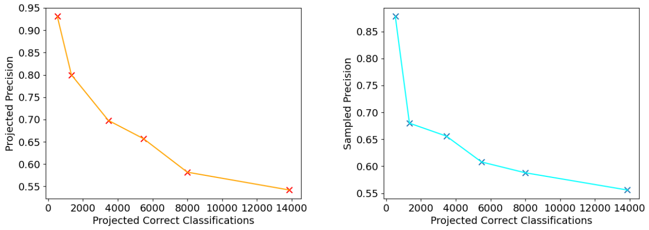
Looking at row 1 vs. row 2, and row 1 vs. row 3, we see that individually, both adding heatmap and KNN improve the classification precision in terms of both sampled precision and projected precision. As expected, when we apply both heatmap and KNN (see row 4), the precision is further improved. Comparing results between row 4, 5, and 6, we see that as we make the KNN decision rule more strict, precision also goes higher. Moreover, when we look at the results from row 1 to row 6, we can see that as precision increase, the projected number of animals correctly classified goes down.

To help interpret the results, we have plotted both sampled and projected precision against the projected number of animals correctly classified in Figure 3. It shows that potentially, our pipeline can be applied in many different use cases. By adjusting the modifications, the user could choose the desired operating point depending on whether they want to maximize the number of animals found, or maximize the classifier accuracy which could reduce human labor for further verification.

Finally, row 7 presents the case with extra images as the support set. We can make direct comparison between row 7 and row 4, since both have the same modifications except row 7 has extra support images. Contrary to our expectation, extra support images do not offer any significant improvement over only having 10 images per species. Therefore, we suspect that quality of the support set may be more important than quantity of the support set; "what makes a good support set?" is a topic worth further investigation.

TABLE VII: Classification Result with Various Modifications, Data Shipment 2

Modifications			Results		
Heatmap Rejection	KNN	Extra Support	Projected Correct Count	Sampled Precision	Projected Precision
No	No	No	13,868	0.556	0.542
Yes	No	No	5,482	0.608	0.657
No	3/5	No	7,997	0.588	0.582
Yes	3/5	No	3,460	0.656	0.698
Yes	4/5	No	1,338	0.680	0.800
Yes	5/5	No	526	0.878	0.931
Yes	3/5	Yes	4,359	0.564	0.544



(a) Projected Precision

(b) Sampled Precision

Fig. 3: Precision vs. Projected Total Correct Classification

TABLE VIII: Classification Result with Various Modifications, Data Shipment 5

Modifications			Results		
Heatmap Rejection	KNN	Extra Support	Projected Correct Count	Sampled Precision	Projected Precision
No	No	No	5144	0.256	0.469
Yes	No	No	2814	0.448	0.706
No	3/5	No	1884	0.184	0.298
Yes	3/5	No	1573	0.388	0.638
Yes	3/5	Yes	1895	0.342	0.469

Now. we will shift our focus to shipment 5, the unseen/out-of-distribution scenario, and results are shown in Table VIII. Unlike what we observe with shipment 2, the results here are not entirely as expected. Comparing row 1 and 2, we can see that heatmap filtering still improves precision significantly. However, by comparing row 1 vs. row 3, or row 2 vs. row 4, we see that KNN does not help increase precision, and is in fact detrimental to the classification result. Our conclusion is that heatmap, which does not rely on labeled data, can universally improve detection and classification results, while KNN is more influenced by the labeled support set and may not be effective with out-of-distribution data.

With out-of-distribution data, having extra support images does slightly improve KNN-based results (row 3 vs. row 5). However, if we compare row 2 vs. row 5, we see that extra support images still do not offset the detrimental effect KNN caused with unseen data.

VI. Conclusion and Future Work

In this paper, we pointed out the lack of study on processing true in-the-wild camera trap videos. Most current work either only focuses on a single aspect or requires large effort by human annotators to create training and evaluating data. To fill the void for researchers who cannot afford a large amount of human labor, we proposed a processing pipeline that requires minimum human involvement, which can also be adjusted to accommodate the user's need between the number of animals found and the amount of human labor needed.

Our pipeline consists of an existing detection tool, a few-shot learning-based classifier, and modifications that mimic the role of human annotators to either reject false detection or make species classification. Our detection experiments in Section V-A demonstrated the effectiveness of a heatmap detection filter in rejecting false detections during detection phase. Our classification experiments in Section V-B yielded mixed results: for in-distribution data (where both support set and unknown images are from shipment 2), our pipeline produced promising results, showing that labeling 10 images per species can achieve reasonable classification results. For out-of-distribution data (where the support set are from shipment 2 while unknown images are from shipment 5), the general classification performance became much worse — heatmap filtering still helped improve the results while the KNN decision rule was lackluster.

In conclusion, we proposed a end-to-end processing pipeline for large, unlabeled camera trap data, and showed its strengths and weaknesses under different scenarios. We also proposed evaluation protocols for estimating classification performance on unlabeled data.

Nevertheless, our work here is not entirely done, and there are still many aspect that can be further investigated. For example, in Section V-B, we ask the question "what makes a good support set for few-shot classification?" In addition, the classifier still faces difficulty with similar looking species such as duiker vs. oribi or between several primate species (baboon, green monkey, patas monkey).

References

[1] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff 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.

[2] Ilja Pavlovs, Kadir Aktas, Egils Avots, Alekss Vecvanags, Jevgenijs Filipovs, Agris Brauns, Gundega Done, Dainis Jakovels, and Gholamreza Anbarjafari, "Ungulate detection and species classification from camera trap images using retinanet and faster R-CNN," *Entropy*, vol. 24, pp. 353, 02 2022.

[3] Avirath Sundaresan, Jason Parham, Jonathan Crall, Rosemary Warungu, Timothy Muthami, Jackson Miliko, Margaret Mwangi, Jason Holmberg, Tanya Berger-Wolf, Daniel Rubenstein, Charles Stewart, and Sara Beery, "Adapting the re-id challenge for static sensors," *Workshop Camera Traps, AI and Ecology*, 2023.

[4] Sara Beery, Dan Morris, and Siyu Yang, "Efficient pipeline for camera trap image review," *arXiv preprint arXiv:1907.06772*, 2019.

[5] Praneet Singh, Edward J. Delp, and Amy R. Reibman, "End-to-end evaluation of practical video analytics systems for face detection and recognition," *Electronic Imaging*, vol. 35, no. 16, pp. 111–1–111–1, 2023.

[6] Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[7] Alexander Gomez Villa, Augusto Salazar, and Francisco Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," *Ecological Informatics*, vol. 41, pp. 24–32, 2017.

[8] Sara Beery, Grant Van Horn, and Pietro Perona, "Recognition in terra incognita," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[9] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar, "Do ImageNet classifiers generalize to ImageNet?," in *International conference on machine learning*. PMLR, 2019, pp. 5389–5400.

[10] Stacy Lindshield, Stephanie Bogart, Mallé Gueye, Papa Ndiaye, and Jill Pruetz, "Informing protection efforts for critically endangered chimpanzees (Pan troglodytes verus) and sympatric mammals amidst rapid growth of extractive industries in Senegal," *Folia Primatologica*, vol. 90, pp. 124–136, 03 2019.

[11] Jill Pruetz, P. Bertolani, Kelly Boyer Ontl, Stacy Lindshield, Mack Shelley, and Erin Wessling, "New evidence on the tool-assisted hunting exhibited by chimpanzees (Pan troglodytes verus) in a savannah habitat at Fongoli, Sénégal," *Royal Society Open Science*, vol. 2, 04 2015.

[12] Haoyu Chen, Stacy Lindshield, Papa Ibnou Ndiaye, Yaya Hamady Ndiaye, Jill D. Pruetz, and Amy R. Reibman, "Applying few-shot learning for in-the-wild camera-trap species classification," *AI*, vol. 4, no. 3, pp. 574–597, 2023.

[13] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales, "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.