iWildcam Animal Species Classification based on Camera Trap Images

Final Report

Advisor: Prof. Joshua Blumenstock

Akshay Punhani, Rahul Nangia, Vikramank Singh
University of California, Berkeley

Abstract

Camera trap images are increasingly used for remotely observing wildlife and have been regarded as one of the most powerful tools for wildlife research. These images are analyzed to monitor the biodiversity and population density of animal species. However, camera traps installed in national parks around the world result in millions of images that need to be visually analyzed, in order to extract insights that can be used in ecological analyses. To add to that, we are faced with an interesting challenge of identifying & classifying a species in a new region that the system may not have seen in previous training data. Expanding the scope of the system from specific regions where the training data was collected to nearby areas is necessary to build a robust automated animal species classification model.

1 Introduction

Extracting information from motion sensor images is expensive, time-consuming and involves manual work. We demonstrate that information can be automatically extracted using deep learning (Y et al., 2015) techniques leveraging the computation power thats available with GPUs today. We train a deep convolutional network model to first identify if theres an animal present in the images and if thats true, the model performs animal species classification on 21 species. The model is currently trained on 150,000 images from both the Caltech Camera trap dataset and the Idaho dataset. We were able to achieve an accuracy of 93% for the empty-non empty classification task and 440,000 for the animal species identification task. Based on our results, we can conclude those deep learning technologies now and in the future could ensure high volume, inexpensive collection of ecological data specifically about the number and type of animals in the wild.

2 Background & Related Work

There have been many attempts to automatically identify animals in camera-trap images. We came across two implementations of machine learning models that used state of the art deep convolutional neural networks for this task. In the first implementation (Tabak et al., 2018), machine learning models were trained using convolutional neural networks with the ResNet18 architecture and 3,367,383 images from the Serengeti National Park Database (Swanson et al., 2015) to automatically classify wildlife species from camera trap images obtained from five states across the United States. he trained model achieved 98% accuracy at identifying species in the United States. The testing was performed on dataset from Canada to achieve 82% accuracy and correctly identified 94% of images containing an animal in the dataset from Tanzania.

The closest work to ours is Norouzzadeh et al who also evaluate Deep Convolutional Neural Networks with a two stage pipeline approach. Apart from the species-identification task, they also also attempt to count animals, describe their behavior, and identify the presence of young. Specifically for their species identification task, they created a two-stage pipeline by having the VGG model from the empty vs. animal experiment classify whether the image contains an animal and, if it does, having the ensemble of models from the second stage label it. The best model for the second task was identified as the ResNet-152 trained on the Serengeti National Park dataset. As part of the two stage pipeline , the first task of detecting images that contain animals was able to perform with an accuracy of 96.6%. While on the animal identification task, their model with 43%

confidence threshold was able to achieve a top-5 accuracy of 99.1% and top-1 accuracy of 93.8%. In contrast, with the same approach of a two stage pipeline, we have attempted to train a model that will be able to generalize better than all the previous models in this domain. With the goal of automating the animal detection process for National Parks around the world, we have identified a state of the art model that is able to learn features for it to generalize much beyond the training data.

3 Datasets

3.1 Caltech Camera (CCT) Traps

The Caltech Camera Traps (Beery et al., 2018) training set originally includes a total of 196,157 camera trap training images from 138 different camera locations in Southern California. The set of training classes is: 'bobcat', 'opossum', 'coyote', 'raccoon', 'dog', 'cat', 'squirrel', 'rabbit', 'skunk', 'rodent', 'deer', 'fox', 'mountain_lion', 'empty'. The sequence of images are collected at approximately one frame per second. The training set consisted of 131457 empty images and 64842 non-empty images i..e only around 32% of images contained animals from the above mentioned species. Also, the dataset is highly imbalanced meaning that some species such as deers. are more frequent that species such as mountain lions. This imbalance is problematic for deep learning techniques as they become heavily biased toward classes with more data. If the model just predicts the frequent classes most of the time, it can still get a very high accuracy without investing in learning rare classes, even though the test could have a higher percentage of these infrequent classes. Another issue with the dataset and its further analysis is that some camera trap images had more than one animals captured and hence humans labelled the images with two animal species. There were 1.9% of such images present in the dataset. These were removed from the dataset. (M et al., 2012)

3.2 Idaho Camera Traps

Similarly, the Idaho National Park dataset consists of a total of 25713 camera trap images. The set of training classes includes deer, squirrel, small_mammal, racoon, coyote, rabbit, bobcat, fox, black_bear, skunk, prongs, bison, bighorn_sheep, cat, mountain_lion, mountain_goat, rodent. The training set consisted of 25713 images all of which were non-empty unlike the CCT. Also, this dataset too is highly imbalanced with the deer being most frequent with 4929 images and elk having just 7 instances of being captured on the camera traps. Unlike the CCT dataset, the Idaho dataset did not contain more than one animal label for each image.

Architecture	# of Layers
VGG-16	$16 + 2^{\#}$
VGG-19	$19 + 2^{\#}$
ResNet-18	18
ResNet-50	50

Table 1: We employ different deep learning architectures to infer which one works the best and to be able to compare the difference between accuracies come from different architectures.

4 Experiments and Results

We found that a two-stage pipeline outperforms a one-step pipeline: in the first stage a network solves the empty vs. animal task (Task I), i.e. detecting if an image contains an animal; in the second information extraction stage, a network then reports information about the images that contain animals.

4.1 Architectures

Different DNNs have different architectures, meaning the type of layers they contain (e.g. convolutional layers, fully connected layers, pooling layers, etc.), and the number, order, and size of those layers (9). In this work, we test 5 different modern architectures at or near the state of the art (as shown in Table 1) to find the highest-performing networks and to compare our results with (Norouzzadeh et al., 2018). We only trained each model one time because doing so is computationally expensive and because both theoretical and empirical evidence suggests different DNNs trained with the same architecture, but initialized differently, often converge to similar performance levels.

4.2 Preprocessing and Training

The original images in the dataset are 1536 x 1048 pixels, which is too large for current state-of-the-art deep neural networks owing to the increased computational costs of training and running DNNs on high-resolution images. We followed standard practices in scaling down the images to 256 x 256

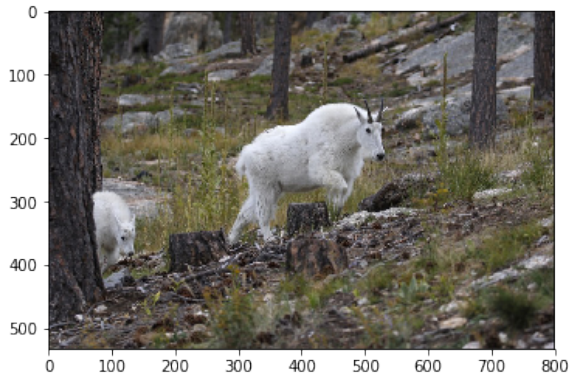

Figure 1: Original Image of a Mountain Goat

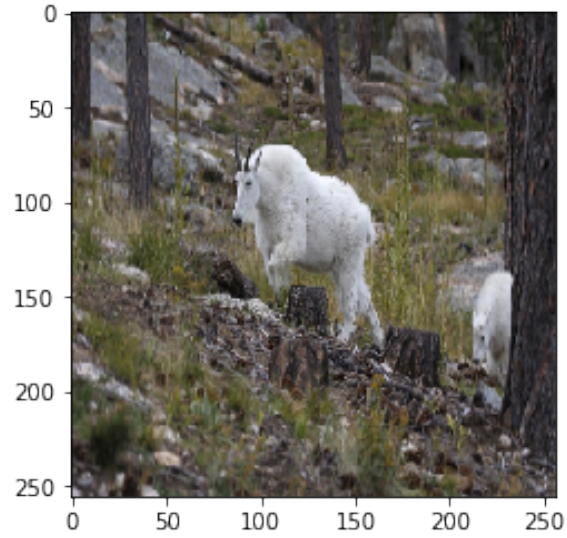

Figure 2: Flipped Image of a Mountain Goat

pixels. The images in the dataset are color images, where each pixel has three values: one for each of the red, green, and blue intensities. We refer to all the values for a specific color as a color channel. After scaling down the images, we computed the mean and standard deviation of pixel intensities for each color channel separately and then we normalized the images by subtracting the average and dividing by the standard deviation. This step is known to make learning easier for neural networks.

4.3 Data Augmentation

We perform random cropping, vertical and horizontal flipping, rotation, sheer enhancement, and contrast modification to each image. Doing so, we provide a slightly different image each time, which can make the network resistant to small changes and improve the accuracy of the network

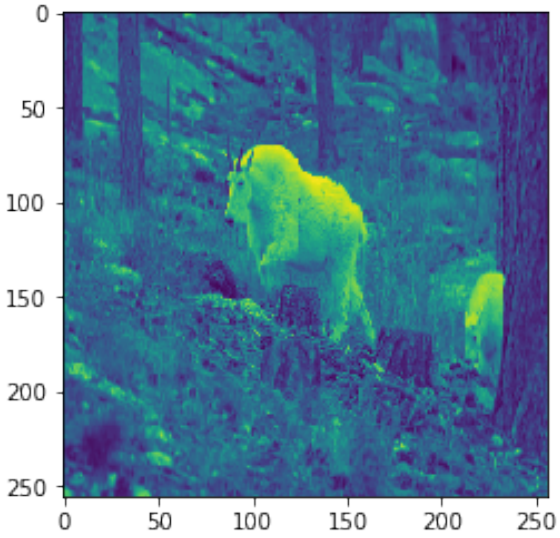

Figure 3: Colorless Flipped Image of a Mountain Goat

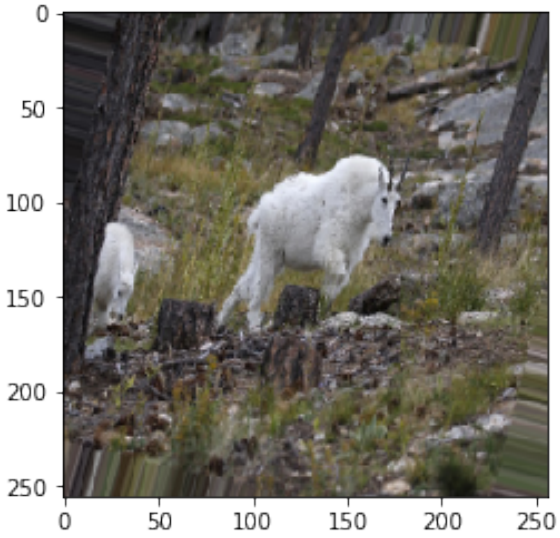

Figure 4: Rotated Image of a Mountain Goat

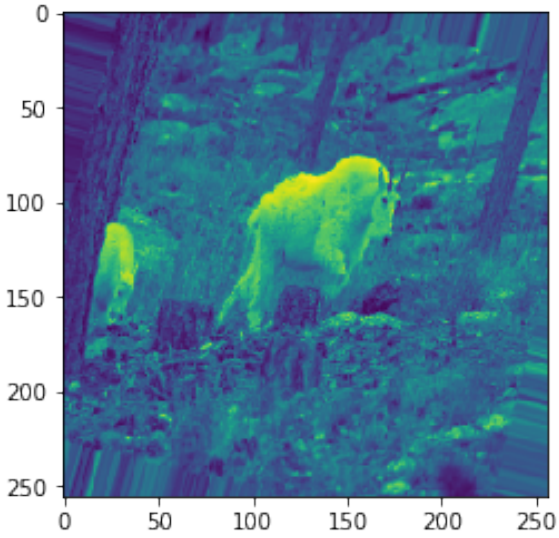

Figure 5: Colorless Rotated Image of a Mountain Goat

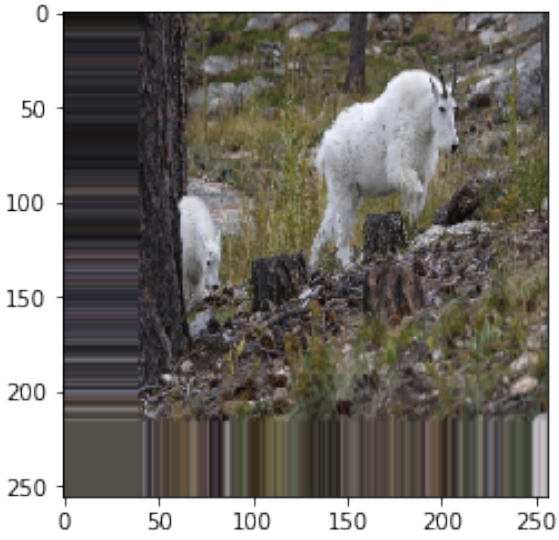

Figure 6: Shifted Image of a Mountain Goat

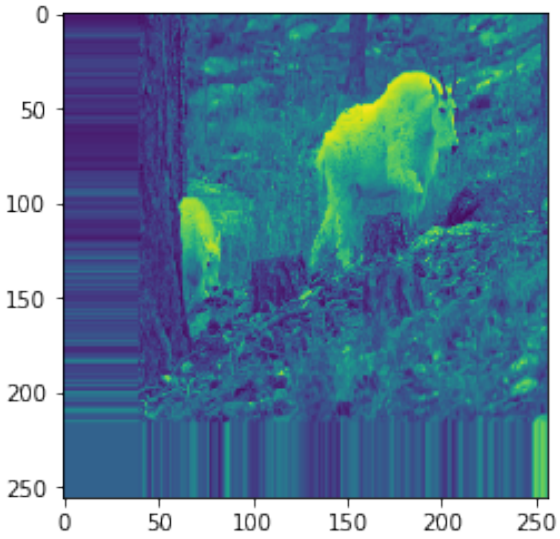

Figure 7: Colorless Shifted Image of a Mountain Goat

Architecture	Accuracy
VGG-16	60.24
VGG-19	58.45
Resnet-18	*
Resnet-50	*
Resnet-101	*

Table 2: Accuracy of different models on Task 2: Identifying Animal Species.

4.3.1 Training

We train the networks via backpropagation using Stochastic Gradient Descent (SGD) optimization with momentum. We used the Keras framework with Tensorflow backend for our experiments. The SGD optimization algorithm requires several hyperparameters. We train models for Task 1 for 20 epochs and models in Task 2 for 53 epochs with the learning-rate decay policy. We checkpoint the model after each epoch and at the end, we report the results of the most accurate model on the test set.

4.4 Task I: Detecting Images That Contain Animals

For this task, our models take an image as input and output two probabilities describing whether the image has an animal or not (i.e. binary classification). We train 5 different neural network models as show in Table 1. Because 75% of the CCT dataset is labeled as empty, to avoid imbalance between the empty and non-empty classes, we take

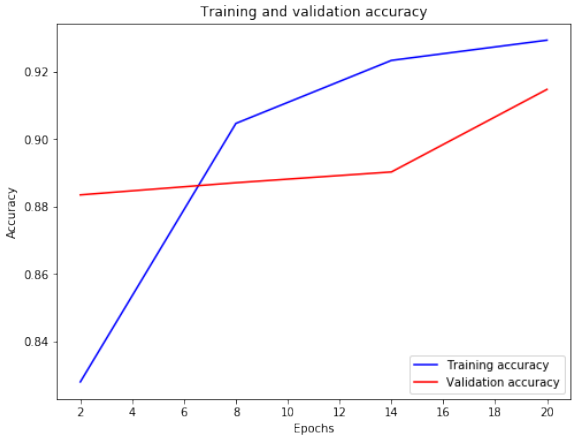

Figure 8: Performance for Task I: Detecting Images that contain animals

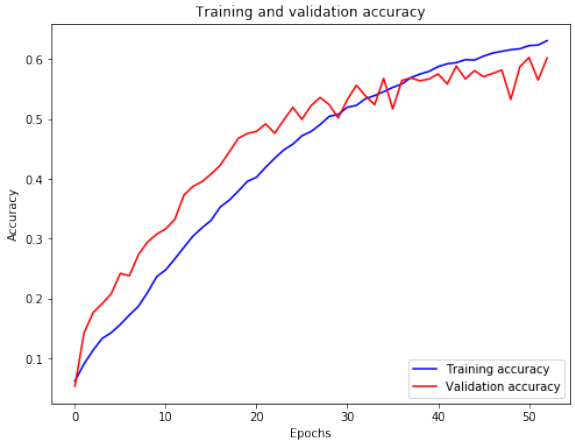

Figure 9: Performance for Task II: Identifying Species

all 131457 non-empty images and randomly remove 25000 "empty" images. This dataset is then split it into training and test sets.

Since the CCT dataset contains labels for only capture events (not individual images), we assign the label of each capture event to all of the images in that event. All the architectures achieve a classification accuracy of over 93% on this task. The VGG model achieved the best accuracy of 93.8% as shown in Table 2.

4.4.1 Task II: Identifying Animal Species

For this task, the corresponding output layer produces the probabilities of the input image being one of the 21 possible species. As is traditional in the field of computer vision, we report top-1 accuracy (is the answer correct?) and top-5 accuracy (is the correct answer in the top-5 guesses by the network?). The latter is helpful in cases where multiple things appear in a picture, even if the ground-truth label in the dataset is only one of them. The top-5 score is also of particular interest in this work because AI can be used to help humans label data faster (as opposed to fully automating the task). In that context, a human can be shown an image and the AIs top-5 guesses. As we will report below, our best techniques identify the correct animal in the top-5 list 79.2% of the time. Providing such a list thus could save humans the effort of finding the correct species name in a list of 21 species over 99% of the time, although human user studies will be required to test that hypothesis.

The best performing model(VGG-16) for Task 2 has 60.24% top-1 and 79.2% top-5 accuracy, while the second best model (VGG-19) obtains 58.45% top-1 and 81.84% top- 5 accuracy.

5 Discussion and Future Work

There are many directions for future work, but here we mention two particularly promising ones.

- Studying the actual time savings and effects on the accuracy of a system hybridizing deep neural networks and teams of human volunteer labelers. Time savings should come from three sources: automatically filtering empty images, accepting automatically extracted information from images for which the network is highly confident in, and by providing human labelers with a sorted list of suggestions from the model so they can quickly select the correct species, counts, and descriptions. However, the actual gains seen in practice need to be quantified. Additionally, the effect of such a hybrid system on human accuracy needs to be studied. Accuracy could be hurt if humans are more likely to accept incorrect suggestions from deep neural networks, but could also be improved if the model suggests information that humans may not have thought to consider.

- Harnessing transfer learning to automate animal identification for camera-trap projects that do not have access to large labeled datasets. The challenge in such cases is how to train a model without access to many labeled images. Transfer learning can help, wherein a deep neural network is trained on

a large, labeled dataset initially and then the knowledge learned is repurposed to classify a different dataset with fewer labeled images. We found that transfer learning between ImageNet and CCT was not helpful, but ImageNet contains many human-made categories and the features learned to classify human-made objects (e.g. computer keyboards or Christmas ornaments) may not help when classifying animals. Previous transfer learning research has shown that it works better the more similar the transfer-from and transfer-to tasks are. Transferring from one animal dataset to another one may prove more fruitful. Experiments need to be conducted to verify the extent to which transfer learning from the SS dataset or others can help automate knowledge extraction from other camera-trap projects with fewer labeled images.

6 Conclusion

In this empirical work, we tested the ability of state-of-the-art deep neural networks to automatically extract information from images in the CCT dataset - camera trap images of wild animals. We first showed that deep neural networks can perform well on the CCT dataset, although performance is worse for rare classes.

Perhaps most importantly, our results show that employing deep learning technology can save a tremendous amount of time for biology researchers and the human volunteers that help them by labeling images. A substantial amount of human labor can be redirected to other important scientific purposes and also makes knowledge extraction feasible for camera-trap projects that cannot recruit large armies of human volunteers. Automating data extraction can thus dramatically reduce the cost to extract valuable information from wild habitats, likely revolutionizing studies of animal behavior, ecosystem dynamics, and wildlife conservation.

Tables 3 and 4 show the baseline (Norouzzadeh et al., 2018) and the accuracies achieved for the two tasks when VGG and Resnet 152 models are used.

Task	Top 1 Accuracy	Top 5 Accuracy
Task 1	(VGG) 96.6%	N.A
Task 2	(Resnet 152) 93.8%	99.10%

Table 3: Baseline Accuracy (Norouzzadeh et al., 2018) for two tasks.

Task	Top 1 Accuracy	Top 5 Accuracy
Task 1	(VGG) 93.8%	N.A
Task 2	(Resnet 152) 60.24%	79.20%

Table 4: Accuracy achieved for two tasks.

References

Sara Beery, Grant Van Horn, and Pietro Perona. 2018. Recognition in terra incognita.

Mohri M, Rostamizadeh A, and Talwalkar A. 2012. Foundations of machine learning.

Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S. Palmer, Craig Packer, and Jeff Clune. 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning.

Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. 2015. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna.

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

LeCun Y, Bengio Y, and Hinton G. 2015. Deep learning. nature.