Categorizing Animals in the Wild

Shanu Vashishtha
UMass Amherst
svashishtha@umass.edu

Abstract

Animal species have succinct visual variations. In the biological classification hierarchy, inter genus variations are starker compared to inter species variations. Wildlife ecologists over the world collect numerous camera trap images in wildlife preserves for the purpose of studying migration patterns and designing effective ecosystems for wildlife human interactions. To extract meaningful information at the moment a forest ecologist has to browse through the collection of images which contains many false positives and filter the interesting ones manually. A computer vision tool that identifies these animal species will reduce hours of manual workload. In this work, we investigate the problem of classifying animal species given a camera trap image.

1. Introduction

Animal species classification is a fine-grained visual classification task. We observe a wide variation in visual characteristics of animals. There are variations in genus (a bear has a much larger body structure than a cat) as well as species (Siberian tigers have paler orange fur than other tiger species and brown instead of black stripes compared to Sumatran or Bengal tiger). These variations make the task of identifying and classifying them in images correctly a major computer vision problem.

In parallel to this growing requirement of a tool within the wildlife community has been the development of neural networks as a classifier. Various architectures of neural networks have achieved remarkable performances on the Imagenet classification task which comprises of 1000 different objects. In comparison, common animal species only have about 30 different variants for classification in a given region although the differences in between classes are fairly tenuous in nature.

One of the drawbacks of these neural network based models however is their inability to generalize to new regions. The models even though are good at learning features to classify animal images from one region fail to perform well when exposed to animal images from a new region. In this work, we explore the direction to make the existing models more robust. The first step to tackle this problem is to collect training data from one region and test them on images from nearby areas where the species found might not completely overlap.

While doing this, we are faced with some new challenges apart from the existing ones which we need the model to tackle. There can be previously unseen species in the test data or other image challenges as well. These include illumination, motion blur, small region of interest, occlusion, perspective changes, different weather conditions, camera malfunctions and other temporal changes manifested in the way backgrounds change across an image at a given camera trap location.

In this work, we explore techniques to overcome these challenges in images and present a comparative study to understand the effect of these techniques on the final model performance. All the experiments are carried out on the iWildCam 2019 dataset. We first describe the dataset in detail followed by the experiments, results and a discussion in the following sections.

2. Dataset

For this challenge, we use the iWildCam 2019 dataset. The dataset statistics are present in Tab.1. We

Split	Images	Locations	Classes
Train	196,157	California South(138)	14
Test	153,730	Idaho(100)	23

Table 1. iWildCam 2019 dataset

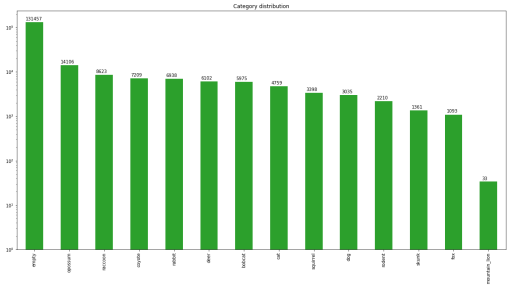

Figure 1. Distribution of classes in the training set

present a distribution of classes of the training set in Fig.1. The dataset is provided as part of a Kaggle competition hosted here - `https://www.kaggle.com/c/iwildcam-2019-fgvc6/overview`.
As can be noticed, the test set has new classes not seen in the training set. Amongst the classes present in the training set, the distribution is highly skewed as well. The final classification labels for the images are one of the following - empty, deer, moose, squirrel, rodent, small_mammal, elk, pronghorn_antelope, rabbit, bighorn_sheep, fox, coyote, black_bear, raccoon, skunk, wolf, bobcat, cat, dog, opossum, bison, mountain_goat, and mountain_lion. Some of the sample images are shown in Fig.2.
Although the original versions of the training data provided are 87 GB and 153GB, a smaller version of these images are provided as well where the image width has been resized to 1024 pixels. These are 27 GB and 18 GB in size respectively and have been used in this work.

3. Related work

One of the beginning points of exploration of this area has been to detect animals in images. As part of the Data Science for common good program [2], we worked on detecting animals from the images captured. However, one of the challenges we faced was to actually find annotated images from the wild. We had access to data from ecologists working at The Nature Conservancy but our investigation pointed out that a lot of the images had incorrect labels. Another challenge we faced during the project was that the images were captured in bursts and so a lot of them were labelled incorrectly because the animal had already cleared the camera field of view by then. The Snapshot Serengeti dataset [8] suffers from the same drawback even though it has a large number of images in its collection.
Existing Natural world datasets such as iNaturalist[3], CUB200[6], LeafSnap [4] have images collected by humans. However, these differ from Camera trap images. Camera traps are fixed at one location and hence the background of the images don't change much. Another important fact is that there is no human bias (such as good lighting conditions) in the images since the camera is motion triggered.
The work by [7] illustrates one of the first applications of neural networks to solve the species identification problem. They describe a two-stage pipeline for the task where at the first step, they solve the animal vs empty task and then in the next stage use the animal images from the first step to classify the animal species present. However, they only experiment with transfer learning and training from scratch on datasets with the same labels in both the training and the test set.
An alternative approach to the two stage pipeline is to use an existing detector that detects the animals in an image. The region proposals extracted can then be fed to a network to classify the animal images. The work by [1] explores such an approach to identify wild gorillas in the Republic of Congo. They perform face detection using a fine-tuned YOLOmodel resulting in a sequence of candidate regions of interest within each image. Each candidate region is then processed up to the pool5layer of the BVLC AlexNet Model for feature extraction. Finally, a linear SVM trained on facial reference images of the gorilla population performs classification of the extracted features to yield a ranked list of individual identification proposals.
Another step in this direction has been taken by Microsoft [5] with them releasing several of their trained detectors and classifiers for the task of identifying species in camera traps. Their Megadetector model performs very well on a wide variety of data. However, the models are not hosted for everyone to play around and can only be evaluated on external data us-

Figure 2. Example images from the iWildCam dataset

ing their provided APIs.

4. Approach

4.1. Baseline

The competition page provides a baseline model which is an Inception Resnet V2 classifier trained with no class rebalancing or weighting with an input size of 299x299. The model is a full image classifier and didn't use a detector. We use this as a starting point to improve our final performance.

4.2. Feature Extraction and Fine tuning

Transfer learning has proven to be effective for many problems. Since a neural network pretrained on the ImageNet dataset has learned useful features, evaluating its performance on a new dataset has become the default approach. We use the Resnet18 model to fine tune the last classification layer initially and then update the weights of all other layers as a first step to improve the baseline.

4.3. Grayscale Images

The most basic approach we looked into was converting the images into grayscale before training. It was hypothesized that since varying background (green vegetation, snowy, arid rocks, day-night variations) was an issue, converting them to grayscale will benefit the classification task.

4.4. Mixup as a regularizer

Mixup is one of the recent techniques proposed to make neural networks learn a more linear model. In this approach, we feed a convex combination of images with their one hot vector encodings to the model for training. The idea is that this acts as a regularizer during training and helps tackle the generalization problem in deep models. Our modified input becomes

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

where $\lambda \in [0, 1]$ and $(x_i, y_i), (x_j, y_j)$ are sampled randomly from the training data.

4.5. Using Bounding box proposals

Techniques described till now focused on improving the standalone classification model. We also experiment with using a detector to obtain the bounding box proposals. The challenge provided the top 100 boxes and associated confidences for the images in the dataset using a Faster-RCNN model with Inception-Resnet-v2 backbone and atrous convolution. We incorporate these into our model as well.

4.6. Focal loss

Training a model with bounding box proposals gives many easy examples and we penalise the training process to make the model focus on hard examples using this modified loss function.

Figure 3. Some failure images from the finetuning experiment. The first label is the prediction from the model performing feature extraction while the second label is the prediction after the finetuning process

Figure 4. Some failure images from the grayscale experiment

5. Experiments and Discussion

The evaluation metric we follow is the F1 score. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0 [9]. Thus, our aim is to maximize the F1 score in our experiments. Since F1 is more relevant for binary classification task, in this work we report the macro F1 score. To calculate the macro F1 score, F1 score will be calculated for each class of animal (including "empty" if no animal is present), and the final score will be the unweighted mean of all class F1 scores.

For fine-tuning and feature extraction, we resize our images to 224*224 width and height. We also normalize across the three channels with the mean values. While conducting the grayscale experiment, we modify the first convolution layer of Resnet18 as we have only one input channel. We train the model using Adam optimizer with an initial learning rate of 0.01 and use cross-entropy loss. We split our training set into 80:20 to create a validation set to obtain the best hyperparameters. We perform the same augmentations to the test set images as well. The results obtained for our experiments are presented in Tab.2.

After feature-extraction, we obtain an F1 score of 0.117 on the public portion of test data and 0.097 on the private portion of test data (68% of the total). Upon fine-tuning, we obtain a F1 score of 0.123 on the public portion while 0.086 on the private portion. This was striking since although the model performed better on one portion of the test set, it didn't do well on the entire test set. Some of the failure images are present in Fig.3. The most common mis-prediction was for the empty case in the images. We felt that the standalone classifier failed at the task of detecting the animals and hence, we decided to use the bounding-box proposals in our later experiments.

For the Grayscale images, we finetuned the entire network on grayscale images. With this experiment, the aim was to remove the effect of day/night images. Our experiment yielded a public F1 score of 0.109 and a private F1 score of 0.087. The performance degrades compared to our finetuning with normal images. This implies that finetuning with grayscale images wasn't that helpful. However, as a way to incorporate some robustness into our model we can randomly grayscale some images instead of using all grayscale images.

For the Mixup experiment, we obtained a score of 0.109 on the public test data and 0.099 on the private test data. This is still below the baseline we are trying to improve upon but is a small improvement over our grayscale experiment.

For the final experiment, we used the bounding box proposals to train our model and combined all the previous techniques. We used a mixup with $\lambda = 0.1$, grayscale probability of 0.01 and passed on the region proposals to the model. We used the provided bounding boxes for each of the images. These are used as labels to train for the classification task. This provided a score of 0.179 on the public set and 0.169 on the private set. As can be seen, incorporating the proposals led to a massive improvement in the final performance of the model. This is intuitive because we are focusing the model to become good at one task instead of two sub-tasks. This technique is similar to the one proposed in [6] with the only difference here being that we don't just focus on the bounding box proposals for faces.

Model	public F1	private F1
Baseline	0.125	0.115
Feature extraction	0.117	0.097
Fine-tuning	0.123	0.086
Grayscale	0.109	0.087
Mixup	0.109	0.099
Bounding box	0.179	0.169

Table 2. F1 scores obtained for different experiments

5.1. Kaggle leaderboard

Although our best model was able to beat the baseline provided for the challenge, it featured amongst the top 20 only. The best entry obtained a score of 0.361 and 0.399 on the public set and private set respectively. While browsing through some of the submissions made by other competitors who featured in top 10, the author came across many new ideas which have been listed in Sec. 7.1.

6. Failures

We implemented focal loss in our code but the model didn't converge when we tried to train using that. We couldn't figure out if there was a bug in our implementation or something else during training and hence, don't have results for that experiment.

7. Conclusion

In this work, we wanted to explore the task of classifying animal species. We utilized an existing dataset namely iWildCam 2019 where we have labelled images provided to us. The current literature pointed out that we have good systems in place to classify images in one region. However, they perform poorly when we deploy those systems in new regions. The provided dataset is exactly meant to study this problem in detail where the training set and test set don't have the same class labels.

We started off with simple image manipulation ideas. The first experiment was just finetuning existing architecture and using grayscale images to tackle some obvious image differences. However, those didn't seem to work very well for our problem.

To make our model more robust, we experimented with mixup as a regulatization. Although we noticed some improvement with using regularization, the performance gain wasn't significant when we tested it compared to our previous results.

Finally, we decided to incorporate a detector's output into our model since our results showed that the mispredictions were for images where no animals were present. We used the provided detector outputs and were able to obtain our best results with this modification.

The results were better than the baseline model provided on the Kaggle leaderboard and retrospectively could be placed amongst the top 20 submissions.

7.1. Future Explorations

The following unexplored ideas came up during the course of the project and will be explored later on -

- Incorporating zero shot learning as a technique to handle the classes which are missing from the training set but are present in the test set
- Use Synthetic data generated by Microsoft AirSim for the missing classes and experiment its effect on the final accuracy
- Using ResNeXt and EfficientNet architecture for the task which is the state of the art model on the ImageNet benchmark at the moment
- Using an ensemble of trained models to improve the classification accuracy

The code for the work can be found here - `https://drive.google.com/open?id=1nEKwhqx8b-xRsU0q3cAGqC1s1TVUjJMj`

References

[1] C.-A. Brust, T. Burghardt, M. Groenenberg, C. Kading, H. S. Kuhl, M. L. Manguette, and J. Denzler. Towards automated visual monitoring of individual gorillas in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2820–2830, 2017.

[2] Center for Data Science, UMass. Data science for common good. `http://ds.cs.umass.edu/industry/data-science-common-good`, 2019. [Online; accessed 19-December-2019].

[3] G. V. Horn, O. Mac Aodha, Y. Song, A. Shepard, H. Adam, P. Perona, and S. J. Belongie. The inaturalist challenge 2017 dataset. *CoRR*, abs/1707.06642, 2017.

[4] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*, 2012.

[5] Microsoft. Ai for earth. `https://github.com/microsoft/CameraTraps`, 2019.

[6] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, C. Packer, and J. Clune. Automatically identifying wild animals in camera trap images with deep learning. *CoRR*, abs/1703.05830, 2017.

[7] 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*, 115(25):E5716–E5725, 2018.

[8] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer. Data from: Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna, 2015.

[9] Wikipedia contributors. F1 score — Wikipedia, the free encyclopedia, 2019. [Online; accessed 19-December-2019].