CS230

Counting the Number of Wild Animals in a Sequence of Images Recorded by Camera Trap

Adrian Ma*
ama5@stanford.edu

Abstract

Camera traps facilitate the automatic collection of images of wild animals that allow ecologists and conservation biologists to monitor biodiversity and to estimate the population density of wildlife. The goal of this project is to automate the task of counting the numbers of wild animals in sequences of images recorded by camera traps. This report compares the performance of six model specifications, with LSTM being the most promising candidate. Since the models are trained on a small training set of 1,424 labelled sequences, semi-supervised learning offers a potential solution for unleashing the power of LSTM.

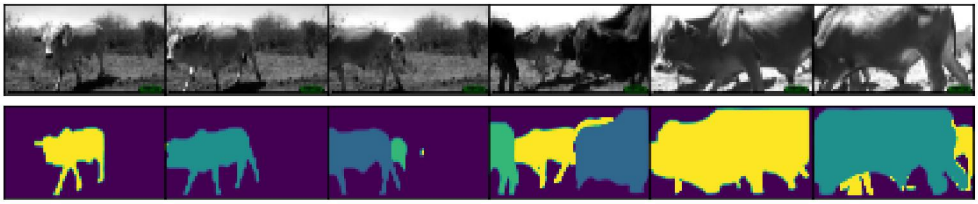

Figure 1: An annotated sequence with nine cows in the herd spreading across six images, along with the corresponding instance masks produced by MegaDetector v4 and DeepMAC.

1 Introduction

Camera traps are equipped with heat or motion detectors that facilitate the automatic collection of sequences of images of wild animals. The vast amount of image data allow ecologists and conservation biologists to monitor biodiversity and to estimate the population density of wildlife.

Once motion is detected, a camera trap takes a number of images, generally between 1 and 10 frames that are at least 1 second apart. As illustrated in Figure 1, there could be temporal discontinuities as animals move in and out of the camera's field of view. Because of the discontinuities, traditional tracking methods are not appropriate for counting the number of animals in a sequence of images.

In addition, images recorded by camera traps present many challenges that need to be overcome in order to achieve accurate results. Images can be poorly illuminated, especially at night or under poor weather conditions. Fast-moving animals may appear blurry. Small animals and natural camouflage present small regions of interest that are hard to spot. Animals occasionally come very close to the camera so that only some body parts are recorded in images. Cameras and detectors may malfunction or may be triggered by non-animal movement, such as wind or vehicles. These factors may compromise image quality and complicate the task of counting the number of animals.

*The author thanks Kevin Yu, Elaine Sui and Grace Lam for providing constructive comments.

CS230: Deep Learning, Spring 2022, Stanford University, CA. (LaTeX template borrowed from NIPS 2017.)

Deep learning has been successfully applied to analyze the vast amount of image data recorded by camera traps around the world [1]. Since 2018, the workshop on Fine-Grained Visual Categorization (FGVC) has been hosting an annual iWildCam competition focusing on various aspects of analyzing camera trap image data. This year, the iWildCam 2022 competition is the fifth annual camera trap challenge that focuses entirely on counting the number of animals.

The purpose of this report is to compare the performance of various model specifications, which will provide the foundation for future work. The rest of this report is organized as follows. Section 2 describes the data source and image processing using MegaDetector v4 and DeepMAC. Section 3 discusses the considerations behind the model architecture design choices. Section 4 provides the details on model specifications. Section 5 presents the model performance results. Section 6 concludes with a discussion of the directions for future research.

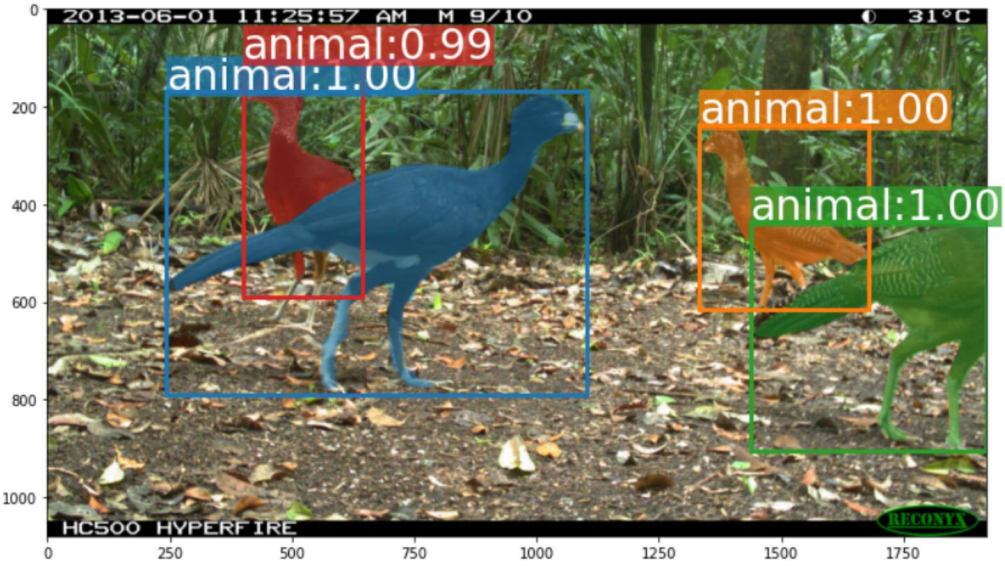

Figure 2: Bounding boxes produced by MegaDetector v4.

2 Data

The dataset is prepared for the iWildCam 2022 competition [2] using images provided by the Wildlife Conservation Society (WCS). The full dataset contains 47,320 sequences consisting of 261,428 images. However, count annotations on only 1,780 sequences are available. These annotated sequences are divided into 80% training and 20% test set. This paper focuses on the labelled sequences only. Further extensions will consider semi-supervised learning to incorporate unlabelled sequences.

The images are processed using MegaDetector v4 and DeepMAC as the building blocks for animal detection. Microsoft AI for Earth MegaDetector provides an open-source animal detection model that has been used for wildlife monitoring by over 30 organizations worldwide such as the Wildlife Conservation Society [3]. It provides a general and robust camera trap detection model. MegaDetector v3 detects animal and person classes, while MegaDetector v4 adds a vehicle class. DeepMAC, which is short for Deep Mask-Heads above CenterNet [4], is designed to produce accurate instance segmentation masks for unseen classes. When combined with detections from MegaDetector, the model is able to generate instance mask for each detected animal, as illustrated in Figure 1.

As illustrated in Figure 2, MegaDetector and DeepMAC output the locations of the bounding boxes and confidence levels, which can be vectorized for each image in the dataset. These vectors become the inputs to the sequence models.

3 Model Architecture Design Choices

The model architecture is guided by the following considerations regarding the dataset and use case:

- Given the small size of the labelled dataset, model parsimony is an important consideration. While some initial model specifications explore the application of convolution layers to instance masks, these networks are too big for a training set of 1,424 labelled sequences. Thus, the models in this report take the bounding box coordinates and confidence as inputs. Moreover, the bounding boxes provide sufficient information for the purpose of counting the number of animals as the detailed features in the instance masks are not necessary for counting.
- To mitigate over-fitting, dropout layer is applied in all model specifications. Moreover, early stopping in training is exercised. Batch normalization is not used because bounding box coordinates and confidence are expressed as floating point numbers between 0 and 1. The neural networks in this project are quite shallow so that internal covariate shift is not observed.
- The number of animals in the dataset is an integer between 0 and 10. The problem could be formulated as multi-class classification. Because some unlabelled sequences contain more than 20 animals, modelling the target variable as an integer presents a more robust formulation that can be generalized to the unlabelled dataset. The internal outputs are floating point numbers, and the final model predictions are rounded to integers. However, rounding is not applied in training because the gradient of the loss function would be flat on rounded numbers.
- The mean squared error (MSE) is used as the loss function as it punishes large deviations from the annotated counts. As shown in Figure 3, the counts skews towards small numbers with few sequences of large numbers of animals. The mean absolute error (MAE) serves as the metric for evaluation in the iWildCam 2022 competition. This is linear in the difference between prediction and ground truth. Nevertheless, MSE is used as the loss function as it puts more weights on large deviations than MAE loss to counteract the skewness in the dataset.

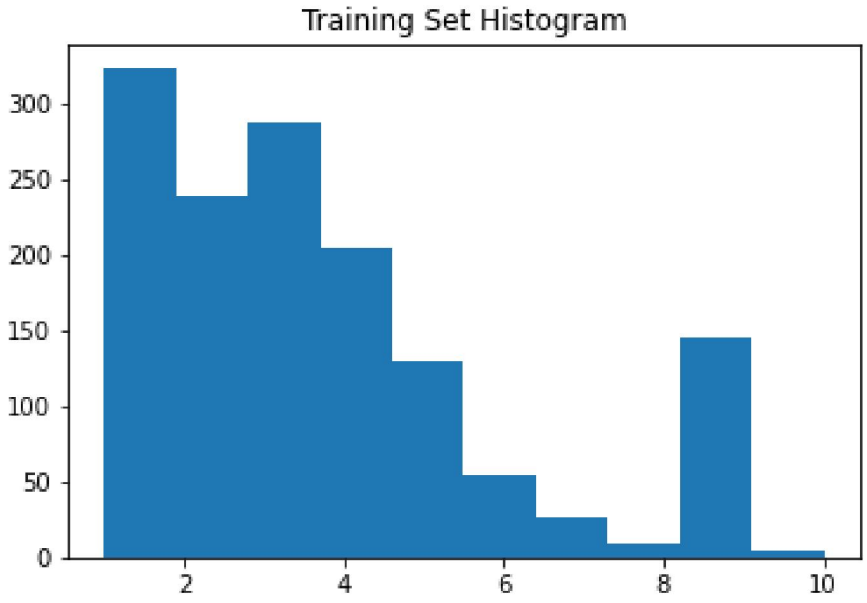

Figure 3: Histogram of the Number of Animals in the Training Set.

4 Model Specifications

This section discusses the six model specifications in this experiment. Further details regarding the models can be found in Figure 4.

Model	Linear	RNN	RNN mixed	LSTM	LSTM mixed
# Params	1041	706	942	1741	1612
Layers	Flatten	RNN(250)	RNN(250)	LSTM(1000)	LSTM(1000)
	Dropout	Dropout	Dropout	Dropout	Dropout
	Linear(20,50)	Linear(30,10)	Linear(5,10)	Linear(30,10)	Linear(30,5)
	ReLU	ReLU	ReLU	ReLU	ReLU
	Linear(1,20)	Linear(1,30)	Linear(1,5)	Linear(1,30)	Linear(1,30)
			Flatten		Flatten
			Dropout		Dropout
			Linear(5,10)		Linear(5,50)
			ReLU		ReLU
			Linear(1,5)		Linear(1,5)

Figure 4: Model Specifications.

4.1 Baseline Model

The first-place solution to the iWildCam 2021 competition constitutes an appropriate benchmark [5] that is taken to be the starting point of this project. The algorithm uses MegaDetector v4 [6] to generate bounding boxes. The maximum number of bounding boxes with confidence at least 0.8 in any image of a sequence is taken to be the prediction of the number of animals.

This approach analyzes the images separately and then takes the maximum number of animals. The drawback is that it ignores the sequential nature of the images in a sequence. Taking the maximum number of bounding boxes works well for animals that stay within the camera's field of view throughout the sequence. However, if a herd of animals move in front of the camera, the full herd may not appear in any single image. As illustrated in Figure 1, the herd of nine cows can only be spotted across six images. It is important to note that the images in a sequence are not necessarily evenly spaced in time. For instance, the sequence in Figure 1 lasts for 57 seconds, with a 53-second time lapse between the third and fourth images. This time lapse allows the first group of cows to leave and the second group of cows to move in the camera's field of view. Hence, the maximum underestimates the total number of animals in this sequence. In such cases, sequence model is more appropriate than taking the maximum. These considerations motivate the following sequence model specifications.

4.2 Sequence Models

Using the MegaDetector, a sequence of images is transformed into a sequence of tensors. Each tensor contains the bounding box coordinates and confidence of the detected animals in an image. The numbers of images in the labelled sequences range from 1 to 10. For each sequence, the top five images with the most detected animals are included. This approach is preferred to padding all sequences to 10 images as the number of tensors would be doubled compared to the current approach. Recurrent neural network (RNN) and long short term memory (LSTM) take the tensors as inputs and predicted the counts in a many-to-one architecture.

4.3 Linear and Mixed Models

In addition, a linear model specification stacks two fully connected layers with a ReLU activation layer in between. Two mixed model specifications combine the baseline model with LSTM and RNN.

The model predictions are weighted averages of the baseline model and a sequence model. The weights are generated by a two-layer full connected layers with a ReLU activation layer. The purpose of these two mixed models is to explore ways to combine the baseline model with sequence models.

5 Results

The key metrics are accuracy and MAE. Accuracy is defined as the percentage of count annotations that are correctly predicted by the model. MAE measures the average absolute value of the difference between the predicted and actual numbers of animals in a sequence. The results are presented in Table 1.

LSTM has the highest accuracy in training. However, its accuracy drops significantly for the test set, which is a clear indication of over-fitting. The number of parameters in LSTM is quite large compared to the number of training sample. Hence, even the inclusion of a dropout layer and early stopping do not address the issue.

	Accuracy		MAE	
Model Specification	Train	Test	Train	Test
Baseline	53%	52%	0.90	0.99
Linear	42%	44%	0.92	0.99
RNN	43%	38%	0.78	1.21
RNN mixed model	53%	52%	0.90	0.99
LSTM	75%	46%	0.28	1.28
LSTM mixed model	57%	47%	0.57	1.19

Table 1: Model accuracy and mean absolute error (MAE).

6 Future Work

This report is only the starting point. LSTM is a promising model specification but suffers from over-fitting despite the inclusion of a dropout layer. The most likely reason is the small training set. Semi-supervised learning offers a potential solution for fully unleashing the power of LSTM.

Bibliography

[1] Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W Mathis, Frank van Langevelde, Tilo Burghardt, et al. Seeing biodiversity: perspectives in machine learning for wildlife conservation. *arXiv preprint arXiv:2110.12951*, 2021.

[2] iWildCam 2022. `https://github.com/visipedia/iwildcam_comp`, 2022.

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

[4] Vighnesh Birodkar, Zhichao Lu, Siyang Li, Vivek Rathod, and Jonathan Huang. The surprising impact of mask-head architecture on novel class segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7015–7025, 2021.

[5] Fagner Cunha. First place solution to iwildcam 2021: Count the number of animals of each species present in a sequence of images. `https://github.com/alcunha/iwildcam2021ufam`, 2021.

[6] Microsoft AI for Earth. MegaDetector. `https://github.com/microsoft/CameraTraps/blob/main/megadetector.md`, 2022.