Assignment 3 - Object Recognition and Computer Vision

Paul Jacob
Ecole Polytechnique - Master MVA
paul.jacob@polytechnique.edu

Abstract

This assignment consists in a Kaggle competition with the rest of the class. The objective is to produce a model that classifies birds of the test dataset with the highest possible accuracy. Here I present the different steps of my method.

1. Introduction

The main challenge of this classification contest is the small amount of data examples that we have to train our network. Thus, an efficient approach should include efficient preprocessing of the images, data augmentation, and transfer learning. I explain how I combine those elements to achieve a 83% accuracy score on the public Kaggle leaderboard.

2. Data Preprocessing & Augmentation

Looking at the images in the dataset, one can realize that not all birds are displayed with the same visibility. This issue invites us to look for a way to detect the birds and crop the images accordingly. Thus, I use a *Mask R-CNN* model with a *ResNet-50-FPN* backbone [1], pre-trained on COCO train2017.

Using this model, I detect the birds in each image and crop around the most probable bird bounding box (with the highest confidence). For data augmentation purposes, I choose to crop at 3 different scale levels: with 10%, 20% and 50% margins around the bounding box. Only one image of the test set does not have a bird detected in it: I choose to ignore it. When looking manually at the cropped images, the results seem pretty accurate on bird detection. The rest of the work is done on those cropped images.

To compensate the small amount of images, I apply a large data transformation pipeline before feeding the images to the network: I resize them to size $(224, 224)$ or $(299, 299)$ when using *Inception v3*. Then, I apply hue and saturation variations, horizontal flips with probability $p = 0.25$, perspective transformations, small random rotations and a color normalization.

3. Classification Models

I use different pretrained models on ImageNet: *Resnet-101*, *Resnet-152*, *Inception v3*, *Densenet-201* and *ResNeXt-101-32x8d* [2, 4, 5, 3]. To avoid overfitting, I freeze most of their deep layers, and only train the high level ones, such as the fully connected classifier and the last convolutional layers before it. I chose *Adam* for the optimizer, with a learning rate of 0.0001, and use a batch size of 128. Everytime the validation loss stops decreasing for 3 epochs, I divide the learning rate by 2. I train on 50 epochs and keep the weights after the epoch with minimum validation loss for evaluation time.

When training and evaluating these models on the raw images without cropping around the birds, the validation accuracy never exceeds 87%. However, on the cropped images, the validation accuracy reaches 90% for most of the trainings and can go up to 95%, which is why I decided to keep working on this cropped dataset.

4. Majority voting

Models detailed in section 3 led to pretty good and similar accuracies on the validation set (between 90% and 95%). To further enhance the results and increase robustness at test time, I started different trainings and store the test predictions of the models with a validation accuracy over 93%. I apply a hard majority voting process to only keep the most predicted label for each bird. Uploading those predictions on Kaggle led to a public score of 83% (*Resnet-152* alone gave 80%).

5. Conclusion

My approach on this challenge combines image preprocessing, data augmentation, transfer learning and majority voting. It led to a Kaggle public accuracy of 83%, which is the $14th$ score on the public leaderboard. My code is attached as a Jupyter Notebook and runs on Google Colab.

To further enhance the results, one could try to retrieve more specific features about birds using unsupervised frameworks (e.g. autoencoders or clustering) on large bird datasets.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1

[3] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[5] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016. 1