Assignment 3 - Bird Species Classification

Hippolyte MAYARD
ENS Paris-Saclay
hippolyte.mayard@ens-paris-saclay.fr

Abstract

In this project, we aim to producing a model that gives the highest possible accuracy at classifying Bird species from the Caltech-UCSD Birds-200-2011 dataset. In this report, we discuss the different methods that have been used to obtain the best accuracy score.

1. Introduction

The provided dataset is composed of both a training set of 1082 images and a validation set of 103 images, allocated to their respective bird species. The training set is well balanced, while the validation is almost well balanced. The test set, on which the model is evaluated, is composed of 517 images that we have to classify. In the following section, we describe the different techniques we used to obtain the best accuracy on the test set.

1.1. Data augmentation

In order to leverage our training, we tried many different data augmentation scenario on the training set. To find the best augmentation scenario, we used the transformation provided by the Pytorch Data Loader: RandomAffine transformations (rotation, translation, scale and shear), ColorJitter transformations (in order to change the brightness, the contrast and the saturation), RandomHorizontalFlip and RandomVerticalFlip. Also, we used the ImgAug external library [1] that provides many interesting transformation such as AdditiveGaussianNoise, GaussianBlur or CoarseDropout.

1.2. Using Fast R-CNN to crop the images

The different models that we trained on the original data led to bad results. This phenomenon could be explain by the fact that the images very noisy and not centered on the birds. To avoid this problem, we performed object detection in order to extract the part of the image that is interesting in our classification problem. We used a pre-trained Faster-RCNN InceptionResNetv2 from TF-Hub [4] in order to localize which part of the image contains a bird and extracted the part that had the higher score. The code that has been used to achieve this task as been adapted from [5].

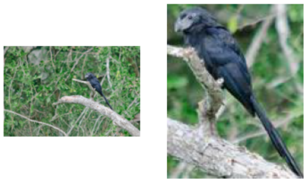

Figure 1. Left: original image - Right: image after RCNN localization and crop

1.3. Transfer Learning and Training

Regarding the architectures, several combinations of different architecture have been used. The architectures that performed the best accuracy on the test set are the following: InceptionResnetv2 (imported from ModelZoo [2], Resnext101_32x8d and Resnet152 (both imported from Pytorch [3]), all pre-trained on ImageNet.
All of these 3 models were trained on the original data and the task 1 resulting dataset that is center on the birds. We used fine-tuning, that is by unfreezing some layers of the last convolutional blocks, and a SGD with momentum as optimizer with a learning rate decay (further details are given in the implementation).
At the end of the training, each of the model has been evaluated on the original dataset and the cropped dataset. Finally, the prediction of an image is decided by a vote accross the 6 previous results.

1.4. Conclusion

The different methods that have been described in this report allowed us to obtain a 0.81290 accuracy on the Kaggle public score. However, this accuracy could be increased by using other data augmentation techniques. We tried to train a Conditionnal Wasserstein GAN to create images related to a specific label, but the results were not satisfying enough.

References

[1] https://imgaug.readthedocs.io/en/latest/index.html. 1

[2] https://modelzoo.co/model/pytorch-cnn-finetune. 1

[3] https://pytorch.org/docs/stable/torchvision/models.html. 1

[4] https://tfhub.dev/. 1

[5] https://www.tensorflow.org/hub/tutorials/object$_detection.$ 1