Bird image classification competition

Fatma Moalla
Ecole Normale Supérieure & CentraleSupélec
fatma.moalla@student.ecp.fr

Abstract

Bird image classification competition is a Kaggle challenge based on Caltech-UCSD Birds-200-2011 dataset that aims to predict the class of the birds within a small subset of the original dataset. The challenge is organized for Computer Vision (Recvis19) course in the 2019 MVA Master.

1. Introduction

This paper presents the methods we used to detect each bird and classify it into one of the 20 species of the subset used for the competition. The main goal is to get the highest accuracy score possible on the mystery test set.

2. Challenging Dataset

Train, validation and test sets present multiple bird images, however, the majority can be ambiguous or misleading. In fact the test set presents hidden (Fig.1) or vey small birds, a hand holding the bird, bird catching a snake (Fig.2). To tackle this ambiguity problem, first we used basic transformation techniques (Greyscaling, flipping...) and applied them to the training set. Second, we cropped testing images by using bounding boxes resulted from models pretrained on **PASCAL VOC dataset** in order to detect the birds.

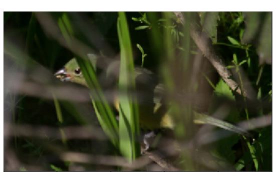

Figure 1. Image from test set: Hidden bird

3. Bird Detection Model

The idea of this technique, inspired by Y.Cui paper[4], is to detect, first, the position of the bird by using the bounding boxes generated either by **FasterRCNN** using VGG16, **YOLOv2** or **YOLOv3** pretrained on the **PASCAL VOC dataset** that contains the class 'bird'. Then compare these algorithms and select the bounding box (Bbox) that detects the bird with the highest score. After, we crop the images based on this chosen bounding box (Fig.2 & Fig.3).

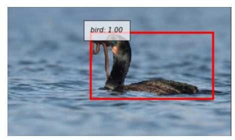

Figure 2. Bbox by YOLOv3

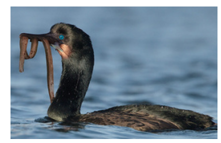

Figure 3. Cropped image

4. Bird Classification Model

Once the cropped images obtained (here only on test set, training is uncropped), we perform a classification by bench-marking several CNNs using Pytorch[2]. The selected CNNs were **Resnet101**[1] and **Resnext**[3]. We also used the pretrained version of these two algorithms on the **1000-class Imagenet dataset**. We trained these models, using **50 epochs**, a decreasing learning rate every **7 epochs** (by 0.1) and a Stochastic Gradient Descent Optimizer.

5. Results

Accuracy on	Train	Val	Test(30%)
Resnet101	0.6460	0.8544	0.7677
Res101(crop_test)	0.6460	0.8544	0.7290
Resnext	0.6327	0.8432	0.7677

Resnext and **Resnet101** produce similar results. However, test accuracy of cropped dataset is lower, meaning that we should also crop the training set to improve the model.

References

[1] K.He and X.Zhang. Deep residual learning for image recognition. *CVPR*, 2015.

[2] N.Inkawhich. Finetuning torchvision models. `https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html`, 2018.

[3] S.Xie, P.Dollár, and K.He. Aggregated residual transformations for deep neural networks. *CVPR*, 2017.

[4] Y.Cui and S.Belongie. Large scale fine-grained categorization and domain-specific transfer learning. *CVPR*, 2018.