Object recognition and computer vision - TP3

Samuel DIAI
Ecole nationale des ponts et chaussées
diai.samuel@gmail.com

Abstract

This document details the methods and results I had by performing the classification of 20 classes of birds for the course "Computer Vision and object recognition". I achieved an accuracy of 0.8129 *on the public leaderboard.*

1. Introduction

The dataset we were supplied with is a subset of the original Caltech-UCSD Birds-200-2011 dataset, reduced to 20 bird classes, instead of the original 200 bird species. There are 1082 training images and 103 validation images.

2. Preprocessing

2.1. Segmentation

The bird pictures have different shapes and within a given species, birds exhibit varying sizes and poses. The main issue I faced was occlusion by twigs or leaves. It was sometimes hard (even for a human) to detect a bird. While birds are always in the focus of the picture, there may be larger objects also in the foreground, such as the trunk of a tree. This motivated me to do segmentation of each image.

For that purpose, I used a deep learning segmentation algorithm : a pre-trained faster R-CNN [1] algorithm, trained on the COCO image dataset. The COCO dataset contains a bird class, so it was easy to crop birds, with this pre-trained model. For each picture, I generated the bounding boxes corresponding to the best confidence value for the bird class. Only a very small number of images were not segmented (0.9% of the train + validation dataset.).

2.2. Re-sampling

I noticed that accuracy achieved was much higher on validation set than on test set. I guessed the initial validation set was not general enough and quite simple. So, I concatenated the training set and the validation set and then I re-sampled it randomly into two new sets, with a split (train/val) percentage of 0.85.

3. Model

I used several state of the art pre-trained (on ImageNet) classification algorithm (in the top 10 benchmark for ImageNet):

- Resnext [3](101_32x8d or 101_32x32d versions)
- EfficientNet [2](b4 and b5 versions)

For each of them, I replaced the final fully connected layer with a Linear layer of shape (input, 512), ReLU, batch normalization 1D, and a final linear layer of shape (512, 20). I started by tuning only the last layer with the other weights frozen. But I noticed better scores when I unfroze all layers after some epochs.

4. Pseudo Labeling

As we were allowed in the rules to implement methods using other bird datasets but without any labels, I decided to implement a pseudo labeling algorithm on the iNaturalist dataset(restricted to the 20 bird classes). This algorithm consists in passing all unlabeled samples through a pre-trained model on the labeled data. Once the new labels are obtained, I re-train my model on the labeled and pseudo-labelled data.

5. Results

Table below shows the accuracy obtained by my models.

	Training	Validation	Kaggle
Resnext	98%	93%	80%
EfficientNet	98%	94%	79%
Resnext+Pseudo Labeling	**97%**	**92%**	**81.2%**
EfficientNet+Pseudo Labeling	97%	93%	79%

Table 1: Accuracy of the models

Pseudo labeling did not improve considerably the score. I guess the pseudo-labeled images were not very complex and did not necessarily help the algorithm to detect bird species on very complex images (trees, low contrast etc...).

References

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 1

[2] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 1

[3] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017. 1