Object recognition and computer vision: Assignment 3
MVA Fall 2019

Antoine Moulin
ENS Paris-Saclay
antoine.moulin@ens-paris-saclay.fr

Abstract

The Caltech-UCSD Birds-200-2011 dataset is usually used for fine-grained image classification. Unlike the ImageNet dataset, it has a large intra-class variance and a small inter-class variance, which makes it challenging for classification. The goal of this assignment is to produce a model that gives the highest possible accuracy on the test set. The first part deals with the preprocessing and augmentation on the dataset, while the second one is about pre-trained models and ensemble methods.

1. Dataset

Here, only a subset that contains 1732 images (1185 for training and 517 for testing) and 20 different classes is considered.

1.1. Preprocessing

When looking at the images, one can notice that in some cases, the bird occupies only a small part of the image. Hence, besides normalizing and resizing images (to 224×224 or 320×320 in my case, a bigger size seems better but it is heavier), it seems to be a good idea to **crop the images so that only the bird is preserved**. For this purpose, as it is not possible to use the provided boxes, one can use a pre-trained Mask RCNN to find them (which yields better performance than single-pass networks such as YOLO). Out of the 1732 images only around 20 raised an issue (non-detection or bad-detection), which I cropped manually.

Another idea is to **pad the images to make them square**, so the birds' aspect is preserved when resizing. However, this did not yield much better results.

Finally, the last idea is to apply an **adaptive histogram equalization** so to make the images with same lightning condition and to avoid over-brightness that could arise from the usual histogram equalization. But I did not have the time to test this idea.

1.2. Data augmentation

For the data augmentation, I have used: **random horizontal flip**, **random rotation** (20 degrees) and a **gaussian blur**. I also added a **coarse dropout** to try simulating objects that could hide part of the bird (e.g. branches, leaves). The two last did not seem to have a huge impact.

2. Approaches

2.1. Fine-tuning

Using pre-trained models (usually trained on ImageNet) is a good way to yield good performance. Indeed, most of the images share features, e.g. the first layers of a network can detect edges, which are present in most of the images. Hence, using a pre-trained model can be seen as a "good initialization". The first network that I used was ResNet50, to which I added different layers at the end: one or two FC layers, one or two Dropout layers (e.g. with a rate of 0.9 for the first and 0.5 for the second, to avoid overfitting). The difference between one or two FC layers was not large, even though it seems better with two FC layers.

At first, I had two training phases: one during which only the last layers are trained and a second one during which the last block of the network is unfreezed so it can be trained as well. The accuracy obtained was around 0.7, but I obtained better results by unfreezing the whole network and using two different learning rates during the optimization (with Adam optimizer), e.g. 10^{-4} for the last layers, and 10^{-5} for the pre-trained network.

A good thing would have been to use models pre-trained on iNaturalist instead of ImageNet, as it is much closer to our dataset than ImageNet.

2.2. Ensemble methods

As I trained several models (e.g. with different backbone network, image size or augmentation), I tried bagging as well as a voting rule. For this purpose, I split the labeled dataset into five folds and trained five models with a different validation set. This is how I got 0.819 accuracy.