Object Recognition and Computer Vision: Assignment 3

Bouatay Wassim
Telecom-Paris
wassim.bouatay@telecom-paris.fr

Abstract

The assignment is about classifying a subset of Caltech-UCSD Birds-200-2011 dataset. In this report, I will detail in the first part, the preprocessing of the images and in the next part, I will explain the approach used, which was based on transfer learning and bagging.

1. Preprocessing

The provided dataset contains 1185 images in training and validation and 517 images in the test-set. These images are divided into 20 classes of birds. The very few number of input images present a problem in deep learning. So, the main idea was to use a pretrained network.

The provided validation was too small and not significant (the gap between validation and test was about 20%), making it harder to fine-tune the parameters. So, I had to transfer some images from the training to the validation to have an 80% - 20% split.

Using the transforms of Pytorch, the images were normalized and resized to 300x300 (the bigger the image the better accuracy).

1.1. Data augmentation

To alleviate the problem of the small dataset. I augmented the data using the tools provided by Pytorch. After observing the data, it is notable that the birds have different orientations in the images. So, I used random rotation [-30°, 30°] and I also used random horizontal flip with a probability of 0.5.

1.2. Cropping

The various background may disturb the model from learning the right features. The birds are also located in different places and with different sizes. Sometimes the bird takes less than 5% of pixels in the image. To help the model, I used Faster R-CNN pretrained on the 'COCO' dataset, to determine the position of the bird in the image. Then, I cropped the image using the bounding box and saved the new image.

2. Architecture

For this problem, I used Resnet50 trained on ImageNet with a backbone composed of two fully connected layers with an output of 64 and 20. I had also frozen the first 7 layers and unfrozen (trained) the remaining layers along with the backbone to adapt the Resnet50 to the problem I have.

I trained this model for 100 epochs, using the SGD optimizer with a momentum of 0.8. I used a small learning rate of 0.006 with a scheduler that multiplies it by 0.8 if the validation accuracy doesn't improve for 5 epochs. I used also early stopping if the network doesn't improve for 25 epochs, and save the model corresponding to the best validation accuracy.

Finally, I trained 5 different networks of this model independently and I summed their output. Then similarly for the case of a single network, I take the class having the maximum output. I had to use multiple networks because I noticed a certain variance in the validation between [89% and 91%] every time I retrain the model. So, taking several ones helps to reduce this variance and to improve the results.

The mentioned parameters were fine-tuned using a Grid-search.

3. results

	Validation	Test
Final Model	89.92%	81.93%
Final Model Without Bagging	89.57%	78.71%
Final Model Without cropping	90.65%	73.55%
Final Model Without unfreezing	79.39%	65.16%
Final Model Without augmentation	86.17%	77.42%

Table: Validation and test accuracy in different models

The Final Model is the model using all the mentioned techniques. We can see in the table how much each technique helps improving the test and/or the validation. We also notice, that even though we have almost the same validation without cropping and without bagging, these techniques have an important impact on the test.