Object Recognition and Computer Vision
Assignment 3

Eya GHAMGUI
Master MVA
eya.ghamgui@telecom-paris.fr

Abstract

The aim of this assignment is to determine the classes of a subset of the ***Caltech-UCSD Birds-200-2011 bird*** *dataset. This dataset contains photos of 20 bird species. It is divided into training, validation, and test data. In this report, I will first describe data pre-processing. After that, I will detail the classification model and its parameters. Then, I will finish by providing the obtained performances.*

1. The dataset

The dataset contains 1082 images in the training set, 103 in the validation set and 517 in the test set. Using the transforms of PyTorch, the images were normalized and resized to 256 x 256.

2. Data Pre-Processing

2.1. Calibration of validation and training data sets

In order to have better parameter tuning, we need to increase the size of the validation dataset. To do this, I added to this dataset images from the training dataset so that the total number of images is divided into 75% of the data for the training data and 25% of the data for the validation data.

2.2. Create new images by cropping

The quality of the images in the dataset has a huge impact on the performance of the model. Having a lot of irrelevant structures in the background of the image will make the model distributed by these elements. Therefore, I chose to create new images by cropping the object (the bird) from the original images using Mask-RCNN network. For a given image, this model returns the objects bounding boxes, classes, and masks. Furthermore, this model is pre-trained on the 'COCO' dataset which contains several objects including birds. I tested two approaches: the first is to use this new dataset instead of the original dataset and the second is to use the two merged datasets together.

2.3. Data augmentation

Data augmentation is useful to improve performance and outcomes of models by forming new and different examples to train datasets. One of the easiest ways to go about it is to work with the simple transforms from PyTorch. I used in this assignment random rotation with angle equal to 20°, random horizontal flip with probability 0.5, random affine with a degree equal to 20° and color jitter by changing the brightness and saturation.

3. Architecture of the model

In this part, I used the ResNet34 as a pre-trained model. In addition, I changed the last fully connected layer (fc) to two fully connected layers, a Tanh activation function and dropout with probability 0.6. In the forward function, I added a block to the model to fine-grained the classification. Indeed, this approach is used to differentiate minor categories like bird species.

The learning is based on two steps. I used in both steps the ADAM optimizer. In the first step, we freeze the ResNet weights and biases before the (fc) layers and leave only the modified (fc) layers unfrozen. Then, we train the model over 20 epochs. This step trains the weights and biases of the added layers. In the second step, we unfreeze all layers and re-train the entire network over 20 more epochs. In this step, I added a stopping criterion on the loss values.

To get better performance, I tried several combinations of hyperparameters to finetune the model. I found that the best parameters are learning rate = 10^{-4}, weight decay = 10^{-5}, step size = 20 and gamma = 0.1.

4. Results

Training the previous model using the dataset, we obtain the following results for the validation data:

	F1-Score	**Precision**	**Recall**	**Accuracy**
without fine-grained (cropped data)	86.1983 %	87.1059 %	86.1386 %	86.1386 %
with fine-grained (cropped data)	87.0724 %	87.7264 %	86.7987 %	86.7987 %
with fine-grained (Both data)	83.7781 %	84.3306 %	83.6634 %	83.6634 %

I found that the fine-grained technique improved the performance of the model on the validation dataset from 86.13% to 86.79%. Adding the initial images to the cropped image dataset, decrease the performance on the validation dataset with an accuracy equal to 83.66%. However, the last model gives the best performance on the leaderboard with an accuracy equal to 79.35%.