Assignment 3: Kaggle competition – Birds recognition

Ilyass RAMDANI
ENPC
6-8 avenue Blaise-Pascal, Champs sur Marne
Ilyass.ramdani@eleves.enpc.fr
http://www.ilyassramdani.com

Abstract

First, I would like to express my special thanks for giving me the excellent opportunity to participate to the course of Object recognition and computer vision as open candidate. Even if I didn't get in the top rankings, this enjoyable experience, helped me to know about so many new things.

I will present all the different strategies I tried and the results it gave.

1. Motivation and problem definition

1.1. What is the problem you are trying to solve?

The objective is to build a model with which performs the best classification.

2. Different strategies implemented & results

2.1. CNN

First we tried building a simple CNN as we knew from the course that they were invariant and prone to give the best results. 4 layers of conv2D+MaxPool and 2 layers of networks.
We first twicked the different parameters in all directions, added convolution layers, the number of nodes per layer etc… It was often prone to over fit the data in the training set but was very lacking in the validation set as we couldn't go over 0.42 in the Kaggle score.

2.2. CNN with augmented data

Having a very small amount of images, we first thought of using augmentation data techniques. However, with the CNN we were still having an upper bound of 50% in accuracy. And even with high dropout and regularization, it doesn't perform better.

2.3. Transfer learning with ImageNet

We decided to use ImageNet as DataSet as it is a large visual database designed for use in visual object recognition.We tested many algorithms to see which one best performs : ResNet, Xception, DenseNet121, _169, VGG, etc.. The best results were performed by DenseNet169 and Xception who gave more or less the same results (63%).
The strategy is always the same: we set the DenseNet part as not trainable, we add a layer of 4096 nodes and a last layer. Both perform very well on the training set that they over fit very quickly (loss=0.02 in less than 10 epochs, accuracy = 100%). However, in the validation set, accuracy around 80% and loss =0.60.
That's when I performed my best score in Kaggle (0.67).

2.4. Transfer learning with augmented data

We then added the same augmentation as we used on the CNN: Width and height shift, rotation, brightness… in the preprocessing step in order to see how our model react. It performs a little better but not skyrocketing: in the validation set: accuracy around 84% and loss =0.50. But once tested on kaggle, we still obtain c.0.65 in score.

2.5. Transfer learning with ImageNet and NABirds

We took 200 classes from NaBirds and tried to train the (DenseNet169 + Dense 4096 nodes) on the Nabirds BD in order to use that same algorithm and do transfer learning on our problem.
The results are even worst than before, we had a loss of 0.7, an accuracy of 0.75 in the validation set and a score that couldn't exceed 0.52 on Kaggle.

3. Conclusion

This assignment was overall very interesting and I learned many new things. I reached a limit in terms of results (and the limit of GPU on colab). Unfortunately, with lockdown it's difficult to share with the other classmates which often lead to better ideas. It would be very interesting to have some of the best practices for this assignment.