MVA RecVis 2019 Challenge : Fine grained bird classification

KILANI AL HOUCEINE
Ecole Normale Supérieure- Paris Saclay
houcinekila@gmail.com

Abstract

There has been a lot of improvements in the Computer Vision domain concerning classification of images. But Fine Grained Visual Categorization (FGVC) is still a current problem that is treated in many researches and papers. In this practical, we had to classify images of birds into 20 classes where their differences are very subtle and very slight, with very few shots data per category. In addition to that, the bird to classify is sometimes hard to identify in the picture due to the background and/or occultation (tree branches, tree leaves...). Here we present a method of treating this classification problem that achieved a score of 84% accuracy in the public test set on the Kaggle leaderboard of the MVA RecVis 2019 Challenge.

1. Introduction

To solve this FGVC problem, we have to identify what are the main tasks to do by identifying the constraints that make this problem such a hard one. There is the lack of training images (**2.1**), the bird localisation in the image (**2.2**), the classification algorithm/NN architecture used (**2.3**).

I want to point out that, unlike some other classmates, I did not use pretrained models on iNaturalist and used only models pretrained on ImageNet (iNaturalist dataset contains some CUB images and thus it boosts the performance but I consider this going against the rules as the supervisors told explicitely not to use models pretrained on birds).

2. Implemented solutions

2.1. Lack of training data

As mentionned in the **Abstract**, there is a lack of data in the classes, so it is hard to train a NN from scratch. I solutions of Domain adaptation [6] in order to find images of a source domain that semantically "close" to our target domain (will lead to close feature representations) then start from that pretrained model on the source domain and apply it on our domain. Finding optimal Data augmentation policies for our dataset [2] and Fewshot learning solutions [5]. But because of a lack of time, I decided to go on a simple data augmentation when training but using the "imgaug" library that contains richer image augmentations and that was also easily compatible with Pytorch transformers.

2.2. Bird Localisation

Many opensource object detection algorithms can be found online, I tried some of them and ended up using the same logic as [1] to detect birds with high confidence, crop and zoom on them to finally store them to be used as input for my model.

2.3. Classification algorithm

I began to explore ideas about attention mechanisms for images (just like NLP) in order to detect features and important parts of images [3]. But it did not yield to good results when applied on our dataset (maybe a problem of implementation ?) so I had two options: continue debugging this solution or try someting else. I decided to go on a safer base by taking a pretrained ResNet152 model (no real justification for the architecture choice, in the beginning I wanted to take the smallest architecture for a quick training, so I tried with the ResNet18 but apparently it was too shallow so I opted for the ResNet34 but it was not good either) where I retrained the last bloc of convolution layers and the classification/fully connected layer. I also wanted to try some other architecture and perform a majority voting or weighted voting but got out of time as well.

3. Submitted result

In the private cross validation set, constituted of 103 pictures, we see a big score (locally I was reaching 90% accuracy but the LB accuracy was 71%) but the problem was that it was not a good proxy for the generalisation accuracy as it was not big enough and the birds there were easy to spot and find whereas some birds in the test set are very hard to locate. These two issues make it hard to know what is the real reason behind the low performance : was it overfitting ? was it because the birds were not well located ? both of them ?

In the end, I ended up using majority voting between :

- This model (pretrained ResNet152 on ImageNet, Augmentation on cropped images, SuperConvergence to regulate the learning rate)
- Taking the last output layer weights of this NN (before fully connected to 20 neurons) and applying an XGboost classifier (Standard parameters)
- Taking the last output layer weights of this NN (before fully connected to 20 neurons) and applying a logistic regression (Standard parameters)

This gave me a LB score of 84% but I know this is not the goal of the competition (That, I believe and hope, was using NN to perform the classification and cope with the difficulties of the dataset and just not getting a big LB score and explaining/justifying our choices).

4. Conclusion

There was a room for improvement, if I had more time to devote to this competition (honestly I did not manage very well my time -choosing a lot of courses, courses that had short deadlines -Altegrad-). And I believe that if I had used the 2 allowed submissions daily to get a hint of what were the actions to take in order to improve my model it would have been better (Ended up submitting only 5 times).

Some other ideas to build a better model :

- Use voting (hard or soft) between more models
- Finetune more layers of the pretrained models used in order to capture more specific and discriminant features of the birds.
- More epochs
- Use SuperConvergence [4] in order to adjust learning rate while training
- Learn the augmentation policies from the data with AutoAugment like said before

References

[1] A. Benamira. https://github.com/adrienbenamira/bird_classification_mvachallenge. 2018.

[2] C. et al. Auto augment: Learning augmentation strategies from data. 2019.

[3] H. et al. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. 2019.

[4] S. et al. Super-convergence: Very fast training of neural networks using large learning rates. 2017.

[5] W. et al. Few-shot adaptive faster r-cnn. 2019.

[6] X. et al. D-sne: Domain adaptation using stochastic neighborhood embedding. 2019.