Bird Classification on Pytorch

ResNet-50 / VGG19 / ViT_H_14

GALAGAIN Calvin
calvin.galagain@ens-paris-saclay.fr

Abstract

The problem that we are going to deal with consists of an image classification. More specifically, we will try to classify twenty species of birds extracted from CUB-200-2011 [1]. Through this work, we will see data augmentation techniques to deal with a database with a low amount of data. Then, we will see that Transfer Learning and Fine Tuning make it possible to obtain the most efficient models.

1. Data Processing

The database that we use contains few images and are images taken in real conditions. This means that there are a lot of differences between the images whether in terms of position, size or even color and, this is true even if it's the same species.

To remedy this problem, we will start by looking for methods of extracting our birds. For this, we can use several models like Detectron2 *[2]* or YOLO *[3]* which have been trained on a lot of data. Of course, the birds are not directly classified, it is only to extract the birds to know the interesting regions of the image *[Fig.1]*. The first problem that we encounter will be the false detections that we can either ignore, or we consider the entire image or, with a little time, we correct by hand. To go further, we could use a segmentation of the birds *[Fig.2]* but for this work, we decided to simply keep the bounding boxes to keep information about the environment of the bird.

So we recreated the database and replaced each image in the folder with the previous extraction obtained. To have more data, we decided, for each training and validation image, to randomly duplicate them slightly. So, by slightly enlarging the extraction box or by slightly rotating the image, we have different images.

As a result of this, the images are changed in terms of brightness, contrast or even by adding Gaussian noise. This allows you to still have slight differences between the images and not to overfit too much on some.

2. Models

The data being very different, it is futile to create a model from scratch to try to classify them. Indeed, it is much simpler and much more efficient to reuse the knowledge of models that are trained to classify harder problems. Indeed, these models have good results because they are able to extract features and then analyze them. What we are going to do is pick up mining skills and specialize with our birds.

Thus, to better compare the performances, it is interesting to test several architectures of models. We will thus recover the weights and architectures of ResNet50 *[4]*, VGG19 *[5]* and vit_h_14 *[6]* (which is a Vision Transformer). The goal is not to train the first layers of these networks and to modify only the specialization layers. After a rough specialization on our database, we can lower the learning rate and train more layers. But, in our case, performance no longer improves once the specialization is done.

3. Evaluation

Once the training is finished, we predict the classes for each model. Based on validation, the performance between VGG19 and ResNet50 is similar and the Vision Transformer is a bit better without showing signs of overfitting *[Fig.3 & 4]*.

To balance the performance, we decide in this work to weight the predictions of each model by the accuracy value obtained on the validation basis. Thus, the performances are averaged and this avoids local overfitting. We previously had 83.2%, 82.8%, 92.9% of accuracy for respectively ResNet50, VGG19 and ViT to 90.1% if all of the predictions are averaged.

On Kaggle, two submissions are therefore published, the performance of our Vision Transformer and the weighted average performance.

ANNEXE

References

[1] Wah, C. and Branson, S. and Welinder, P. and Perona, P. and Belongie, S. ***The Caltech-UCSD Birds-200-2011 Dataset***, 2011

[2] Yuxin Wu and Alexander Kirillov and Francisco Massa and Wan-Yen Lo and Ross Girshick, ***Detectron2***, 2019

[3] Redmon, Joseph and Farhadi, Ali, ***YOLOv3: An Incremental Improvement***, 2018

[4] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, ***Deep Residual Learning for Image Recognition***, 2015

[5] Simonyan, Karen and Zisserman, Andrew, ***Very Deep Convolutional Networks for Large-Scale Image Recognition***, 2014

[6] Alexey Dosovitskiy and Lucas Beyer and Alexander Kolesnikov and Dirk Weissenborn and Xiaohua Zhai and Thomas Unterthiner and Mostafa Dehghani and Matthias Minderer and Georg Heigold and Sylvain Gelly and Jakob Uszkoreit and Neil Houlsby, ***An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale***, 2020

Figures

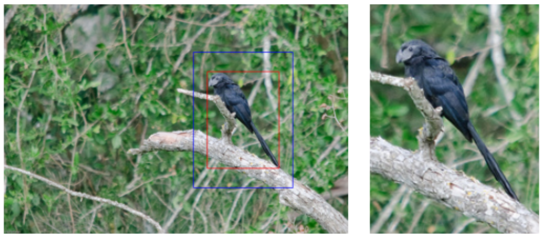

Fig.1 LEFT: original bounding box of a bird is Red and extension is apply to obtain the blue / RIGHT: the result of extracting

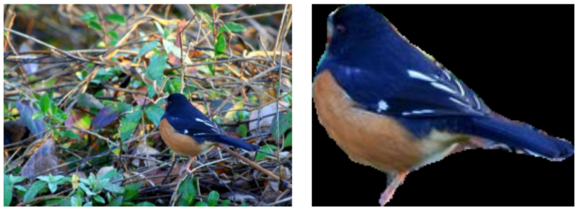

Fig.2 segmentation of a bird with Detectron2

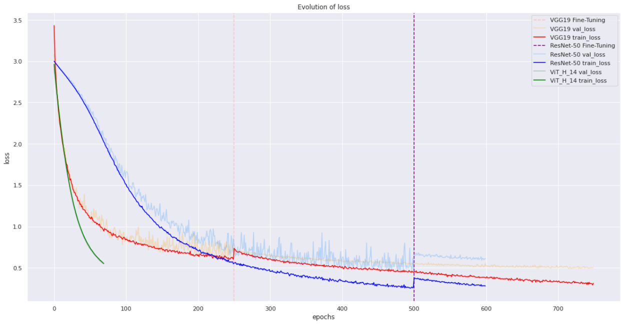

Fig.3 Evolution of the Loss of each model

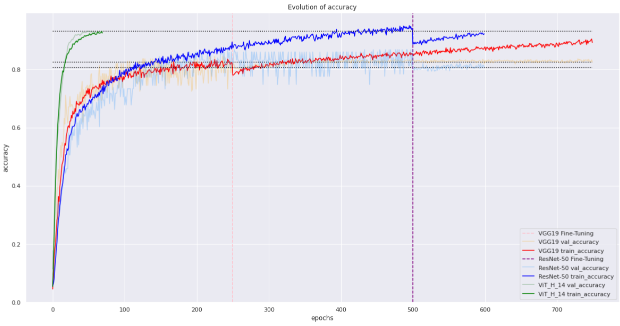

Fig.4 Evolution of the Accuracy of each model