Efficient Fine-grained visual categorization on a subset of CUB - 200-2011

Adrien BENAMIRA
CentraleSupelec
Paris-Saclay
adrien.benamira@supelec.fr

Abstract

Fine-grained visual categorization (FGVC) aims to distinguish objects in subordinate classes. For example, birds images are classified into different breeds of birds, such as Painted Bunting, Lazuli Bunting, Bobolink. The problem is therefore to classify images with different background, whose bird is on the one hand hard to find and on the other hand the difference between these birds is slight. All with little data per class : around 50. Here we used a features extraction proposed by Cui and al. (2018) [1] to embed the image. This embedding and few tricks permits to achieve 0.90322 on test set of the Kaggle Competition.

1. Introduction

For fine-grained visual categorization (FGVC) the quality of the embedding of the image and the quality of the bird detection are decisive. In a first time we implement a bird detection algorithm thanks to Niitani and al. (2017) [2] project [1]. Then, to realise the embedding, we use the open source project[2] provided by Cui and al [1]. So for every image, we have two vectors representation : 1 vector bird with background, 1 without. We can concatenate this vector or not. Finally, we classify results with linear regression or 3 layers perceptron layers.

2. Pipeline

In this section, we present our pipeline scheme that achieves top performance on the challenging Kaggle MVA-RecVis-2018 :Bird image classification competition.

2.1. Bird detection

Niitani and al. (2017) [2] provides many open source model for object detection and after few experimentation, models FasterRCNNVGG16, SSD300, SSD512, YOLOv3, YOLOv2 seems to have the best performance. Bird is class 2 and for every detection, we want to control the fact that the algorithm cropped where there is no bird. So we crop on the bounding boxes with the highest a rate of confidence if this rate is higher 0.9. This prepocessing takes 12 hours.

[1]https://github.com/chainer/chainercv

[2]https://github.com/richardaecn/cvpr18-inaturalist-transfer

2.2. Embedding

Cui and al. (2018) [1] provides open source model for picture embedding in tensorflow. It treats images of size 299x299 and output a vector, called after features, of dimension 2048. It takes one minute to treat 150 images.

2.3. Classification

At this stage we have 2 features represention for every image. We concatenate it and then we classify. We use two methods : linear regression with Scikit-Learn and 3 layers perceptron model with pytorch. I train it during 70 epochs, with batch of 8, learning rate of 0.0001, momentum of 0.9.

2.4. Results

They are in Table 1.To go further, we can get the closest neighbour of each image missed in validation set.

	Linear Regression	Perceptron
Validation set	0.94175	0.95151
Test set	0.90322	0.90322

Table 1. Results.

3. Conclusion

We present a pipeline which achieves excellent results on Kaggle Bird image classification competition.

References

[1] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie. Large scale fine-grained categorization and domain-specific transfer learning.

[2] Y. Niitani, T. Ogawa, S. Saito, and M. Saito. Chainercv: a library for deep learning in computer vision. In *ACM Multimedia*, 2017.