Bird classification from CUB-200-2011 dataset

1. Introduction

The goal of this paper is to show a way of classifying 20 bird species from the CUB-200-2011 dataset [5]. Only a part of the dataset have been use, making in total 1082 pictures of birds for the train set and 103 for the validation set. On top of the bird species, segmentation maps of the birds were given for the train and validation set. **I present here a two-step classification method: the first step is the segmentation of the image and the second one is the classification of the given image knowing its segmentation.** The code for this method is available [3].

2. Data prepocessing

The segmentation part give us the pixels where the bird should be in the image. The information is used to build two kinds of data input for the neural networks of the classification part: crop from the raw image and 4D tensors composed of the raw image and the segmentation map.

The data augmentation was craft with the purpose of enlarging the scope of the train set and to avoid over-fitting. The data augmentation is as follow. First, the color of the raw image is changed by jittering a bit around its values, by applying a gaussian blur, by adjusting the sharpness and by modifying its constrast. Then, the image was randomly flip along the horizontal axis and a rotation within -30, +30 degrees was done. This rotation keeps all the pixels of the original image inside the rotated frame. After that, the image was resized to fit the neural network input requirements. A special care was taken to keep the aspect ratio to show the neural network the right proportion of the birds. Finally, the image is normalized according to the values of the training set.

3. Models

The segmentation part is done by the detectron2 network [6]. This neural network was trained to classify the objects in a picture and output the segmentation map of this object. Since the segmentation maps were looking good I decided not to spend much time on this part and to switch to the classification part.

As described earlier, the neural networks trained to classify are trained either with the cropped images or with the 4D tensors. In both case, 6 different architectures of the neural network were tried: ResNet, VGG, DenseNet AlexNet SqueezeNet and EfficientNet-B7. EfficientNet-B7 is the one that performs the best on ImageNet according to the developpers of PyTorch [4]. They were taken pretrained from ImageNet. I also tried to pretrain them from a combined dataset of monkey species [1] and butterfly species [2] but it did not led to better results. The last layer of the networks was removed and a new layer was added with random weights.

In order to adapt the architecture of the networks when the input is a 4D tensor, the first layer of each network was replaced by another layer with the same characteristics (same kernel size, same padding...) but allowing the input to have 4 channels. Note that the weights for the first 3 channels were kept from the pretrained weights.

At the end of the different trainings, an ensemble model was built on top of the best 3 models trained from crops and the best 3 models trained from 4D tensors. The ensemble model consists on a voting system where the probabilities of an image to belong to a class are summed up among the different models and the resulting prediction is the class with the highest sum.

4. Training

All the models were trained on Google Colab. The segmentation data was coming from the ground truth. The cross entropy loss was used with some weight decay to avoid over-fitting. Adam optimizer was used with a learning rate scheduler. This scheduler divides by two the learning rate when the validation loss has not decreased for a few steps. At the end of the training, the best weights of 80 epochs is selected. To define what are good weights, I defined a score. This score is the sum of the rank on the validation loss and the rank on the validation accuracy. Indeed, it is important to focus on the validation accuracy since our goal is to have a good testing accuracy. The validation loss also tells us how confident is the model, which gives an insight on the generalization property of the network.

5. Results

The accuracies of the networks on the validation set using the segmentation network are shown in figure underneath. The fact that the ensemble model is able to improve the validation accuracy to **93.2** % shows that the different networks have learnt to classify birds in a different way.

	ResNet	VGG	DenseNet
crop	88.3	85.4	85.4
4D tensor	88.3	90.3	90.3

6. Discussions

A interesting idea would have been to train a segmentation model and a classification model at the same time using a loss on the segmentation map and a loss on the output probabilities. The difficulty with this idea is to find a neural network already trained for a similar problem. Doing it on a virgin model would not have worked because we have too few data.

To improve the ensemble model event more, we could have extracted the ouputs of the second last layers of the best models and then, we could have trained a classifier on top of that. Nonetheless, I think that the bottle neck here

is the classifiers themselves. With more time, it could have been interesting to see how researchers tried to solve this problem.

References

[1] 10 monkey species from kaggle dataset. https://www.kaggle.com/slothkong/10-monkey-species. 1

[2] Butterfly image classification 50 species from kaggle dataset. https://www.kaggle.com/gpiosenka/butterfly-images40-species/activity. 1

[3] Github repository. https://github.com/theovincent/birdClassification. 1

[4] Pytorch documentation. https://pytorch.org/vision/stable/models.html. 1

[5] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1

[6] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 1

7. Author

Made by Théo VINCENT