Fine-Grained Birds Classification (Assignment 3)

Matthieu Futeral-Peter
Master MVA - ENSAE Paris
matthieu.futeral.peter@ensae.fr

Abstract

The problem is to classify 20 bird species extracted from the Caltech-UCSD Birds-200-2011 dataset. There are 1702 images of which 517 are test images and 1185 are train and validation images. The goal is to develop a method that has best accuracy score on the test set. Programs of this project are available at my git.

1. Introduction

Fine-grained classification is a particularly difficult task. It aims at constructing an algorithm capable of differentiating between very similar species. To complete this task, after preprocessing steps, I cropped the images around the birds and gave the cropped images to a stacked classifier.

2. Preprocessing steps

First, I resampled the validation set because it was unbalanced so it made the model selection biased. I randomly split the 1185 train and dev images, keeping 15% for the dev set.

2.1. Crop birds

Then, I cropped the train, validation and test images around the birds in order to reduce the impact of the scene on the prediction and to have a model as scale invariant as possible. To do so, I used the official Mask R-CNN PyTorch implementation [4] and built a program to crop the 33 out of 1702 images on which Mask R-CNN failed to detect any birds. All the cropped images were resized at 224x224 in order to fit most networks input resolution and to be close to the actual mean width and height values (resp. ≈ 210, ≈ 180).

2.2. Data Augmentation

During training, all the images are horizontally flipped with probability 0.5 and vertically flipped with probability 0.5. The purpose is two-fold : not only it increases the amount of data but it also makes the model more robust to changes of orientation in the image.

3. Method

I trained a Vision Transformer [1] stacked with Inceptionv3 in a supervized way, both models are pre-trained on ImageNet. I used the 16x16 patch size and 224x224 large ViT implementation by Pytorch-Image-model[1] and the implementation of Inceptionv3 by torchvision. The output of these models are then concatenated and given to a layer with 512 hidden units which is then given to a classifier. I used Adam optimizer with 2e-5 learning rate scheduled by cosine annealing and 1e-4 weight decay (without bias weight decay). To handle limit of GPU RAM, I implemented gradient accumulation and then performed backward update every 2 batches (with batch size = 4). To prevent the model from overfitting, I used early stopping[2] with patience 5 and 0.33 dropout at each layer. This method results in an accuracy of 0.97 and an average cross entropy loss of 0.014 on the dev set after 5 epochs and 0.87741 accuracy on the public leaderboard which is awarded by 3rd place.

4. What I tried and did not work

Augmenting the dataset with NABirds unlabeled images and applying semi-supervized learning techniques with loss smoothing such as pseudo-labelling and consistency regularization [3] or training API-NET (Attentive pairwise interaction) [5] were tried but none of these methods improved either validation loss and accuracy or public score. Other data augmentation techniques such as CutOut, Random Rotation, Gaussian blurring or Super Resolution algorithm[3] [2] led to poor performances.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

[1] https://github.com/rwightman/pytorch-image-models

[2] https://github.com/Bjarten/early-stopping-pytorch

[3] https://github.com/dongheehand/SRGAN-PyTorch

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *Int. Conf. Learn. Represent.*, 2021. 1

[2] C. Ledig et al. Photo-realistic single image super-resolution using a generative adversarial network. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 105–114, 2017. 1

[3] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 1

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

[5] Peiqin Zhuang, Yali Wang, and Yu Qiao. Learning attentive pairwise interaction for fine-grained classification. In *AAAI*, pages 13130–13137, 2020. 1