MVA RecVis 2021: Bird image classification competition

Loïc Magne
ENS Paris-Saclay

Abstract

The Fine-Grained Image Classification task focuses on differentiating between hard-to-distinguish object classes such as species of birds. Here the challenge aims at classifying species of birds, using a small subset containing 20 species of birds of the CUB-200-2011 dataset [7]. This task is challenging due to the low amount of label training data, and the difficulty to differentiate certain species of birds. An object detection network is first used to find the bird in the image, then vision transformers are used to classify the bird species.

1. Bird Cropping

A quick glance at the dataset allows to realize that there is a lot of variance in the bird images: some birds are nicely cropped, but some are partially hidden, some are in the background, some are hard to find. To normalize that, I first crop every images around birds using the YOLOv5 model [3]. YOLOv5 is a family of object detection architectures and models pretrained on the COCO dataset, based on the YOLO [4] object detection architecture. I used the YOLOv5 pretrained large model out of the box, which already has a bird category. Only the most likely bird is kept from each image. This method allowed me to crop bird in every images.

2. Image Augmentation

Test set and train/val set have slightly different distributions, and results may vary a lot between validation set and test set. To improve generalization and avoid overfitting, an intensive data augmentation pipeline is used with a combination of spatial augmentations (rotation, shift, scale, distortion) and pixel level augmentations (blur, downsampling, noise, contrast).

3. Model

A vision transformer is fine-tuned to classify the cropped bird images. Several recent architecture (ViT [2], DEiT [6], BEiT [1]) are tested using the HuggingFace [8] library. Based on experiments, I selected the BEiT model [1]. It is a recent self-supervised vision transformer, trained in a BERT-fashion where some image patches are randomly masked into the backbone Transformer. The pre-training objective is to recover the original visual tokens based on the corrupted image patches. Large-size BEiT obtains 86.3% only using ImageNet-1K, even outperforming ViT-L with supervised pre-training on ImageNet-22K (85.2%).

The goal of using a self-supervised architecture was to be able to use unlabeled dataset like NABirds or iNaturalist. Sadly it takes a lot of resources (16 V100 GPU are recommended) to pre-train such an architecture in a self-supervised way, hence I only finetuned the model in a supervised way. A dense linear layer is added on top of the transformer to classify the 20 species.

4. Training details

Several tricks are used to train the model, some coming from [5]. Here are the details of the training parameters:

- SGD optimizer, momentum 0.9, lr 1×10^{-2} with cosine decaying scheduler, batchsize 8, weight decay 1×10^{-5}
- Gradient clipping to 1 to prevent gradient problems
- Automatic Mixed Precision is used to make training faster although the impact for 'small' models is not significant
- Cross Entropy is reweighted based on some biais that the model often fall into

5. Results

Here are the obtained results using techniques described above:

	Validation Set	Public Test Set
BEiT baseline	98.05%	92.9%

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers, 2021. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, 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, 2021. 1

[3] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Abhiram V, Laughing, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Jebastin Nadar, imyhxy, Lorenzo Mammana, AlexWang1900, Cristi Fati, Diego Montes, Jan Hajek, Laurentiu Diaconu, Mai Thanh Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. https://github.com/ultralytics/yolov5, Oct. 2021. 1

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. 1

[5] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers, 2021. 1

[6] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention, 2021. 1

[7] 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

[8] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. 1