Kaggle bird classification challenge report

Marius Schmidt-Mengin
marius.schmidt.mengin@gmail.com

Abstract

This report outlines two techniques that I used to address this few-shot fine-grained classification problem. The first one is based on metric learning. The second one is based on a 2-layer neural network appended to the last feature layer (before pooling) of a state-of-the-art convolutional neural network.

1. Data preparation

I found the provided validation set to be too small and not representative enough of the test set. Therefore, I added images from the train set to the validation set in order to get 20 validation images for each bird species.

Furthermore, to make the classification task easier, I used an object detection model to obtain the bounding box of the bird in each image. Fortunately there is a "bird" class in COCO. I used EfficientDet-D7x[1] [5], which is currently state-of-the-art on the COCO benchmark.

2. Classifying without training

I used pre-trained feature extractors[2] to obtain embeddings for all images. Following [3], I computed class-specific "prototype" embeddings: for a given class c, I gathered all embeddings from the training set of class c, L_2-normalized them, computed their mean embedding, and re-normalized it. Then, for a given test image, its embedding is compared to all class prototypes by cosine similarity. We infer the class probabilities by taking the softmax over those similarities. It is also possible to do this without normalizing and by replacing cosine similarity by L_2 distance but I got slightly worse results. I did this for several pre-trained models as listed in table 1. Without training, it is possible to achieve accuracies of more than 90%. This is probably due to the fact that there are several bird classes in ImageNet. I then combined the predictions of all these models. I achieved the best validation accuracy of 94% (82.52% on the public test set) by averaging the probabilities of all models listed in table 1. I also tried to learn a linear classifier ontop of those embeddings, but I did not get than more 95% validation accuracy, as well as fine-tuning the whole network including the backbone, but even with strong augmentations [1, 8, 9], I was not able to outperform the approach I describe in the next section.

Pre-training	ImageNet		Noisy Student		AdvProp [6]	
EfficienNet-B0	86.00	77.3	88.25	78.8	87.50	77.1
EfficienNet-B1	88.25	79.2	87.75	81.5	87.75	79.3
EfficienNet-B2	87.00	80.0	87.75	82.4	89.25	80.3
EfficienNet-B3	87.25	81.7	88.00	84.1	90.00	81.8
EfficienNet-B4	89.00	83.2	**88.50**	85.3	**91.50**	83.2
EfficienNet-B5	91.50	84.0	88.50	86.1	90.75	84.3
EfficienNet-B6	87.25	84.5	87.00	86.4	87.75	84.8
EfficienNet-B7	**92.25**	**85.0**	87.25	**86.9**	88.75	**85.2**

	Birds	ImageNet
ViT base patch 16 image size 224	91.00	79.35
ViT base patch 16 image size 384	92.00	84.21
ViT base patch 32 image size 384	92.50	81.65
ViT large patch 16 image size 224	**92.75**	82.7
ViT large patch 16 image size 384	92.75	**85.16**
ViT large patch 32 image size 384	91.00	81.51
ViT small patch 16 image size 224	86.50	77.86

Table 1. Accuracies of state-of-the-art pre-trained models [2, 4, 7, 6]. Black text color: validation accuracies for the bird dataset (without fine-tuning). Grayed out text: ImageNet test accuracies as reported in the papers.

3. Classifying features

I got my best result using EfficientNet-B5 with Noisy Student [7] weights. This model has 86.1% accuracy on ImageNet, which is currently one of the best. I inferred all images to obtain their feature maps of size 15x15 and 2048 channels. Then, I trained a small neural network on these features. The neural network is made of a few layers: dropout (drop rate 0.5), 3x3 convolution with 512 channels, batch normalization, ReLU, global max-pooling, dropout (drop rate 0.5), linear. I optimized the network with SGD

[1] https://github.com/zylo117/Yet-Another-EfficientDet-Pytorch
[2] https://github.com/rwightman/pytorch-image-models

(learning rate = 10^{-3} and momentum = 0.9). This approach gave me 96% accuracy on my validation set and 85.161% on the public test set.

References

[1] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 3008–3017. IEEE, 2020. 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. In *https://arxiv.org/abs/2010.11929*. 1

[3] Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5822–5830. IEEE Computer Society, 2018. 1

[4] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. 1

[5] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10778–10787. IEEE, 2020. 1

[6] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 816–825. IEEE, 2020. 1

[7] Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. IEEE, 2020. 1

[8] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6022–6031. IEEE, 2019. 1

[9] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 1