Object recognition and Computer vision: Assigment 3

Arthur Pignet
arthur.pignet@mines-paristech.fr

Abstract

The objective of this assigment is to produce a model that performs the best on a bird classification task. Due to the small size of the given dataset, relative to the complexity of the task, I decided to focus on features extraction.

1. Introduction

The dataset is constitued of pictures of 20 different species extracted from the Caltech-UCSD Birds-200-2011 bird dataset [7]. The data is spltted into 3 parts, with 1082 training samples, 103 validation samples, which will be never used for training, and 517 test samples, of which the labels are not given. The amount of training data is really small, and the end to end training of a network is not an option. Thus I focused on features extraction. First of all I tested an approach based on scattering wavelet network, followed by a small CNN, with poor result. Then I used networks which have been pretrained on ImageNet. At last I tried a weakly supervised solution by training in an unsupervised way an autoencoder on a much bigger dataset, and then I used the encoder part as features extraction for a deep classifier. We will use as a baseline the small network provided and trained, whose validation accuracy is 12.6%.

2. Wavelet Scattering network

The first method is based on the observation that most of the filters learn by the first convolution layers are often wavelet filters. Stephane Mallat developed this idea, and highlighted that scattering wavelet transform are powerful feature extractors as they are covariant by small deformations [5]. I used a python library developed by the DIENS DATA team, named Kymatio [1] to implement the scattering transform. It is worth noting that it improves the val accuracy, however the test accuracy was not convincing at all. I conjecture that the bird is often a really small part of the image, something even behind a branch.

Method	val. acr. (%)	test acc. (%)
Scattering + CNN	46.60	23.87
Fine-tuned ResNet	90.29 %	70.32
Auto-encoded features	88.34	70.96
VGG	84.46	68.38

Table 1. Accuracy results obtained by the different methods described. The test accuracy is the one reported by the Kaggle submission process.

3. Fine-tuning of a Resnet pretrained on ImageNet

I downloaded a version of Resnet50 [4] which have been previously trained on ImageNet [2], and retrained it on my small dataset. I used early stopping to stop the training, ie I stop the training when the validation loss start to increase [8]. The val and test accuracy achieved where really good, and as of now the best I achieved.

4. Using as features extractor an encoder trained on unlabelled data.

Starting from the fact that I have really few data, I decided to use the bird part of the iNaturalist 2019 dataset *without the labels*. I built an autoencoder [3] based on VGG13 [6] architecture, where the encoder was initialized with pretrained weights on ImageNet. The autoencoder is trained on the iNaturalist Birds dataset, with 47,489 samples, and then fine-tune the encoder on my small labelled dataset. In practice I trained the autoencoder for 10 epochs, and then 15 epochs for the encoder and classification parts.

5. Conclusion

The method that yields the best result was the last one. The difference is no significant, but I think that this approach generalize better. I am convinced that the method could be of interest, but need further adjustments, with more time. For instance, I wanted to use superpixel segmentation to find the bird on the picture and design a loss which weights more the error on the bird for the training of the auto-encoder.

References

[1] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, Joan Bruna, Vincent Lostanlen, Matthew J. Hirn, Edouard Oyallon, Sixin Zhang, Carmine-Emanuele Cella, and Michael Eickenberg. Kymatio: Scattering transforms in python. *CoRR*, abs/1812.11214, 2018. 1

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 1

[5] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, Apr 2016. 1

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 1

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 1

[8] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, Aug 2007. 1