Bird Classification Challenge - HW3

Eva ZAGURY
eva.zagury@ens-paris-saclay.fr

Abstract

This report is an overview of my work for the classification challenge on a subset of the Caltech-UCSD Birds-200-2011 bird dataset.

Model name	Validation	Test
ResNext-101-32x8	92%	81%
Densenet-169	90%	72%
RestNet-121	89%	63%
AlexNet	87%	68%

Table 1. Models performances

1. Introduction

After a quick analysis of the provided dataset, the training set only contains 1087 images with a low informative value. This motivates a data augmentation, and a cropping of the background as it does not appear as a discriminating factor. We will then use a pretrained model to achieve classification. Code is provided on a Kaggle notebook, and we use Pytorch as a deep learning framework.

.

2. Data Preprocessing

We first apply some transformations on the data to increase it informational value.

2.1. Bird detection and Image cropping

We use Mask R-CNN [1] pre-trained on COCO dataset to crop the images on the birds. We select the bounding box showing with highest probability a bird. With a detection threshold of 0.85, model fails to detect birds in only 2.76% of cases. This step enables a 'smart' data augmentation as it prevents the algorithm from learning irrelevant features.

2.2. Data augmentation

We create new artificial data by applying random transformations (brightness, rotation and vertical flip) to the training data. This technique is useful to make the model invariant to translation, size, flip or illumination.It also helps to prevent from overfitting.

3. Model Presentation

We use a ResNext-101-32x8d model [3], pretrained on ImageNet . This model architecture is inspire by ResNet architecture (four main stages of convolution and a fully connected layer), but the Blocks itselves differs. ResNeXt block uses a "split-transform-merge" strategy, which means it aggregates a list of transformations. I also exposes a new dimension, cardinality (size of set of transformations). After unfreezing the first 3 blocks and the fc, adding a final classification layer, we train the model for 16 epochs on the concatenated dataset, with an SGD optimiser. Finally, as recquired, the loss function will be Cross Entropy.

3.1. Discussion

Several other pretrained models were also tested, with poorer performances. Table presents results on pretrained models which showed best performances.
Another approach to improve the model was to perform feature extraction, which consists in only updating the final layer weights. We tried then to progressively unfreeze each layer (finetuning) and compare performances. We finally decided to unfreeze last layers from Layer 3, as we observed that the first layers of a CNN contain generic features that are not specific to a particular task.

4. Conclusion

The approach presented gives a – accuracy on the public Kaggle leaderboard, and run in quite reasonable time (less than an hour). This approach is originally inspired by Human classification method (which consists of an adaptative suppression effect (hence neurons activation) in the most anterior part of the LO complex (High order area, modelized as the last layers of the model) [2].

References

[1] K.H.et al. "mask r-cnn". 2018. arXiv:1703.06870v3. 1

[2] Craig Weiss and John F. Disterhoft. *Encyclopedia of Social Measurement*, volume 1. 2005. 1

[3] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. 1