RecVis HW3 approach and obtained results

Dorian Gailhard

A. Data preparation

The initial cropping size of 64x64 pixels could not allow to get past 20% of accuracy with the baseline model and was expanded to 256x256 pixels inputs, and finally to 224x224 pixels inputs to ensure compatibility with ResNet.

The training dataset was then augmented using torch.transforms.AutoAugment with the IMAGENET policy, ie the application of the combinations of small transformations - gaussian blur, translations, shearing, small rotations, color jittering, etc... - on the training data with magnitudes found optimal on the ImageNet dataset.

B. Choice of the model

The model was initially the baseline model - consisting of 3 convolutional layers with maxpooling between them followed by 2 linear layers. The highest validation accuracy obtained with the model was 40% so the decision was taken to start again from an ImageNet pre-trained model.

Two models were tested : VGG19 and ResNet152.

C. Training procedure

Initially, the optimizer was a classic stochastic gradient descent with decreasing learning rate - with an exponentiel scheduler with parameter 0.9 ie at each epoch the learning was decreased by 10% - and a momentum of 0.5, but was eventually changed to an Adam optimizer which seemed to give better and quicker results. A weight decay factor of 1e-3 was finally added to escape overfitting.

The training procedure was divided into two phases. First, we locked all parameters but the ones responsible for the classification - the last block ”classifier” for VGG and the last linear layer ”fc” for ResNet - which we optimized for the bird classification problem. When the loss couldn’t be further improved, all the parameters were unlocked and optimized to gain some additional percents of accuracy.

D. Final results

With a model based upon VGG, the highest validation accuracy obtained was 85% which translated into a 72% accuracy on Kaggle.

The model based upon ResNet achieved a 87% validation accuracy which translated into a 75% accuracy on Kaggle.

E. Possible improvements

Due to constraints on GPU memory, it was not possible to implement heuristics such as training multiple models on subsets of the training data and averaging their output as the global output, which could likely greatly improve accuracy. Random Forests - training multiple decision trees on new small datasets sampled from the original data and combining them by a majority vote - was another heuristic that was not tested.

Image segmentation - localizing the bird and zooming on it to feed a clear image of it instead of the whole reduced picture to the classification model - was not tested, but it might not be so efficient as context data - in other words the species of trees visible on the image and other clues about the environment where the photo was taken - which is correlated to the bird species - birds species do not live in all environment - is lost in the segmentation. This problem could be solved by training two model : one to identify the characteristics of the bird given by segmentation and a second one to identify the environment of the picture, the two feeding a final classification model combining the information from the two. This way it might be possible to combine context data and a clearer picture of the bird.