# Assignment 3: Image classification

Paul-Emile Zafar
paul-emile.zafar@telecom-paris.fr

## Abstract

*This challenge aims at classifying 20 bird species from a subset of the Caltech-UCSD Birds-200-2011 bird dataset. The data is split in a train & validation part of around 1200 labelled images and a test part of 500 unlabeled images.*

## 1. Approach overview

I decided to use convolutional neural networks (CNN) as they have proven their efficiency on image classification tasks for the past years. Given the small amount of data, transfer learning methods using available and reliable pretrained models (AlexNet, Vgg, ResNet, Inception, ...) seems to be a good option to experiment with, given their capability in image classification, and the similarity between the original ImageNet dataset and our. After inspecting the data, we can see they are images of birds in all positions, with natural colors. Moreover, it seems that regions of their body that seem important for identification are sometimes occluded (face, eyes, breast).

## 2. Pretrained architectures

I decided to test several model implementations available in the Pytorch library: ResNet, Inception, Vgg16, DenseNet and EfficientNet. The fully connected layers of each model was replaced by one or two untrained fully connected layers, with ReLu activation and Dropout to avoid overfitting (when I had necessary time to experiment with it on the given architecture). I couldn't lead experiments an all architectures to a full extent, but a ResNet50 with the 5 first convolutional layers frozen, two classifying layers with ReLu, Adam optimizer (with a learning rate of $10^{-4}$), and step learning rate scheduler gave the best results with around 90% accuracy on the validation set (after the data crops modifications described below) and about 76% on the test set. I experimented with the other models during the remaining time and reached between 75% and 85% on the validation set.

## 3. Dataset and data augmentation

I split the data in new training and validation set (85%/15%) to get more accurate validation scores and prevent unbalanced datasets.

The images have also different sizes, and the target bird is often in different positions, scales and places in the images whereas ImageNet examples are zoomed on the target object. I thus used a Fast R-CNN implementation to detect and crop a square zone of the target bird in each image. This gave the biggest improvement on validation accuracy.

Several on-the-fly classic image transformations where added for data augmentation (image resizing to 224x224 for pretrained models, contrast and intensity jittering, random rotations and flips, blur, noise, perspective). These transformations (except noise and blur) gave a noticeable improvement of the validation accuracy.

## 4. Multiple models aggregation

In order to further improve the results, I tested several voting methods for the different models I trained. The softmax function was applied to normalize the different outputs into probabilities, and hard and soft voting was tested with four trained models (based on Resnet50, DenseNet, Vgg16), using majority vote to make a prediction. Those gave worse results than with the best Resnet50 based model. This might be because all models could not be trained as thoroughly as the best one. I could have tested weighted votes to better identify model confidences. I finally tried made the models vote only when the preferred model was not confident enough ($max(output) < 0.9$) for the maximum output. This improved the test score to 82% which is the best result I achieved overall.

## 5. Conclusion

Pretrained architectures gave good results on the data, however, preparing and cropping was a necessary step in the pipeline to get more decent results. To further improve these results, I would experiment on the different pretrained architectures to better tune the derived models, and explore more sophisticated ways of voting for model ensembles, as some of which I tested were promising.