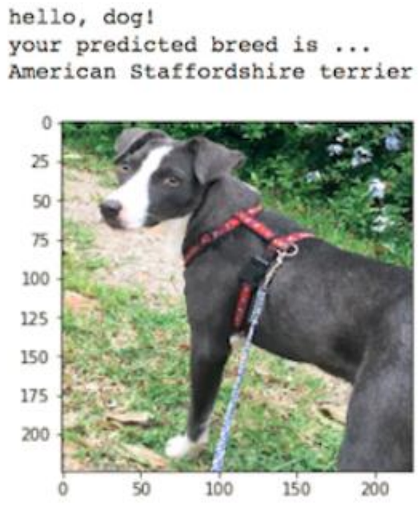
Capstone project proposal
Dog breed classification

Author: Matija Varga

Date: December 3rd 2020

Introduction

This project will result in an app within a Jupyter Notebook that will classify images. If there is a dog in the image, it will identify its breed. If there is a human in the image, the app will say to which dog breed does a human resemble. If there is neither a dog nor a human, the app will report that no dog or human could be detected in the image. Here is an example how the program should work:



```
hello, dog!
your predicted breed is ...
American Staffordshire terrier
```

Figure 1: Example output of the final application.

Domain background

The problem of classification of images belongs to the field of deep learning and here we will tackle it using convolutional neural networks. There are two distinctive parts of such a network: convolutional and fully connected neural part. The convolutional part decomposes and again recombines parts of an image through several layers. Output of the convolution is an input for the fully connected neural layer. The parameters of the convolutional part (e.g. values of the filter kernels) and parameters of the fully connected neural part (e.g. weights) are obtained in a training procedure (backpropagation and gradient descent using a optimizer). An overview and further literature on convolutional networks can be found in a review by Lecun et al.[1]

Problem statement

The problem consists of the following tasks: 1) finding a human face in an image, 2) finding the most similar breed to that of the human. If no human faces are detected, it is necessary to

[1] Yann Lecun et. al. Nature 2015, Figure 2.: link

determine if there is a dog in that image. If there is neither a dog nor a human in the image, a message should be displayed. The problem of dog breed classification is challenging due to the wide variety of classes in the classification problem (>100). In addition to that, the within class variance is large (e.g. different positions of dogs in an image) and between-class variance is small (e.g. different colors of Labradors).

Datasets and inputs

Dog and human datasets are provided by the organizer of this course. They consist of 13233 human images and 8351 dog images. Images have three color channels, different sizes and aspect ratios. Potential problems might arise from different aspect ratios of dog images and the fact that some pre-trained models only work with square images, i.e. the cropping might result in loss of relevant information (e.g. head of a dog). Images are split in train, validation and test sets and stored in different folders. Validation dataset is used after each epoch to determine whether there is an improvement in the model's performance and whether it should be saved. Finally, the performance of the dataset is reported on a test set. The dataset is not heavily unbalanced with most classes having 30-50 samples. The following image shows the distribution of classes in dog breed dataset:

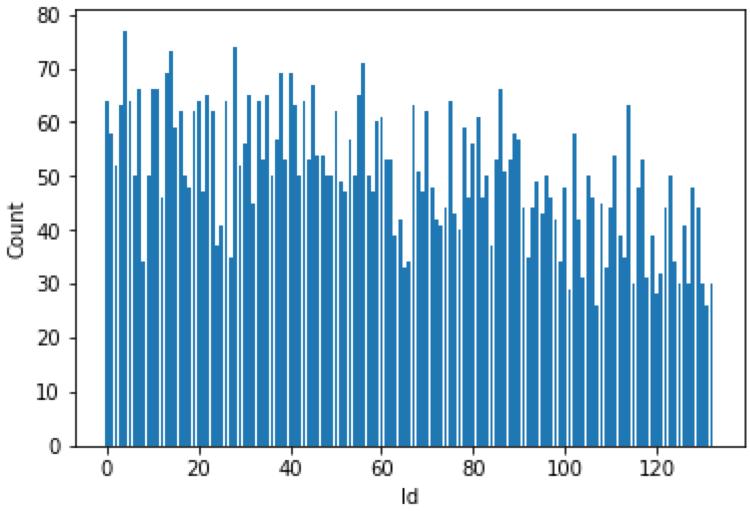


Figure 2: Distribution of dog breed (classes) within the dataset. Courtesy of [2]

Solution statement

[2] Jing Xian Lin: link

For the problem of dog breed classification one will need to use large networks with many convolutional layers (>10). To speed up the training procedure (with resources provided within the MLE course), we will use an existing human face detector (Haar feature-based cascade[3]) and pre-trained detectors for image classification such as VGG16[4]. The solution will be presented in three steps. The first step is to quantify the accuracy of a human face detector on a dataset of dog and human images. Then, another classifier is built that should determine a dog breed. Quantitative assessment (accuracy) of the dog classifier is obtained on a test set with dogs. The last step is to build an app which fulfils the requirements in the *Problem statement*.

Benchmark model

As a benchmark model for the most challenging part - dog breed classification, we will first construct a simple convolutional network with 2-3 convolutional layers and 2-3 fully connected layers. The goal here is to set a benchmark on the lowest value of accuracy and also to understand how complex should the final classifier be. It is expected that a simple benchmark classifier will have accuracy of ≈10%.

Evaluation metrics

We use accuracy as a performance for assessing both the benchmark and the final model (dog breed classification). For assessing the performance of human face detection we test the classifier on both human and dog images and report the accuracy. In that test (human face, no human face) the accuracy on dog images can be considered as a true negative rate.

Project design

The project is divided in seven phases:

1. Detecting human faces in images.

2. Detecting dogs in images using a pre-trained model such as VGG16.

3. Creating a simple (benchmark) classifier of dog breeds from scratch.

4. Creating the final dog breed classifier using transfer learning on a pre-trained convolutional neural network.

[3] P. Viola, M. Jones, Conf. on Computer Vision and Pattern Recognition, 2001, link
[4] K. Simonyan and A. Zisserman, arXiv:1409.1556

5. Writing an app according to the problem statement.

6. Testing the app in Jupyter Notebook with the provided and own images.