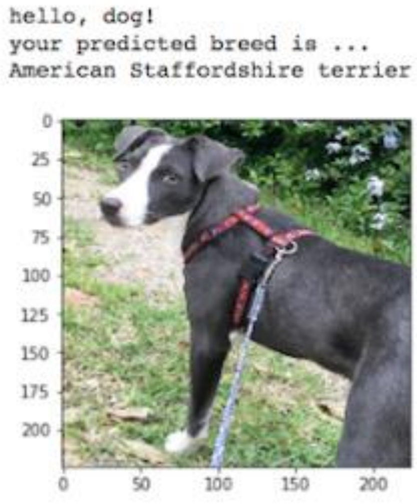
Capstone Proposal

Machine Learning Engineer Nanodegree

Ayran Olckers
Udacity
December 28, 2019



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

Domain Background

Convolutional Neural Networks (CNN) project “Dog Breed Classifier” is a Udacity suggested Project. This project hopes to classify dog breeds from images. This is a fine-grained classification problem: most of the breeds share similar body features and overall structure, so differentiating betIen breeds is a difficult problem (Such as four legs, tail, fur, etc.). Furthermore, there is low inter-breed and high intra-breed variation; in other words, there are relatively few differences betIen breeds and relatively large differences within breeds, differing in size, shape, and colour. The aim is to develop an algorithm which has code that will accept any user supplied image as an input in order to identify a dog breed from image set, If a dog is then detected in the image, it will output an estimate(likelihood) of the dog’s breed. If the supplied image of a human, the code will identify the resembling dog breed.

This problem is not only challenging but also its solution is applicable to other such classification and image recognition problems. For instance, the methods used to solve this problem would also help identify breeds of cats and other animals. Any set of classes with a relatively small variation within it can be solved as a fine-grained classification problem. In the real-world, an identifier like this could be used in biodiversity studies, helping scientists save time and resources when conducting studies about the health and abundance of certain species populations. Breed prediction can also assist in identifying dogs that need medical care. Ultimately, I have found dogs to be the most interesting class to experiment due to the large data set, alas I could have chosen any popular animal, but I also hope to expand my understanding of the fine-grained classification problem.

Problem Statement

The goal is to build a model capable of identifying which breed (label) a dog belongs to, based only on its image which is a fine-grained classification problem. There are several factors that make the problem of dogs classifying a challenge such cross breeds and shared similar features. The purpose of the project is to use a convolutional neural network (CNN) for the classification of dog breeds which allows computers to automatically extract hierarchies of features from raw pixels. Transfer learning approach will be applied to increase the accuracy which focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. Since the feature extraction process is the most complex modelling challenge, reusing it allows you to train a new model with less computational resources and training time.

Datasets / Inputs

The datasets, which I plan to use are provided by Udacity in pre-filled Jupyter notebook (*dog_app.ipynab*) which includes 8351 dog & 13233 human images
<u>File descriptions :</u>
● /dog_images: Dog Images
● /lfw: Human images
The most important file is */dog_images* training set images, there are 8351 training examples. In this case, it was trivial, as Udacity provided 1.08Gb of dog images spanning 133 different breeds, already in a proper file structure means files are segregated by training, validation, and testing, and further segregated within these folders by dog breed.

Solution Statement

This capstone project proposed breed classification and identification based on a convolutional neural network. The model is extended by applying transfer learning on the given datasets that shifts pre-trained Visual Geometry Group (VGG) model to the next model and consequently increased the resilience and efficiency of the model. Subsequently, the model is tested by the given data sets to validate its accuracy by returning the corresponding dog breed of human or dog images.

Benchmark Model

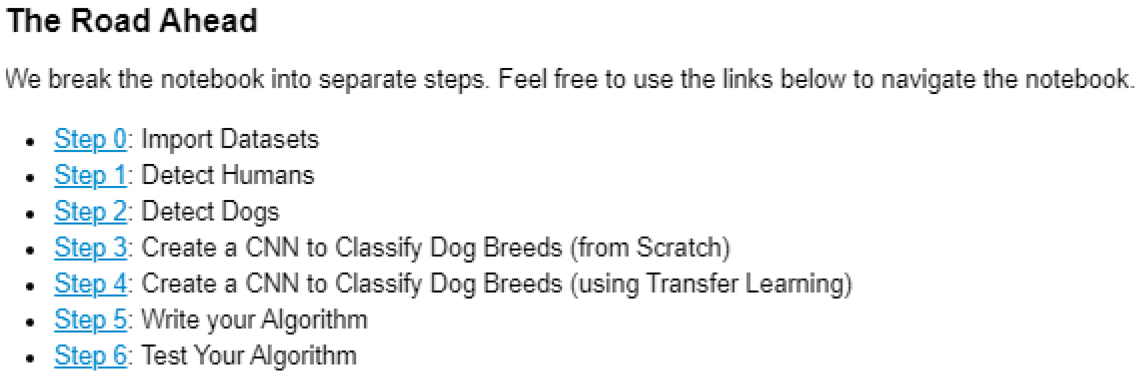
For the sake of benchmarking, I will restrict the CNN till 100 Epochs and extend by using Transfer learning which trains and validates the classification of dog breeds. Reason for this is, conventional face recognition methods have been surpassed by deep learning methods based on convolutional neural networks (CNNs) with high accuracy and robustness acquired by learning from actual deviations appearing in the images.

Evaluation Metrics

I am only going to focus on an accuracy score. The goal of what I am looking to do here is: I want to see how I can do at classifying breeds of dogs and evaluate the quality of the classifier by asking it to predict labels for a new set of images. I will then compare the true labels of these images to the ones predicted by the classifier and Accuracy will be able to tell us in a simple and easy-to-understand way how Ill our deep learning model is performing in this regard.

Project Design

The official Project design criteria used in ***dog_app.ipynab*** defined as following:



The Road Ahead

We break the notebook into separate steps. Feel free to use the links below to navigate the notebook.

- Step 0: Import Datasets
- Step 1: Detect Humans
- Step 2: Detect Dogs
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)
- Step 5: Write your Algorithm
- Step 6: Test Your Algorithm

Step 1: use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images
Step 2: use a pre-trained model to detect dogs in images.
Step 3: Create a CNN that classifies dog breeds. You must create your CNN from scratch (so, you can't use transfer learning yet!), and you must attain a test accuracy of at least 10%. In Step 4 of this notebook, you will have the opportunity to use transfer learning to create a CNN that attains greatly improved accuracy.
Step 4: use transfer learning to create a CNN that can identify dog breed from images. Your CNN must attain at least 60% accuracy on the test set.
Step 5: Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,

- if a dog is detected in the image, return the predicted breed.
- if a human is detected in the image, return the resembling dog breed.
- if neither is detected in the image, provide output that indicates an error.

Step 6: In this section, you will take your new algorithm for a spin and Test it!

References:

http://ijarcet.org/wp-content/uploads/IJARCET-VOL-5-ISSUE-12-2707-2715.pdf
https://www.rivas.ai/pdfs/mulligan2019dog.pdf
http://pytorch.org/docs/
https://medium.com/@hazutecuhtli/udacity-dog-breed-classifier-96da5e492f96
https://habr.com/en/post/447732/
https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
https://programqa.com/question/53942331/
https://github.com/pytorch/pytorch/issues/15563
https://s3-us-Ist-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip
https://s3-us-Ist-1.amazonaws.com/udacity-aind/dog-project/lfw.zip
https://www.youtube.com/watch?v=umGJ30-15_A
https://www.researchgate.net/publication/325384896_Modified_Deep_Neural_Networks_for_Dog_Breeds_Identification
https://arxiv.org/pdf/1811.00116.pdf