Machine Learning Engineer Nanodegree
Capstone Proposal

Mohammed Aydid Hasan

January 21st, 2020

Dog Breed Classification

Domain Background:

Image classification has multiple uses and one of the most used skill that is using by most of the manufacturing companies as well as tech companies. In fact, in medical science, image classification is a hot topic. Industries are using images of different affected areas to identify diseases. Here, dog breed classification used the same technique. Majority of dogs are often difficult to classify by simply looking, and breed identification is important when rescuing dogs, finding them forever homes, treating them, and various other furry situations. The goal of this capstone project is to apply Deep Learning techniques to the classification of dog breeds where the code will accept any user to supply image as an input. If a dog is detected in the image, it will provide an estimate of the dog's breed. And if a human is detected, it will provide an output where it reflects the result and if neither is detected in the image, it provides output that indicates an error. But mainly our concern is to identify dog breeds from any image.

Problem Statement:

The main objective of this project will be to use Deep Learning techniques to build a classifier that could first detect whether a dog or a human was detected in the provided image and then, classify the detected dog into one of over 100 dog breed categories (and in case of a human, what dog breed the detected human looks like). When given an image sample in a computer readable format (such as a .jpeg/.png file), we want our model to be able to determine if it contains one of the target class with a corresponding likelihood score. Conversely, if none of the target image were detected, we will be presented with an unknown score. This project uses Convolutional Neural Network with Transfer learning to classify the dog breeds. For achieving the accuracy, I used RestNet50 as a transfer learning method.

Datasets and Inputs:

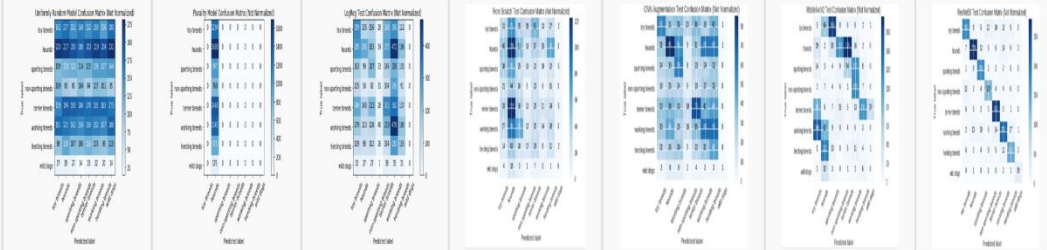
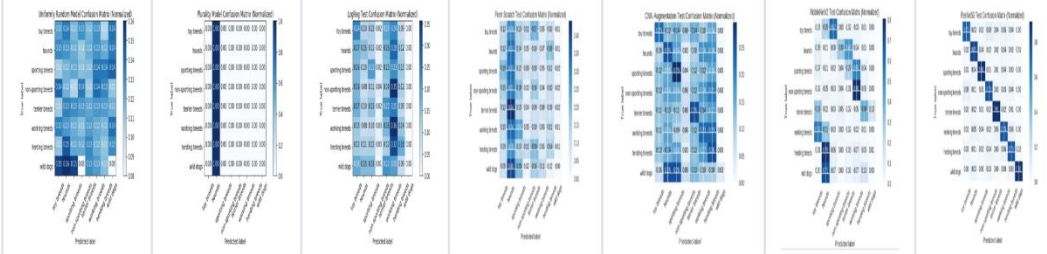
The data set imported from sklearn provided in a Udacity's workspace consists of 8351 dog images in 133 different categories. The project will be done at Udacity's workspace (GPU-enabled) with TensorFlow in Keras. To work for a model, we need to split the dataset into training set, validation set and testing set. The model will be tested on 100 images of each (human and dog). There will be 6680 training dog images, 835 validation dog images and 836 test dog images. The human face detector will be relied on OpenCV's Haar Feature-based Cascade Classifiers. It will be trained on 13233 human faces which will be imported from sklearn. And the dog detector will be built on a pre-trained ResNet50 model. Its purpose is to confirm if an image is a dog or not. Input should be any sort of computer readable image.

Solution Statement:

To train a Deep Neural Network with these data sets and make predictions, I will try it by building a simple CNN from scratch with the help of transfer learning which in this case will be RestNet50. CNN uses relatively little pre-processing compared to other image classification algorithms. It takes advantage of the hierarchical pattern in data and assembles more complex patterns using smaller and simpler patterns. With the help of CNN, we can use the large amount of data more effectively and accurately. Therefore, CNN is typically applied for Image Classification from an input image with a single object and to an output as a class label from a list of object categories. I will use 5 convolutional layers where the filter size will be increased from 16 to 32 to 64 to 128 to 256 as it is a standard practice in CNNs. And I will select 15 epochs because that is all it will take to get desired accuracy.

Benchmark Model:

For this problem, I will try to beat the existing model by CNN using RestNet50 transfer learning. From the image below, we can see that the accuracy score for RestNet50 model is 83.28%. I will try to make it more accurate for the same model by using different hyperparameters and tuning.

Model:	Random	Plurality	Logistic	CNN From Scratch	CNN with Data Augmentation	MobileNetV2	ResNet50
Categorical Loss	30.2838	27.7437	1.9771	5.8862	2.5033	0.7060	0.6864
Accuracy Score	12.32%	19.67%	20.26%	22.54%	22.88%	79.51%	83.28%
Time	< 1 sec	< 1 sec	19.6 mins	17.8 mins	18.5 mins	20.8 mins	50.3 mins
Confusion Matrix							
Confusion Matrix (Normalized)							

Source: https://hljames.github.io/dog-breed-classification/

Evaluation Metrics:

The evaluation metric for this problem is simply the Accuracy Score. The most popular metrics used in CNN are “accuracy”, “binary accuracy”, “categorical accuracy”, “sparse categorical accuracy”, etc. But as we are classifying dog breeds, so we can’t use binary or sparse categorical accuracy. Accuracy is the ratio of number of correct predictions to the total number of input samples. However, it works well only if there are equal number of samples belonging to each class. In this project, there are total 133 dog breed as class labels. Based on the distribution of training/validation/testing selected, the classes were approximately evenly distributed, except a couple of classes. Therefore, we should be able to use “accuracy” metric in training.

Project Design:

First identify the different data types in our dataset and what pre-processing needs to be done to make it uniform.

Data Splitting
Split the data into a training set and validation set with an 80-20 split.

Model training and evaluation
I will start with the simple model architecture first before training and evaluating it. Then iterate this process trying different architectures and hyper-parameters to reach an accuracy score we are happy with.

References:

1. https://machinelearningmastery.com/handwritten-digit-recognition-usingconvolutional-neural-networks-python-keras/
2. https://www.kaggle.com/sdelecourt/dog-breed-classification
3. https://medium.com/@das.kirtirjasaswini/dog-breed-classifier-using-cnnmodel-data-science-capstone-project-8207a43321eb
4. https://mc.ai/dog-breed-classification-using-cnn/
5. https://hljames.github.io/dog-breed-classification/