Machine Learning Engineer Nanodegree **CHANG YONG LIK**
CAPSTONE PROJECT PROPOSAL **2021 May 27**

CNN Project: Dog Breed Classifier

Domain Background

The purpose of this project is to classify different dog breeds. There are a lot of dog breeds in the world, so distinguishing each dog breed is a challenging task. In this project, we are going to design a system and algorithm to solve the classification problem of dog breed. We would need to classify 133 dog breeds using convolutional neural networks (CNN). After completing this project, I am planning to build a simple phone app with Flutter using Tensorflow Lite. I hope I can learn how to combine and fully utilize the cross disciplinary industry knowledge of software development and machine learning.

Problem Statement

The aim of this project is to build a model that can accept an image and then distinguish whether that image is a dog or a human. If a user submits a dog photo, the model has to correctly predict its breed; and if a user submits a human photo, the model has to correctly predict the most resembling dog breed. So, we can divide the task up into two parts:

Dog face detector: Given a Human face image, model will predict its breed

Human face detector: Given a Human face image, model will predict most resembling dog breed

Datasets and Inputs

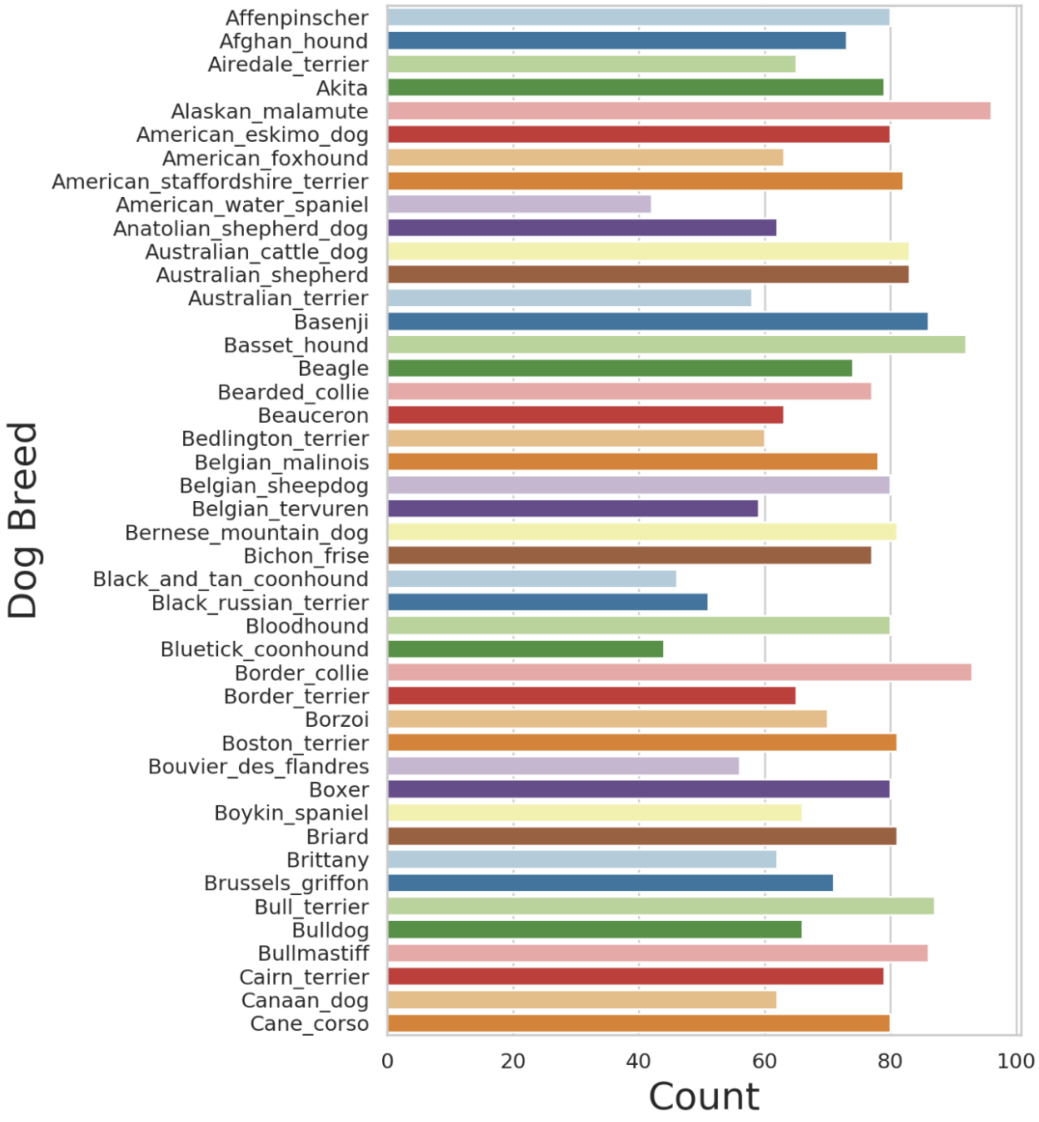
Inputs
Images of either dogs or humans

Dataset
The dataset is provided by Udacity and comprises two parts - dogs and humans.

The dogs dataset is composed of:

- Training: 6680 images
- Validation: 835
- Test: 836
- Total images: 8351
- Classes (Dog Breeds): 133

Dog images are separated into 3 folders, namely train, test and valid. The folders are then further separated into 133 folders each representing the total number of dog breeds. Here is a distribution of dog breeds downloaded from Udacity.



Dog Breed Distribution

The human dataset is composed of 13233 human images. Each image is grouped together in a folder named after its human owner. There are a total of 5749 folders provided. Images are distributed evenly among the folders.

Before we begin training, all of the images are resized to 244x244 and normalized before being used with the model. (CNN works best when the inputs are normalized to a range of 0~1. Then again, most algorithms do.)

Solution Statement

The problem at hand is a classification problem. This type of problem can be easily solved by using CNN. We can extract the features efficiently from an image with CNN. To further increase our work efficiency, we would implement the following solutions.

- Human face detection - Utilizing OpenCV's Haar Cascades[1] classifier to detect face.
- Dog image detection - Utilizing transfer learning (VGG16[2]) that was trained on an ImageNet challenge.
- Breed Classification - We could further increase our training efficiency by using a pretrained Resnet-101[3].

Benchmark Model

We can break down the workflow in the following with respective benchmark model to compare:

- VGG156 model used for dog breed detection should have a high accuracy of more than 60%.
- The CNN model that we will build from scratch should have an accuracy of more than 10%.

We will try to beat the VGG158 with the CNN model that we build from scratch.

Evaluation Metrics

This is a multiclass classification problem of dog vs human. In this classification problem, we will use multiclass log-loss to evaluate the model. The dataset is imbalanced, so we would need a good metric other than accuracy to gauge the performance of the model. Precision and recall are great choices for model evaluation.

Project Design

The project would be divided in the following steps:

1. Performing exploratory dataset analysis (EDA).
2. Detect human faces using PenCV's Haar cascade classifier.
3. Detect dogs using pretrained VGG16 model.
4. Create a CNN to classify dog breeds

5. Combining dog and human detectors - First check if the image is dog, if so return dog breed prediction. If human face is detected, return the most resembling do breed, else return error.
6. Test the system and algorithm with some random images found online.

References

1. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
2. https://neurohive.io/en/popular-networks/vgg16/
3. https://neurohive.io/en/popular-networks/resnet/
4. https://www.kaggle.com/c/dog-breed-identification