ML Engineer Nanodegree Capstone proposal: **Dog Breed Classifier**

1. Domain background

It is a challenging task to recognize dogs according to their respective breed.
There are hundreds of breeds in existence which are grouped into
10 distinct groups (Group 1-10) according to physical characteristics. This
shows that classification is a complex task as remembering all the breeds and
distinguishing between similar breeds cannot be easily done by humans. Therefore, the
need for this to be done by a computer is necessary. The task at hand
involves image recognition and classification which uses machine learning techniques to
aid computer vision. A convolutional neural network (CNN) will be
used to accomplish this task.

2. Problem statement

The aim of the project is to build a pipeline to process real-world, user-supplied
images. The algorithm will identify an estimate of the dog's breed given an
image. When the image is of a human, the algorithm will choose an estimate of
a dog breed that resembles the human. If neither a dog or a human is detected,
then an error message is output. Therefore, the models in place should be
capable of detecting a dog or human in an image, classify the dog to its breed
and classify a dog breed that the human resembles.

3. Datasets and inputs

All the datasets used to train, test and validate the CNN model are provided by Udacity.
There is the dog dataset containing 133 breeds and each breed contains 8 images.
There is also the human dataset which contains images, first names and last names.
There are a total of 13233 human images and 8351 dog images. The human dataset will
be used to detect human faces in images using OpenCV's implementation of Haar
feature-based cascade classifiers. The dog dataset is used to detect dogs in images

using a pre-trained VGG-16 model. The datasets of human and dog images are justified for this project given the problem statement above.

4. Solution statement

The project will first use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces from the images supplied. Next, a pre-trained VGG-16 model with trained weights on ImageNet, a large popular dataset for image classification, will be used to detect dogs in the user-supplied images. The pre-trained model should be checked that it returns indexes from 151 to 268 inclusive, that is, it must include categories from 'Chihuahua' to 'Mexican Hairless'.A CNN model is then built from scratch to classify dog breeds, that is, transfer learning cannot be used just yet. This model should surpass a test accuracy of 10% set by Udacity because the model is being built from scratch so classifying similar breeds can be a challenge however transfer learning will greatly improve this. Finally, a transfer learning will be used with a ResNet50 model to significantly boost the accuracy of the CNN model. It should surpass the 60% test accuracy set by Udacity.

5. Benchmark model

The pre-trained VGG-16 model I with trained weights on ImageNet, a large popular dataset for image classification, will be used as the benchmark model. ImageNet has over 10,000,000 URLs that link an image containing an object from one of 1,000 categories. The pre-trained model should be checked that it returns indexes from 151 to 268 inclusive, that is, it must include categories from 'Chihuahua' to 'Mexican Hairless' as stated in the solution statement.

6. Evaluation metrics

Accuracy will be the main metric used to test both the benchmark model and the solution model. Accuracy $= (TP + TN) \div (TP + TN + FP + FN)$

True Positive(TP),
True Negative(TN),
False Positive(FP) & False Negative(FN).

7. project design

The workflow for the project to approach the solution is given below:

- **Step 0:**
  The datasets for the human and dog images are imported and the total number of dog and human images is determined.
- **Step 1:**
  OpenCV's implementation of Haar feature-based cascade classifiers is used to detect human faces from the datasets.
- **Step 2:**
  THE pre-trained VGG-16 model is used to detect dog images from the datasets.
- **Step 3:**
  CNN model is created from scratch to classify the images. Test accuracy should be greater than 10%. Resizing and cropping the image. Augmentation through rotations.
- **Step 4:**
  CNN model is created using transfer learning to classify the images. Test accuracy should be greater than 60%. Resizing and cropping the image. Augmentation through rotations. CNN layers are utilized for feature extraction and image generalization. The dropout layer ensures that overfitting is dealt with and the linear classifier converts the extracted features to a classified type.
- **Step 5:**
  An algorithm is developed that returns the predicted breed if a dog is detected. If a human is detected, it should return the resembling dog breed. If neither dog or human is detected, then an error message is output.
- **Step 6:**
  Testing the solution model with images from the datasets to see the model in work.