MACHINE LEARNING ENGINEER NANODEGREE

CAPSTONE PROJECT: DOG BREED CLASSIFICATION USING CNN

PROPOSAL BY:
ROHAN BHAGWATKAR

1. **Domain Background:**
   Image classification is one of the most exciting topics in the research domain. Multi-Class classification is a complex task. The task at hand requires us to classify dogs into their different breeds. This is a very tough task when tried to be done using human eyes because every human does not possess that amount of knowledge and accuracy in classifying the breed. This can be easily done by training a CNN model and asking it to predict the breed for us. A CNN model is chosen as it works efficiently with image data.

2. **Problem Statement:**
   We need to develop a model which when trained can produce accurate results in predicting the dog breed from the photo that has been provided as input to the system. If the photo is of a dog then the model will predict its breed else if it will try to predict the most resembling breed to the provided photo.

3. **Datasets and Inputs:**
   The dataset is supplied by the Udacity MLE Nanodegree Program.

TYPE	DOG (Images)
Train	6680
Test	836

Valid	835
Total	8351

Human dataset size: 13233 (5749 dirs)
The dataset is imbalanced. Some image processing needs to be done beforehand to make all images of the same size before passing them through our model

4. **Solution Statement:**
   To complete this project, we will first use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces from the images. Now to detect dogs, the VGG-16 model will be used with trained weights. This will prove to be a good model to detect dogs as it is already trained on huge datasets. At the end of this process, we will pass the image through the model and predict the breed.

5. **Benchmark Model:**
   The least expected accuracy from the CNN model that is built from the scratch is 10% and the model that we will be developing using transfer learning should have an accuracy of at least 60%.

6. **Evaluation Metrics:**
   Accuracy can be considered as a metric when the dataset is balanced. Because of the imbalance in the dataset, accuracy is not a good indicator here to measure the performance. We will take log loss into account to evaluate the model.

7. **Project Design:**
   Step 1: Import datasets and required libraries.
   Step 2: Pre-process the image data and segregate it into a train, test, and validation dataset.
   Step 3: Use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces from the datasets.

Step 4: Use a pre-trained VGG-16 model to detect dog images from the datasets.
Step 5: Model a CNN to classify dog breeds from scratch and train, validate, and test the model.
Step 6: Model, train and test a CNN to Classify Dog Breeds using Transfer Learning
Step 7: Final algorithm includes combining results of dog and human detector models and then we should finally be able to predict the dog breed.

8. **References:**

1. Pytorch Documentation
2. CNN
3. OpenCV
4. Image Classification