Domain Background:

Dog Breed Classification is a very specific application of convolutional neural networks. It falls under the category of fine-grained image classification problem, where inter-class variations are small and often one small part of the image considered makes the difference in the classification. This methodology requires a significant amount of images as training data and substantial time for training and achieving higher accuracy on the classification. Transfer Learning can be used to overcome this problem. Through Transfer learning, a pre-trained model can be fine-tuned to perform classification on image datasets that may be outside the domain of the pretrained model. Due to the limitation of image datasets available for dog breeds, we propose to use transfer learning to perform Image Classification of different dogs.

Problem Statement:

Developing an algorithm that could be used as part of a mobile or web app. At the end of this project, my code will accept any user-supplied image as input. If a dog is detected in the image, it will provide an estimate of the dog's breed. If a human is detected, it will provide an estimate of the dog breed that is most resembling.

Datasets and Inputs:

Dog images data set was provided by Udacity which can be found here:
https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip
Human faces data set was also provided by Udacity can be found here:
https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip
There are three folders in the datasets ie train, valid and test which are used for training, validation and testing the model respectively. The lfw dataset can be used for human face training and can also be used for testing the dog breed classifier for an estimate of the dog breed.

Solution Statement:

For solving this problem statement, the following steps were taken as a roadmap:

- 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: Writing the Algorithm
- Step 6: Testing the Algorithm

Benchmark Model:

There are various research papers based on the Dog Breed Classifier, some of them are models designed from scratch as Dog Breed Identification which uses computer vision and machine learning techniques to predict dog breeds from images. It predicts the correct dog breed on its first guess 52% of the time; 90% of the time the correct dog breed is in the top 10 predictions.

Another paper Using Convolutional Neural Networks to Classify Dog Breeds which aims to use a convolutional neural network framework to train and categorize dog breeds. I approach this first using CNNs based on LeNet and GoogLeNet architectures.

I also went through one paper which uses Transfer Learning to classify the dogs. Transfer Learning for Image Classification of various dog breeds which solves the problem of huge dataset requirement by using the Transfer Learning as through Transfer learning, a pre-trained model can be fine-tuned to perform classification on image datasets that may be outside the domain of the pretrained model. They have used Google's Inception-v3 model trained on 100,000 images covering 1000 different categories and achieved an overall accuracy of 96% on the images taken from Google.

Evaluation Metrics:

Here the model can be evaluated based on the accuracy obtained on the Test dataset. Furthermore, I will test my model against the few random images from my computer and compare it with the previous results.

Project Design:

For designing the appropriate model, I followed the instructions mentioned in the classroom for the dog breed classifier:

- Step 0: Import Datasets: As the datasets are already provided in the classroom, so there was nothing to be done here.
- Step 1: Detect Humans: For detecting the humans, first I used the haarcascades as suggested in the classroom, which 98% accuracy on human faces dataset and 17% of the dog's images are also detected as human faces, as the accuracy was low I decided to try another method, so I used the dlib inbuilt library to detect human faces which given me an accuracy of 100% on human faces and 5% of the dogs faces are detected as human faces.
- Step 2: Detect Dogs: For detecting the dogs I used the pretrained vgg16 model in pytorch and achieved an accuracy of 95% on dogs dataset and 0% of human faces are detected as dog face, furthermore as an optional method I tried Resnet50 from Keras which improved the accuracy to 100% on dogs dataset.
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch): For this step, I designed a CNN model with 5 CNN layers with max-pooling layers and after flattening I applied two-layer perceptron which I trained from scratch to obtain a minimum accuracy of 10%.
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning): Now I will use transfer learning to create a CNN that can identify dog breed from images.
- Step 5: Writing the Algorithm: In this step, I will 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 an output that indicates an error.
- Step 6: Testing the Algorithm: For testing the algorithm I will use at least six images which consist of at least two human and two dog images.