# Udacity ML Engineer Nanodegree

## Capstone Proposal

John Nikhil Kollanoor
September 7th, 2020

### Domain Background

Machine learning techniques has accelerated the rate at which humans analyze and extract meaningful insights, predict certain events and recognize certain patterns. Image recognition and pattern detection is one of the most studied aspect of machine learning techniques.

Even though the field of computer vision exists since 1960s with early algorithms such as Canny edge detection by John F Canny, the increase in computing capabilities over the years gave algorithms and techniques such as Deep learning, Gradient Boosting the means to solve complex problems only humans were known to solve with a great accuracy. With advent of these techniques along with available open source libraries and cheap computing power image classification and recognition is no longer a sponsored research project at universities but a project for anyone who is passionate about solving problems and experiment with the free tools available out there to create useful models.

Using deep learning techniques, I intend to classify different breeds of dogs by training a model with labeled images of dogs with their respective breeds and also identify human faces.

### Problem Statement

Often times people wonder what breed a dog is when they see one walking looking quite happy and wagging its tails. Some of us have quite a bit of information about this and are able to recognize the breed immediately. But what about technology doing the work for us and recognizing the dog breed with a decent accuracy? That would be great! In this project, I intend to classify the dog breeds by training a model using labeled images of dogs.

### Datasets and Inputs

The dataset available has 13233 images and 8351 dog images provided by Udacity for the purpose of this project.

Dog images dataset are partitioned into train, test and validation directories. Each of these directories has 133 dog breeds. The data is unbalanced as the training data(images) for each breed is not the same. Also, the size of the images vary.

Human images dataset is of the same size (250 x 250).

Dog images are used to train and classify dog breeds. Human images are trained to retrieve the closest resembling dog breed.

## Solution Statement

CNN model is trained and implemented to solve this problem since its great at pattern recognition and analyzing images. Pretrained model from OpenCV is used to detect human face and to identify the dog and breed pre-trained VGG16 model is used. Transfer learning method is used as this would significantly reduce the training effort and prevent reinventing the wheel.

## Benchmark Model

CNN model built from scratch must have at least 10% accuracy. This is to have a significant result as probability of a random guess yielding a correct answer is 1/133 which is less than 1%

CNN model created using transfer learning must have an accuracy of at least 60% as we use information that already is processed and modeled.

## Evaluation Metrics

To balance accuracy, precision and recall we use F1-Score as the evaluation metric.

## Project Design

The following steps are implemented

1. Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.
2. Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.
3. Create dog detector using pretrained VGG16 model.
4. Create a CNN to classify dog breeds from scratch, train, validate and test the model.
5. Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101. architecture. Train and test the model.
6. Write an algorithm to combine Dog detector and human
    - If dog is detected in the image, return the predicted breed.

- If human is detected in the image, return the resembling dog breed.
- If neither is detected, provide output that indicates the error.

## References


1. Resnet101: https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101
2. Pytorch Documentation: https://pytorch.org/docs/master/
3. ResNet Overview - https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035