Capstone Project
Machine Learning Engineer Nanodegree

Bhadresh Savani

# Dog Breed Classifier using CNN

**Domain Background:**

Its well-known real-world problem. Given an image of a dog we need to identify the breed of dog. If human image is given as an input, we need to find a dog breed resembling dog breed. It's a multiclass classification problem which uses supervised machine learning. I choose this project because it involves work to build an end to end pipeline. I wanted to build a complete application which takes image from UI and show correct prediction.

**Problem Statement**

To build an application which takes input from UI either dog image or human image, predict a dog breed resembling to input image and show the result on UI to the User.

- 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 output that indicates an error.

**Datasets and Inputs**

The dataset we are going to use here is provided by Udacity. We have 13233 images of human and 8351 dog images. We have 133 classes of dog.

The data is distributed in folder according to the classes so that dataloader can identify image class based on folder name for the dog data. While for Human data. Images are stored in folders according to Human names.

**Solution Statement**

First, we will identify image as if its human image or dog image using Opencv based image processing. Our **Convolution Neural Network (CNN)** based deep learning network build using Pytorch framework and trained using Training data will be used to identify features and patterns from the given image and identify the dog breed out of 133 classes having similar facial pattern.

**Benchmark Model**

- We will build a basic CNN based Model with few layers. It should be able to give us at least 10% accuracy on test set. If not, we will add more model and change model Hyperparameter to achieve this much prediction accuracy.
- We will use pretrained CNN based models like Resnet to take advantage of transfer learning. Such Networks are trained on huge data and having larger structure so it's more capable of identify pattern from the data. We will try to achieve at least 60% accuracy on test set.

**Evaluation Metrics**

We are having a greater number of classes and imbalanced data accuracy will not be good indicator in such situation. We will use log loss to get class with largest score.

**Project Design:**

Step1. Import libraries and data. Perform image pre-processing and augmentation and create train, test and validation set.

Step2. Detect Human using Opencv based Haar Cascade detector.

Step3. Build a VGG16 based dog detector model

Step4. Create a CNN to Classify Dog Breeds (from Scratch)

Step5. Create a CNN to Classify Dog Breeds (using Transfer Learning)

Step6. Write an Algorithm to return the following thing based on below three things:

1. if a **dog** is detected in the image, return the predicted breed.
2. if a **human** is detected in the image, return the resembling dog breed.
3. if **neither** is detected in the image, provide output that indicates an error.

References:


1. *Original GitHub Repository: https://github.com/udacity/deep-learning-v2pytorch/blob/master/project-dog-classification/ Dataset to be downloaded from the link provided in the notebook*
2. *https://medium.com/@fzammito/whats-considered-a-good-log-loss-in-machine-learning-a529d400632d*
3. *Resnet101: https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101*