# Dog Breed Classification

***A project proposal for the Udacity Machine Learning Engineer Nanodegree***

**Author:** Gavin Ayres **Date:** April 2021

## Domain Background

This project is primarily concerned with the field of computer vision and specifically in an applied setting. The task at hand, classifying dog breeds, is a challenging one, even for humans. However, it is precisely this challenge which makes it an important illustration of the power of deep learning approaches to computer vision. It is a particular aim of this project to achieve significant accuracy on the test set (>60%), with our dog breed classifier, as a means of illustrating the speed to development of deep learning approaches to certain problem representations. The motivation for this project choice is primarily because the author has limited previous experience in computer vision tasks and would like to further develop some proficiency.

## Problem Statement

The problem to which this project will provide a solution is that of dog breed classification. More precisely, given an input image of a dog we aim to assign it to a specific breed. If we are shown an image of a human we will assign it the dog breed of highest association. The solution will therefore be required to implement two high level features, filtering input images to ones we can appropriately classify and then assigning *dogs* and *humans* to breeds.

## Datasets and Inputs

The datasets required for this project consist of, as one would expect, images of dogs and humans(1). Necessarily, these datasets are separated by species. The *dogs* dataset is broken up into train, test and validation folders with subfolders consisting of breeds and their images. The amount of images that make up each breed directory are imbalanced and consist of different size and quality. The *humans* dataset are subsetted by name and again vary in size and quality.

## Solution Statement

We outlined previously that our solution would require two high-level features. We will now go into further detail. Starting from the end of our pipeline, the breed classifier we will train will be a convolutional neural network (CNN), the exact architecture will be decided upon via experimentation on the test data set. The component to identify dogs will be a pre-trained VGG-16 model(2). We will use Haar feature-based cascade classifiers(3) to identify humans.

## Benchmark Model

We have an implicit benchmark of ~10% accuracy on the test set (outlined in the notebook template). Random chance (1 in 133) will set a baseline of <1%. The CNN must have at least 60% accuracy on the test set.

## Evaluation Metrics

We will use categorical cross entropy as our loss function for the multi-class classification problem of assigning images to dog breeds. We will also consider typical classification metrics, such as accuracy, for each class.

## Project Design

This project will be completed upon successive completion of the following steps;

- Importing required datasets and libraries.
- Implement OpenCV(4) pre-trained face detector (Haar feature based classifiers).
- Implement PyTorch(5) pre-trained VGG-16 model to detect dogs.
- Build and train CNN to classify dog breeds.
- Implement dog breed classifier using transfer learning (adapting a pre-trained computer vision model from pytorch).
- Assemble detectors into an algorithm which executes logic depending on the input image. Specifically, if a dog is detected in the image we will return the predicted breed; if a human is detected in the image we will return the associated breed; finally, if neither are provided we will return an error.
- A final qualitative evaluation of the trained model on sample images.

## References


1. Udacity GitHub repository with notebook template; https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification.
2. Simonyan, K. and Zisserman, A. Very Deep Convolutional Neural Networks for Large Scale Image Recognition. (ICLR 2015) https://arxiv.org/pdf/1409.1556.pdf.
3. Viola, P. and Jones, M. Robust real-time face detection. (International Journal of Computer Vision 2004) https://www.face-rec.org/algorithms/boosting-ensemble/16981346.pdf.
4. OpenCV; https://docs.opencv.org/3.4/index.html.
5. PyTorch; https://pytorch.org.