

## **Capstone Project Proposal:** **Dog Breed Classifier**

### **Domain Background:**

Image classification has been a problem that interested researchers for many years and most attempts at coding a solution failed until Yann LeCun introduced Convolutional Neural Networks (CNNs).

CNNs are what revolutionized image classification because they mirror our own neurons in the way they work meaning they're able to recognize patterns through analyzing images.

Sources: [Wikipedia](#), [Siraj Raval](#), [Ksenia Sorokina](#)

### **Problem Statement:**

The problem at hand is image classification using Convolutional Neural Networks (CNNs). We'd like to be able to give an image as an input and have a model predict whether the image corresponds to a dog, a human or neither. Additionally, if the input is a dog image, we'd like for the model to predict the breed. If the input is a human, we'd like the model to predict the closest dog breed to that human.

### **Datasets and Inputs:**

The dataset we're using is the dog dataset and human dataset provided in the Dog Breed Classifier project.

These datasets are images of dogs and humans that we can use to train, test and validate our model. The dog dataset contains images for 133 different breeds. It should be enough to be able to recognize a dog and its breed. The human dataset is smaller in size since human classification is not the main focus of this project.

Links to the datasets: [dog dataset](#), [human dataset](#)

### **Solution Statement:**

To solve this problem, we will be using a Pytorch neural network to train a model on our training datasets.

We will then test our model and be able to determine the level of accuracy through the rate of true predictions. Ultimately, we want a model that can be deployed on a web app and for a user to be able to input an image and have the model predict whether it's a dog, a human or neither.

### **Benchmark Model:**

We can compare our model to the pre-trained dog detector model given in the project.

We can also compare to a model I found on Kaggle using the Keras API. The model has an accuracy of 98%. I find that it will be interesting to see if I'm able to achieve the same level of accuracy with Pytorch. To view the Kaggle project in question click [here](#).

### **Evaluation Metrics:**

We will evaluate how our model performs with an accuracy score:

Accuracy = true predictions / total predictions

We will then compare this score to the benchmark models' accuracy scores.

### **Project Design:**

I will attempt to solve this classification problem by:

- Importing the datasets
- Testing the face detector algorithm
- Implement dog detecting code on a pre-trained model
- Assess the dog detecting model
- Create a CNN to Classify Dog Breeds (from Scratch)
- Train, test and validate the model
- Create a CNN to Classify Dog Breeds (using Transfer Learning)
- Write code that takes an image's directory as input and returns:
  - Whether the image is of a dog, human or neither
  - In the case of a dog predict the breed
  - In the case of a human predict the closest dog breed
  - In case it's neither return an error