Milestone Report

Computer Vision is a field of Artificial Intelligence and Computer Science that deals with how computers can be made to gain an understanding of digital images and videos. With the advances in machine learning technology, innovations in Image Processing are emerging rapidly and being leveraged by businesses all around the world. Applications of Computer Vision include:

1. Boredom killers such as your favorite Snapchat filter that use facial recognition and image processing to distort your face and/or overlay images
2. Potential lifesaving uses such as Medical startups that claim they'll soon be able to use computers to read X-rays, MRIs, and CT scans more rapidly and accurately than radiologists.
3. And borderline unethical uses such as Facial Recognition in police body cams

With so many emerging and interesting uses of Computer Vision, my goal is to compare different machine learning algorithms from scikit-learn and Keras tasked with classifying drawings made from Google's Quick Draw! dataset. By doing so I will showcase my abilities processing and analyzing the doodles through image classification and neural networks. This project is meant for companies that are interested in leveraging the Computer Vision methods used in my project in business.

Data

Google has capitalized on the use of crowdsourcing to label over 50 million drawings with their online game "Quick Draw!". It gives its users 20 seconds to draw one of 345 different classes that range from an aircraft carrier to a zig-zag. Recently, they have open-sourced all their data and I will select 10 images to build my classifier. The data we used is a 28x28 grayscale bitmap in numpy .npy format of the simplified version of each drawing.

Other datasets we could have used include:

- The *raw dataset* that contains a two letter country code of where the drawing originated, the drawing in a JSON array representing the vector drawing that details each stroke in the drawing and the timing it took to complete each stroke, whether the word was recognized by the game, the label, and the date/time of when the drawing was created.

- A *simplified version* of the raw dataset that has timing information removed and has the drawings positioned and scaled into a 256x256 region.

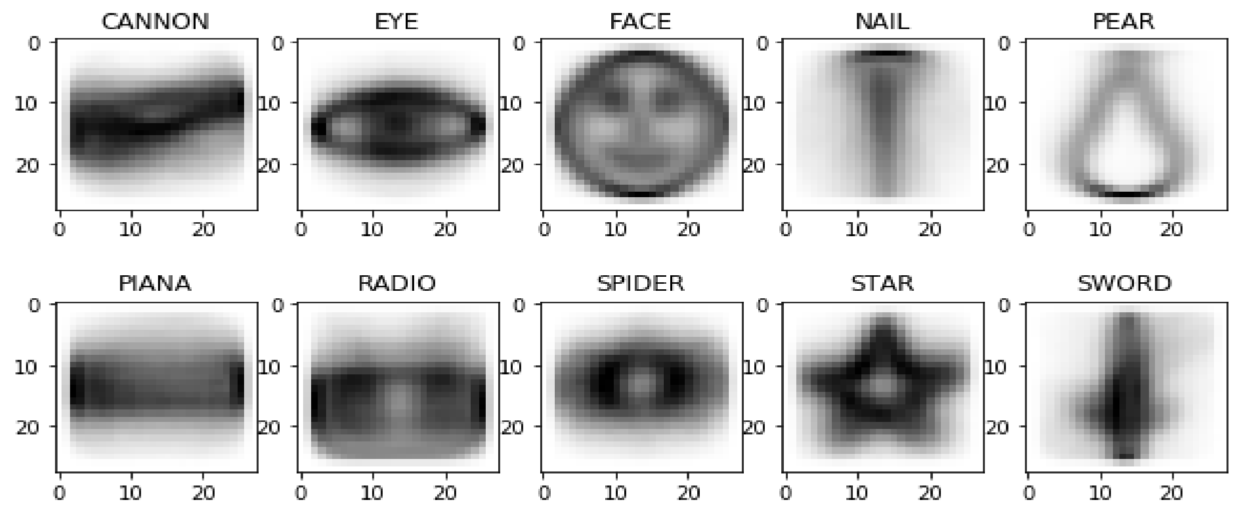
The **data wrangling** process consisted of:

- Loading each class separately into their own numpy array
- Adding the class label to the array of each sample in the dataset.
- The samples were then aggregated and split into training and testing sets.
- The features were normalized between 0 and 1, as a means to "center" the data and have all pixels on the same scale. This weights all features equally in their representation.

Initial Findings

We started our exploratory analysis by taking the first image in each classes' dataset and plotting them along side a histogram of the pixel intensities. The histogram showed that each doodle's pixels were mainly grouped at the two extreme of intensities with very few falling between. Most pixels in the dataset are completely white, along with another set of pixels that are completely dark, with relatively few in between. If we wanted to, we could probably replace each pixel with a binary 0 or 1 with very little loss of information.

From there we plotted the average doodle for each class and the average doodle of all classes together. The average doodle for each class really highlighted where the important pixel will be for each class.


CANNON
EYE
FACE
NAIL
PEAR
PIANA
RADIO
SPIDER
STAR
SWORD