

# MSAI-349-Final-Project - Identifying Sketches

---

**Harrison Bounds, Andrew Kwolek, Sharwin Patil, Sayantani Bhattacharya**

## Task Description

The task for this final project is to create a Convolutional Neural Network that can identify sketches from the user. The user will first draw the sketch using a digital canvas. This image will then be used as the input to the CNN. The goal is provide a probability that predicts the sketch the user has drawn. This is a classification task, where each labeled sketch the model is trained on represents a different class.

## Dataset

The dataset for this project is provided by online on a website called "Papers with code". It is called "Sketch", and is available to download. The dataset contains ~20,000 sketches distributed over 250 object categories (which are our classes). Each image is simple enough that there should be no issue with the quality of the user's input sketch (unless you have a really bad artist). Each class has around 80 images, which could prove to be a potential problem since that is not a large number. If we run into a high loss function, we can experiment with data techniques that allow us to increase the size of this training set. Each image is 1111x1111 pixels. Since this is a larger image for processing, we will be tedious with our convolutions.

## Project Execution

### Machine Learning Techniques, Steps to Complete the Project

- Build the user interface to draw your sketch. This sketch drawn by the user needs to be the same dimensions as the sketches in the dataset. We will most likely use PyGame for this step, unless there are better libraries that can provide a digital canvas.
- Create the Convolutional Neural Network to take in the sketches from the dataset and predict what category it belongs to via probability. Ideally, we would like the image to pass through the CNN everytime the canvas is updated, so it can predict in real time. This is of course when we test.
- Train the model on the training set, which is a partition from the total dataset. Also use a validation set to tune the hyperparameters. A few hyper parameters include number of epochs, batch size, learning rate, number of hidden layers, size of hidden layers, etc.
- Predict correctly (Hopefully)!

### Interpretation of Analysis and Results

- We will interpret our results via a cross-entropy loss function built into the model. This loss function will tell us the error "distance" between the predicted sketch and the actual label. We will then plot this on a Loss vs Number of Epochs graph to easily interpret how our model is learning.