Classification Models for Sketch Drawings

CS221: Project Proposal, Spring 2019

Aparajita Dutta (apdutta) and Sarah Saki Robinson (srobins2)

1 Overview

The goal of this project is to build an artificial intelligence that is able to reliably identify hand-drawn doodles. To achieve this, we will apply a variety of image classification techniques to a dataset of hand-drawn doodles. We will then tune and evaluate the different models to pick the one with the best classification performance.

2 Data

For this project , we will work with the Google QuickDraw dataset [1]. The dataset contains a large number of simplified drawings, centered and formatted to a 28×28 grayscale bitmap. Data is available for over 300 different objects, but for this project we will use a subset containing 22 objects: Apple, Candle, Door, Leaf, Ice Cream, Horse, Hat, Firetruck, Drums, Computer, Butterfly, Birthday Cake, Bicycle, Banana, Umbrella, Soccer Ball, Snowman, Smiley Face, Rainbow, Pencil, Peanut, and Panda. We will restrict our data to N number of images per class, where N is a hyperparamter that will be tuned going forward. The final, formatted data will be split in a 60:20:20 ratio into the training, development and tests sets respectively.

Figure 1 illustrates some sample input images from the dataset and the expected output labels for each.

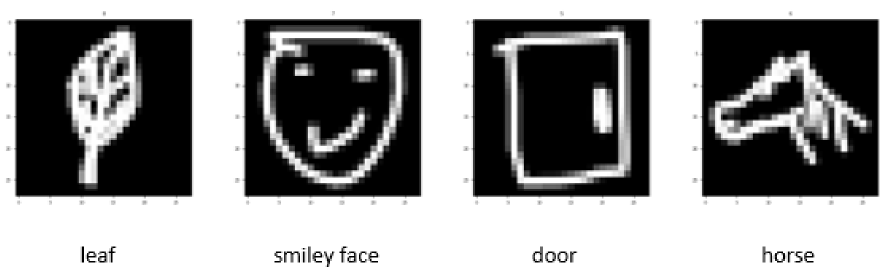


Figure 1: Sample input/output behavior for the classification algorithm

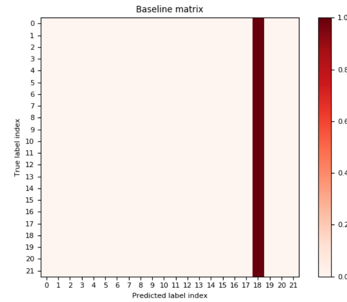
3 Approach

3.1 Evaluation metrics

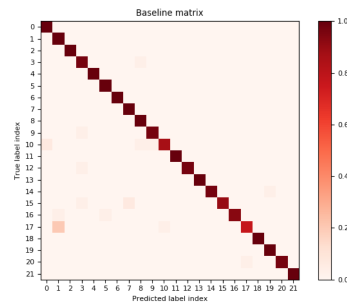
For evaluating the different models on our dataset, we will use the accuracy and log-loss. We will also plot confusion matrices for the models to visualize the models' performance. Since we are maintaining a class balance by taking a constant number of images per class, these metrics should allow good evaluation of the models.

3.2 Baseline and Oracle

For our baseline, we loaded the data with $N = 20000$ images per class, and implemented a majority classifier. This achieved 4.447% accuracy with log loss = 33.003 on the development set. For the oracle, we simulated our image classification models through real-human classification. We printed out 30 individual images for each of our 22 classes and hid the true labels. We shuffled all the images so the order was random and proceeded to identify the correct class for each image. With this model, we achieved 96.818% accuracy with log loss = 1.099. Figure 2 illustrates the confusion matrices for the baseline and oracle. The gap between the baseline and oracle was expected, considering the baseline predicted a single class for our whole development set. As we saw with our oracle, even a human was unable to correctly identify all images correctly. Correctly classifying unclear images will be a challenge for our image classifier and we hope to address this using the different models described in Section 3.3.



(a) Baseline Model



(b) Oracle Model

Figure 2: Confusion Matrices

3.3 Models

We will implement and compare different image classification models to analyze which model performs best on the drawing data. We will start with simple machine learning classifiers like Logistic Regression and Naive Bayes, then move on to deep learning methods such as Convolutional Neural Networks and Multi-layer Perceptron. We also plan to implement the Reinforcement Learning algorithm for classification described in [2], to see how it performs on this dataset.

3.4 Demonstration

To demonstrate this project, we will build an application that lets users create a drawing of one of the 22 classes chosen for this project. It will then convert the image to the right format and apply the best classification model, chosen from our experimentation, to predict what has been drawn.

3.5 Stretch Goals

If time permits, we would like to explore image generation algorithms on the dataset, similar to [3] which uses Recurrent Neural Networks to complete incomplete drawings.

4 References

[1] J. Jongejan, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg, "The Quick, Draw! - A.I. Experiment", `https://quickdraw.withgoogle.com/data`, 2016.

[2] M.A. Wiering, H.V. Hasselt, A.D. Pietersma, and L. Schomaker, "Reinforcement learning algorithms for solving classification problems", pp. 91-96, 2011.

[3] D. Ha and D. Eck. "A neural representation of sketch drawings." arXiv:1704.03477, 2017.