Deliverable 1

Abd-El-Aziz Zayed

February 9, 2021

1 Project

The project I will be tackling for the rest of the semester is one I will call Smart Sketch. It is basically a replica of Quick, Draw by Google. I might migrate the project to a mobile application if time permits where the user can draw with their finger and my algorithm tries to categorize the doodle.

2 Dataset

The dataset has been open sourced and is offered in several different formats. The files we will use for our data are a group of .ndjson files. Each file contains hundreds of thousands of examples (sketches) of an object we can train our model to detect. Each sketch is labeled with the proper object. The doodle images are NOT stored in memory as just regular images, but as a list of pencil strokes. These strokes are represented by a list of x coordinates, a list of y coordinates and a list of time coordinates. We will ignore the time coordinates in our project. In fact, the timing data will be removed in our data processing phase.

3 Methodology

3.1 Data processing

Lucky me, the data already has been preprocessed for me and is ALMOST ready to use. All the coordinate data have been normalized to between 0 and 256 (exclusive). Also the timing information has been removed. Our job is to further remove the data about the country and the timestamp. Last but not least, all strokes have been simplified using the Ramer–Douglas–Peucker algorithm with $\epsilon = 2.0$. In terms of visualization of our data, we can use pygame to render the strokes on a canvas. We still need to Remove any noise and smooth the edges of our stokes. This can be done using OpenCV.

3.2 ML Model

Many, many ML models can be used for the purpose of this project. This article goes in depth through several ML models and test which is the most efficient. From the data we can immediately see that it is labeled so we will use supervised learning. The most accurate algorithm stated by the article is CNN (Convolutional Neural Networks) with a 91% test accuracy.

4 References

https://github.com/googlecreativelab/quickdraw-datasetget-the-data
https://console.cloud.google.com/storage/browser/quickdraw$_{d}ataset/full/simplified;tab = objects?prefix = forceOnObjectsSortingFiltering = false$
https://www.kaggle.com/harunshimanto/lets-play-with-quick-draw%F0%9F%93%A1-5.Convolutional-Neural-Networks-(CNN-/-ConvNet)