COMP 576 Proposal - Doodle Recognition

Group Members: Jianwei Jin, Chen Zeng, Yuhui Tong

1. Problem Description

"Quick, Draw!" [1] was released as an experimental game to educate the public in a playful way about how AI works. The game prompts users to draw an image depicting a certain category, such as ”banana,” “table,” etc. The task is to build a classifier for the existing Quick, Draw! dataset.

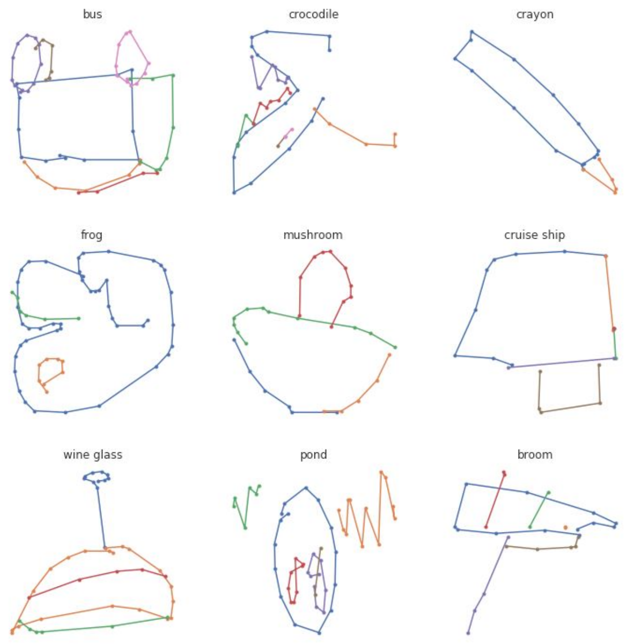

Figure 1. Samples of the Quick,Draw! dataset

The challenges of the problem are:

1. The dataset contains not only the final images but also **time-stamped vectors**, so a good solution should make the most of the time series information.
2. Since the training data comes from the game itself, drawings can be incomplete or may not match the label.
3. The dataset is huge so efficiency matters with the constraint of computational resources.

It's necessary to build a recognizer that can effectively learn from this noisy data and perform well on a manually-labeled test set from a different distribution.

2. Dataset

The Quick Draw Dataset is a collection of millions of drawings across 300+ categories, contributed by players of Quick, Draw! The drawings were captured as **timestamped vectors**, tagged with metadata including what the player was asked to draw and in which country the player was located. The dataset is huge so it will be infeasible to load all the images into memory at once with limited resources we have if they are represented as matrix. Efficient data representation is important for the problem. The dataset actually represents themselves using list of points which save lots of memory since they are doodles and most of them consists of some connected lines.

3. Methodology

Since the nature of this problem is image classification. There already exists lots of efficient methods to accomplish. The most common and effective one are various **CNN based model**. Definitely we'll first investigate this problem starting with CNN. We also plan to use RNN-based models which are well known to excel in situation involving a sequence of values (such as the time-stamped vector in Quick-draw dataset). Specifically, we'll try the **RNN/LSTM** and their variation **Bidirectional LSTM (BLSTM)** and **Bidirectional RNN (BRNN)**.[2] We are also interested in a relative new model named **MobileNets** [3,4] which is known to be more efficient than classic deep learning models.

3.1 BLSTM / BRNN

The main difference between BRNN/BLSTM and RNN/LSTM is that BRNN/BLSTM connects two hidden layers of opposite directions to the same output. [2] With this form of generative deep learning, the output layer can get information from past (backwards) and future (forward) states simultaneously.

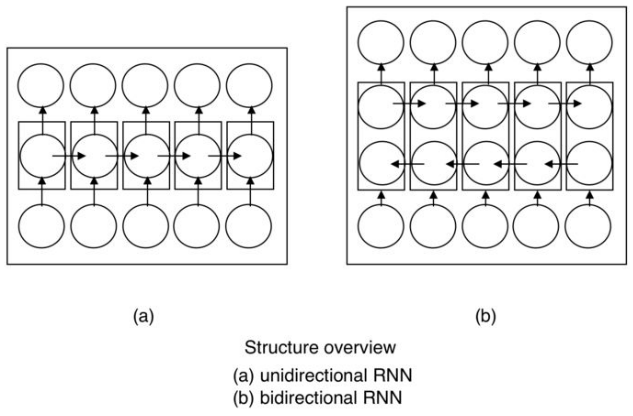

Figure 2. unidirectional RNN and bidirectional RNN.

BRNN are especially useful when the context of the input is needed. For example, in handwriting recognition, the performance can be enhanced by knowledge of the letters located before and after the current letter. It would be interesting to compare the performance of RNN/LSTM and BRNN/BLSTM on the very dataset.

3.2 MobileNet

We would like to investigate **MobileNet** as, compared to CNN, it utilizes the depth wise separable convolutions which drastically reducing computation and model size. [4] Based on a streamlined architecture MobileNet uses depth wise separable convolutions to build light weight deep neural networks. It has two global hyperparameters, i.e., width multiplier and resolution multiplier, which can be adjusted to trade off a reasonable amount of accuracy to reduce size and latency. We'll compare classic deep learning model with MobileNets, and expect it see its performance on our huge datasets (~50 million samples).

4. Tools and platforms

We plan to work mainly on Google Colab using Tensorflow and Keras. Computation intensive task wills be executed on Google cloud platform powered by GPU acceleration.

Reference

[1]https://www.kaggle.com/c/quickdraw-doodle-recognition/
[2]https://towardsdatascience.com/introduction-to-sequence-models-rnn-bidirectional-rnn-lstm-gru-73927ec9df15
[3] https://www.kaggle.com/adarsh1012/mobilenet-lb-0-895-please-upvote
[4] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).