CAPSTONE PROJECT PROPOSAL

**Project Title :** Doodle Recognition System – 4-doodle (art game)

**Course Name :** CAPSTONE TERM II

**Course Code :** 202041.23309-AIDI-2005-01

**Course Facilitator :** Marcos Bittencourt

**Student Names and IDs:** Anish Sharma – 100767348

Dhairavi Patel– 100771583

Karamjit Singh – 100767214

Yogeshkumar Patel – 100767342

**Group Number :** 14

**Date of Submission :** January 24, 2020.

# Executive Summary

Doodle recognition has significant impact in computer vision and pattern recognition, particularly in relation to the handling of noisy datasets. We will be generating a Doodle Recognition Application for this capstone project. The users will be able to doodle on the screen and the application will classify the doodle(s) into one or more of the 345 categories. While the user is drawing, an advanced neural network attempts to deduce the category of the object, and its prognostications improve as the user adds more and more detail. For the purposes of this project, we choose to focus on the classification of the completed doodles in their entirety.

# Rationale Statement

Computers possessing the capability to comprehend quick line drawings will allow for broader forms of expression and communication. These days children love to play on gadgets, and they get familiar with them faster than elders. But the children waste their most of time in doing non-productive activities and just use these devices for fun. So, we came up with an idea to develop a Doodle Recognition game for them. While this is not only a way to meet fun with art, this also encourages creativity in children and help them utilise their time on learning. This game will challenge children to reach the optimum level of basics of drawing as well as help them understand the difference between the various objects, for instance – they will not only learn to draw an apple but also learn how apple look like and how this is different from orange. The objective of the game is simple to teach the children, challenge them to improve and reach next level, and help them utilise their time in doing productive things rather wasting time on other non-productive things.

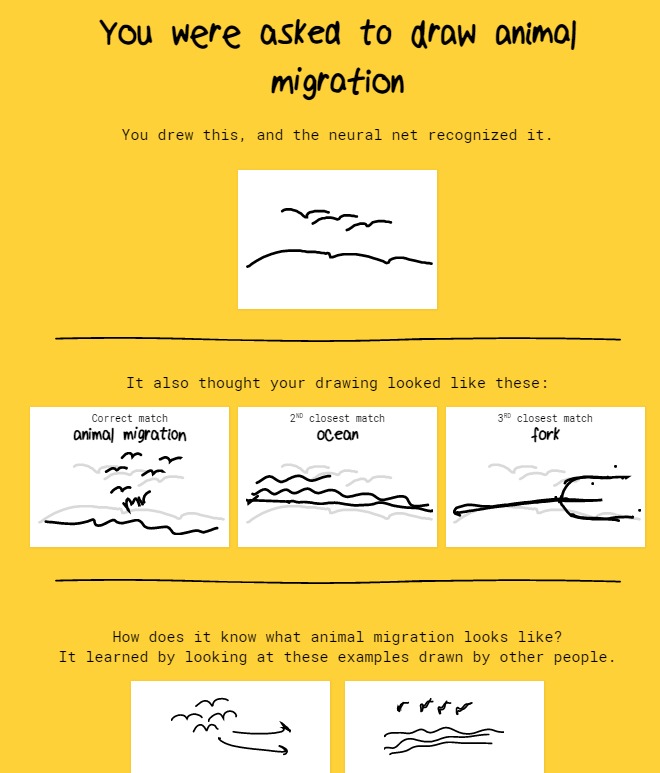
# Key Evaluation Metrics

We evaluate our methods not only with raw accuracy but also with a scoring metric that is more lenient of incorrect predictions.

* **Raw Accuracy** – Raw accuracy is a good measure of a model’s performance; it penalizes harshly for an incorrect prediction (wrong predictions receive 0 points and right predictions receive 1 point). Since we have so many categories, including some that are extremely similar such as “cake” and “birthday cake”.
* **Scoring Accuracy** - scoring metric that is more lenient of incorrect predictions. Thus, predictions are evaluated using Mean Average Precision @ 3 (MAP@3): MAP@3 = 1 U X U u=1 min X (n,3) k=1 P(k) where U is the number of drawings in the test set, P(k) is the precision at cut-off k, and n is the number of predictions per drawing. Put more intuitively, the equation considers the top 3 predictions (P1, P2, P3) that the model makes for a given drawing. It then assigns a score of 1 if Pi is the correct label for the image and a score of 0 if the correct label is not in the top 3 guesses. Note that MAP@1 is equivalent to single prediction accuracy.

# Result

The resulted solution would look like the image given below. As it is shown in the image that 4-doodle game not only predicts the images but also show how the intended drawing should look like. This is a combined way of fun and learning. Hence, this is going to be an interactive solution to bring drawing and gaming on a single platform.



# Data Analysis and Feature Engineering

The whole Data Engineering and Feature Engineering task can be found in the following jupyter notebook file.

<https://github.com/karamjit-singh/four-doodle-dl/blob/master/Data%20Analysis%20and%20Feature%20Engineering.ipynb>

## Benefits of Feature Engineering

In machine learning, the data that is used to train a model determines the quality of the model (ie. If the data used is poor in quality, the model will not be accurate). Due to the great significance of the data, it is vital to generate a dataset that is enhanced to augment the information density of the data that will be utilized. In order to achieve this level of dataset, feature engineering and feature selection are used; meaning that numerous manipulations are made to the dataset in these techniques. The apt features will drastically increase its predictive capability but will also propose the litheness to use less complicated models which are faster to run, as well as easily comprehended. Here are some key points regarding the feature engineering on given dataset.

* The benefit of feature engineering in regard to the dataset of this project will greatly improve pattern recognition.
* The impact will be immediate in various fields such as handwriting recognition, optical character recognition, automatic speech recognition, and natural language processing. The data comes from the game itself, leading to it being very noisy and unsuitable for training the model that will be implemented. The reason for this is that the users have not accurately drawn objects, or have just drawn random scribbles. Feature engineering will be key in this project as filtering out the data will lead to an efficient model, results, and less computing power.
* For this project, the concentration will be on classification of the entire doodle, thus, the latter version of the dataset will be made use of. Each 28x28 pixel image will be treated as a 784-dimensional vector. To test the models, the data will be fragmented into distinctive folds for training, for validation, and for testing. In order to condense computation time and storage of the data, there will be a creation of a reduced subset of the original dataset by unsystematically sampling of the drawings from each classification.

# Key Features in Dataset

This dataset from Google, is named Quick, Draw!. Its crucial qualities are that it comprises over 50 million images spanning through 345 categories. There are various examples for the doodle images. One dataset epitomizes every drawing as a series of line vectors, and another comprises each image in a 28x28 grayscale matrix. For this project, the concentration will be on classification of the entire doodle, thus, the latter version of the dataset will be made use of. Each 28x28 pixel image will be treated as a 784-dimensional vector.

To test the models, the data will be fragmented into distinctive folds for training, for validation, and for testing. In order to condense computation time and storage of the data, there will be a creation of a reduced subset of the original dataset by unsystematically sampling of the drawings from each classification.

Subsequently, approximately examples for the training set and examples of each for the validation and testing set will be acquired. Additionally, the number of drawings in each class is equalized, leaving approximately examples per category in the training dataset.

# Data Requirements

The problem we are trying to solve is recognizing the pattern drawn by the user and let our model identify if it matches with any of the available categories, we trained our model on. Since drawing is a unique creation for every individual, we want our model to recognize the object which even closely resembles the object. To achieve this, we want variety of dataset that highlight various orientations as well as shapes an object can be drawn in. The variety of dataset can be measured in two categories:

* Inter-class variation: This type of variation is within the same label. If the drawing of one whole apple and a half apple are drawn, the model should be able to predict both the images as apple although both images have less similarities.
* Intra-class variation: This type of variation is between different labels. For example, the difference between an orange and full moon is not much in respect to shape, but the model should have enough samples to train on both types to detect the difference successfully.

The more the variation observed in the training set the better the model learns to predict a new drawing. Therefore, the basic requirement to train our model is a variety of labels and enough pictures under each label for our model to train on.

Also, the images need to be within a certain quality limit such that we don’t lose the information due to reduced pixel quality. The size of the image should also be of certain dimension and should not be either too big or too small. This is important as in the later process we may need to use image augmentation which will require rescaling of images and if the image is too small or too big, we may not be able to retain all the necessary information in the rescaled image.

# Data Source

The data source we came across that closely resembles our requirement was the Quick Draw! Google Doodle recognition dataset. This dataset was collected by google as a fun activity to make people aware of the working on AI. In the interactive game designed by google the user is told to draw an object within a limited time frame and then the AI uses real time recognition to guess what the user is drawing through a list of labels. This resulted in a huge dataset which have labels close to 340 and have around 5000 training samples for each label.

The dataset consists of two type of images:

* Raw image – The image is stored in JSON array representing the vector drawing. The array consists of 3 variables x, y, t. Where x and y are the pixel coordinates, and t is the time in milliseconds since the first point. x and y are real values while t is an integer. The raw drawings can have vastly different bounding boxes and number of points due to the different devices used for display and input.
* Processed image – The images from raw dataset are scaled into 256 X 256 region and the variable t is eliminated. The images were then simplified to be used for model training using the following techniques:
* Image Alignment
* Image scaling
* Resampling
* Simplifying strokes

Processed image will be used to train our model. We will take input from the user on a drawing board interface and then that image will be preprocessed similar to the image the model will be trained on and then the image will be converted into pixel vector and pushed into the model to predict.

Data Assumptions

Regarding the dataset we have these are the following assumptions we may consider:

* We assume that the raw image is of such fine quality that even after processing of image, we will be able to extract the required information.
* Since the dataset is user input, there is no way to validate every image being complete and accurate. Therefore, we assume that the image under the label represents the same.
* Given that the dataset was prepared under a given time constraint, we assume that the user was able to represent the image they wanted to draw.
* We also assume that there are no blank or minimal samples collected in the dataset and every time user was asked to draw an object, user performed the required task.
* Since drawing is unique for each individual and may vary in representation based on what a user remember about that object, we assume that the inter class and intra class variation of the images is widely spread along the spectrum for model to train well on multiple representation of same object.

# Data Limitations and Constraints

The Quick,Draw! Dataset that is going to be utilized for this application has numerous limitations and constraints as follows.

* The copious of the drawings that are a part of the dataset are not relevant to the category they are classified in. For example, there are random scribbles present which do not look like a bear if they are characterized under the bear category.
* T these drawings remain to be ambiguous due to various reasons; such as the 20 second time constraint that is present, the presence of cultural differences in comprehending the words, etc. This limitation makes it difficult for the model to be able to classify the doodles accurately, leading to greater chances of misclassification.
* An inevitable constraint that this dataset arises from the method of its collection. It consists of the fact that each and every drawing is considered an input. As soon as the user begins to draw, the program intends to justify it in at least one of the 340 existing categories. Though this has provided a larger dataset to allow more inputs into the model for training purposes, on the contrary, it has also made the dataset noisier.
* A major limitation is the 340 categories and classes. Despite this appearing to be a large number, it limits the drawings in these boundaries, leaving no possibility of the correct classification. Regardless of the possibility existing of a better category for a drawing input, the possibility of it not being an existing category increases the chances of misclassification, and accuracy is also lowered.

Nonetheless, despite the presence of these limitations and constraints, it is possible to over come them through feature engineering and feature selection.