# CAPSTONE TERM 2 Group 14

Quality Assurance and Prototyping

# 

### **Project Title**: Doodle Recognition System

### **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

### **Date of Submission**: March 6, 2020.

# 

# Quality Assurance

Quality Assurance ensures that an organization is providing the best possible product or service to customers. It focuses on improving the processes to deliver Quality Products to the customer.

## Evaluation Report for CNN

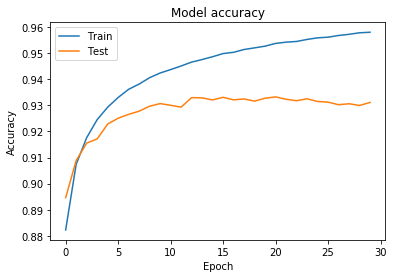
**Classification Accuracy**

Accuracy is one metric for assessing classification models. Nonchalantly, accuracy is the portion of predictions our model got correct. If the accuracy comes out to 0.91, or 91% , it appears that the classifier we are using is doing a great job of identifying the doodles.

While 91% accuracy may seem good at first glance, our model is no better than one that has zero predictive ability to distinguish the doodles from different categories.

Accuracy alone does not provide information on the model when we are working with a class-imbalanced data set, like this one, where there is a significant disparity between the number of doodles classified in each category.

According to our scorecard, the classification accuracy for our CNN model is 93.08%. This model has the highest accuracy from all the models that we have used. The accuracy percentage shows us that although this algorithm will provide us the best result in comparison to the others, there is still a possibility for misclassification of the doodles.



**Validation Loss**

Validation Loss is another method to evaluate our model. Validation loss is not used in updating weights. In the model accuracy graph below, we can see that the accuracy for our train set increases as we add more epochs and the error decreases. The model loss graph below shows that the loss is decreasing for our train dataset, and increasing for our test dataset. This means that our model is slightly under fit.

### 

### 

### Pros and Cons

**Pros**

* Convolutional Neural Networks can capture and learn relevant features from an image /video at different levels similar to a human brain. This is referred to as feature learning.
* CNNs can weight share. For example: If we have one layered CNN with 10 filters of size 5x5. Now you can simply calculate parameters of such a CNN, it would be 5\*5\*10 weights and 10 biases like **5\* 5\*10 + 10 = 260 parameters**. CNN is more efficient in terms of memory and complexity.
* In terms of performance, CNNs outperform NNs on conventional image recognition tasks and many other tasks.
* CNNs are very good feature extractors.Meaning that one can extract useful attributes from an already trained CNN with its trained weights by feeding your data on each level and tune the CNN a bit for the specific task.

**Cons**

* CNN does not encode the position and orientation.
* Lack of ability to be spatially invariant to the input data.
* A convolution is a significantly slower operation than, say maxpool, both forward and backward. If the network is pretty deep, each training step is going to take much longer.

### Google Collab Notebook and GitHub

<https://colab.research.google.com/drive/1BFfJdCU951NEmc7vNFJh2_eYzjo_341_>

<https://github.com/karamjit-singh/four-doodle-dl/blob/master/Four_DoodleCNN_ModelEvalAndPred.ipynb>

## Evaluation Report for MobileNet

**MobileNet Pros**

MobileNets are light weight deep neural networks best suited for mobile and embedded vision applications.

* MobileNets are based on a streamlined architecture that uses depth wise separable convolutions.
* MobileNet uses two simple global hyperparameters that efficiently trades off between accuracy and latency.
* MobileNet could be used in object detection, finegrain classification, face recognition, large-scale geo localization etc.
* Reduced network size - **17MB**.
* Reduced number of parameters - **4.2 million**.
* Faster in performance and are useful for mobile applications.
* Small, low-latency convolutional neural network.

**MobileNet Cons**

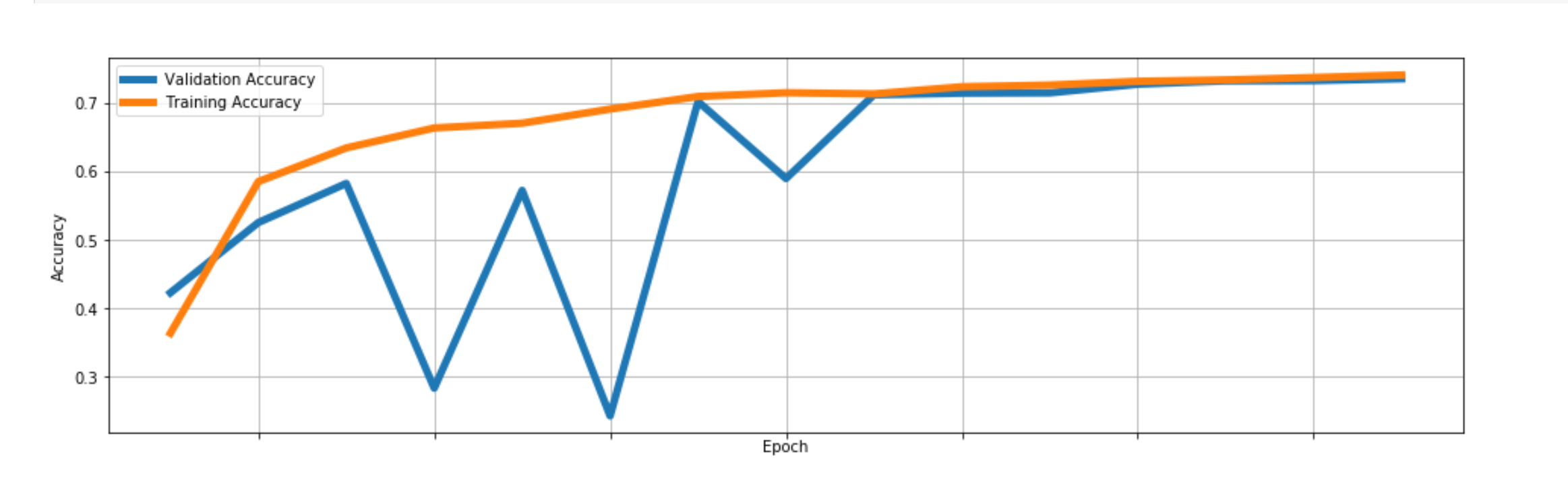
* Has too many hidden layers, this makes it computationally very expensive.
* Even though we have reduced parameters in MobileNet, it is less accurate, in comparison to other networks.

### Kaggle Notebook

https://github.com/karamjit-singh/four-doodle-dl/blob/master/quick-draw-mobilenet-final-evaluation.ipynb

**Classification Accuracy**

According to our scorecard, the classification accuracy for our MobileNet model is 88.90%. This model has the second highest accuracy from all the models that we have used. The accuracy percentage shows us that although this algorithm will provide us a fairly accurate result, there is still a great possibility for misclassification of the doodles.



**Validation Loss**

As the number of epochs increases, the validation accuracy and the training accuracy start to align. Initially, there was a large difference in the accuracy but overtime it decreased.

# 

# Model Prototyping

Evaluation Report

|  |  |  |
| --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy |
| CNN | 95 | 93 |
| Mobile Net | 89 | 73 |

After trying a couple of models and analysing theri pros and cons, CNN fits best to our requirement. We have trained and validated models using CNN algorithms and run a couple of tests to test the model behaviour and predict the drawings. Most of the guesses were relevant to what was drawn.

Here’s the link to the file in the github repository

<https://github.com/karamjit-singh/four-doodle-dl/blob/master/Four_DoodleCNN_ModelEvalAndPred.ipynb>

References: <https://developers.google.com/machine-learning/crash-course/classification/accuracy>

<https://gogul.dev/software/mobile-net-tensorflow-js>

<https://www.guru99.com/all-about-quality-assurance.html>