Robotic Inference Project

Vihaan Akshaay Rajendiran

Abstract—An attempt was made on training two neural networks using DIGITS, One being trained on a given dataset, with three classes .A model accuracy of 75.4098 was achieved with an inference time of around 4.4ms. The second model was trained on generated dataset with three classes which was based on an attempt to solve a real world problem.

Index Terms—Robotic Inference Project, IEEEtran, Udacity, LaTeX, deep learning.

1 Introduction

THE idea that this robotic inference project is modeled after is explained below. There are hundreds of rubik's cube speedsolving competitions that happen across the world every year.Here, the solver is given a scrambled rubik's cube and is asked to place a solved cube in front of an evaluator (a person is made an evaluator for each table and has to do this repetitive task through out the competition). Placing a cube that is unsolved gives the person a penalty. This process of checking if the cube is completely solved currently requires hiring a person who has to sit through out the competition each day to just check if the cube is completely solved which does not require deep enquiry. This process can be automated having a couple of cameras which will check if the cube is completely solved, and make necessary entries that the person otherwise would manually have to do.

Fig. 1. An unsolved Rubik's cube

2 Background / Formulation

The first model trained was with Alexnet CNN architecture because of the type of dataset that was given as well as it being simpler was able to train much faster for the number of epochs that was chosen.Now since the results with this was not good enough for the target, which was a model accuracy of 75 percent, it was saved as a pre-trained model, later it another model was built based on this with different epoch so as to attain the desired result. The third model was also trained from the Alexnet CNN Architecture because of the same reason as well as to match the generated dataset.

3 Data Acquisition

For the first case, data set was provided. Below attached is a glimpse of the dataset that was provided. It was primarily made of data belonging to three classes - 'bottle', 'candy box' and 'nothing'.

Dataset.png

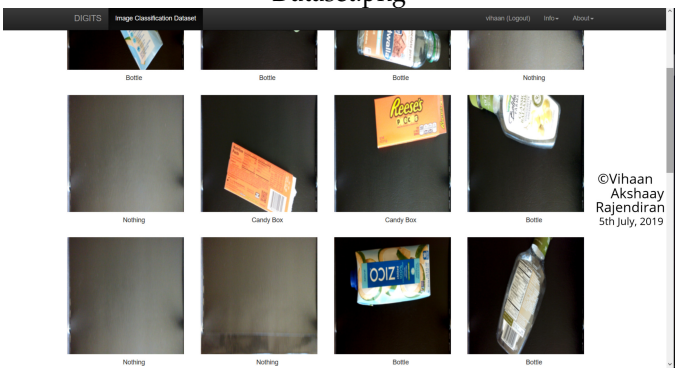

Fig. 2. Provided Dataset

Dataset.png

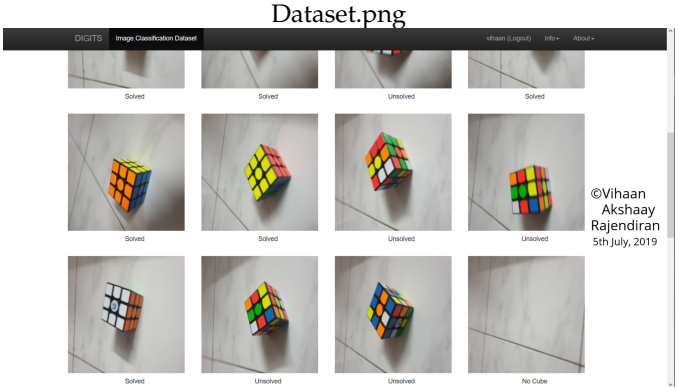

Fig. 3. Generated Dataset

For the rubik's cube model, dataset was generated. A solved cube and an unsolved cube was pictured with different orientations from various angles. These pictures were taken using a phone with a 12mp camera, having the dimension 3000x4000. Now these pictures were scaled down to be used for our model which expects 256x256. The dataset was modified accordingly in the DIGITS platform.

Attached above is a picture of the dataset used for training the model. It was also segregated into three classes - 'solved' , 'unsolved' and 'no cube'. it was also made sure that, except for what has to be focussed, the remaining of the background in every picture was almost kept same. This was done so the model does not use that as a parameter for distinguishing between classes and only majorly uses what we want it to.

4 Results

Two models are ready, trained from the Alexnet CNN Architechture. One was from a given data set and one from a generated dataset. The first task to achieve a model accuracy more than 75 percent with inference time less than 10ms was achieved and the picture is attached below. The results are attached below.

Result.png

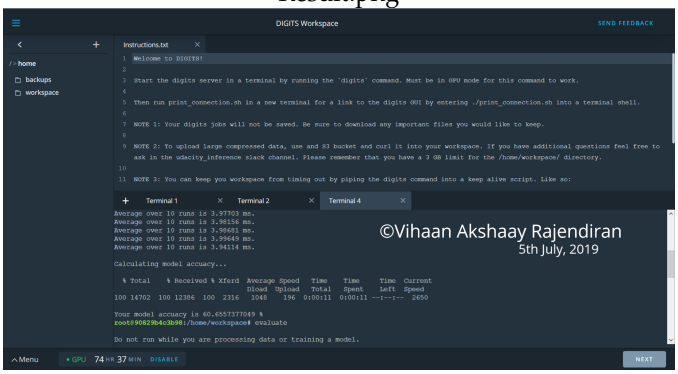

Fig. 4. Evaluate Command Results Of The Successful Model

2.png

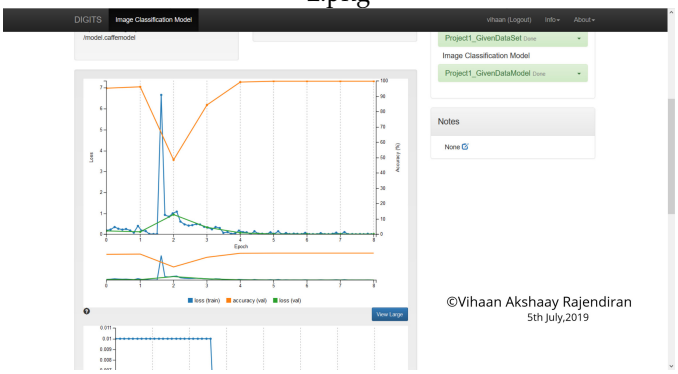

Fig. 5. Graphs of the model

In addition, here are the results to the first model that was trained, yet was not able to hit the 75 percent target.

The graphs of the model that was trained using generated dataset is also attached below.

5 Discussion

Since the created dataset had all the classes with the same background, it worked well.Different models of cube with different surroundings can be worked on with annotated training data for better results. Here we need more accuracy

Result Model 1.png

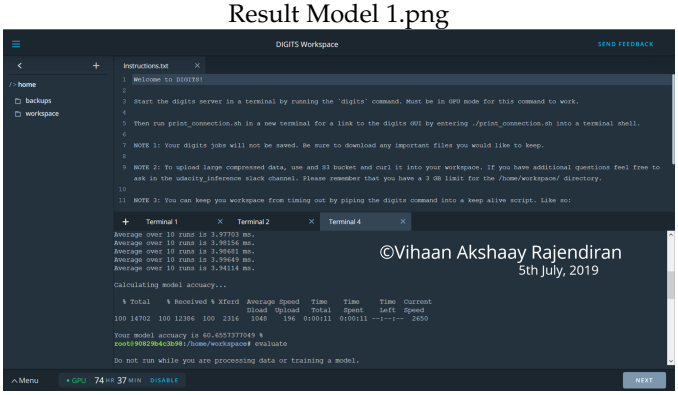

Fig. 6. Evaluate Command Results Of The First Attempted Model

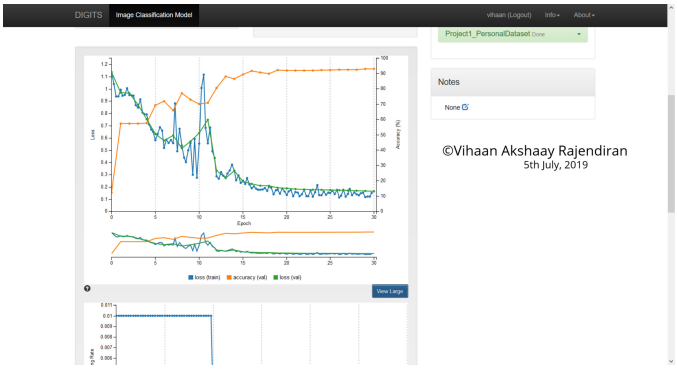

Fig. 7. Graphs Of Rubik's Cube Model

as we should make sure that the cube is solved completely, and do not have to bother much about the inference time. For this, the accuracy could have been better using googlenet, even though it would have taken more time to train the models because of it's CNN Architecture .It also took higher inference time .Inference time can be improved by using better hardware like the jetson TX2 and with better software.

6 Conclusion / Future work

The network was able to train well on the generated dataset, but cannot yet be generalised because here, it was made sure that the background was similar for all the classes, which will definitely not match with the surroundings in competitions. Using a more generalised dataset, will help in better generalising the model. Using various cameras could also help as the cube may not be completely solved, yet look complete from one angle. This project has viable scope in becoming a product soon enough as the computational resources needed is very less and the task that has to be automated is repetitive and can easily work with higher efficiency.