Robotic Inference Project

Mohamed Ewis

Abstract

This project has two classification data models using Deep Neural Network technology. The first one is classifying the bottles, candy wrappers and nothing on a moving belt. The second one classified the traffic light if its Green, Yellow, Red or nothing. The image data are come from public dataset. The project used the two models: AlexNet,GoogLeNet. For the first dataset and AlexNet for the second dataset, the best results were presented in this article.

Introduction

The classification method of modern deep learning era follows a diverse range of decision driven methods for the identification of images. The basic assumption is these approaches is that the images contains one or more features which can be used to train a model in order to predict a label, which in this case is the object in the image itself. Often, the features are represented by the pixel values in different regions of the images and the spatial analysis of the image by neural networks often results in a superior classification model for the class of the object depicted in the image. The classes, which acts as labels, can be detected manually by an analyst a-priori for training the model or it can be automatically clustered using an unsupervised algorithm into sets of prototype classes, where the analyst merely mentions the desired number of categories.

There are two parts in the project:

1. A classification network was built for classifying candy wrappers, bottles or nothing (empty conveyor belt) on a moving conveyor belt.
2. A classification network was built for classifying the different colors of Traffic Lights (Green, Yellow, Red or nothing) which is a public dataset.

Background / Formulation

- During the first inference task, i.e. classification of objects on a moving conveyor belt, both AlexNet [1] and GoogleNet [2] were chosen as they both had good inference rate per image with reasonable accuracy. The models were trained for 5 epochs and a Stochastic Gradient Descent Optimizer with an initial learning rate of 0.001 was used which was decayed to 0.0001 halfway through the training for both of these models and it successfully met the criteria of an inference time of below 10 ms with an accuracy greater than 75 percent.

- During the 2nd inference task, i.e. classification of different colors of Traffic Lights, AlexNet model, same optimizer and initial learning rate were used. The models were trained for 5 epochs using a decaying learning rate as before. The images were kept as RGB images to detect the light color, the only major difference being the color of the traffic light. Increasing the number of epochs did not improve the accuracy.

Data Acquisition:

- The P1 image dataset consists of images of bottles, candy wrappers and no object on a conveyor belt passing under a camera. A swing arm is used to sort all right objects to correct the bins depending on classifying results. The RGB images were scaled to 256 x 256 pixels for effectively using AlexNet and GoogleNet. This dataset was provided by Udacity. Figure 1 shows the sample conveyor belt dataset.

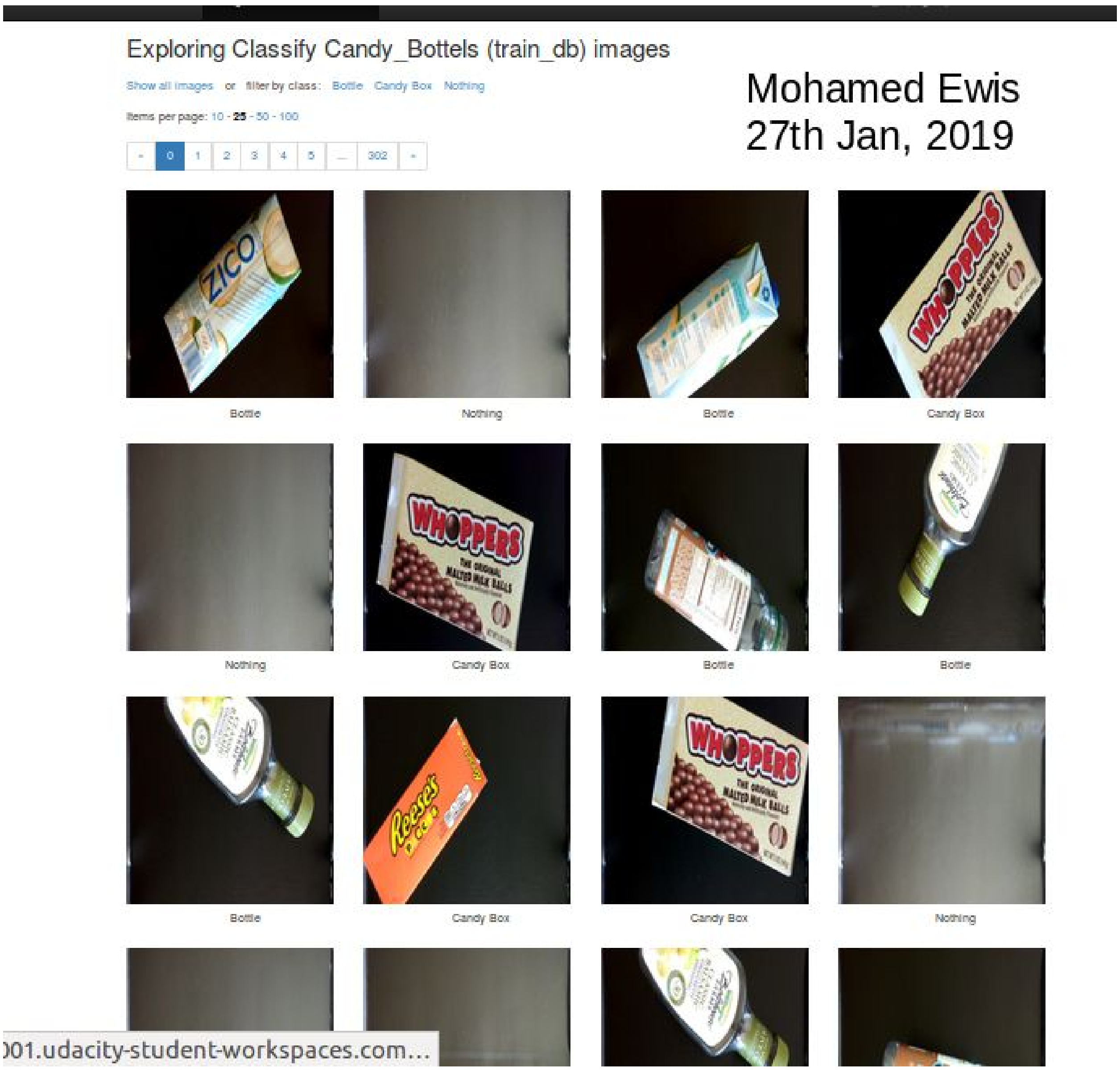


Fig. 1. Conveyor Belt data

- The 2nd dataset, i.e. colors of Traffic Lights namely green, yellow, red and nothing was a public dataset. The RGB images were scaled to 256 x 256 pixels as these were also trained with the help of AlexNet and GoogleNet. Figure 2 shows the traffic lights dataset.

Fig. 2. Traffic Lights data

Model creation

- For the initial inference task of conveyor belt classification:
  - AlexNet Model was built as:

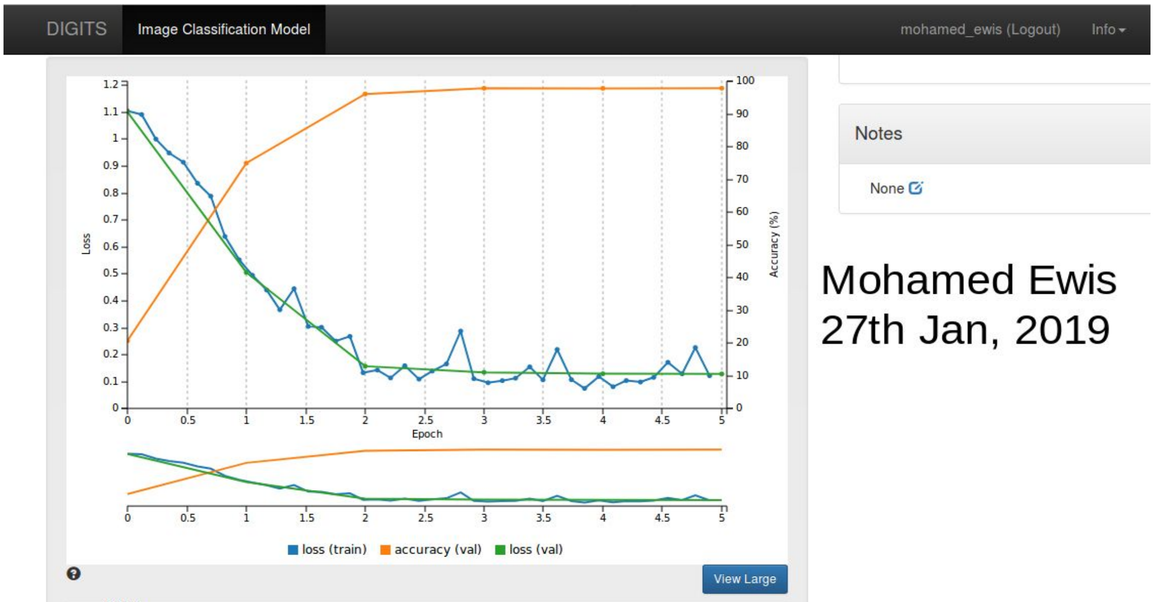


Fig. 3. Conveyor Belt model_AlexNet

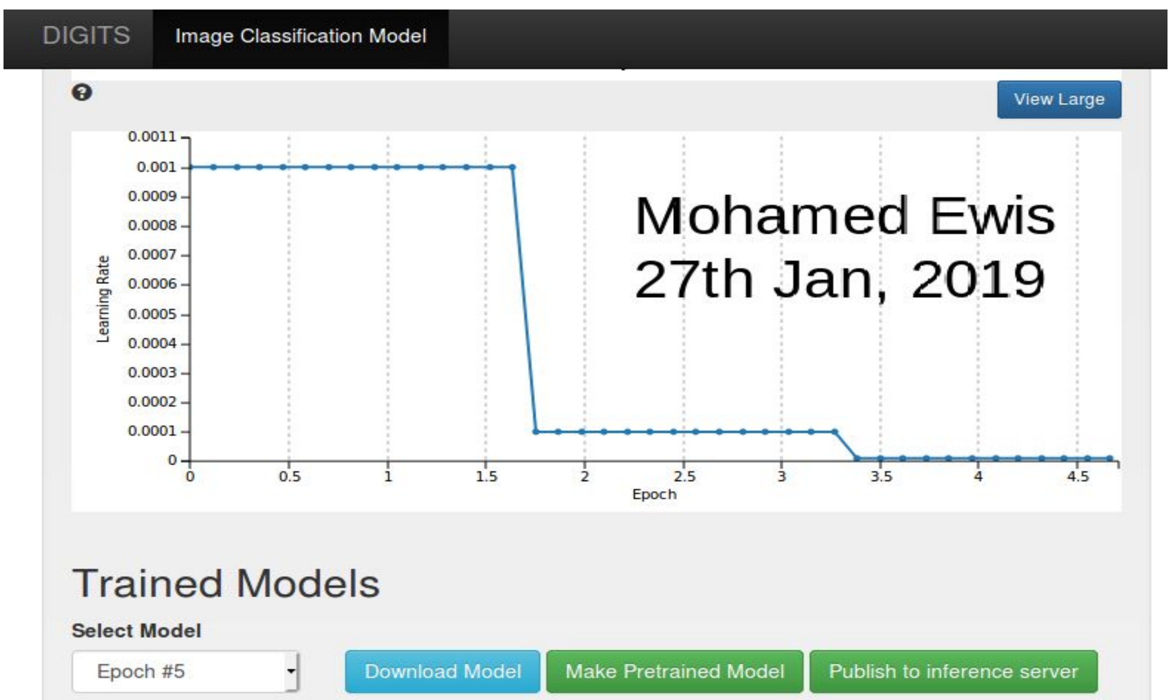


Fig. 4. Conveyor Belt model_AlexNet

- GoogleNet Model was built as:

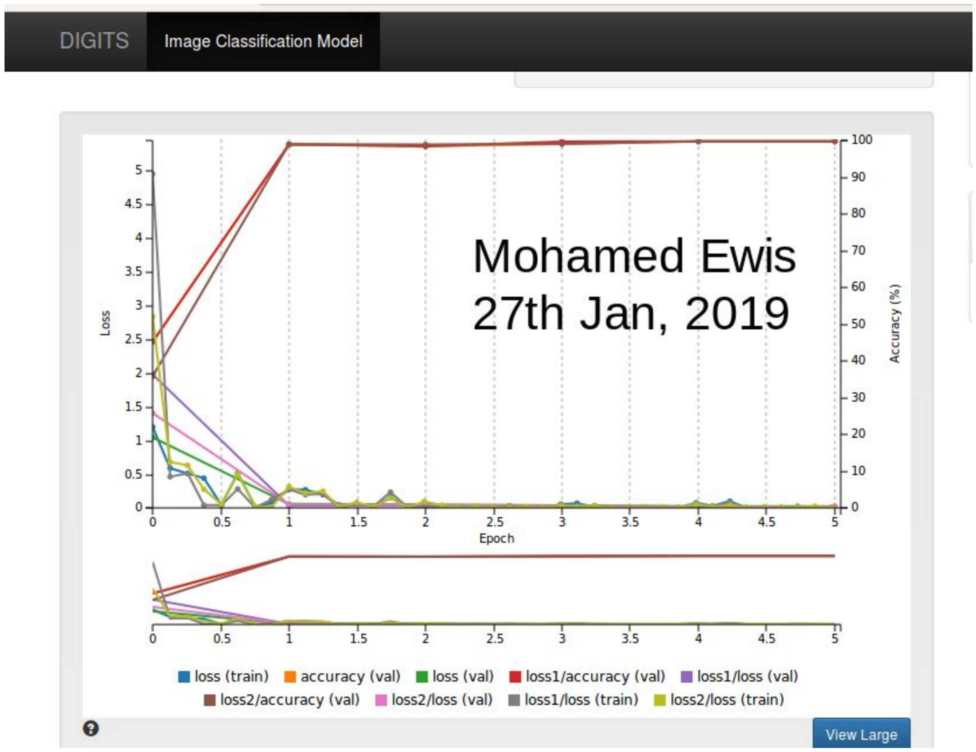


Fig. 5. Conveyor Belt model_GoogleNet

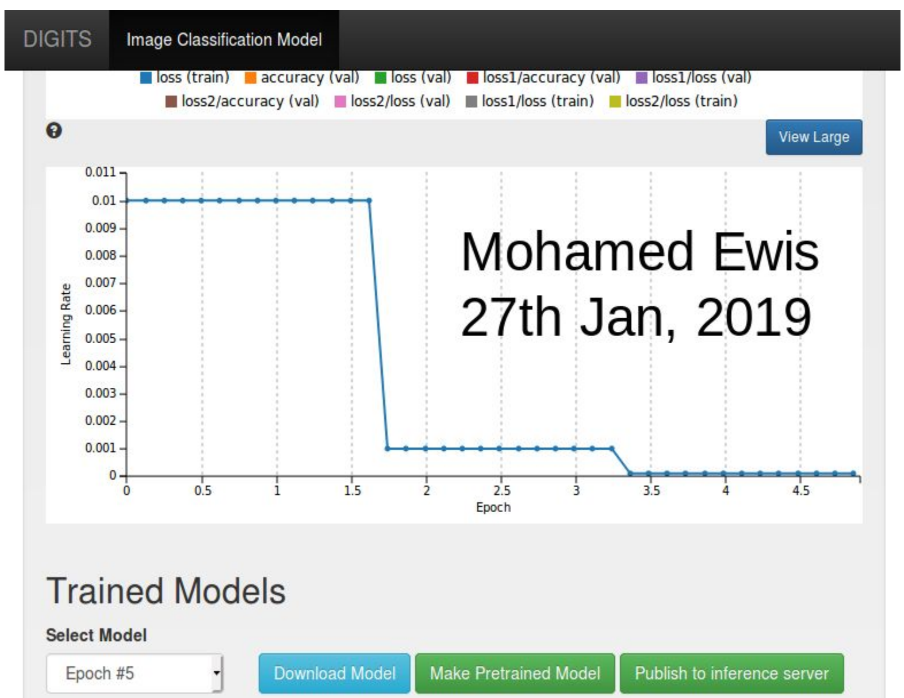


Fig. 6. Conveyor Belt model_GoogleNet

- For the second inference task of Traffic Lights classification:
  - AlexNet Model was built as:

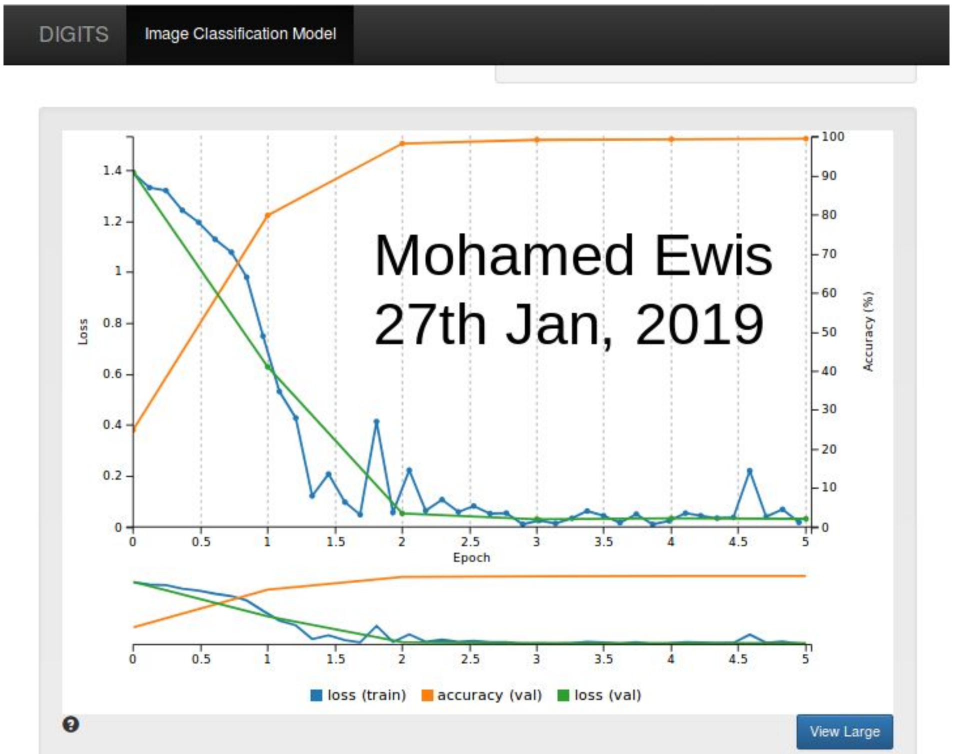


Fig.7. Traffic Lights model_AlexNet

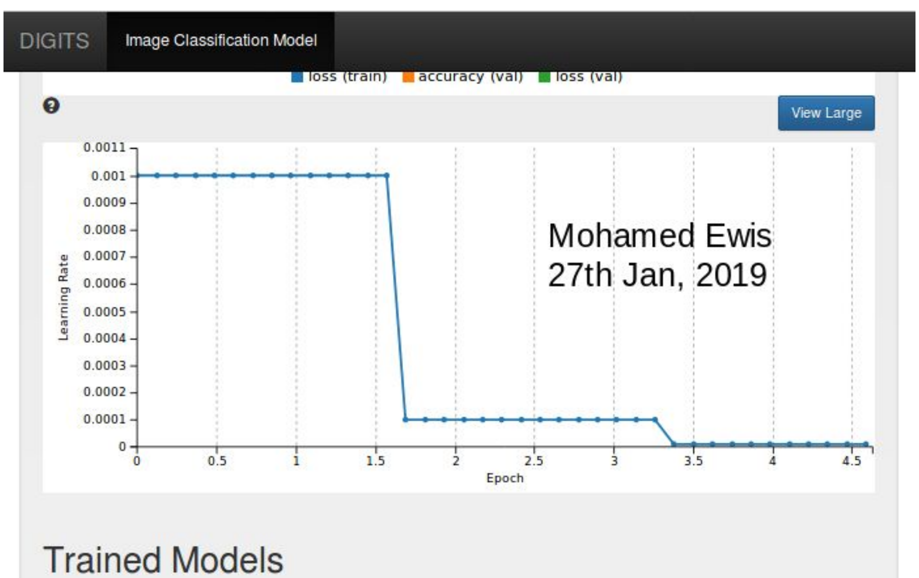
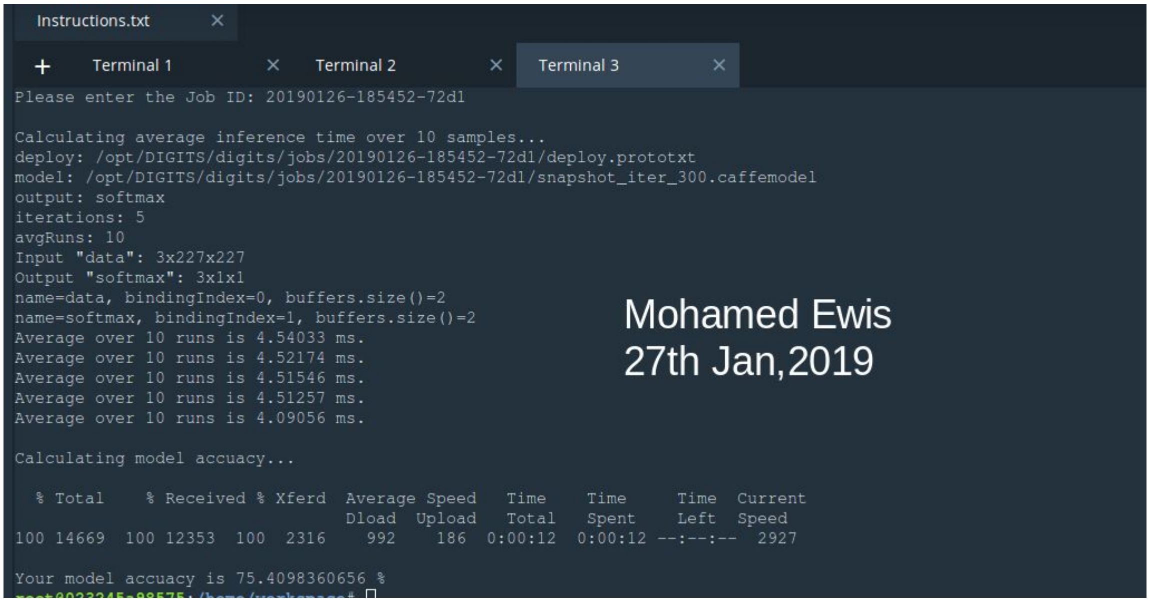


Fig.8. Traffic Lights model_AlexNet

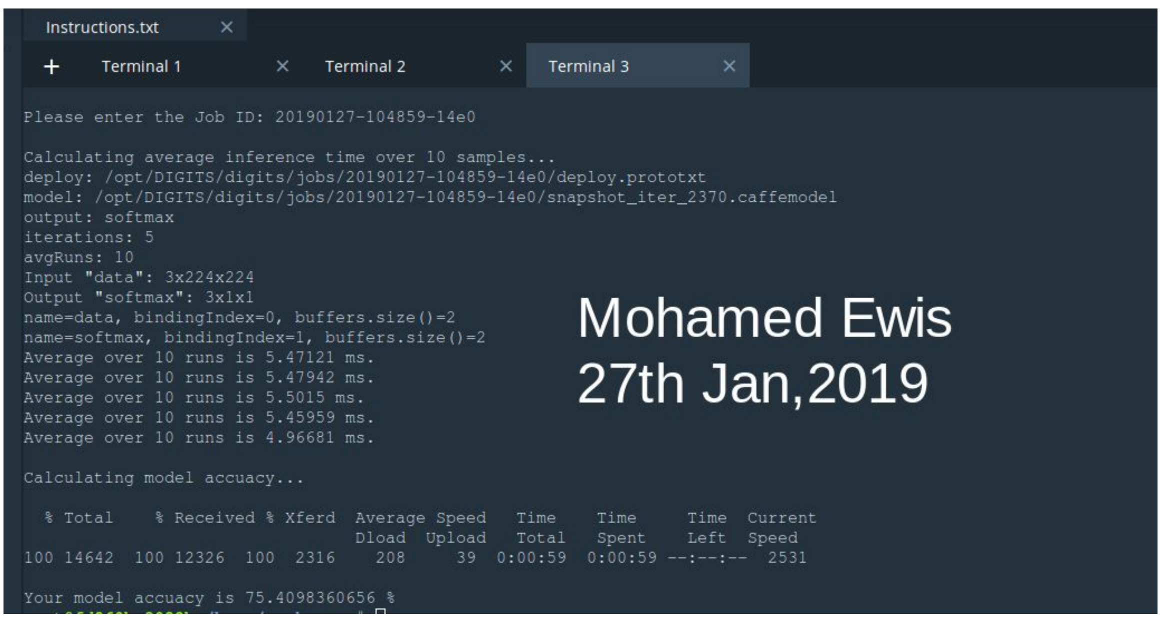
Results

- For the initial inference task of conveyor belt classification, both GoogleNet and AlexNet were used as both produced and accuracy of greater than 75 percent. Figure 3 and 4 denotes the evaluation results for AlexNet and GoogleNet respectively.



```
Please enter the Job ID: 20190126-185452-72d1

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20190126-185452-72d1/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20190126-185452-72d1/snapshot_iter_300.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x227x227
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 4.54033 ms.
Average over 10 runs is 4.52174 ms.
Average over 10 runs is 4.51546 ms.
Average over 10 runs is 4.51257 ms.
Average over 10 runs is 4.09056 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14669  100 12353  100  2316    992    186  0:00:12  0:00:12 --:--:--  2927

Your model accuacy is 75.4098360656 %
```


Fig. 9. AlexNet Inference_Conveyor Belt model



```
Please enter the Job ID: 20190127-104859-14e0

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20190127-104859-14e0/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20190127-104859-14e0/snapshot_iter_2370.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x224x224
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 5.47121 ms.
Average over 10 runs is 5.47942 ms.
Average over 10 runs is 5.5015 ms.
Average over 10 runs is 5.45959 ms.
Average over 10 runs is 4.96681 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14642  100 12326  100  2316    208     39  0:00:59  0:00:59 --:--:--  2531

Your model accuacy is 75.4098360656 %
```


Fig. 10. GoogleNet Inference_Conveyor Belt model

It is observed that AlexNet produces an average inference time of a little over 4 ms and GoogleNet produces an average inference time of around 5 ms. Both produces an approximate accuracy of 75.41. So in contest, AlexNet performs better than GoogleNet with respect to inference time. So AlexNet was chosen as the final contender.

- However, for the 2nd inference task of traffic lights classification,AlexNet and GoogLeNet models, both were built and tested. Compare both of them, the AlexNet model has better results. This article only shows AlexNet model result.

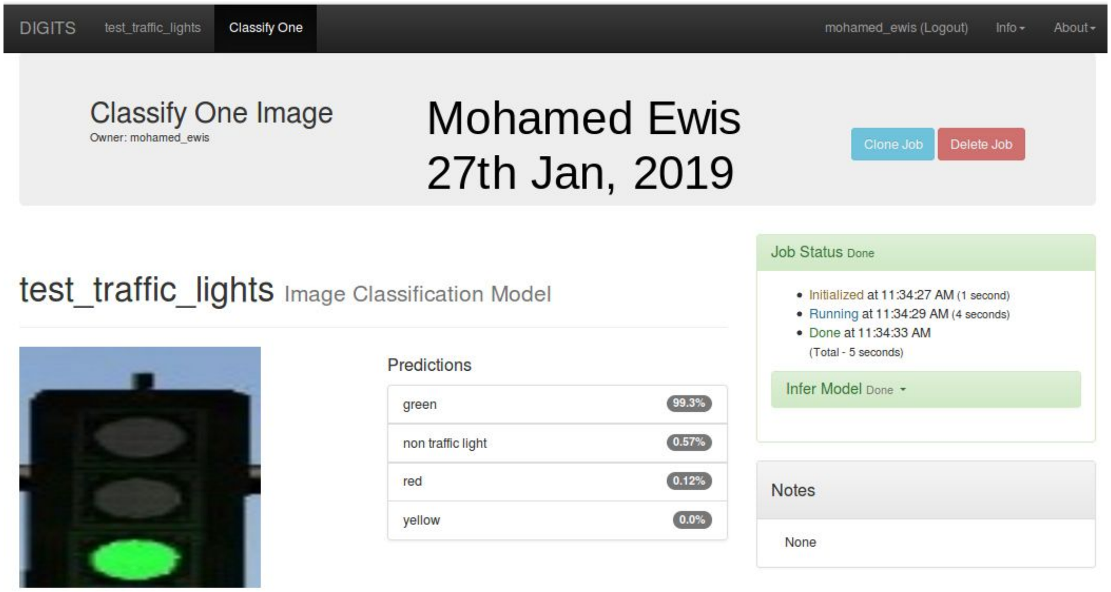


Fig. 11. AlexNet Inference_Traffic Lights model_Green

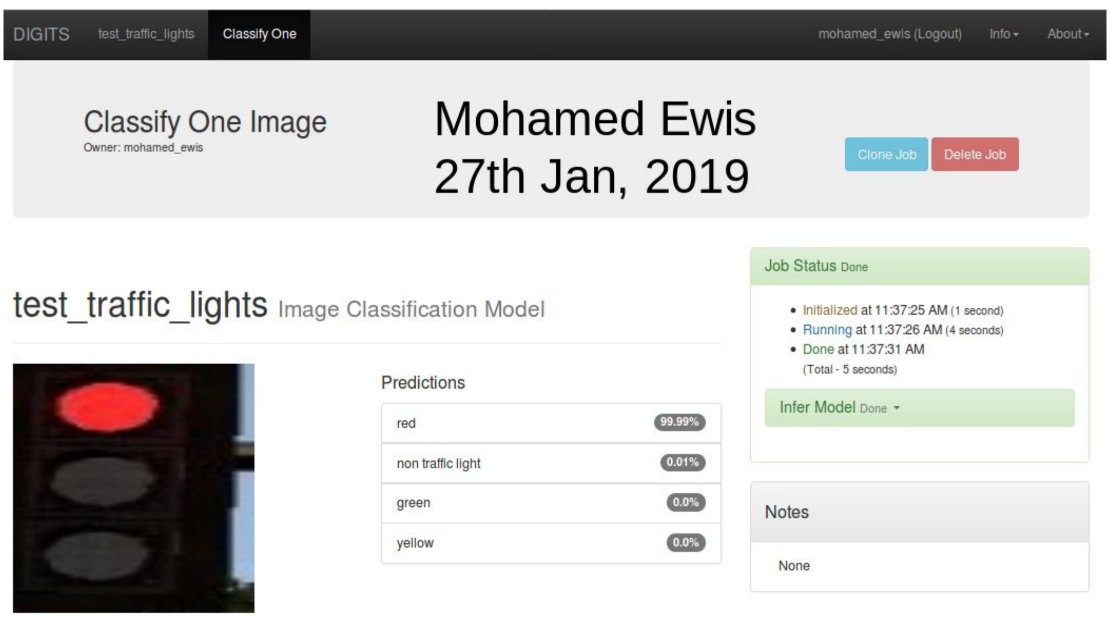


Fig. 12. AlexNet Inference_Traffic Lights model_Red

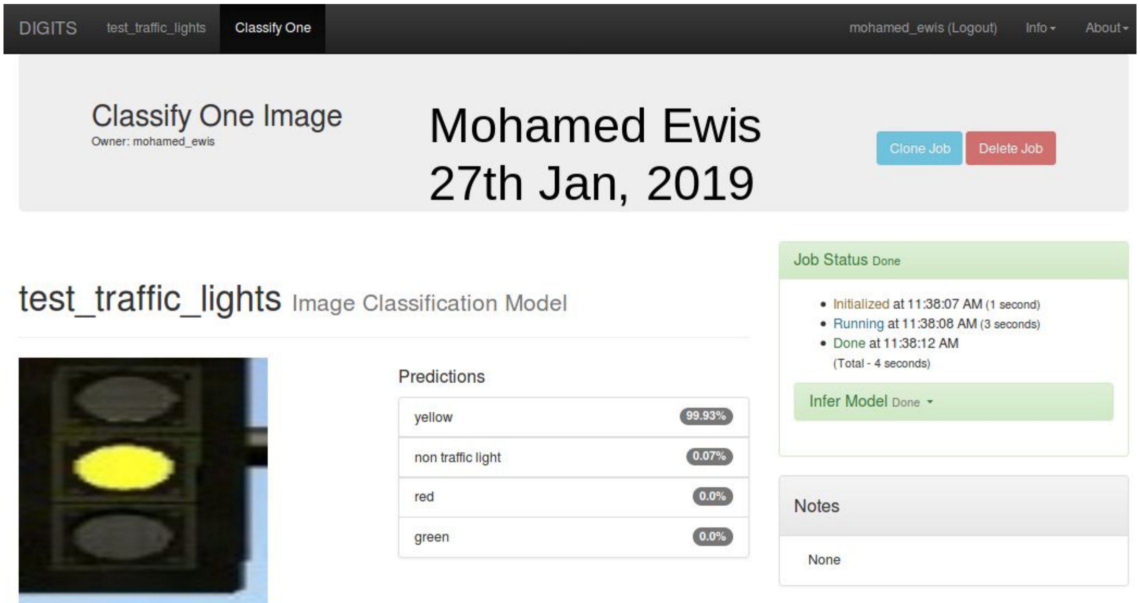


Fig. 13. AlexNet Inference_Traffic Lights model_Yellow

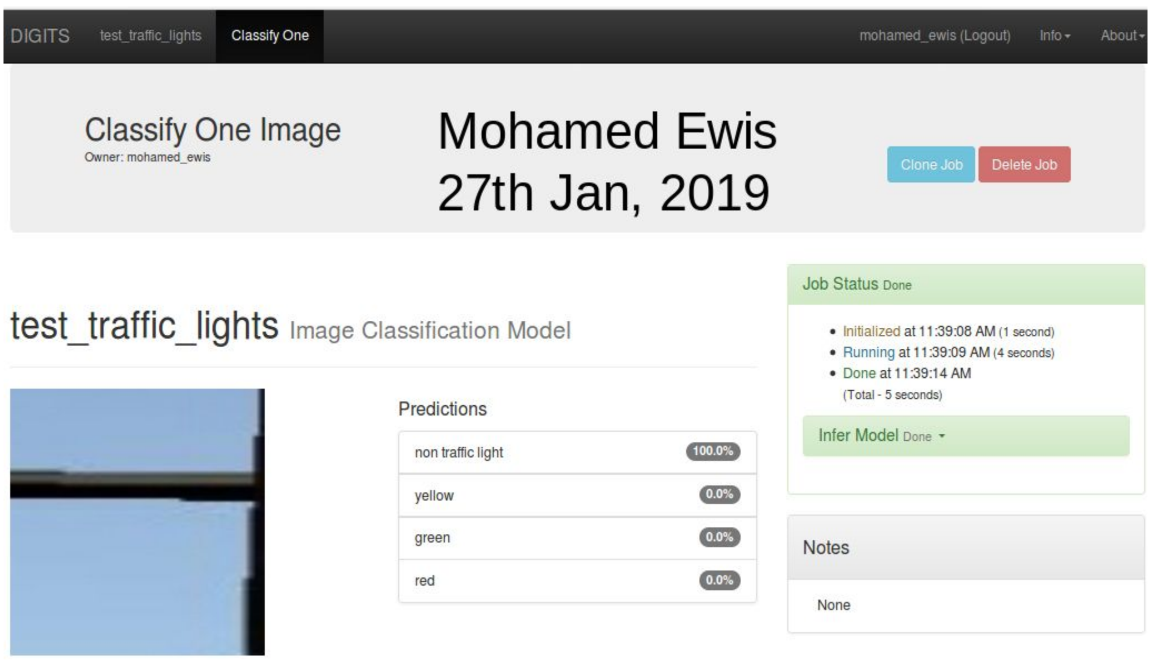


Fig. 14. AlexNet Inference_Traffic Lights model_No Light

Discussion

In the first inference task, an extremely high validation accuracy of 100 percent was obtained. For the 2nd task, i.e. classification of Traffic Lights, similar level of accuracy was obtained (99 percent) because the sample size was big enough. Also, the images have a good resolution. Also, during the 2nd inference task, the background was a solid colored background which improves the training data. 4 objects were correctly classified out of 4 in the test set. So the results were significantly acceptable.

Conclusion

- **P1 moving belt image classification**
  Both AlexNet and GoogLeNet models were used with P1 dataset provided by the lesson in moving belt image classification, the results achieved the requirement of Udacity lesson (least 75 percent accuracy and an inference time of less than 10 ms.

- **Traffic Lights image classification**
  Both AlexNet and GoogLeNet models were tested, the AlexNet model was better results at same epoch number = 5.
  The AlexNet model (with epoch = 5, Batch size = 32, Learning rate = 0.001 and test dataset = 1% of total images) achieved the
  goal (All four classes accuracy are 99-100%).

Future Work

1. Install Nvidia DIGITS system on local PC instead of using cloud GPU resource, the way is no time limitation to implement and test different projects and models.
2. Include testing object detection and segmentation implementation, and deploying the model on Jetson TX2 board and testing them in real world environment.
3. Using Nvidia DIGITS system to build and test the stock market trading system to classify and detect trading pattern.

References

[1] Nvidia, DIGITS workflow https://developer.nvidia.com/digits .
[2] Siddharth Das, CNNs Architectures:LeNet, AlexNet, VGG, GoogLeNet https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5 .