Robotics Inference

Salabson Salabiya Isa

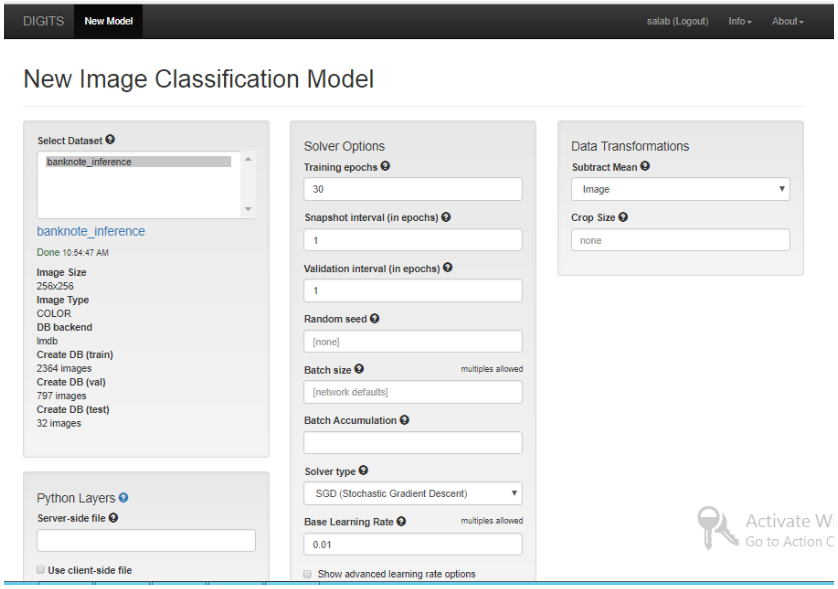
Abstract - This paper discusses project that that uses NVIDIA DIGITS workflow to train two deep neural networks to perform inference. The first one classify bottles, candy wrappers and nothing on a moving belt and the training dataset was supplied by Udacity while the second network classify three Nigeria's banknotes which include 200, 500 and 1,000 Naira and it is student own idea and the dataset also collected by the student.

1.0 INTRODUCTION

The traditional method of optical sensor has been used detect and identify different banknotes, however, use of neural to classify banknote will offer more effective and flexible alternative. In this project GoogleNet neural network is utilized to train and classify three Nigeria's banknote denominations as well as supplied data by supplying thousands of sample images. This enable these networks learn important color, geometric and pattern information from the images and make prediction accordingly.

2.0 BACKGROUND/FORMULATION

This project is an image classification problem which requires supervised learning approach. Since supervised learning means predicting input values based on the label of the training examples that you have previously provided. Therefore, the GoogleNet is used because is a good model for solving image classification problems and it has a good inference per image with considerable accuracy. The parameters used are Stochastic Gradient Descend (SGD) with the learning of 0.01 and 30 epochs for bank note classification while P1 (supplied data) classification parameters are Stochastic Gradient Descend (SGD) with the learning of 0.001 and 5 epochs.



New Image Classification Model

Select Dataset
banknote_inference

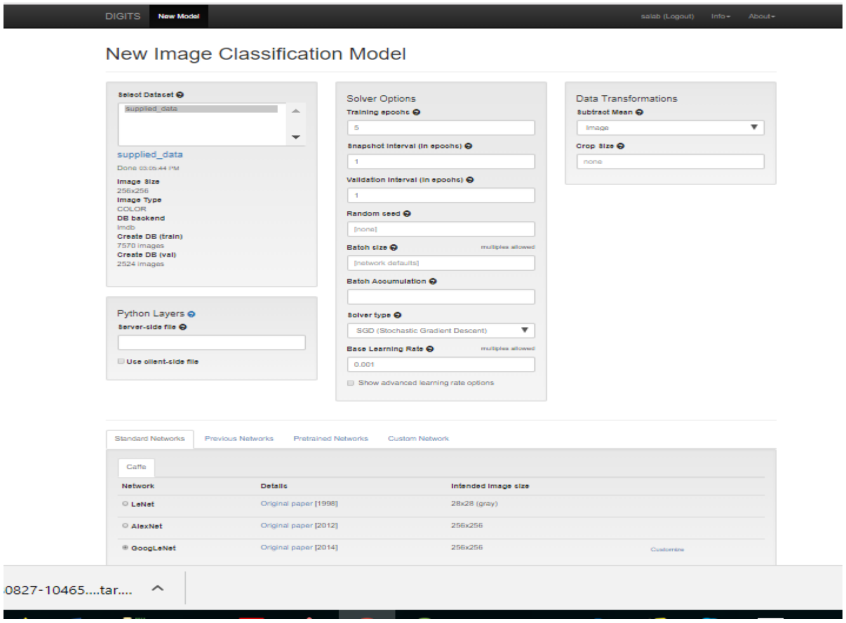
banknote_inference
Done 10:54:47 AM
Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
2364 images
Create DB (val)
797 images
Create DB (test)
32 images

Python Layers
Server-side file
□ Use client-side file

Solver Options
Training epochs: 30
Snapshot interval (in epochs): 1
Validation interval (in epochs): 1
Random seed: [none]
Batch size (multiples allowed): [network defaults]
Batch Accumulation:
Solver type: SGD (Stochastic Gradient Descent)
Base Learning Rate (multiples allowed): 0.01
□ Show advanced learning rate options

Data Transformations
Subtract Mean: Image
Crop Size: none

Figure 1: Banknote creation model Parameters



New Image Classification Model

Select Dataset
supplied_data

supplied_data
Done 03:05:44 PM
Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
7570 images
Create DB (val)
2524 images

Python Layers
Server-side file
□ Use client-side file

Solver Options
Training epochs: 5
Snapshot interval (in epochs): 1
Validation interval (in epochs): 1
Random seed: [none]
Batch size (multiples allowed): [network defaults]
Batch Accumulation:
Solver type: SGD (Stochastic Gradient Descent)
Base Learning Rate (multiples allowed): 0.001
□ Show advanced learning rate options

Data Transformations
Subtract Mean: Image
Crop Size: none

Standard Networks | Previous Networks | Pretrained Networks | Custom Network

Caffe

Network	Details	Intended image size	
○ LeNet	Original paper [1998]	28x28 (gray)	
○ AlexNet	Original paper [2012]	256x256	
● GoogLeNet	Original paper [2014]	256x256	Customize

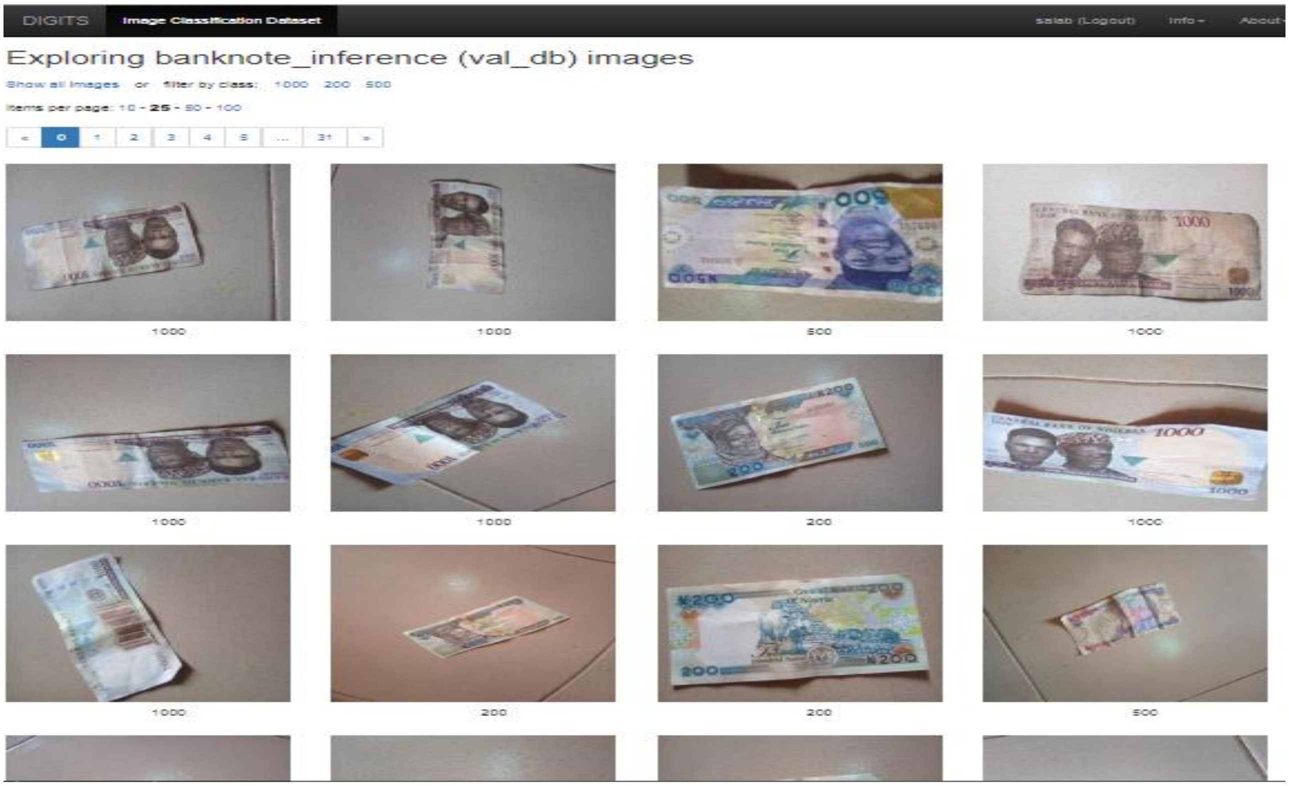
Figure 2: Supplied data Model creation parameters

3.0 DATA ACQUISITION

P1 dataset is provided by Udacity while banknote images dataset were taken with Infinix Hot 4 Pro Android phone and used Augmentor python library to generate more additional images in order to have enough data for better network accuracy. The images are RGB color and were resized to 256x256.

Below are details for the images collected for the network training:-

	200 note	**500 note**	**1000 note**
Phone Images	322	311	260
Augmented images	804	851	650
Total	**1,126**	**1,162**	**910**



Exploring banknote_inference (val_db) images

Show all images or filter by class: 1000 200 500

Items per page: 10 - **25** - 50 - 100

Figure 3: Banknote sample images

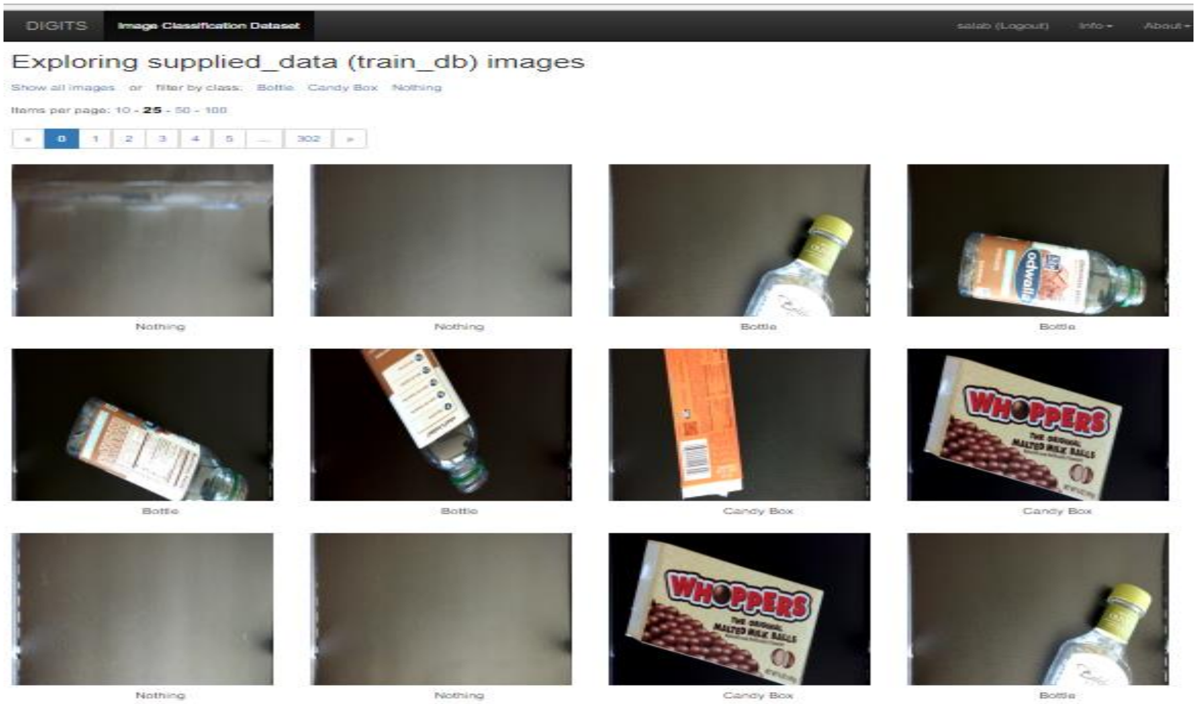
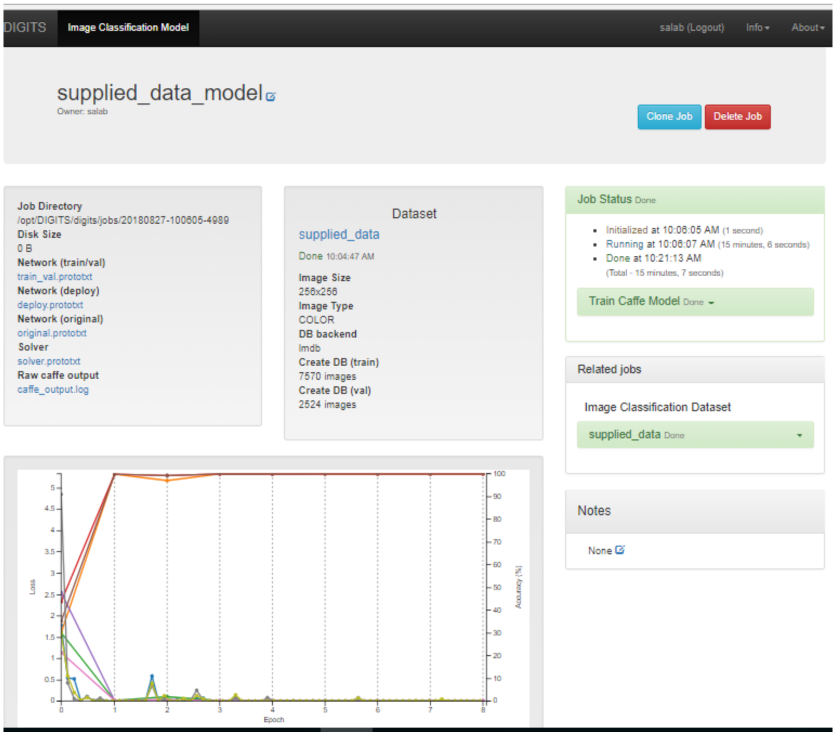

DIGITS
Image Classification Dataset
salab (Logout)
Info
About
Exploring supplied_data (train_db) images
Show all images or filter by class: Bottle Candy Box Nothing
Items per page: 10 - 25 - 50 - 100
« 0 1 2 3 4 5 ... 302 »
Nothing
Nothing
Bottle
Bottle
Bottle
Bottle
Candy Box
Candy Box
Nothing
Nothing
Candy Box
Bottle

Figure 4: (P1) Supplied data sample images

4.0 RESULTS

The network with the supplied data was trained with GoogleNet and 100% accuracy achieved during the training. Upon evaluation, the model meets the required accuracy with 75.41% accuracy and the inference time was 5.06 ms which also meets the required inference time of below 10 ms.

On the other hand, the banknote's model was able to achieve accuracy of 99.87%.



supplied_data_model

Owner: salab

Job Directory
/opt/DIGITS/digits/jobs/20180827-100605-4989
Disk Size
0 B
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

Dataset

supplied_data

Done 10:04:47 AM

Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
7570 images
Create DB (val)
2524 images

Job Status Done

- Initialized at 10:06:05 AM (1 second)
- Running at 10:06:07 AM (15 minutes, 6 seconds)
- Done at 10:21:13 AM
(Total - 15 minutes, 7 seconds)

Train Caffe Model Done

Related jobs

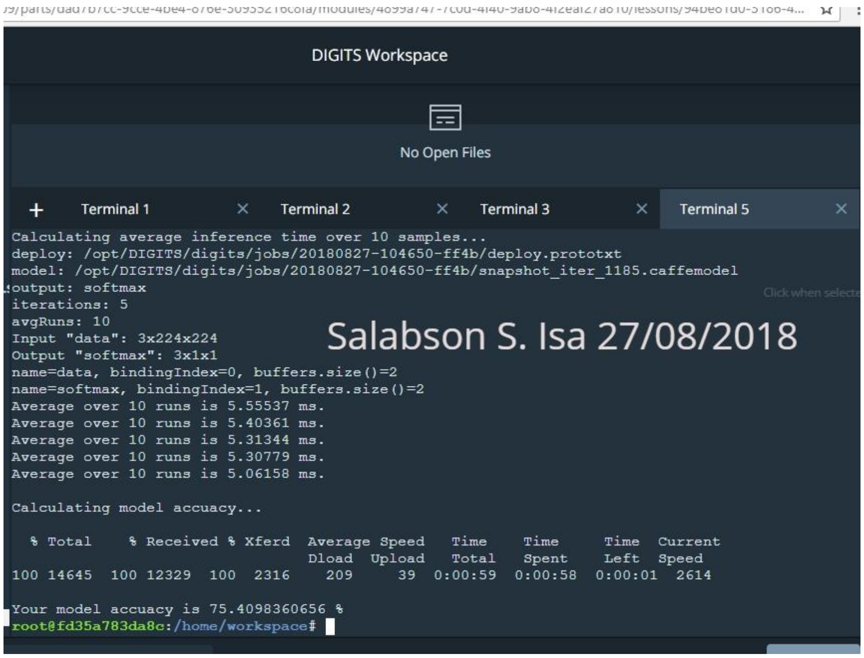
Image Classification Dataset

supplied_data Done

Notes

None

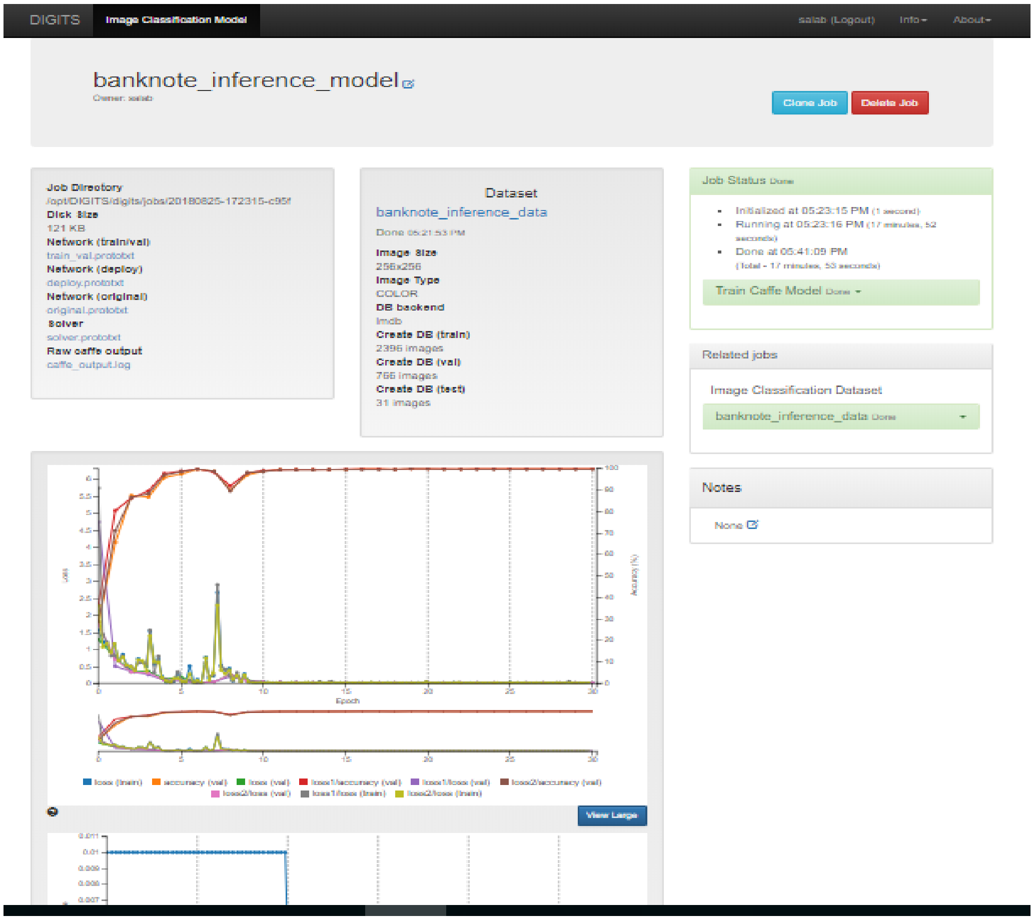
Figure 5: Suppled data training graph


J9/parts/dad7b7cc-9cce-4be4-876e-5095521bc0fa/modules/4899a747-7c0d-4140-9ab8-412ea127a610/lessons/94be61d0-5166-4...
DIGITS Workspace
No Open Files
+
Terminal 1
Terminal 2
Terminal 3
Terminal 5
Click when selecte

```
Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180827-104650-ff4b/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180827-104650-ff4b/snapshot_iter_1185.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 5.55537 ms.
Average over 10 runs is 5.40361 ms.
Average over 10 runs is 5.31344 ms.
Average over 10 runs is 5.30779 ms.
Average over 10 runs is 5.06158 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14645  100 12329  100  2316    209     39  0:00:59  0:00:58  0:00:01  2614

Your model accuacy is 75.4098360656 %
root@fd35a783da8c:/home/workspace# 
```

Salabson S. Isa 27/08/2018

Figure 6: Supplied data evaluation accuracy



banknote_inference_model

Owner: salab

Job Directory
/opt/DIGITS/digits/jobs/20180825-172315-c95f
Disk Size
121 KB
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

Dataset
banknote_inference_data
Done 05:21:53 PM
Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
2396 images
Create DB (val)
766 images
Create DB (test)
31 images

Job Status Done

- Initialized at 05:23:15 PM (1 second)
- Running at 05:23:16 PM (17 minutes, 52 seconds)
- Done at 05:41:09 PM (Total - 17 minutes, 53 seconds)

Train Caffe Model Done ▾

Related jobs

Image Classification Dataset
banknote_inference_data Done ▾

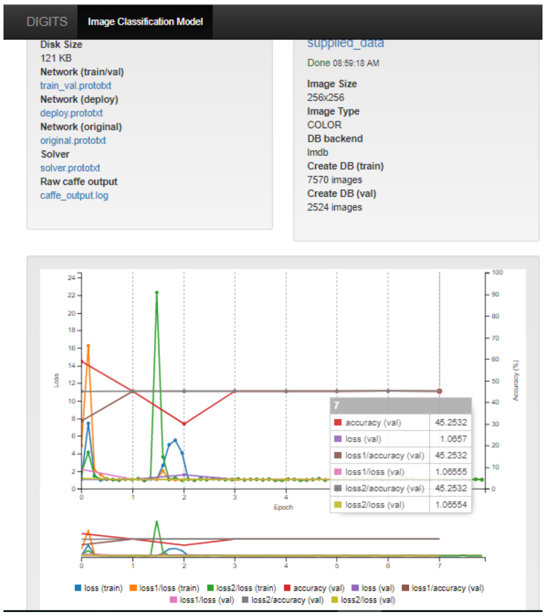
Notes

None

Figure 7: Banknote training graph

5.0 DISCUSSION

Both models achieved reasonable level of accuracy because considerable amount of sample image datasets were provided for the training and also right parameters were chosen. For example, initially when learning rate of 0.01 was used on the supplied data model, the model achieved accuracy of only 45.25% compared to 100% accuracy when learning rate of 0.001 was used. In the same vein, when Adam optimizer was used, upon evaluation only 73.77% accuracy was realized compared to 75.41% accuracy when Adam optimizer was replaced by Stochastic Gradient Descent.



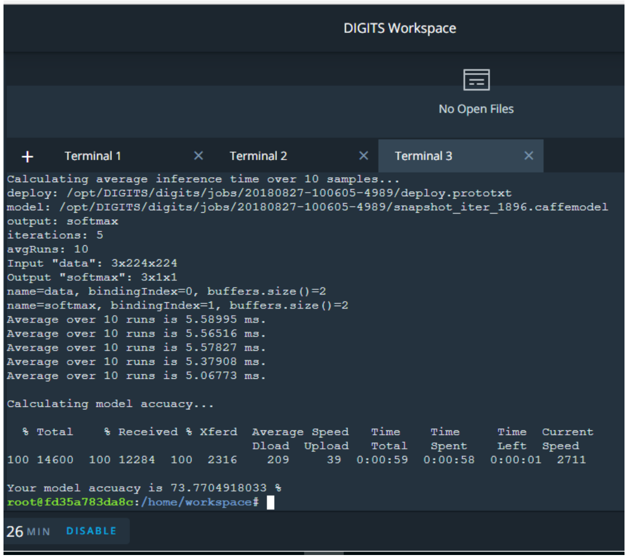
Done 08:59:18 AM

Disk Size
121 KB
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
7570 images
Create DB (val)
2524 images

7	
accuracy (val)	45.2532
loss (val)	1.0657
loss1/accuracy (val)	45.2532
loss1/loss (val)	1.06555
loss2/accuracy (val)	45.2532
loss2/loss (val)	1.06554

Figure 8 When 0.01 learning rate was used



```
Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180827-100605-4989/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180827-100605-4989/snapshot_iter_1896.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 5.58995 ms.
Average over 10 runs is 5.56516 ms.
Average over 10 runs is 5.57827 ms.
Average over 10 runs is 5.37908 ms.
Average over 10 runs is 5.06773 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14600  100 12284  100  2316    209     39  0:00:59  0:00:58  0:00:01  2711

Your model accuacy is 73.7704918033 %
root@fd35a783da8c:/home/workspace#
```

Figure 9: When Adam optimizer was used

6.0 CONCLUSION/FUTURE WORK

GoogleNet model was used to design two inference networks, one with P1 dataset provided by udacity and the required accuracy of at least 75% accuracy and inference time of below 10 ms are achieved. The other was built with own collected dataset and achieved accuracy of 99.887%.

Because the data collection was tedious, second model was limited to classifying only three banknotes from the available banknotes.

In future, all available banknotes will be included and the network will extrapolated to build counterfeit banknote detection system.

REFERENCES

[1] Siddhart Das: CNN Architectures:LeNet, Alexnet, VGG, GoogleNet, ResNet and more.., https://medium.com/@sideral/ cnn architectures-lenet-alexnet-vgg-googlenet-resnet-and more , 2017.

[2] Marcus Bloise: Data Aumentation, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton: ImageNet Classification with Deep Convolutional Networks, NIPS 2012