Inference project

Ammar Yasser

Abstract—Inference is one of the most important steps to implement deep-learning ideas after various trials to find the right and suitable model for the application. In this report, Nvidia DIGITS will help to find the perfect model for our idea. First, an example will be introduced for using DIGITS and evaluating the models. Then, the idea of banknotes will be presented going through the steps of data collection and implementation and discussion on the result and future work.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, deep learning.

1 Introduction

DIGITS (the Deep Learning GPU Training System) is a web-app for training deep learning models. Classification is one of these model which DIGITS can help to create using well-known networks like ALexNet and GoogleNet. the project idea is to classify some of Egyptian banknotes which will help blind people to identify their cash-flow and make it easy for them to deal with money in everyday situations. The idea will be initially implemented on 200LE, 100LE and 50LE using both AlexNet and GoogleNet to find the most accurate model to deploy it in the embedded frame.

2 DIGITS Tutorial

With the available data-set provided by the workspace ./data/P1-data/ for classifying 2 different objects: bottle , candy box and also noting notation. Using both AlexNet and GoogleNet with 5 epoches and initial learning rate 0.01. for the following result, the both Nets worked very well by achieving 75 percent of accuracy

2.1 AlexNet and GoogleNet

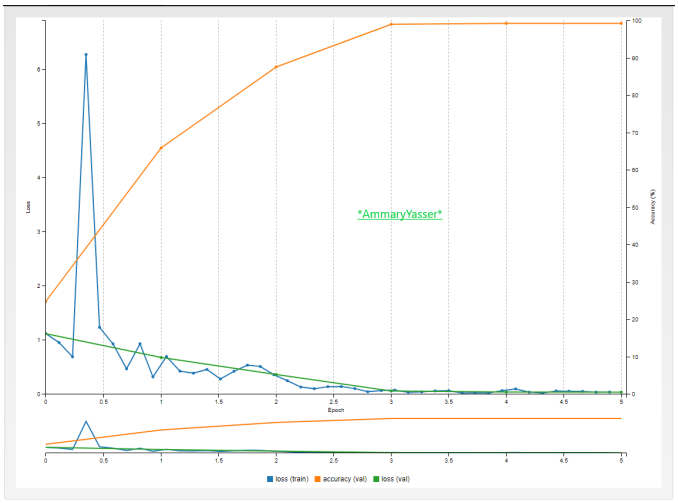

Fig. 1. AlexNet P1 epochs.

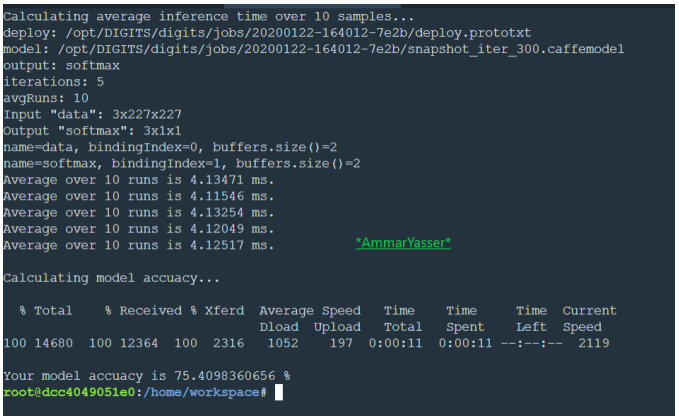

Fig. 2. AlexNet P1 Evaluate.

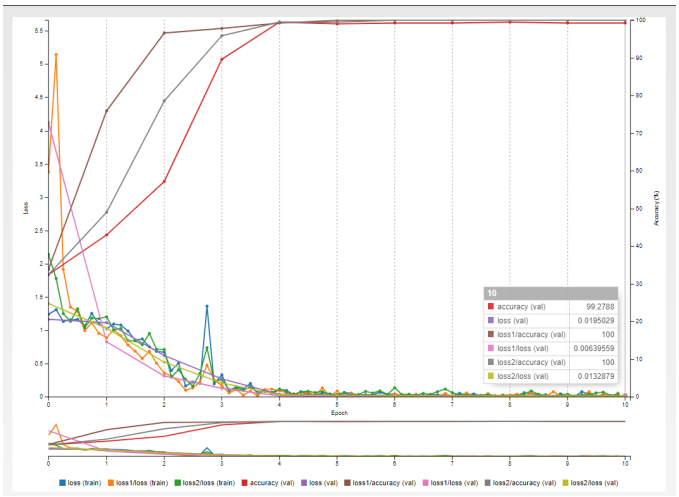

Fig. 3. GoogleNet P1 epochs.

3 Inference Idea

For this section, Idea of Egytian banknotes will follow the same pattern to find the classification model. But first, Data should be collected.

```
model: /opt/DIGITS/digits/jobs/20200122-162119-f9cb/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 5.45191 ms.
Average over 10 runs is 5.42851 ms.
Average over 10 runs is 5.41675 ms.
Average over 10 runs is 5.0007 ms.
Average over 10 runs is 4.91562 ms.          *AmmarYasser*

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14657  100 12341  100  2316    212     39  0:00:59  0:00:58  0:00:01  2491

Your model accuacy is 75.4098360656 %
root@dcc4049051e0:/home/workspace#
```

Fig. 4. GoogleNet P1 Evaluate.

3.1 Data Collection

first, Making a short video where the banknote appears with different angles and orientations. each video is almost 40sec with 30 fps. Using VLC media player 'scene filter' option to catch the frames from the video and save the images within a specific folder under the name of the classified banknote. then, Using https://bulkresizephotos.com/ to resize the images to 256x256 size to be perfect inputs to the Nets.

Fig. 5. One Hundred.

3.2 Training

by uploading the data-set to the work space and train it by both AlexNet and GoogleNet.

4 Results

by testing some images by the model, GoogleNet is showing a good performance than AlexNet. Models can be found in folder on Google Drive : https://bit.ly/3aDFGxX

5 Conclusion / Future work

concluding, the GoogleNet can be used for the implementation. it also shows a good accuracy with high speed. for future, the data set should be bigger to contain the whole Egyptian banknotes then, Designing a good physical platform with inference device to be easily used.

References

Fig. 6. One Hundred resized.

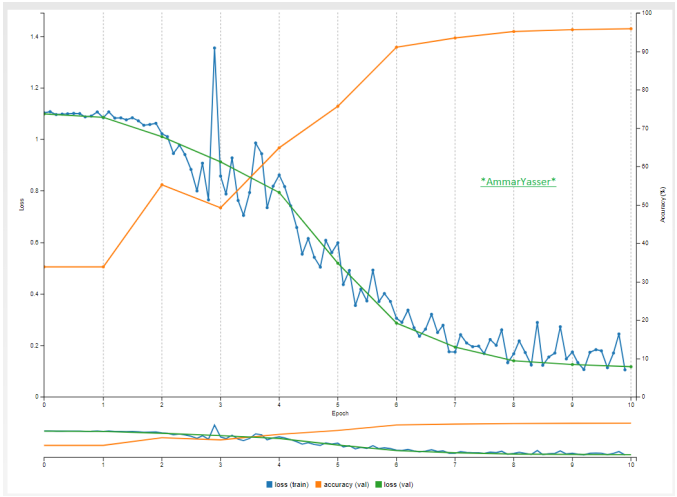

Fig. 7. AlexNet.

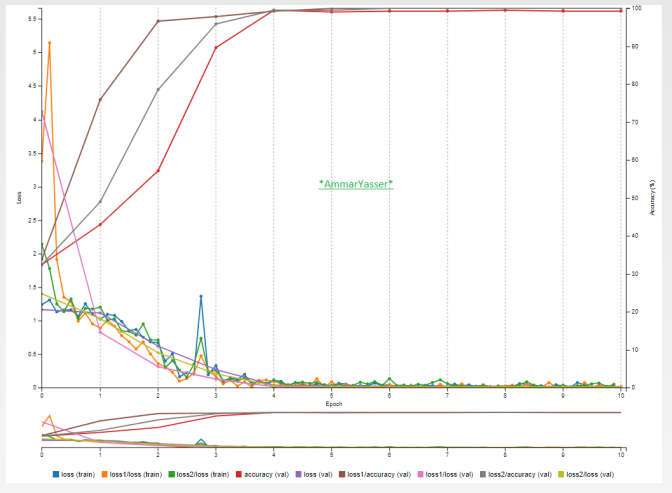

Fig. 8. GoogleNet.

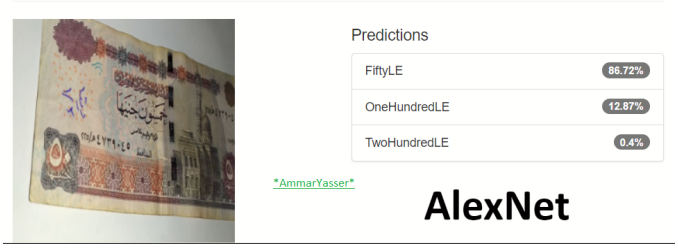

Fig. 9. AlexNet prediction.

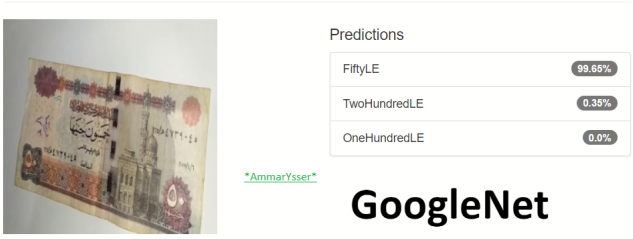

Fig. 10. GoogleNet prediction.