Robotic Inference Project

Angelakis Panagiotis

Abstract—In this project, two Convolutional Neural Networks are designed and tested for classification tasks, using NVIDIA DIGITS workspace. The first goal is to design and train a Neural Network on data supplied to us, achieving at least 75% accuracy and inference time under 10 ms. After we are tasked with collecting our own dataset for classification and designing a new neural network suitable for this dataset.

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

1 Introduction

In many automatic industrial settings, we want fast and accurate detection of defective items with visual means. In many cases, a human is still required for this task which is a laborious and error prone task. With the advances in Pattern recognition using Neural Networks, now it is possible to automate this task, with an accuracy higher of a human. For this task accuracy is not enough, we want fast response from our classifier if we want it to use it for real time tasks. Thus we require an embedded system for a efficient recognition pipeline, with a capable gpu. Here we will try to detect defective bayonets in a production line. Nvidia Digits workspace allowed us to quickly design and tune our networks and datasets in a graphical environment.

2 Background / Formulation

For image classification NVIDIA DIGITS provide 3 award winning CNN's (AlexNet, GoogleNet, LeNet), with the ability of modifying the architecture of the Fully Connected layers or hyper-paramaters. GoogleNet was chosen for our task due to its high accuracy and inference time while being the most efficient (accuracy per operation), reaching the required 75% accuracy and under 10ms inference time. For my collected dataset GoogleNet was initially chosen, but as it reached 99% accuracy due to the robustness of my dataset a smaller and faster Network architecture was chosen AlexNet to improve on inference time.

DNN	Top1 Accuracy	Operations	Parameter	Inference	Power	Memory	Info Density
Network name	%	G-Ops	M	fps	W	MB	%accuracy/ Mparams
AlexNet-BN	57	2	60	50	10.9	310	1.0
GoogleNet	68	3	7	33	10.7	200	9.7
VGG-16	71	31	135	6	11.8	850	0.5
ResNet-101	76	16	45	10	12.9	300	1.7
Inception-v3	78	12	25	12	11.6	200	3.1

Sampling of metrics for architectures running ImageNet inference on a Jetson TX1

Fig. 1. Compare different architectures

- Supplied dataset : GoogleNet, SGD optimizer, 0.005 learning rate, 25 epoch
- Collected dataset : AlexNet, SGD optimizer, 0.005 learning rate (exponential), 25 epoch

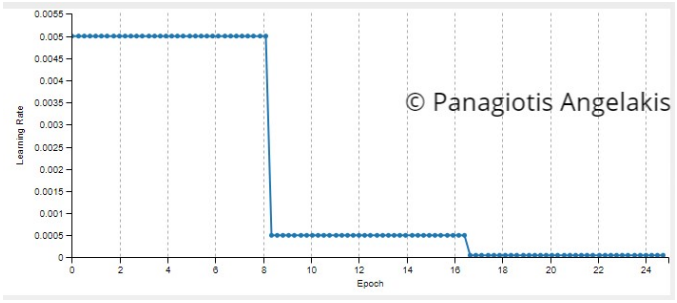

Fig. 2. Learning rate step down

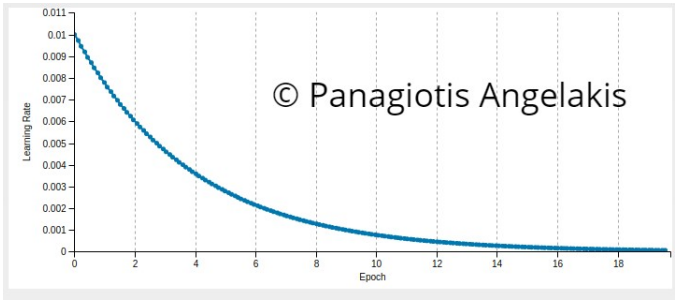

Fig. 3. Learning rate exponential decay

3 Data Acquisition

3.1 Supplied Dataset

This dataset is supplied to us (8-bit RGB PNG 500X500), it consist of three categories.

- Bottle (4568)
- Candy box (2495)
- Nothing (3031)

3.2 Collected Dataset

My inference idea is to trainonets a network to distinguish defective bayonets from good bayonets or the absence of them. The images were produced from a phone camera and afterwards they were resized to 256x256 8-bit RGB PNG which is the default input size of the AlexNet and GoogleNet. Phone cameras provide high resolution images which maintain more information in the downsize process.

Fig. 4. A sample of the supplied dataset

Some examples of the collected images can be seen below along with the total number of pictures (approximately 500 per class).

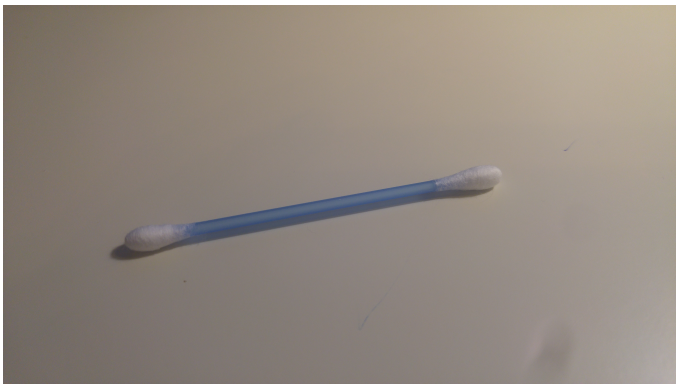

Fig. 5. A normal bayonet

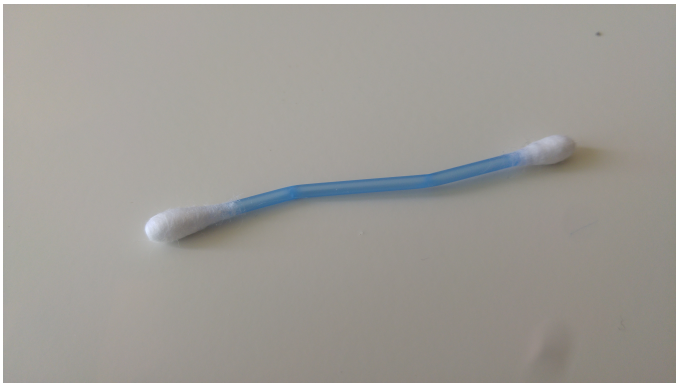

Fig. 6. A Defective bayonet

4 Results

4.1 Supplied dataset Results

The accuracy in the training set was almost 100% with low validation loss, then this network was tested with an evaluation script, which test our network on a new test

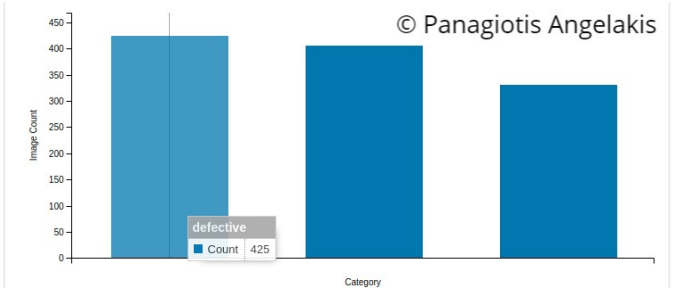

Fig. 7. Number of images per class

dataset the accuracy on the new dataset was 75.4% with inference time 5.5ms satisfying the project requirements. Images can be seen below:

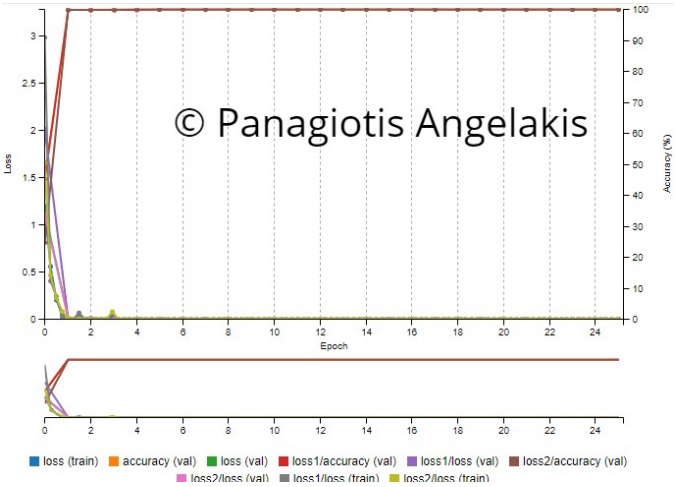

Fig. 8. Training accuracy and Validation loss on the supplied dataset

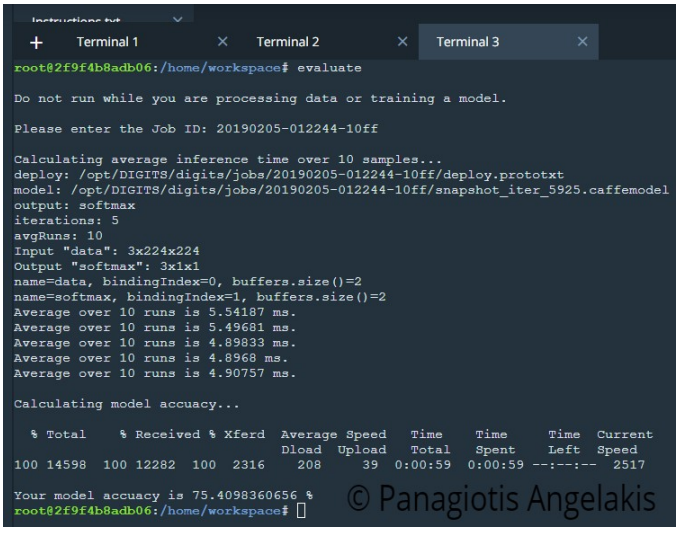

Fig. 9. Evaluate command

4.2 Collected dataset results

Again the accuracy in the training set was close to 100%, although this migh not be the case on new samples or

real world examples, the validation loss was also low with GoogleNet performing better than AlexNet in this regard, however AlexNet was chosen due to its smaller size of paramaters and layers resulting in faster inference time (although we cant measure it) Below we can see the photos depicting our results:

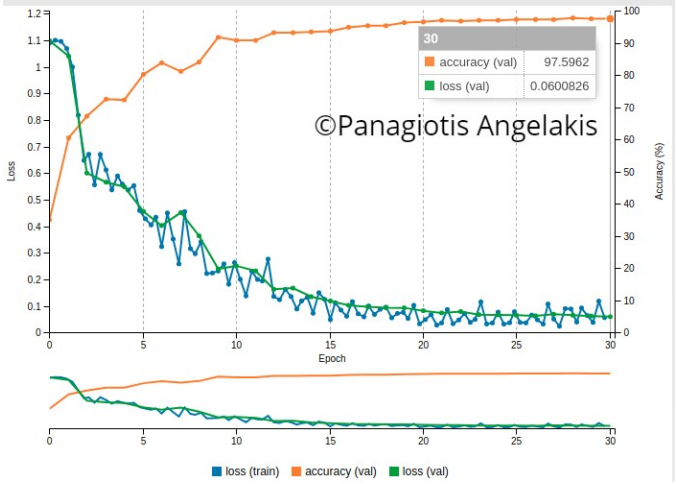

Fig. 10. Training accuracy and Validation loss on the collected dataset (AlexNet)

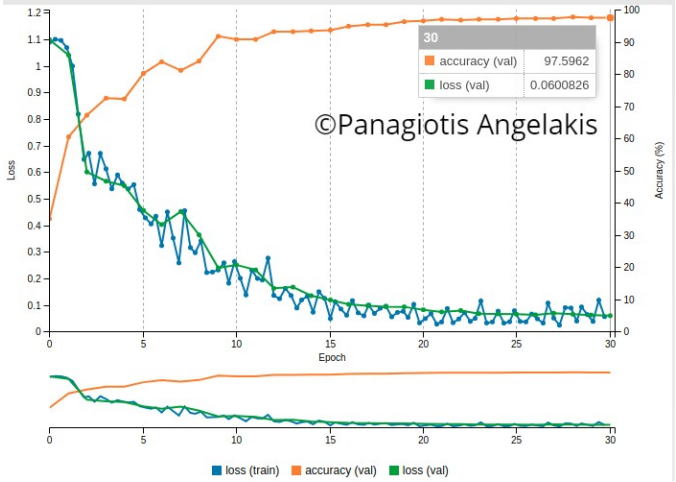

Fig. 11. Training accuracy and Validation loss on the collected dataset (GoogleNet)

Fig. 12. Prediction accuracy on a test image

5 Discussion

For both datasets high training accuracy was achieved with a low validation loss, however this accuracy may drop on objects that it wasnt trained on like different candy boxes, bottles or bayonets and extra care should be given depending on the particular application we want to deploy our system, this should be addressed with additional data that cover these areas. This is apparent, when we see the 75% accuracy on the test set provided to us, although this is a good indication that our network is not overfitting,it scored lower than on the training/test dataset.

6 Conclusion / Future work

With the help of the NVIDIA DIGITS workspace we quickly built our dataset and models, achieving satisfactory results on the dataset provided to us with 75% accuracy on a test dataset and inference time 5.5ms meeting the requirements, after a dataset of defective and normal bayonets was collected using a phone camera and GoogleNet and AlexNet were tested, AlexNet was chosen due to its simplicity and low inference time, due to the robustness of the dataset a high accuracy was achieved on the training/test dataset, although some additional fine tuning may be needed for real world applications. In the future, a detection network will be built to detect the location of specific objects, while a rover automatically maps and navigates the environment as part of my final graduation project for my university.