

Robotic Inference

Seyfi Gozubuyuk

Abstract—Deep Learning is widely used in object classification. Convolutional Neural Networks improved the success of the computers on object detection, recognition, and classification. They are commonly used in Self-Driving Cars, robotics and other fields of machine learning. There exist models such as LeNet, AlexNet, and GoogleNet. Nvidia Digits provides a web interface to utilize these models and create other models. It is possible to upload data and test it with these models. This project includes different classification tasks on image datasets.

Index Terms—Robot, IEEETran, Udacity, L^AT_EX, Deep Learning, Convolutional Neural Networks, Nvidia Digits.

1 INTRODUCTION

THE goal in this project is to utilize Nvidia Digits application to perform classification on image datasets. There are two datasets. The first one is the provided data that contains bottle and candy box images. Besides, there exist samples that do not include any.

The second dataset was The German Traffic Sign dataset [1]. There are 43 different traffic signs. This dataset was selected because it is related with self-driving field and it has many different classes.

Nvidia Digits provide an interface to load the data, create a new model or utilize predefined models. Besides, it enables to customize the models and select hyperparameters. During training, it saves the model on every epoch. This saved models can be used as pretrained models to train more.

2 BACKGROUND / FORMULATION

The problem is to classify objects using Deep Learning. Convolutional neural networks are an excellent choice [2]. Backpropagation provided enhanced learning for Multi-layer Neural Networks.

LeCun et al. developed LeNet to classify digits in 1998 [3]. LeNet was a seven-layer CNN. Krizhevsky et al. built a convolutional neural network which has sixty million parameters, half million neurons and five convolutional layers in 2012 [4]. Szegedy et al. trained another deep neural network in 2014 which has 22 layers [5].

2.1 Task 1: Bottle vs Candy-Box

In the first task, AlexNet and GoogleNet were selected to classify the samples of bottles and candy-boxes. The reason for choosing AlexNet is its speed, and the reason for working with the GoogleNet was its higher accuracy. Five different models trained while completing this task. Since adam and RMSprop help the gradient descent to reach the minima faster, these solver types were allied with AlexNet and GoogleNet. Besides the Stochastic Gradient Descent were selected with AlexNet as the default option. Learning rate 0.01 was a reasonable choice to prevent overfitting and from overshooting. The image sizes were compatible with AlexNet and GoogleNet. Therefore there were no changes in

dimensions. The number of epochs parameter has different values to experiment. The results section discuss the details of the models tried and the outcomes.

2.2 Task 2: Traffic Sign

The selected dataset has 39,209 images in 43 classes for training. Besides, there are 12,530 images to test. The image size is varying between 15x15 and 250x250 pixels. This situation forced the use of LeNet. All samples were resized by squashing to 28x28 during loading. Enlarging the images would result in excessive training. From the experience from the first task, the solver type was selected as stochastic gradient descent. The learning rate was again 0.01 since it provided satisfying results in the previous part of the project.

3 DATA ACQUISITION

The first task does not contain any changes on the data, since the models were compatible with image size of 256x256.

The second task works on the traffic signs data. The main reasons are to select a dataset related to autonomous vehicles and to select a dataset with many classes. A problem with the dataset was, it had varying image sizes, and these sizes were small to use AlexNet or GoogleNet. Therefore, the LeNet model was the remaining choice for this part of the project. Since LeNet takes input grayscale images with size 28x28, all images in the dataset were resized to 28x28 and converted to grayscale. Nvidia Digits automatically performed these operations during loading the data. The number of epochs was selected as ten to prevent overfitting. Figure 1 shows samples from 14 different classes.



Fig. 1. Samples From 14 Different Classes

4 RESULTS

4.1 Task 1: Bottle vs Candy-Box

The results for the five different model for the Task 1 are given on Table 1.

TABLE 1
Results on Task 1

No.	Model	Epoc	Solver	Rate	Accuracy
1*	AlexNet1	30	RSMprop	0.01	73.77%
2*	AlexNet1E17	5	RSMprop	0.0001	70.49%
3	GoogleNet	6	Adam	0.01	67.21%
4*	GoogleNetE6	4	Adam	0.01	67.21%
5	AlexNet2	10	SGD	0.01	75.41%

1* RMS decay value for 1.AlexNet1 and 2.AlexNet1E17 were 0.99

2* AlexNet1E17 started from the Epoch 17 of model 1.AlexNet1

4* GoogleNetE6 started from the Epoch 6 of model 3.GoogleNet

Figure 2 shows the screenshot of the console when evaluating the AlexNet2 model for Task1.

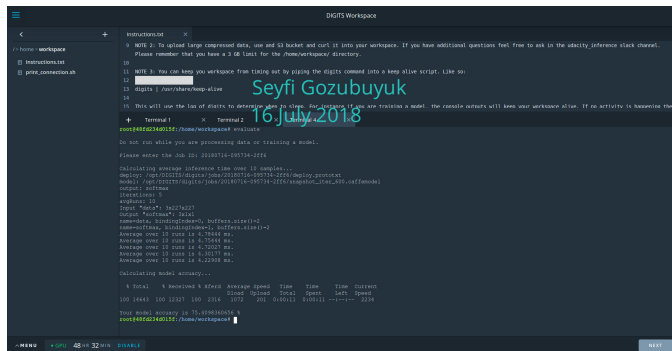


Fig. 2. Screenshot of Task 1 AlexNet2

Figure 3 shows the training graph of successful AlexNet2 model. Figure 4, Figure 5, Figure 6, and Figure 7 shows the classification results of testing AlexNet2.

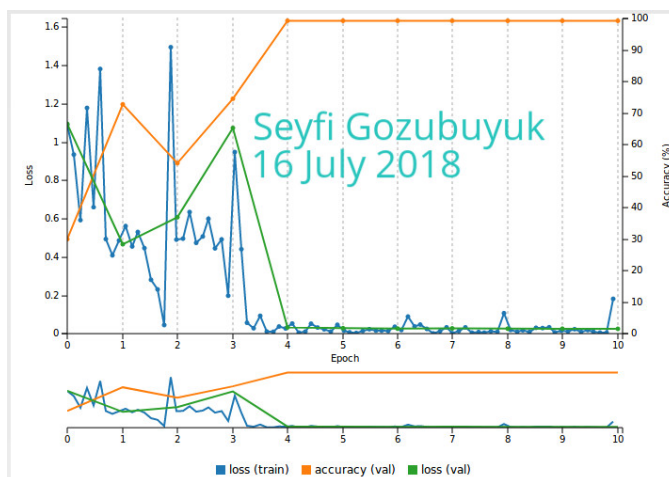


Fig. 3. Training Graph of Task 1 AlexNet2

Figure 8 shows the training graph of unsuccessful AlexNet1 model. The accuracy graph shows that the highest accuracy was obtained at epoch 17.

Path	Top predictions
1 /data/P1_data/Candy_Box/reeses_17.png	Candy Box 99.9% Bottle 0.0% Nothing 0.0%
2 /data/P1_data/Candy_Box/reeses_42.png	Candy Box 99.9% Bottle 0.0% Nothing 0.0%
3 /data/P1_data/Candy_Box/reeses_680.png	Candy Box 100.0% Bottle 0.0% Nothing 0.0%
4 /data/P1_data/Candy_Box/reeses_933.png	Candy Box 100.0% Bottle 0.0% Nothing 0.0%
5 /data/P1_data/Candy_Box/woppers_1192.png	Candy Box 100.0% Bottle 0.0% Nothing 0.0%
6 /data/P1_data/Candy_Box/woppers_1442.png	Candy Box 100.0% Bottle 0.0% Nothing 0.0%
7 /data/P1_data/Candy_Box/woppers_335.png	Candy Box 100.0% Bottle 0.0% Nothing 0.0%
8 /data/P1_data/Candy_Box/woppers_586.png	Candy Box 99.1% Bottle 0.0% Nothing 0.0%
9 /data/P1_data/Candy_Box/woppers_836.png	Bottle 65.7% Candy Box 12.9% Nothing 2.3%
10 /data/P1_data/Bottle/coconut_1449.png	Bottle 99.3% Candy Box 0.0% Nothing 0.0%
11 /data/P1_data/Bottle/coconut_522.png	Bottle 99.9% Candy Box 0.0% Nothing 0.0%
12 /data/P1_data/Bottle/coconut_980.png	Bottle 99.3% Candy Box 0.0% Nothing 0.0%
13 /data/P1_data/Bottle/dressing_1437.png	Bottle 99.3% Candy Box 0.0% Nothing 0.0%
14 /data/P1_data/Bottle/dressing_524.png	Bottle 84.4% Nothing 15.5% Candy Box 0.0%
15 /data/P1_data/Bottle/dressing_983.png	Bottle 99.9% Candy Box 0.0% Nothing 0.0%
16 /data/P1_data/Bottle/odwalla_143.png	Bottle 93.3% Candy Box 5.7% Nothing 0.0%
17 /data/P1_data/Bottle/odwalla_537.png	Bottle 99.9% Candy Box 0.0% Nothing 0.0%
18 /data/P1_data/Bottle/odwalla_995.png	Bottle 99.9% Candy Box 0.0% Nothing 0.0%
19 /data/P1_data/Nothing/Nothing_pic_1378.png	Nothing 100.0% Bottle 0.0% Candy Box 0.0%
20 /data/P1_data/Nothing/Nothing_pic_1768.png	Nothing 100.0% Bottle 0.0% Candy Box 0.0%
21 /data/P1_data/Nothing/Nothing_pic_2159.png	Nothing 100.0% Bottle 0.0% Candy Box 0.0%
22 /data/P1_data/Nothing/Nothing_pic_254.png	Nothing 100.0% Candy Box 0.0% Bottle 0.0%
23 /data/P1_data/Nothing/Nothing_pic_2940.png	Nothing 100.0% Bottle 0.0% Candy Box 0.0%
24 /data/P1_data/Nothing/Nothing_pic_599.png	Nothing 100.0% Candy Box 0.0% Bottle 0.0%
25 /data/P1_data/Nothing/Nothing_pic_989.png	Nothing 100.0% Bottle 0.0% Candy Box 0.0%

Fig. 4. Testing 25 Images with Task 1 AlexNet2



Fig. 5. Testing a Bottle Image Task 1 AlexNet2



Fig. 6. Testing a Candy-Box Image Task 1 AlexNet2



Fig. 7. Testing a Nothing Image Task 1 AlexNet2

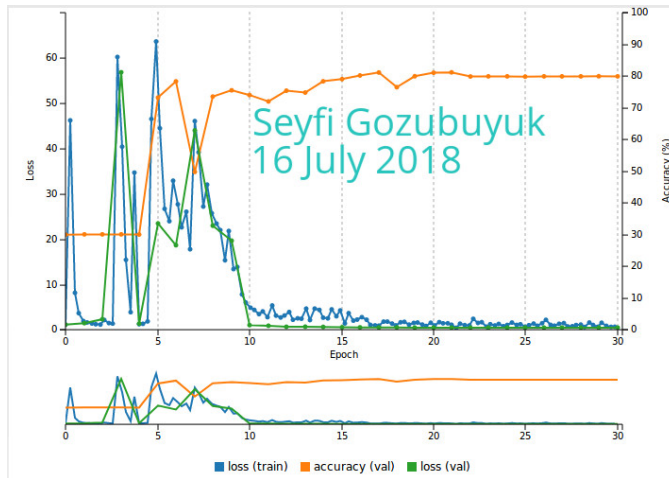


Fig. 8. Training Graph of Task 1 AlexNet1

4.2 Task 2: Traffic Sign

Figure 9 show the training graph of the LeNet for the second task. It can be seen from the graph that, the accuracy did not change significantly after ten epochs. The validation accuracy at tenth epoch was 98.7276% and for the thirtieth epoch 98.7581%. The highest accuracy was 98.7785% for the nineteenth epoch.

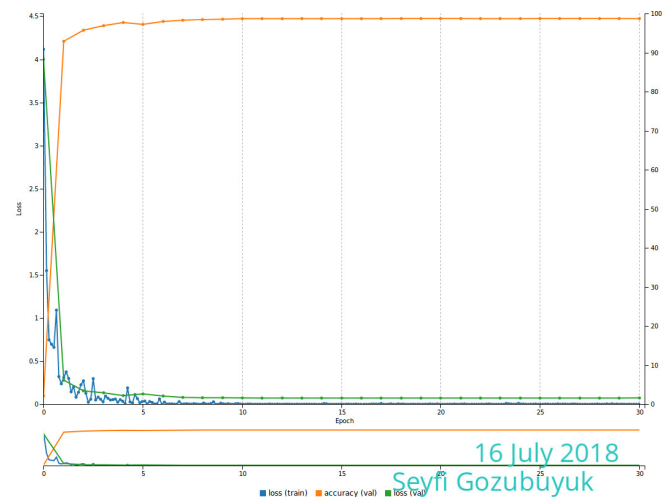


Fig. 9. Training Graph of Task 2 LeNet

Figure 10 shows the top 9 predictions for category. Figure 11, Figure 12, and Figure 13 shows sample difficult predictions.

5 DISCUSSION

The Figure 8 and Figure 3 show that SGD worked better on both the validation dataset and the test dataset for the Task 1. The GoogleNet was not successful for this dataset. The reason was the use of Adam optimizer instead of SGD and the small number of epochs. It is required to either increase the number of epochs or select the SGD optimizer.

For the second task, due to time and hardware limitations, the available datasets were limited. Most of the

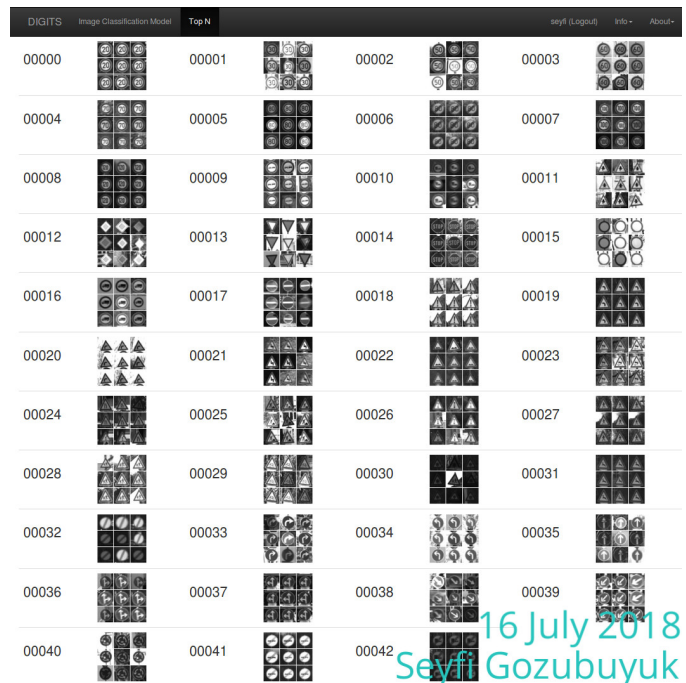


Fig. 10. Top 9 Predictions per Category

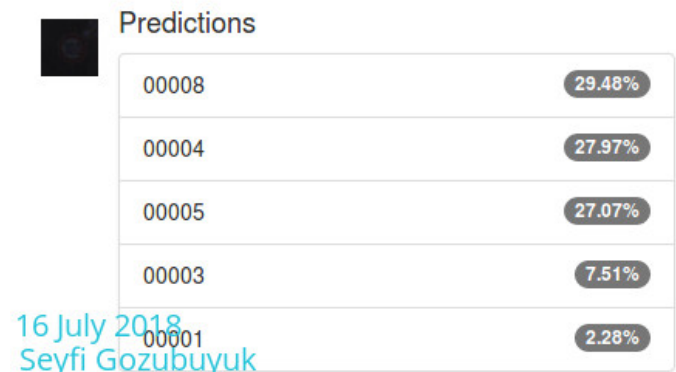


Fig. 11. Traffic Sign Test 1



Fig. 12. Traffic Sign Test 2



Predictions	Accuracy
00003	63.46%
00020	19.61%
00005	12.4%
00025	3.1%
00032	0.44%

- [5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, "Going deeper with convolutions," 2014.

Fig. 13. Traffic Sign Test 3

datasets for autonomous driving have larger than 10 GB data sizes. The Traffic Sign Classification is an old problem, and there are already excellent results. In addition, the size of the images in the dataset made use of AlexNet or GoogleNet irrational. Besides, the LeNet already provide satisfactory results.

6 CONCLUSION / FUTURE WORK

The results of the first task are open for improvement. A deeper network or optimized hyperparameters may provide better accuracy. Due to time limitations, this remains as future work.

For the second part of the project, it is required to access an environment which has more than 10 GB disk space and a GPU with more than 4 GB memory. As a future work, deeper networks will be trained for more complex datasets such as vehicle and pedestrian detection.

Testing the trained networks on a Jetson TX2 is another waiting task.

6.1 Hardware Deployment

The first part of the project was deployed on Udacity Workspace. Udacity Workspace provides a powerful GPU, which offers fast training and enough memory to train large deep networks. The second part of the project was deployed in the local installation of Nvidia Digits. The machine has an I7 CPU, 8GB RAM, and 650M GPU. The configuration was able to train LeNet, and with more powerful CPU and GPU, it will be possible to increase the complexity of the network.

REFERENCES

- [1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IEEE International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.
- [2] CNN, "Convolutional neural networks — Wikipedia, the free encyclopedia," 2018. [Online; accessed 16-July-2018].
- [3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," 1998.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.