Robotic Inference

Saminda Abeyruwan

Abstract—Object detection and classification is an integral part of modern-day robotic applications. In this project, we have leveraged NVIDIA's DIGITS workflow to develop image classification networks on two datasets. We have discussed on the hyperparameter tuning and selection of a deep convolutional neural network architecture. We have analyzed the performance of the models, and finalize the project with our findings and future work.

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

1 Introduction

Object detection and classification is an integral part of robotic applications. In order to develop a successful inference workflow, first, we collect a labeled dataset. The dataset contains the inputs, $\mathcal{X}$, and the output, $\mathcal{Y}$ mappings, e.g., an input can be an image and the output can be the category of interest. Second, we learn a function, $h : \mathcal{X} \mapsto \mathcal{Y}$, where $h(\mathcal{X})$ is a good predictor for the corresponding target. This mapping is known as *supervised learning*. When the target takes a finite number of discrete values, it is known as a classification problem or softmax regression [1].

We have used two datasets in the report: 1) Udacity dataset, that contains labeled images of candy boxes, bottles, and nothing (empty conveyor belt) for the purpose of real time sorting, and 2) a set of selected categories from the Amazon Picking Challenge [2]. In practice, we can only approximate the function, h, therefore, we have used GoogLeNet, a deep convolution neural network, as the function approximator [3]. We have used the implementation available in NVIDIA DIGITS [4] for all our experiments.

2 Background / Formulation

Since 2014, the quality of the high performing convolutional neural networks has improved significantly by utilizing deeper and wider networks [5]. It has been shown that the Inception architecture of GoogLeNet [3] has perform well under strict constraints on memory and computational budget. GoogLeNet yield similar performance to VGGNet, and higher performance than AlexNet [5]. Therefore, we have chosen GoogLeNet for all our experiments.

We have used *Adam* optimizer [6] and the learning rates have been selected from 0.01, 0.001, and 0.001. We have used epochs 3, 5, and 8. For the first dataset, we have used $75 - 25$ train-validation split, while, on the second dataset the splitting has been set to $80 - 20$. We have used image normalizing and $256 \times 256 \times 3$ image squashing, and set all other hyperparameters as it is in the DIGITS workflows.

3 Data Acquisition

We have selected the Amazon Picking Challenge dataset[1] for the second task. The dataset contains 27 categories, and we have selected the three categories: 1) rollodex mesh collection jumbo pencil cup, 2) kong duck dog toy, and 3) cheezit big original. Fig. 1 shows a sample of the selected categories from the Amazon Picking Challenge.

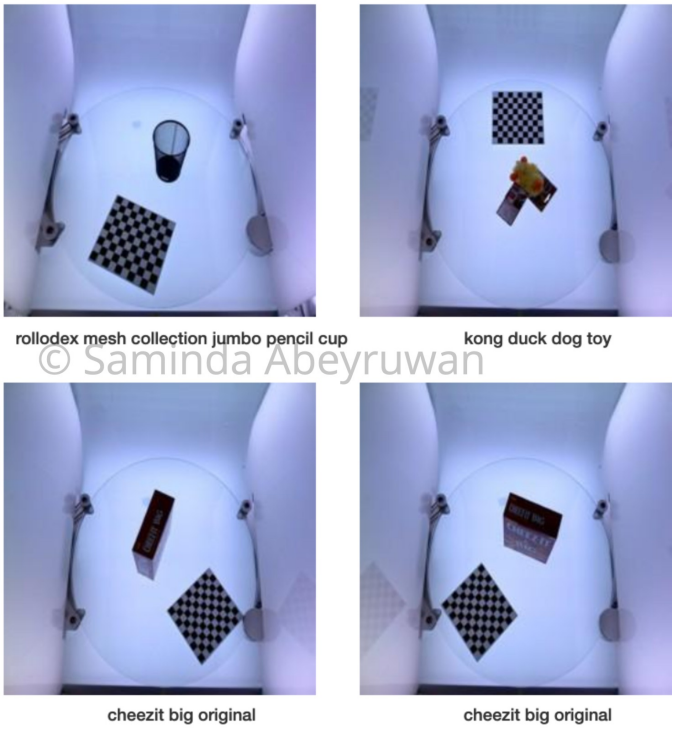

Fig. 1. A sample of the selected categories from the Amazon Picking Challenge.

We have used the raw RGB images from the Primesense sensors. All images are of size $1280 \times 1024 \times 3$, and there are 600 images from each category. In addition, the raw RGB-D data contain raw depth maps, segmentation masks for the RGB images, turntable pose information for each image, and calibration information for each RGB-D sensor. For the task of our project, we have selected only the raw RGB images. When we setup the DIGIT workflow, all the images have been squashed to $256 \times 256 \times 3$. This is a challenging dataset and also addresses one of the open problem in robotic arm manipulation. For other usages of the dataset, the reader is

1. http://rll.berkeley.edu/amazon_picking_challenge/

referred to [7].

4 Results

We have used GoogLeNet with Adam optimizer for all our experiments. The best results for Udacity dataset has been obtained with the learning rate 0.001 for 5 epochs. The model has achieved 100% validation accuracy around 3 epochs (Fig. 2). On the evaluation step, the model has used approximately 5 ms for inference, and obtained a accuracy of 75.4% (Fig. 3).

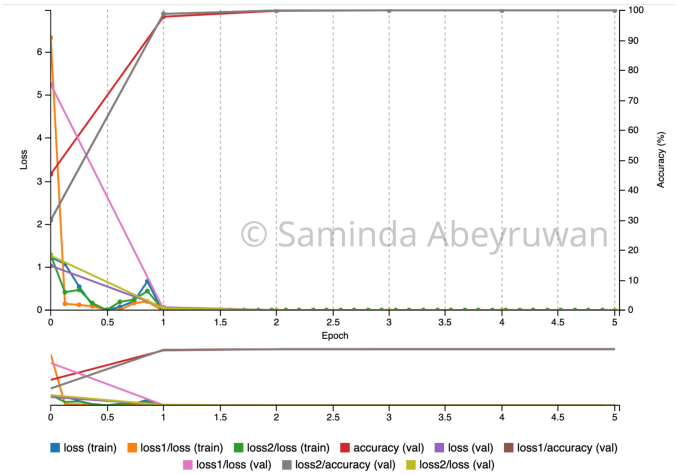

Fig. 2. GoogLeNet performance on Udacity dataset.

```
root@e9ee6f7f6b67:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20190303-011357-5834

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20190303-011357-5834/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20190303-011357-5834/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 5.54562 ms.
Average over 10 runs is 5.6082 ms.
Average over 10 runs is 4.90596 ms.
Average over 10 runs is 4.8683 ms.
Average over 10 runs is 4.86105 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14601  100 12285  100  2316    209     39  0:00:59  0:00:58  0:00:01  2167

Your model accuacy is 75.4098360656 %
```


Fig. 3. The inference results model trained from Udacity dataset.

The best results for the Amazon Picking Challenge has been obtained the learning rate 0.001, and the model has converge to 100% evaluation about 3 epochs. GoogLeNet has been able to isolate the object of interest, and has been able to provide good results for the challenging dataset (Fig. 4).

We have randomly selected a image from each category and observed the inference results. Figs. 5, 6, and 7 show the prediction probabilities for 1) rollodex mesh collection jumbo pencil cup, 2) kong duck dog toy, and 3) cheezit big original respectively.

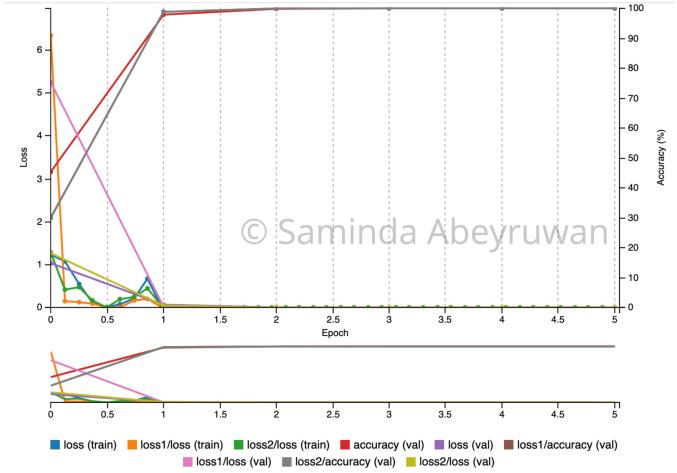

Fig. 4. GoogLeNet performance on the Amazon Picking Challenge dataset.

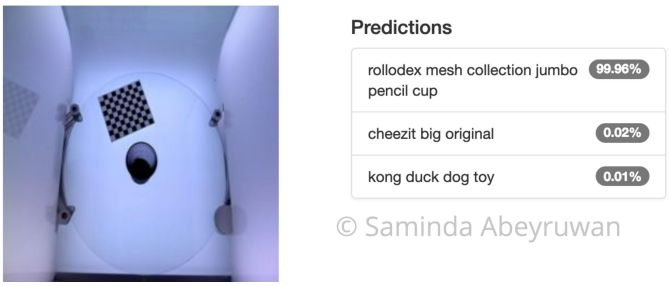

Fig. 5. The predictions for a random instance of the rollodex mesh collection jumbo pencil cup.

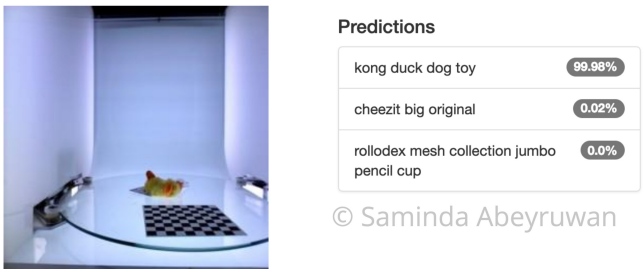

Fig. 6. The predictions for a random instance of the kong duck dog toy.

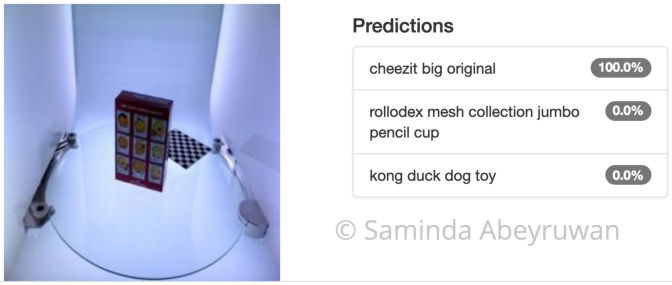

Fig. 7. The predictions for a random instance of the cheezit big original.

5 Discussion

GoogLeNet has obtained 100% accuracy on the validation set for the selected categories of the Amazon Picking Challenge dataset. This is a welcoming sign as the images of interest are occupy relatively smaller area compared the whole image (Fig. 1). Based on the findings, we could assume that 600 images per category is enough for the setup inference problem. The Amazon Picking Challenge dataset contains 27 categories, and it is interesting to observe the performance using all categories.

We have decided to use GoogLeNet because of its performance under strict constraints on memory and computational budget. VGGNet yield similar results to GoogLeNet, but, evaluating the network requires a lot of computation [5]. The next generation of GoogLeNet, Inception-v3, has shown state-of-the-art performance on the ILSVRC 2012 classification benchmark [5].

6 Conclusion / Future work

In this project, we have trained a deep convolutional neural network GoogLeNet on two datasets using DIGITS workflow. We have shown that the model has achieved over 75% accuracy on Udacity dataset. The model has performed well on the Amazon Picking Challenge dataset, and achieved 100% on the validation set.

DIGITS workflow provides a set of convenient tools to quickly prototype a deep model, and to promote such models to production use or deploy on a Jetson TX2 board. In addition, workflows can be set of other tasks, such as segmentation. The Amazon Picking Challenge dataset contains training examples for such tasks, and very well can be used develop robotic applications.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[2] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 509–516, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[4] B. J. Erickson, P. Korfiatis, Z. Akkus, T. Kline, and K. Philbrick, "Toolkits and libraries for deep learning," *Journal of Digital Imaging*, vol. 30, pp. 400–405, Aug 2017.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*, 2016.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization.," *CoRR*, vol. abs/1412.6980, 2014.

[7] K. S. Narayan, J. Sha, A. Singh, and P. Abbeel, "Range sensor and silhouette fusion for high-quality 3d scanning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3617–3624, May 2015.