Robotic Inference using NVIDIA Digits

Pritesh Gudge

Abstract—The Inference project is a part of the Robotics Nanodegree. The objective was to understand and implement a prototype of vision system for a robotics application. Two implementations are provided for evaluation. First, the provided dataset of bottles, boxes and other images is used to train a Google Lenet [1] based network and the results are used to classify relevant images provided in the test suite. An average run time for evaluation is approximately 5 milliseconds and the model accuracy of the model is 75.40% for the given test set. Second, data collection, data conditioning and Google Lenet based model training is used to train a network for food classification.

Index Terms—Robot, Inference, IEEEtran, Udacity, Deep Learning.

1 Introduction

The progress in the field of robotics is accelarating every day. Robots are being used in many applications like manufacturing, search and rescue, exploration etc. Many of the applications require identifying and detecting objects in multiple scenes and backgrounds. Deep learning is being leveraged into robotic vision and inference as it provides a framework for robots to be configured to operate in multiple environments both natural and man-made. Convolutional and reinforcement learning based techniques have been most effective in robotics applications. Previous approaches to robotic inference were based line follower and other approaches for structured operating environments [2]. Deep learning based approaches provide the possibility to provide identification and detection for robotic inference, given that adequate data is available for training the neural networks.

In the first part, the provided images of boxes and bottles are used to train an Google Lenet based network for classification. In the second part, a model is trained to classify between fruits and other foods. One can build a diet-tracking or junk food alert robotic system using such a classification model, possibly built into a refrigerator. It generates an alert if unhealthy food is taken out or stored in the refrigerator. A similar model can be used in a robot which sorts items purchased while grocery shopping. Eg. Bananas should not put into the refrigerator, while other fruits can be.

2 Background / Formulation

The **NVIDIA Digits** environment - shell and GUI was used for structuring the databases and training, validation of the models.

2.1 Bottles and Boxes

For the first part of the project, the 10094 images provided were color RGB images with size 256x256. For simplicity purposes, the networks provided in the digits workspace were used. Two pre-defined networks are available in the Digits workspace which take 256x256 RGB images as input. The training was done with 5 epochs first with Google LeNet with Adam Optimizer. Google LeNet was chosen as it offers state of the art results for ImageNet dataset and as it offers comparatively higher quality in classification results [1] at the cost of slightly higher compuational requirements . GPUs were available for training hence quality was chosen over lower computational requirements in the tradeoff. Adam Optimizer was chosen as it has been established to be more efficient compared to other optimizers [3]. The model with Adam Optimizer accuracy failed to go beyond 50% Refer image: 1

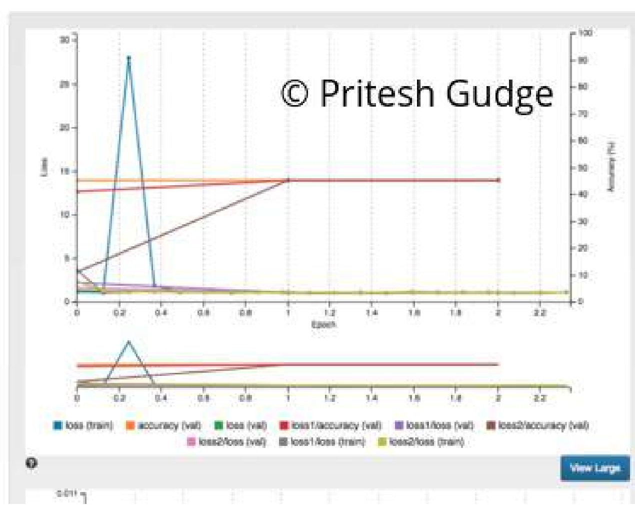

Fig. 1. Google Lenet Adam Optimization Bottless and Boxes: Training Progress Chart

In the next step the same Google LeNet network was used with standard SGD optimizer with 5 epochs and 0.01 learning rate at the start. 25% of the dataset was used for validation. The test dataset was stored separately before training. Once the model was found to be converging, the final model was set to train with an initial learning rate of 0.01 and 10 epochs of the network 22-layer Google LeNet. [1]. The test images were used to determine performance of the trained model. Speed of each individual classification run was also evaluated. Once trained the Google LeNet network model provides quick classification on GPU ¡10ms.

2.2 Food Classification

6020 images were collected with approximately equal number of images for each of the classes:

- Banana
- Guava
- Other foods(Marshmallows)

Google Lenet [1] model was chosen for its effectiveness to classificiation and detection on ImageNet dataset as described above section 2.1. The hyperparameters: 10 epochs and a starting learning rate of 0.01 was chosen for training. 25% of the dataset was used for validation. The test dataset was stored separately before training. The test images were used to determine performance of the trained model. Once trained the Google LeNet network model provides quick classification results on GPU and hence was chosen for this project. Eg. The model can be deployed on a TX2 processor and classification results can be obtained real-time for the robotic platform in its operating state.

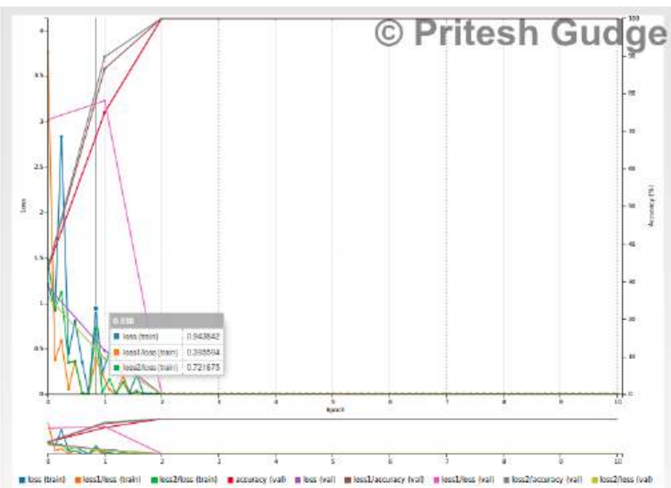

Fig. 2. Food Classification: Training Progress Chart

3 Data Acquisition

3.1 Bottles and Boxes

The dataset was acquired by using a TX2 kit to capture images of boxes and bottles off a conveyor belt. The captured images were provided as standard in the *Digits* workspace. Total of 10094 images were provide out of which 7570 images were used as training set and remaining 2524 were used as the validation set.

3.2 Food Classification

This dataset was collected by capturing images with a standard webcamera: *Logitech 270* in diffused sunlight. The camera was connected to a Lenovo laptop (Figure 3) using the USB port and images were captured by using a script [4]. The python language script used *OpenCV* python library **cv2** to render and transform the images. 2 varieties of guava and 1 variety of banana was taken for the fruits' classes of the dataset. These were the objects readily available while performing the data collection. Also, multi-colored marshmallows were available to describe the non-fruit(junk food) class. 6020 total images RGB images were captured, and resized to size 256x256 before storing into the appropriate classes Ref. Table 1.

Fig. 3. Food Classification: Data Collection Setup

TABLE 1
Fruit Images

 Guava	 Banana

4 Results

4.1 Bottles and Boxes

The accuracy of Google Lenet Trained model with SGD optimizer was nearly 100%. The learning stage loss and accuracy graphs are shown below Refer Figure: 4

The results from the *evaluate* command are shown in the figure 5. The average time for the evaluation runs is around 5 milliseconds which is within the provided constraint of it being below 10 milliseconds. The model accuracy is 75.40% which is above the requirement minimum 75% required from the implementation.

A sample classification run of one of the bottle images is shown below in figure 6. The accuracy for prediction is 99.87%

4.2 Food Classification

The classification performance from the food classification model for the included test images is satisfactory. The classification confidence is above 95% for the images from the

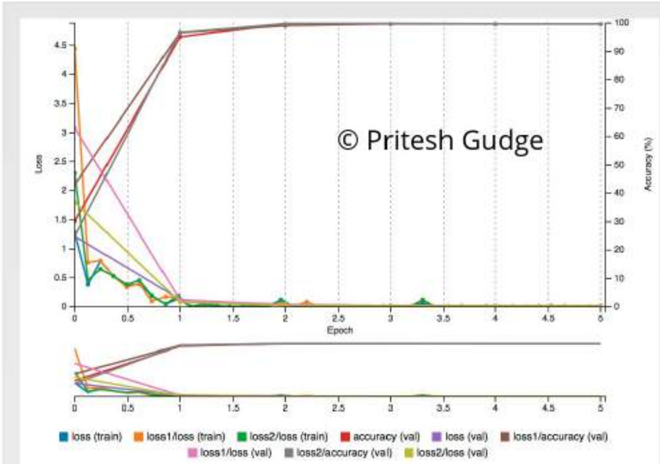

Fig. 4. Boxes and Bottles: Learning Chart

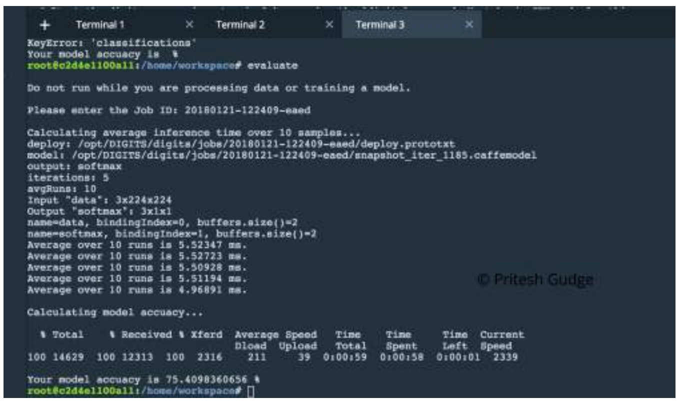

Fig. 5. Boxes and Bottles: "evaluate" Command Result

test dataset. The model appears to be overfitting according the the training chart Fig:2 and the resulting test image classification (Figures: 7 & 8).

5 Discussion

For the first part of the project, the classification confidence is above 75% is satisfactory and meets the given requirements. The avearge evaluation time is also approximately 5 milliseconds. This meets the given requirement for evaluation time to be less than 10 milliseconds. The Google

Fig. 6. Boxes and Bottles: Sample Bottle Evaluation Result

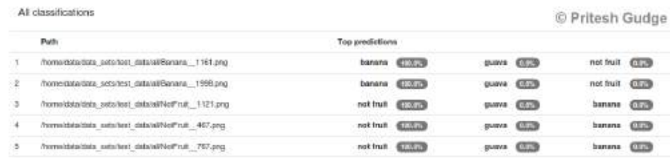

Fig. 7. Food Classification: Classify Many Result Show Overfitting

Fig. 8. Classify Guava: Show Overfitting

LeNet network is optimized for computation on GPU due to parallelization and high scale performance optimization [1].

In the second part of the project, the model is overfitting the training data. Accuracy is near 99% for the training model. Lower number of epochs, probably 3 or 4 epochs may help in preventing overfitting. More data needs to be collected to include different varieties of the fruits and junk food categories. The current model will be performant in a very narrow range of inputs.

6 Conclusion / Future work

The first part of the project achieved the expected result. The evaluation performance was above 75% and the evaluation time less than 10 milliseconds. This falls within the parameters of rubric requirements for the project. This model can be used in the future to build a sorting robot, which will sort objects into various bins. This can be useful for a robot which sorts trash for recycling. Extensive training with a huge variety of data will be required for this inference model to be viable for a commercial grade application.

For the second part, the work documented here achieves the objective of data collection, data conditioning, structuring and training of a model. The results from the evaluation for the limited test set are also satisfactory. This model can be used in a robotic application which will be mounted on a refrigerator/cupboard. When placing store bought items into the refrigerator/cupboard, the camera will capture the item and open the door only if it is healthy. Otherwise it will prompt the user to return the item. Another application can be junk food alert warning. An app can be used to record images of food consumed through the day. If junk food is consumed, the same will be flagged and an alert message will be sent to the user. For both the above applications to become commercially viable, the model will have to be trained on huge and varied dataset which captures all the commonly available processed and unprocessed foodstuffs.

REFERENCES

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.
[2] A. Harom, "Line follower robot," *Universiti Malaysia Perlis*, 2008.
[3] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[4] P. Gudge, "Capture images from webcam for machine learning traning dataset," 2018.