Robotic Inference

Quah Yu Soon

Abstract—This documentation discusses about two different deep neural network models. The first network model was designed to classify correctly the images which might contain bottle or candy box which was placed on conveyor belt. Another deep neural network model based on student own idea which is vehicle classification will also be discussed. GoogLeNet was used in both network models and validation accuracy for first and second network models were 99.92 percent and 71.2 percent respectively. More results will be covered under Result session.

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

1 Introduction

DEEP LEARNING is a state-of-the-art artificial intelligence technology used in image classification,in object detection as well as in segmentation. This technology is about designing appropriate deep neural network and then train it with large amount of relevant image data. The first deep neural network model for a classification task will be trained using data set supplied by Udacity which contains images of candy boxes and bottles located on a conveyor belt. Requirement is to fulfil at least 75 percent accuracy and an inference time of less than 10 ms. The second network model will be solely trained with self-collected image data which allows students to demonstrate their skills in data collection. The chosen idea is to classify vehicles on the road such as bus, car, motorbikes, trucks and vans. Such classification could be useful in road traffic analysis.

2 Background / Formulation

There are three neural network options, LeNet, AlexNet or GoogLeNet. When training the first network model, LeNet was opted out as it supports grayscale images. An attempt with AlexNet with 8 epochs and Stochastic Gradient Descent as solver generated a result of slightly less than 75 percent accuracy which is obtained by running dedicated command on DIGITS server. Another attempt with GoogLeNet with similar hyperparameters could achieve above 75 percent accuracy. Hence GoogLeNet was selected to train with supplied data set. Since it seemed to be able to produce the best result, GoogLeNet was selected for vehicle classification task. Table 1 lists down important hyperparameters defined for training both network models.

TABLE 1: Hyperparameters for training network models

Hyperparameter	First Model	Second Model
Training epochs	8	100
Solver Type	Stochastic Gradient Descent	Stochastic Gradient Descent
Base Learning Rate	0.01	0.01
Learning Rate Policy	Step Down	Step Down
Learning Rate Step Size	33 %	33 %
Learning Rate Gamma	0.1	0.1

3 Data Acquisition

In supplied data set, there were 3 classes, 4568 images containing bottles, 2495 images containing candy boxes as well as 3031 images that do not have any objects on conveyor belt. All images were in a fixed size of 500x500 pixels. 25 percent of entire data set was set apart for validation set, the rest were for training set. Resize transformation 'Squash' was used to resize images to a dimension of 256x256 pixels as to comply with GoogLeNet's input dimension. Below are three images from supplied data set by Udacity.

(a) Candy box. (b) Bottle. (c) Nothing.

Fig. 1: Images from supplied data set.

As images of vehicles are needed for vehicle classification task, photos were taken using smart phone on an overhead bridge during daytime. A total of 130 photos were taken. From these photos, 31 images of buses, 183 images of cars, 41 images of motorbikes, 102 images of trucks and 63 images of vans were extracted through picture cropping. Cropped images were always in different sizes. This data set had also a validation set in same percentage. Resizing cropped images to a fixed dimension of 256x256 pixels was done through resize transformation 'Squash' as well.

Fig. 2: A photo of vehicles taken.

(a) Bus. (b) Car.

Fig. 3: Cropped images of bus and car respectively.

(a) Motorbike. (b) Truck. (c) Van.

Fig. 4: Cropped images of motorbike, truck and van respectively.

4 Results

As being mentioned in Background / Formulation section, the first neural network model have achieved an accuracy of 75.4098 percent and an average inference time of around 5.5 ms as illustrated in Figure 5. Figure 6 shows 99.92 percent of validation accuracy at 8th epochs which is pretty impressive.

Fig. 5: Accuracy and average inference time of first model.

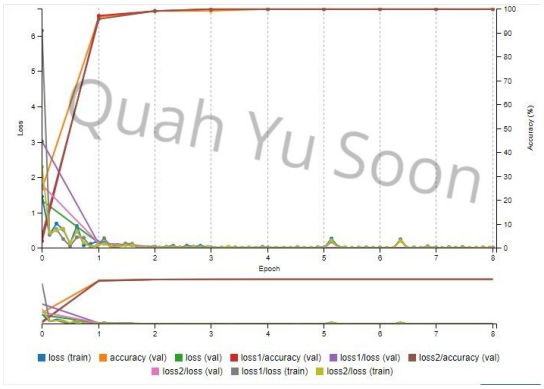

Fig. 6: Validation accuracy of first model.

As for the performance of second network model, validation accuracy achieved 71.2 percent as can be seen in Figure 7. The accuracy result does not look good, but two predictions on a truck and a car images respectively worked good enough. Something have to add is the images of truck and car being used for prediction were not included in training and validation data set. Hence it was good to see the model worked for new data.

Fig. 7: Validation accuracy of second model.

Fig. 8: Prediction on a truck image.

Fig. 9: Prediction on a car image.

5 Discussion

The first network model performed very well as it had a very high validation accuracy of 99.92 percent. One may believe the accuracy on the dataset is extremely good because amount of images prepared for training is a lot. Validation accuracy of second network model was only around 71 percent. This could be mainly due to insufficient amount of images. Inconsistent image size might also lead to lower accuracy rate. In a humble opinion, accuracy is more important than inference time as vehicle classification with high accuracy could produce correct road traffic analysis. Classify wrongly could produce misleading information and subsequently may lead to mistakes in decision making.

6 Conclusion / Future work

The first network model did fulfil the requirements as it achieved an accuracy of 75.4098 percent and an average inference time of around 5.5 ms. As for second network

model which is to be used in vehicle classification task, validation accuracy achieved 71.2 percent and there should be room for improvement on accuracy. Thus it was not ready yet to be a commercially viable product. For future works, collecting more data is a must-do step. Doing some researches on how to make data better for model training could be helpful as well.