Robotic Inference

Naruhiko Nakanishi

Abstract—Nvidia DIGITS utilizes transfer learning from existing pretrained models such as LeNet, AlexNet, and GoogLeNet. In this Robotic Inference project, a network on the supplied data is trained, and a network using the own collected data is also chosen and trained. Upon evaluation, the model for the supplied data meets the required accuracy of above 75 percent (around 75.4 percent), and required inference time of below 10 ms (around 4.6 ms). For the own dataset, the predictions of randomly selected images are mostly good except one of car images.

Index Terms—Robot, Inference, Nvidia DIGITS, DNN, deep learning.

1 Introduction

In 2012, AlexNet was the first DNN architecture to win the ImageNet classification challenge. Ever more complex and accurate DNNs have been developed on the ImageNet benchmark dataset ever since, including VGGNet, ResNet, Inception, GoogleNet, and their many variations. [1]

Nvidia DIGITS (the Deep Learning GPU Training System) is a web application for training deep learning models. It utilizes transfer learning from existing pretrained models such as LeNet, AlexNet, and GoogLeNet.

In this Robotic Inference project, with DIGITS workspace, a network on the supplied data is trained, and a network using my own collected data is also chosen and trained.

2 Background / Formulation

Using Digits, once the data is imported, a training model is chosen. The model has to achieve an inference time of 10 ms or less on the workspace and have an accuracy greater than 75 percent. A trained model is tested by running the command evaluate with the DIGITS server.

One of the created datasets can be selected that you want to train. The number of epochs can be changed as can the seed, batch size, solver, and learning rate.

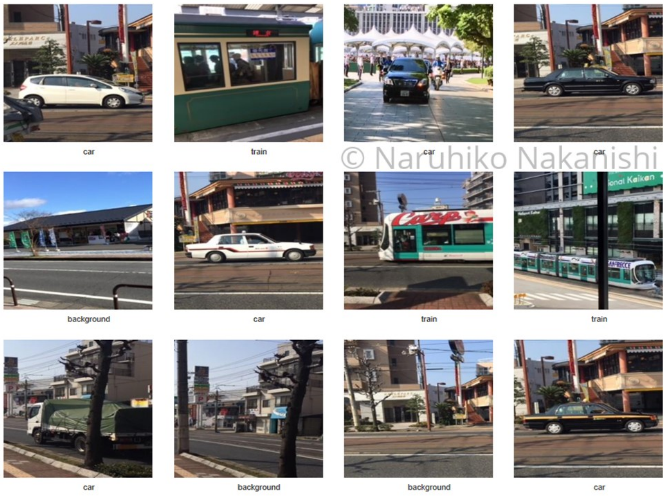
One of the award winning standard networks, LeNet, AlexNet, or GoogLeNet can be selected. In this project, AlexNet is chosen for the supplied data set and then GoogLeNet is chosen for a robotic inference project.

3 Data Acquisition

The supplied dataset are added into DIGITS and trained. In the dataset, there are photos taken from a Jetson mounted over a conveyor belt. The images of candy boxes, bottles, and nothing (empty conveyor belt) are trained for the purpose of real time sorting. This kind of design can be extrapolated to many things that require real time sorting.

For this robotic inference project, a phone is used to collect the dataset of images which are rectangular images at first. A system requires a constant input dimensionality. Therefore, the images are down-sampled to a fixed resolution of 256 x 256.

Fig.1 shows RGB type of the collected images of trains, cars, and background. In the dataset, there are 47 images of trains, 34 images of cars and 35 images of background.



© Naruhiko Nakanishi

Fig. 1. Collected images of trains, cars, and background

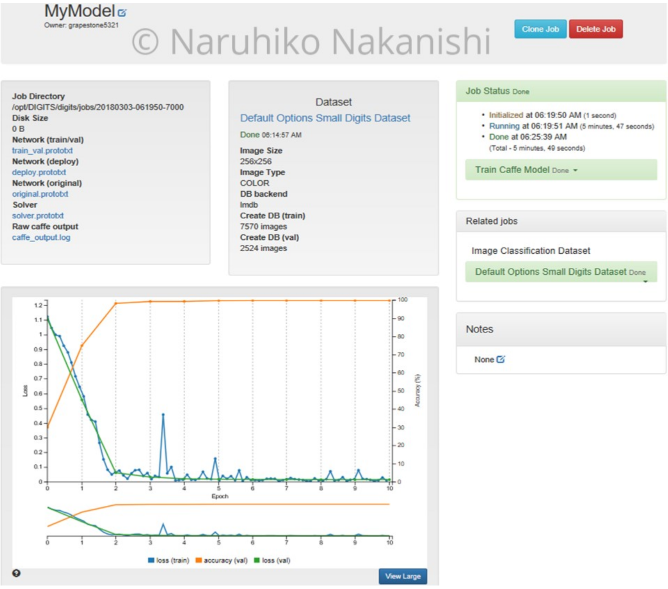
4 Results

Using the supplied data, a network is created and is trained. AlexNet is chosen for the dataset. Fig.2 shows the result of training of 10 epochs and a learning rate of 0.001 with SGD as solver type.

Fig.3 shows a snapshot of the terminal output after running the evaluate command. Upon evaluation, the model for the supplied data meets the required accuracy of above 75 percent (around 75.4 percent), and required inference time of below 10 ms (around 4.6 ms).

GoogLeNet is chosen for the own dataset. Fig.4 shows the result of training of 200 epochs and a learning rate of 0.001 with Adam as solver type.

Fig.5, Fig.6 and Fig.7 show the predictions of randomly selected images. The predictions of them are mostly good except the prediction of bottom car image in Fig.6 that is 20.79 percent for car and is 79.16 percent for train.



© Naruhiko Nakanishi

MyModel
Owner: grapestone5321

Job Directory
/opt/DIGITS/digits/jobs/20180303-061950-7000
Disk Size
0 B
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

Dataset
Default Options Small Digits Dataset
Done 06:14:57 AM
Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
7570 images
Create DB (val)
2524 images

Job Status Done

- Initialized at 06:19:50 AM (1 second)
- Running at 06:19:51 AM (5 minutes, 47 seconds)
- Done at 06:25:39 AM
  (Total - 5 minutes, 49 seconds)

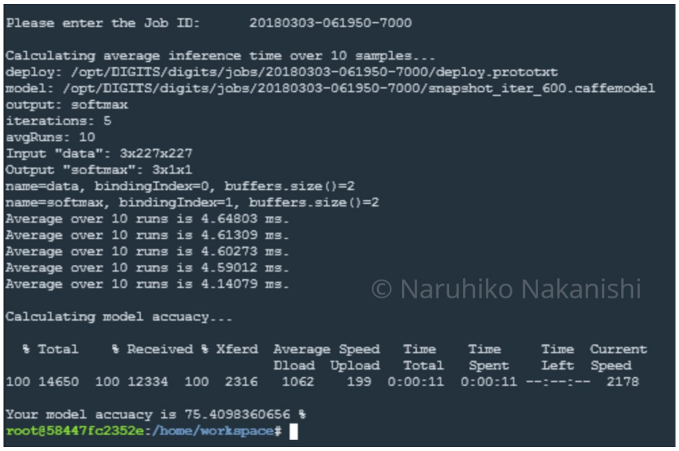
Train Caffe Model Done

Related jobs

Image Classification Dataset
Default Options Small Digits Dataset Done

Notes
None

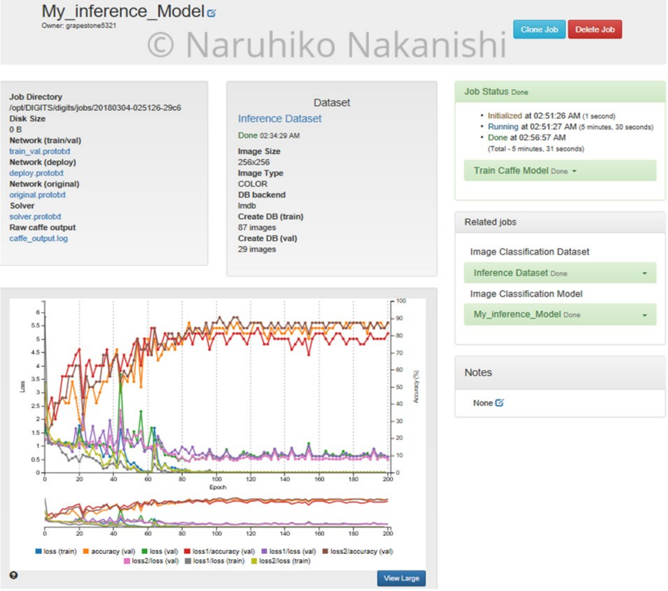
Fig. 2. Training for supplied data



```
Please enter the Job ID:      20180303-061950-7000

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180303-061950-7000/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180303-061950-7000/snapshot_iter_600.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x227x227
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 4.64803 ms.
Average over 10 runs is 4.61309 ms.
Average over 10 runs is 4.60273 ms.
Average over 10 runs is 4.59012 ms.
Average over 10 runs is 4.14079 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14650  100 12334  100  2316   1062    199  0:00:11  0:00:11 --:--:--  2178

Your model accuacy is 75.4098360656 %
root@58447fc2352e:/home/workspace#
```

© Naruhiko Nakanishi

Fig. 3. Evaluation



© Naruhiko Nakanishi

My_inference_Model
Owner: grapestone5321

Job Directory
/opt/DIGITS/digits/jobs/20180304-025126-29c6
Disk Size
0 B
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

Dataset
Inference Dataset
Done 02:34:29 AM
Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
87 images
Create DB (val)
29 images

Job Status Done

- Initialized at 02:51:26 AM (1 second)
- Running at 02:51:27 AM (5 minutes, 30 seconds)
- Done at 02:56:57 AM
  (Total - 5 minutes, 31 seconds)

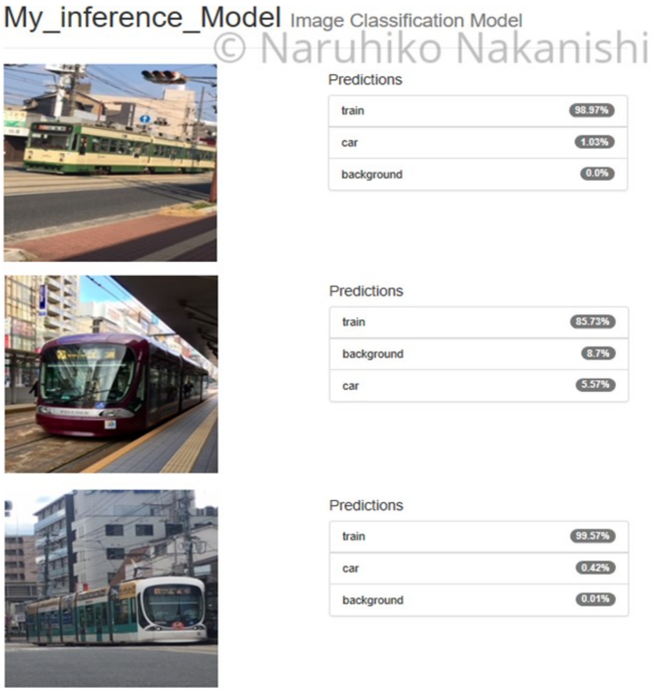
Train Caffe Model Done

Related jobs

Image Classification Dataset
Inference Dataset Done
Image Classification Model
My_inference_Model Done

Notes
None

Fig. 4. Training for the own data



My_inference_Model Image Classification Model

© Naruhiko Nakanishi

Predictions

train	98.97%
car	1.03%
background	0.0%

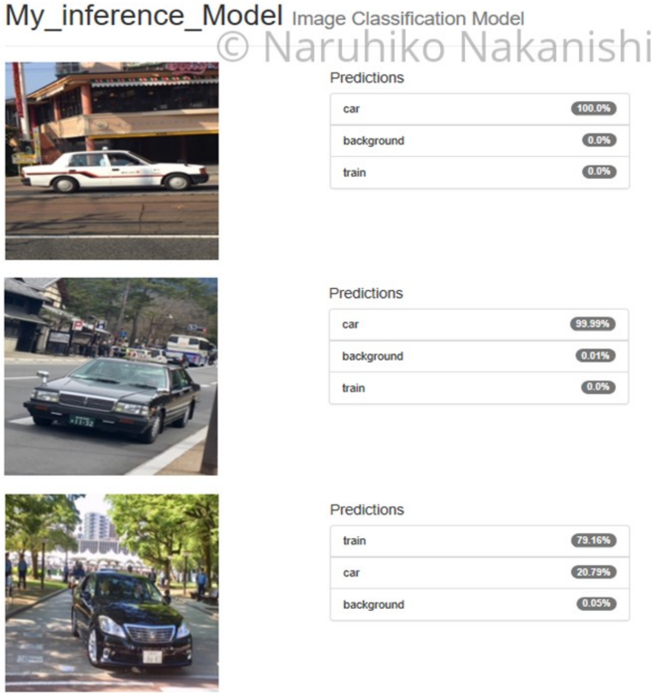
Predictions

train	85.73%
background	8.7%
car	5.57%

Predictions

train	99.57%
car	0.42%
background	0.01%

Fig. 5. train



My_inference_Model Image Classification Model

© Naruhiko Nakanishi

Predictions

car	100.0%
background	0.0%
train	0.0%

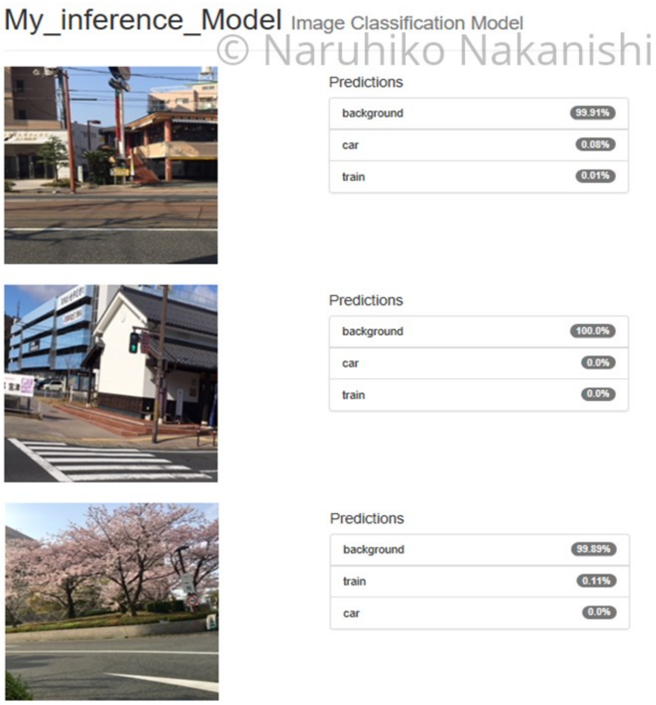
Predictions

car	99.99%
background	0.01%
train	0.0%

Predictions

train	79.16%
car	20.79%
background	0.05%

Fig. 6. car



My_inference_Model Image Classification Model

© Naruhiko Nakanishi

Predictions

background	99.91%
car	0.08%
train	0.01%

Predictions

background	100.0%
car	0.0%
train	0.0%

Predictions

background	99.89%
train	0.11%
car	0.0%

Fig. 7. background

5 Discussion

For the supplied data, AlexNet is chosen. After the training of 10 epochs and a learning rate of 0.001 with SGD as solver type, the model for the supplied data meets the required accuracy of above 75 percent (around 75.4 percent), and required inference time of below 10 ms (around 4.6 ms).

To meet the requirements, parameters should be optimized. For example, the accuracy become 0 percent in the case of a learning rate of 0.01.

For the own dataset, GoogLeNet is chosen. After the training of 200 epochs and a learning rate of 0.001 with Adam as solver type, the predictions of randomly selected images are mostly good.

However the prediction of bottom car image in Fig.6 is not sufficient. A possible reason is that the bottom car image in Fig.6 looks similar to train. The countermeasure for this problem is to increase the number of data.

Inference time is more important than accuracy in regards to the robotic inference project, because how long it takes to react to incoming data is always a consideration in real time systems. The industry is concerned with faster inference and many hardware and algorithm technologies are introduced for the inference acceleration.

6 Conclusion / Future work

Nvidia DIGITS utilizes transfer learning from existing pretrained models such as LeNet, AlexNet, and GoogLeNet. In this Robotic Inference project, with DIGITS workspace, a network on the supplied data is trained, and a network using my own collected data is also chosen and trained.

The supplied dataset are added into DIGITS and trained. The images of candy boxes, bottles, and nothing (empty conveyor belt) are trained for the purpose of real time sorting. Upon evaluation, the model for the supplied data meets the required accuracy of above 75 percent (around 75.4 percent), and required inference time of below 10 ms (around 4.6 ms).

For this robotic inference project, a phone is used to collect the dataset of RGB images of trains, cars, and background. The images are down-sampled to a fixed resolution of 256 x 256.

For the own dataset, the predictions of randomly selected images are mostly good. However the prediction of one of car images is not sufficient. A possible reason is that the car image looks similar to train.

In the own dataset, there are 47 images of trains, 34 images of cars and 35 images of background. The countermeasure for this problem is to increase the number of data.

The result isn't feasible commercially because the prediction of one of selected car images randomly is not sufficient. It is the problem of small quantity of data.

For future improvements, the own data can be generated with the existing data to overcome this problem of limited quantity of data. This methodology of generating the own data is known as data augmentation.

References

[1] Udacity, *Robotics Software Engineer Nanodegree Program Term2 Lesson7: Inference Application in Robotics*. Udacity, 2018.