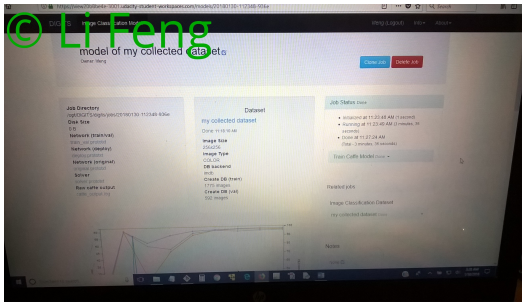
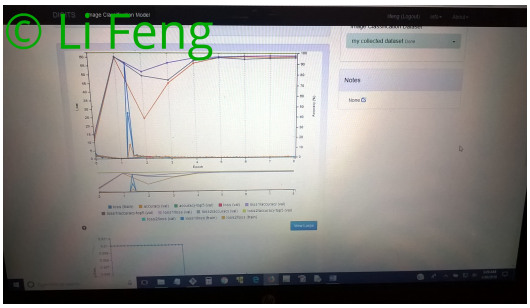
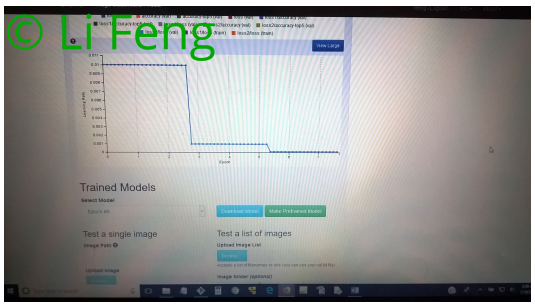
4 Results on Collected Data - Model - GoogLeNet

A (model of my collected dataset) was created using (my collected dataset) with 256x256 image size, GoogLeNet, and 8 epochs. Below are the images of the created model (model of my collected dataset).



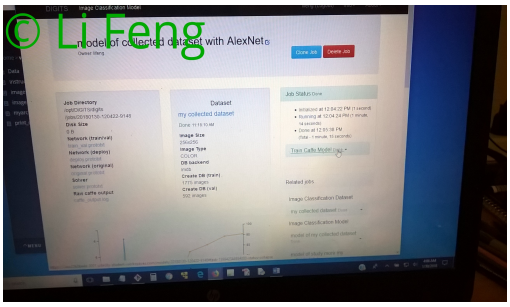


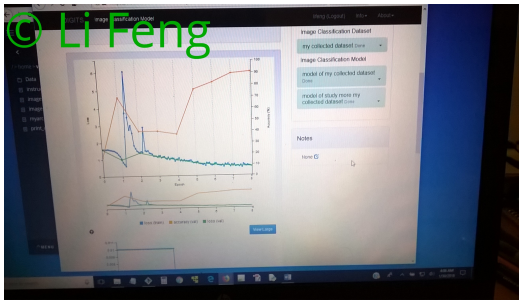


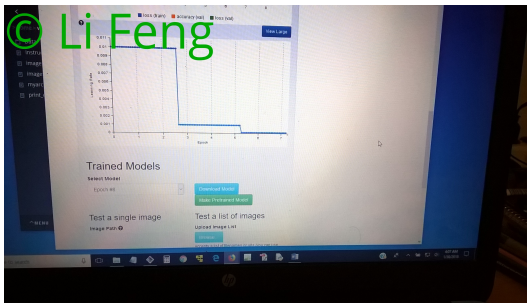
This trained model (i.e. model of my collected dataset) files are located at: ./collected_data/models/googlenet foler which contains: deploy.prototxt, info.json, labels.txt, mean.binaryproto, original.prototxt, snapshot_iter_448.caffemodel, solver.prototxt, and train_val.prototxt. (note: the size of snapshot_iter_448.caffemodel was reduced from 40330 KB to 0 for easy uploading of this project.)

4 Results on Collected Data - Model - AlexNet

A (model of collected dataset with AlexNet) was created using (my collected dataset) with 256x256 image size, AlexNet, and 8 epochs. Below are the images of the created model (model of collected dataset with AlexNet).





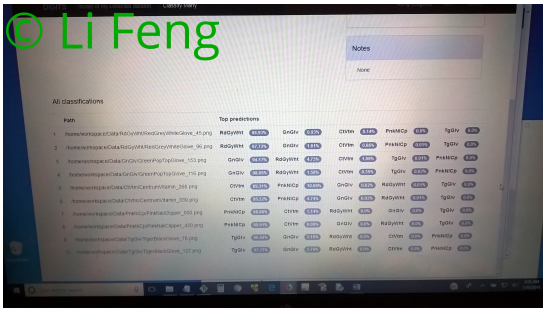


This trained model (i.e. model of collected dataset with AlexNet) files are located at: ./collected_data/models/alexnet foler which contains: deploy.prototxt, info.json, labels.txt, mean.binaryproto, original.prototxt, snapshot_iter_112.caffemodel, solver.prototxt, and train_val.prototxt. (note: the size of snapshot_iter_112.caffemodel was reduced from 222224 KB to 0 for easy uploading of this project.)

4 Results on Collected Data - GoogLeNet - Classification -1

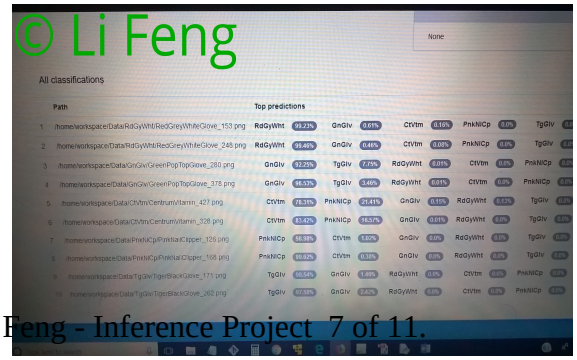
The (model of my collected dataset)
was tested with the (Classify Many Image)
on an image list file which contained
(./Data/RdGyWht/RedGreyWhiteGlove_45.png,
./Data/RdGyWht/RedGreyWhiteGlove_96.png,
./Data/GnGlv/GreenPopTopGlove_153.png,
./Data/GnGlv/GreenPopTopGlove_116.png,
./Data/CtVtm/CentrumVitamin_268.png,
./Data/CtVtm/CentrumVitamin_339.png,
./Data/PnkNlCp/PinkNailClipper_500.png,
./Data/PnkNlCp/PinkNailClipper_420.png,
./Data/TgGlv/TigerBlackGlove_78.png,
./Data/TgGlv/TigerBlackGlove_127.png).
As can be seen from resulting image below, the
probabilities of the correct predictions
were ranged from 99.91% to 89.31%.

(98.93% and 97.72% on RedGreyWhiteGlove #45 and #96.
94.17% and 98.05% on GreenPopTopGlove #153 and #116.
89.31% and 95.22% on CentrumVitamin #268 and #339.
98.86% and 99.91% on PinkNailClipper #500 and #420.
96.84% and 97.21% on TigerBlackGlove #78 and #127.)



The (model of my collected dataset) was tested again with
the (Classify Many Image) on a different image list file
which contained
(./Data/RdGyWht/RedGreyWhiteGlove_153.png,
./Data/RdGyWht/RedGreyWhiteGlove_248.png,
./Data/GnGlv/GreenPopTopGlove_280.png,
./Data/GnGlv/GreenPopTopGlove_378.png,
./Data/CtVtm/CentrumVitamin_427.png,
./Data/CtVtm/CentrumVitamin_328.png,
./Data/PnkNlCp/PinkNailClipper_126.png,
./Data/PnkNlCp/PinkNailClipper_168.png,
./Data/TgGlv/TigerBlackGlove_171.png,
./Data/TgGlv/TigerBlackGlove_262.png). As can be seen
from resulting image below, the probabilities of the
correct predictions were ranged from 99.62% to 78.31%.

(99.23% and 99.46% on RedGreyWhiteGlove #153 and #248.
92.25% and 96.53% on GreenPopTopGlove #280 and #378.
78.31% and 83.42% on CentrumVitamin #427 and #328.
98.98% and 99.62% on PinkNailClipper #126 and #168.
98.54% and 97.58% on TigerBlackGlove #171 and #262.)



4 Results on Collected Data - AlexNet - Classification -1

The (model of collected dataset with AlexNet)
was tested with the (Classify Many Image)
on an image list file which contained
(./Data/RdGyWht/RedGreyWhiteGlove_45.png,
./Data/RdGyWht/RedGreyWhiteGlove_96.png,
./Data/GnGlv/GreenPopTopGlove_153.png,
./Data/GnGlv/GreenPopTopGlove_116.png,
./Data/CtVtm/CentrumVitamin_268.png,
./Data/CtVtm/CentrumVitamin_339.png,
./Data/PnkNlCp/PinkNailClipper_500.png,
./Data/PnkNlCp/PinkNailClipper_420.png,
./Data/TgGlv/TigerBlackGlove_78.png,
./Data/TgGlv/TigerBlackGlove_127.png).
As can be seen from resulting image below, incorrect
predictions were made on CentrumVitamin #268 and
#339, and the probabilities of the correct predictions
were ranged from 100.00% to 36.28%.

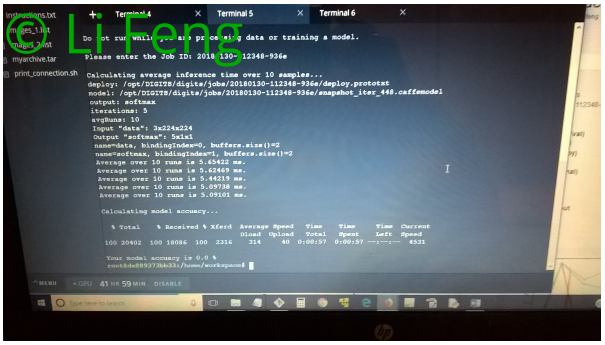
(100.00% and 100.0% on RedGreyWhiteGlove #45 and #96.
36.28% and 38.32% on GreenPopTopGlove #153 and #116.
CentrumVitamin #268 and #339 were predicted to PnkNlCp & RdGyWht.
95.26% and 99.07% on PinkNailClipper #500 and #420.
100.0% and 99.88% on TigerBlackGlove #78 and #127.)

The (model of collected dataset with AlexNet) was tested
again with the (Classify Many Image) on a different
image list file which contained
(./Data/RdGyWht/RedGreyWhiteGlove_153.png,
./Data/RdGyWht/RedGreyWhiteGlove_248.png,
./Data/GnGlv/GreenPopTopGlove_280.png,
./Data/GnGlv/GreenPopTopGlove_378.png,
./Data/CtVtm/CentrumVitamin_427.png,
./Data/CtVtm/CentrumVitamin_328.png,
./Data/PnkNlCp/PinkNailClipper_126.png,
./Data/PnkNlCp/PinkNailClipper_168.png,
./Data/TgGlv/TigerBlackGlove_171.png,
./Data/TgGlv/TigerBlackGlove_262.png). As can be seen
from resulting image below, incorrect predictions were
made on CentrumVitamin #427 and #328 and the
probabilities of the correct predictions
were ranged from 100.00% to 37.11%.

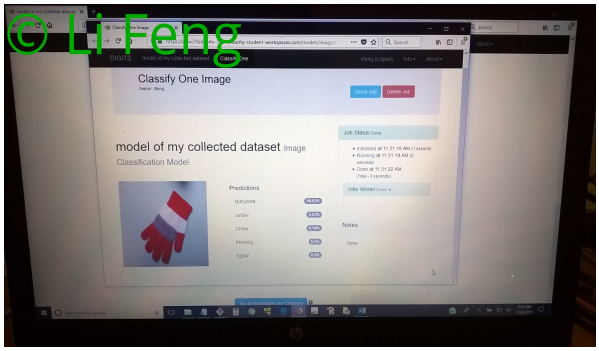
(100.00% and 100.00% on RedGreyWhiteGlove #153 and #248.
37.93% and 37.11% on GreenPopTopGlove #280 and #378.
CentrumVitamin #427 and #328 were predicted to PnkNlCp & RdGyWht.
81.16% and 90.17% on PinkNailClipper #126 and #168.
100.00% and 100.00% on TigerBlackGlove #171 and #262.)

4 Results on Collected Data - GoogLeNet - Classification - 2

The (model of my collected dataset) was tested with the (evaluate) command in a terminal window and the average inference time over 10 run were between 5.09101 ms - 5.65422 ms. The accuracy measurement was inappropriate in this case and can be ignored.

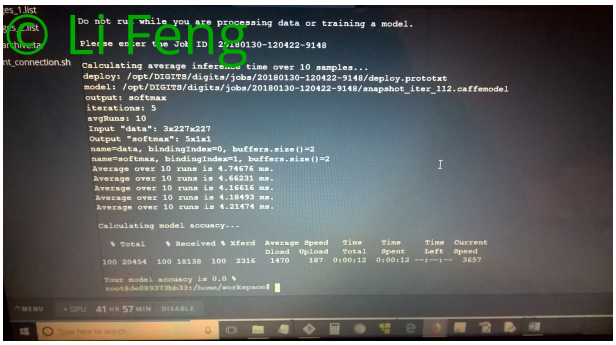


The (model of my collected dataset) was tested with the (Classify One Image) on a red-grey-white glove image and the predictions for RdGyWht was 98.93% as seen in the below image.



4 Results on Collected Data - AlexNet - Classification - 2

The (model of collected dataset with AlexNet) was tested with the (evaluate) command in a terminal window and the average inference time over 10 run were between 4.16616 ms - 4.74676 ms. The accuracy measurement was inappropriate in this case and can be ignored.



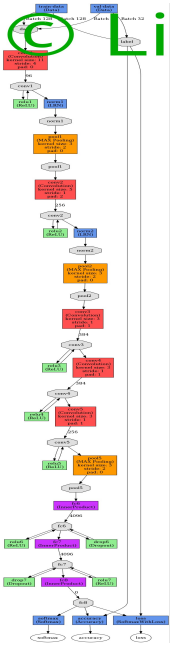
There was no photo taken on (model of collected dataset with AlexNet) tested with the (Classify One Image).

4 Results on Collected Data - Visulization of the GoogLeNet

The (model of my collected dataset) was visulized as following:

4 Results on Collected Data - Visulization of the AlexNet

The (model of collected dataset with AlexNet) was visulized as following:



4 Results on Collected Data - Inference on TX2 with GoogLeNet

The trained model (i.e. model of my collected dataset) using googlenet was (downloaded, tar-ed, and loaded) to Jetson TX2. "imagenet-camera" command was run and the inference on a "Tiger Black Glove" was 96.27% as seen in picture below. The model incorrectly inferred other items.

(referenced commands on TX2:
>NET=~/mygooglenet
>./imagenet-camera --prototxt=$NET/deploy.prototxt --model=$NET/snapshot_iter_448.caffemodel --labels=$NET/labels.txt --input_blob=data --output_blob=softmax
)

4 Results on Collected Data - Inference on TX2 with AlexNet

The trained model (i.e. model of collected dataset with AlexNet) using alexnet was (downloaded, tar-ed, and loaded) to Jetson TX2. "imagenet-camera" command was run and the inference on a "Tiger Black Glove" was 63.96% as seen in picture below. The model incorrectly inferred other items.

(referenced commands on TX2:
>NET=~/myalexnet
>./imagenet-camera --prototxt=$NET/deploy.prototxt --model=$NET/snapshot_iter_112.caffemodel --labels=$NET/labels.txt --input_blob=data --output_blob=softmax
)

5 Discussion

For the supplied data set, GoogLeNet and 500x500 image size were selected, and the trained model satisfied the required inference time of 10 ms or less and accuracy > 75% in the DIGITS workspace. One possible improvement here is to resize images to 256x256 image size to match the intake of image size by GoogLeNet.

For the collected data set, the model built with GoogLeNet performed consistently better in terms of accuracies over the model built with AlexNet. But it used extra 1 ms on the average inference time over the model built with AlexNet.

The model built with GoogLeNet correctly predicted 21 images (i.e. 2 image lists with 10 images each and one image) in the DIGITS workspace. (18 of them were above 90%, 2 were above 80%, and 1 was 78.31%.) The average inference time were between 5.09101 ms - 5.65422 ms. It predicted one item (tiger black glove) correctly with 96.27% accuracy in the TX2 Jetson environment.

The model built with AlexNet correctly predicted 16 images out of 20 images (i.e. 2 image lists with 10 images each) in the DIGITS workspace. (7 of them were 100%, 4 of them were above 90%, 1 was above 80%, 4 of them above 30%, and 4 were incorrectly predicted.) The average inference time were between 4.16616 ms - 4.74676 ms. It predicted one item (tiger black glove) correctly with 63.96% accuracy in the TX2 Jetson environment.

For a soft robot serves motion impaired person, the accuracy definitely outweighs the inference time.

The DIGITS Workspace and it's UI interfaces made the modeling process much easier. And the Inference on the Jetson and dusty-nv/jetson-inference and various commands provided, gave the author an interesting experience.

6 Conclusion / Future Work

The results on the supplied data set satisfied the requirements.

The results on the collected data set using GoogLeNet in the DIGITS workspace were great. However, when run on the TX2 Jetson, it only predicted one item (out of five items) correctly.

The model built with AlexNet neural network performed worse (in general) than the one with GoogLeNet.

Further work can be done to collect more images, augment images, train more, play with parameters, and update neural network layers and values. (GoogLeNet has many layers and AlexNet has many parameters.)

This project achieved what the author attempted, but it is not a commercially viable product yet.

References

[1] Krizhevsky, Sutskever, and Hinton. ImageNet Classification with Deep Convolutional Neural Networks

[2] Canziani, Culurciello, and Paszke. An analysis of deep neural network models for practical applications

[3] Lana Lazebnik. Convolutional Neural Network Architectures: from LeNet ResNet. http://slazebni.cs.illinois.edu/spring17/lec01_cnn_architectures.pdf

[4] Siddharth Das. CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more .... https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

[5] Li, Johnson, and Yeung. Lecture 9: CNNs Architectures. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf

[6] Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, and Rabinovich. Going Deeper with Convolutions, https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf

[7] Adit Deshpande. The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3). https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html