Robotic Inference Project

Mohamed Sayed Antar
Robotics Software Engineer
mohamedsayed99.ms@gmail.com

March 15, 2019

ABSTRACT

working on a supplied data from a camera fixed above a conveyor belt and a collected data using a mobile camera, first using the supplied data with Nvidia DIGITS workflow to train a model by tuning the hyperparameters and choosing the best network to achieve a specific inference time and accuracy, second using the collected data for an inference idea and training a model using Nvidia DIGITS to deploy this model on an embedded system like Jetson TX2 board.

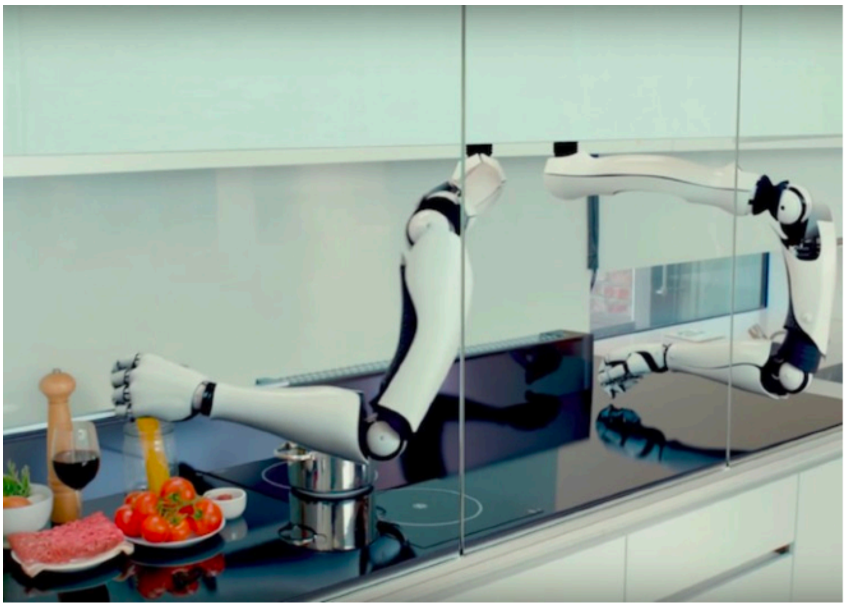
Keywords Deep Learning . Neural Networks . Object recognition . Nvidia DIGITS.

1 Introduction

the robotic kitchen is now an interesting part of robotics which full of challenges, due to the routine of life nowadays we are very busy because of the tight schedules, so it's difficult to take care of our food, people tend to approach fast-food instead of preparing healthy meals at home which leads to severe diseases.

nowadays there are some cooking robots which enable us to cook the food.

they are fully automated robots, so they have to capture the world around them using several kinds of sensors and cameras. in turn classifying images around the robot and determining each kitchenware is a core element of the robot perception process, using Nvidia DIGITS workflow with a collected photos of spoons or forks or even no thing to train a network for the classification processes.



2 Background / Formulation

several types of DNN’s have been developed on the ImageNet benchmark dataset like AlexNet, VGGNet, ResNet, Inception, GoogleNet and their many variations.

The increased accuracy is the result of breakthroughs in design and optimization, but comes at a cost when computation resources are considered.

The following table provides a sampling of the results (values are approximated from graphs in the paper), including a derived metric called information density. The information density is a measure of the efficiency of the network, or how much accuracy is provided for every one million parameters that the network requires.

DNN	Top1 Accuracy	Operations	Parameter	Inference	Power	Memory	Info Density
Network name	%	G-Ops	M	fps	W	MB	%accuracy/ Mparams
AlexNet-BN	57	2	60	50	10.9	310	1.0
GoogleNet	68	3	7	33	10.7	200	9.7
VGG-16	71	31	135	6	11.8	850	0.5
ResNet-101	76	16	45	10	12.9	300	1.7
Inception-v3	78	12	25	12	11.6	200	3.1

Note that only the results based on a batch size of one are included. In most cases, the batch size provides a speedup in inference time but maintains the same relative performance among architectures. However, an exception is AlexNet, which sees a 3x speedup when going from 1 to 64 images per batch due to weak optimization of its fully connected layer.

for the supplied data collected using Jetson TX2 camera above a conveyor belt AlexNet gave a good results in this case however the other DNN’s may give more accurate results.

for the collected data, using about 350 image per class GoogleNet gave more accurate results than AlexNet using 0.001 learning rate.

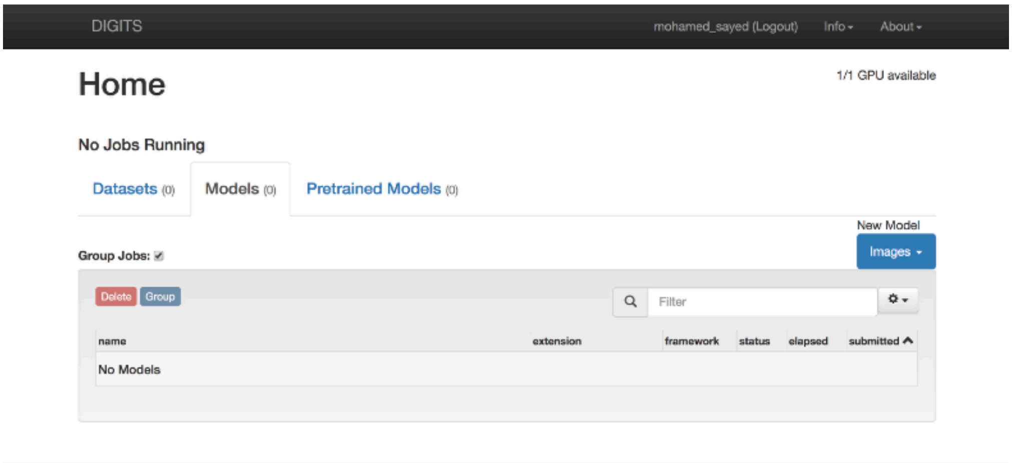
3 Data Acquisition

<u>for the supplied data</u> there are 7570 images for the following 3 classes:

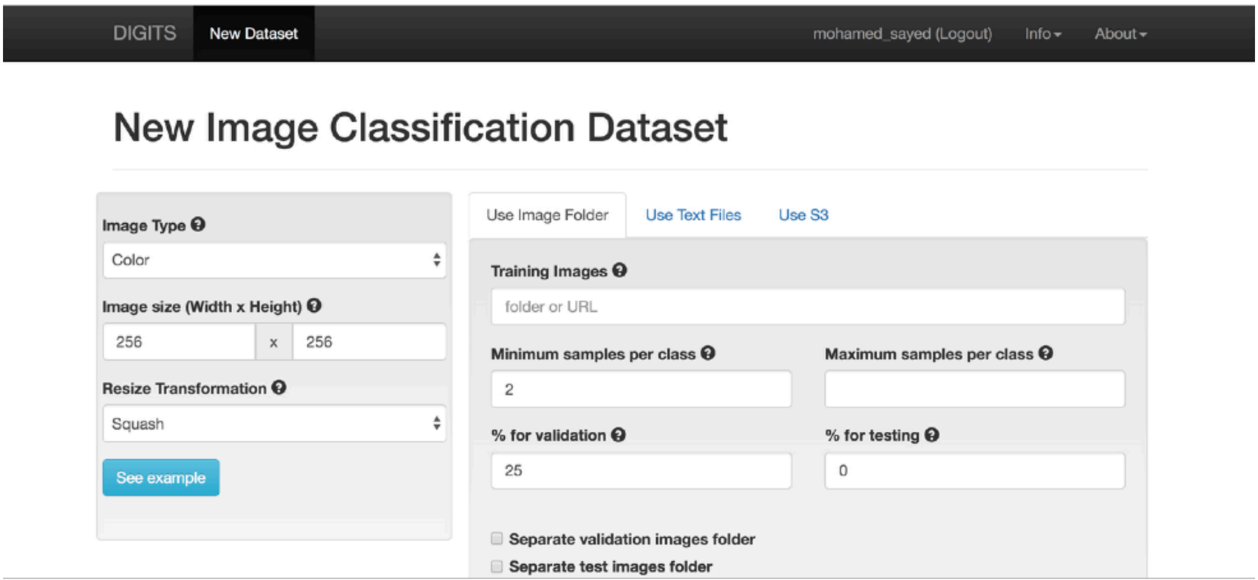
1. Bottle
2. Candy Box
3. Nothing

the data looks like this photo —>

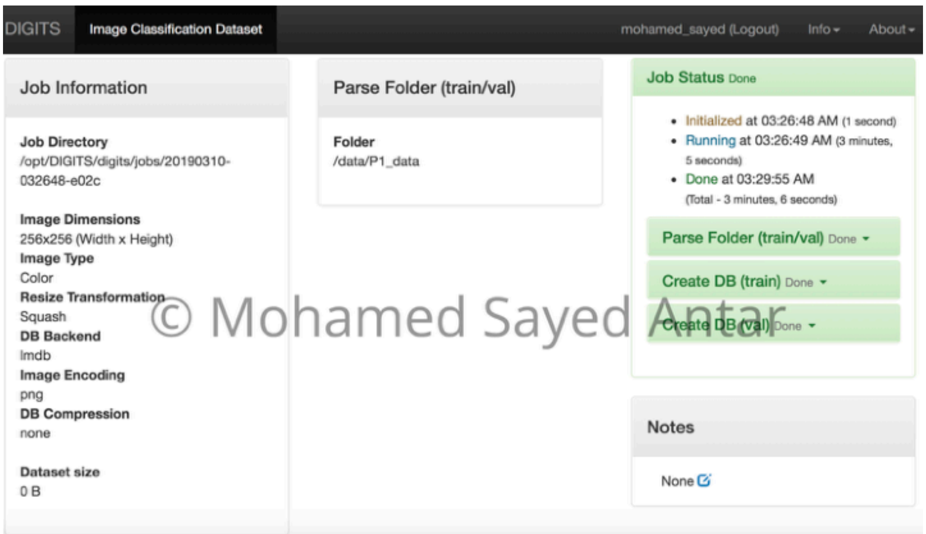
first by opening the DIGITS workspace we should see something like this



- by choosing `images` then `classification` we will see something like so

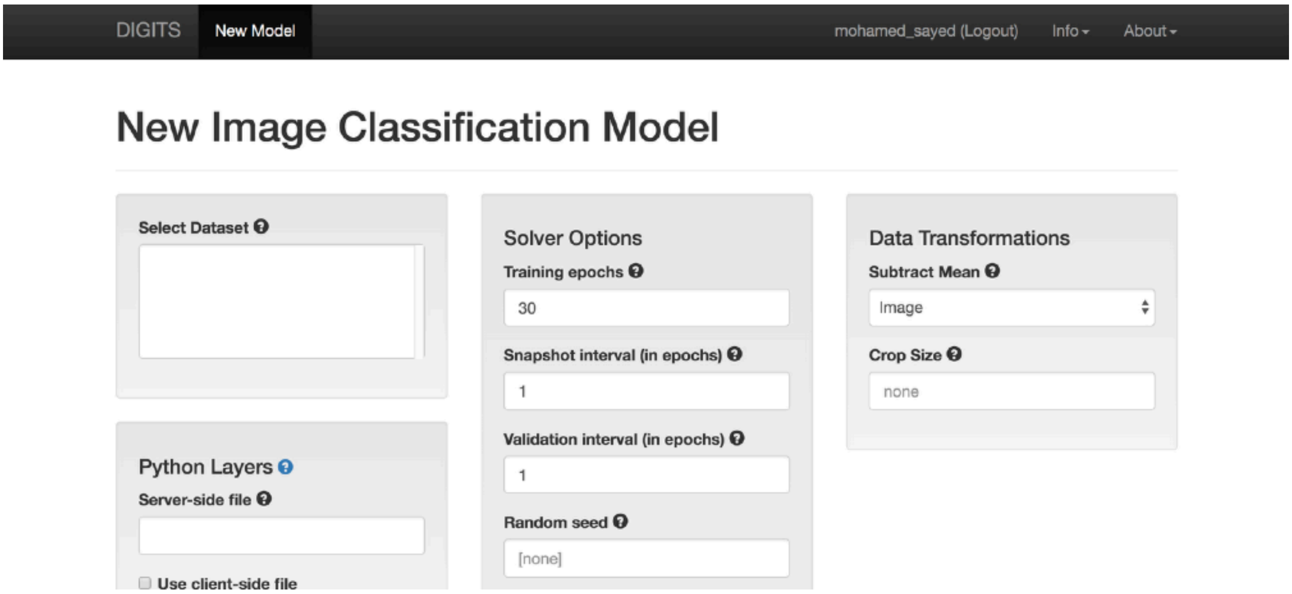


- adding our dataset url in this case using udacity workspace it will be `/data/P1_data`
- more setting can be done here like Minimum samples per class or maximum samples per class or even the percentage of validation photos and testing photos
- then by giving the dataset a name and choosing create :

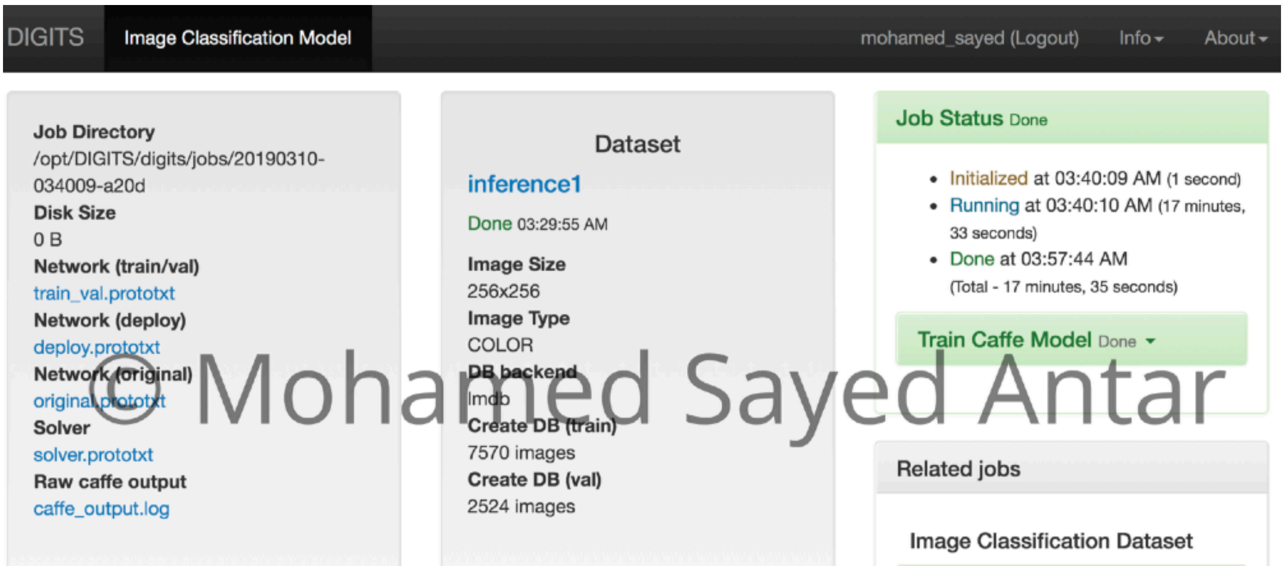


the model

- by choosing `images` then `classification`



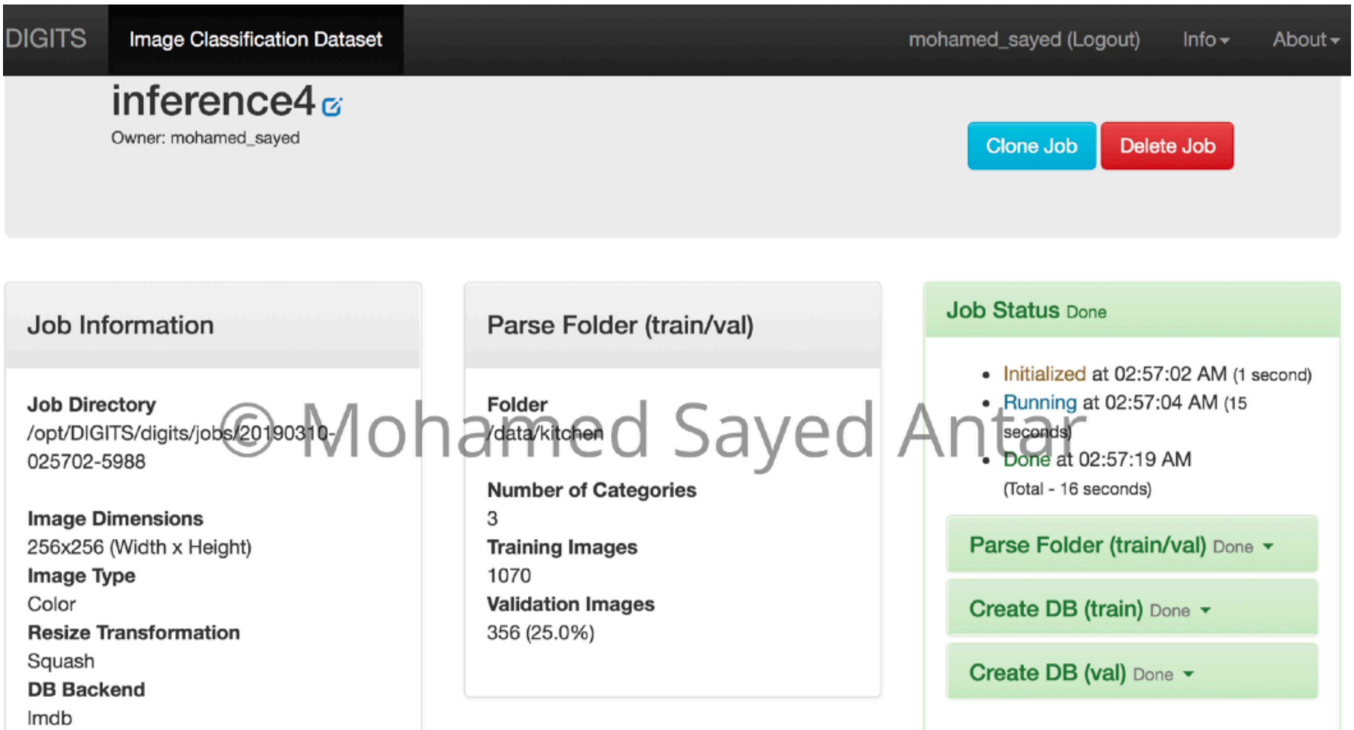
- now we can set the number of epochs, Snapshot interval, Base Learning Rate, Validation interval ....
- by choosing one of the 3 networks provided LeNet, AlexNet and GoogLeNet in this case AlexNet gave a good results.
- giving the model a name then create.



<u>for the Collected Data</u> there are 1070 images for the following 3 classes:

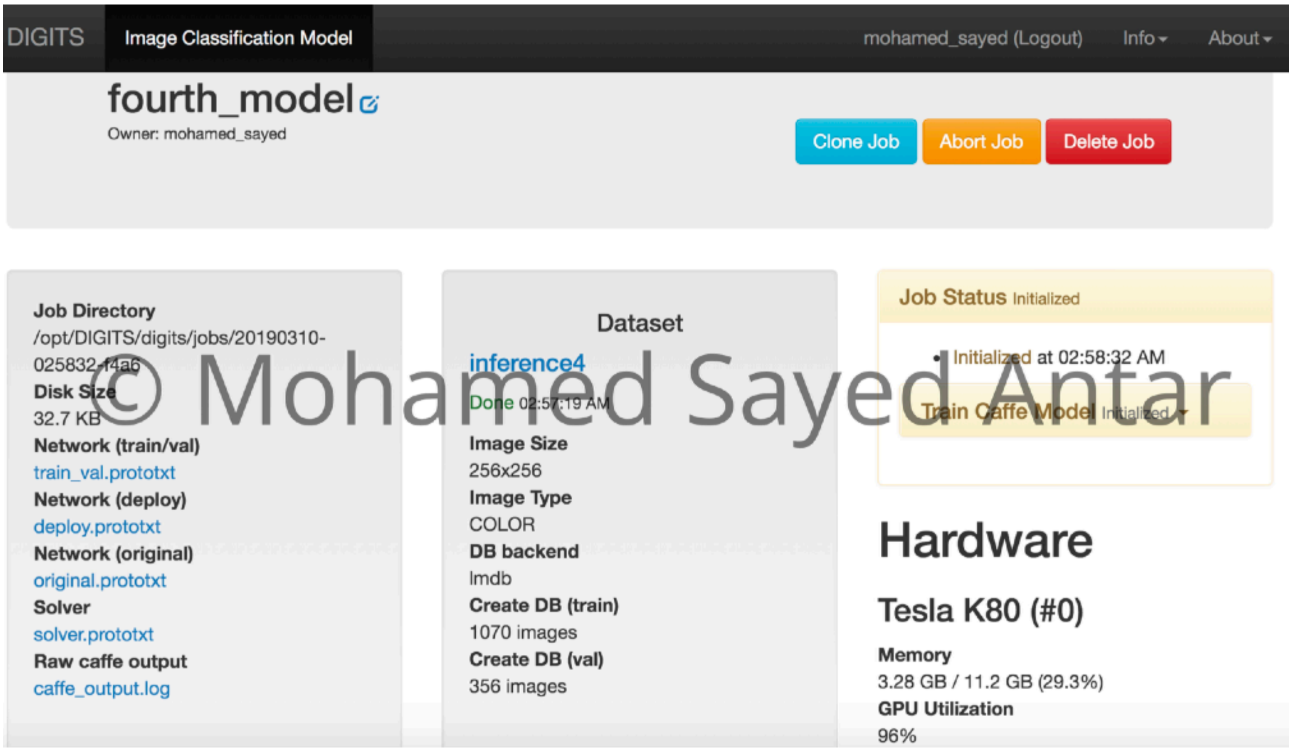
1. spoon
2. fork
3. no-thing

- the collected data using mobile camera then by resizing the collected data to be 256*256
- by choosing `images` then `classification`.
- adding our dataset url in this case using udacity workspace it will be `/data/kitchen`
- as previous more setting can be done like Minimum samples per class or maximum samples per class or even the percentage of validation photos and testing photos
- then by giving the dataset a name and choosing create :



the model

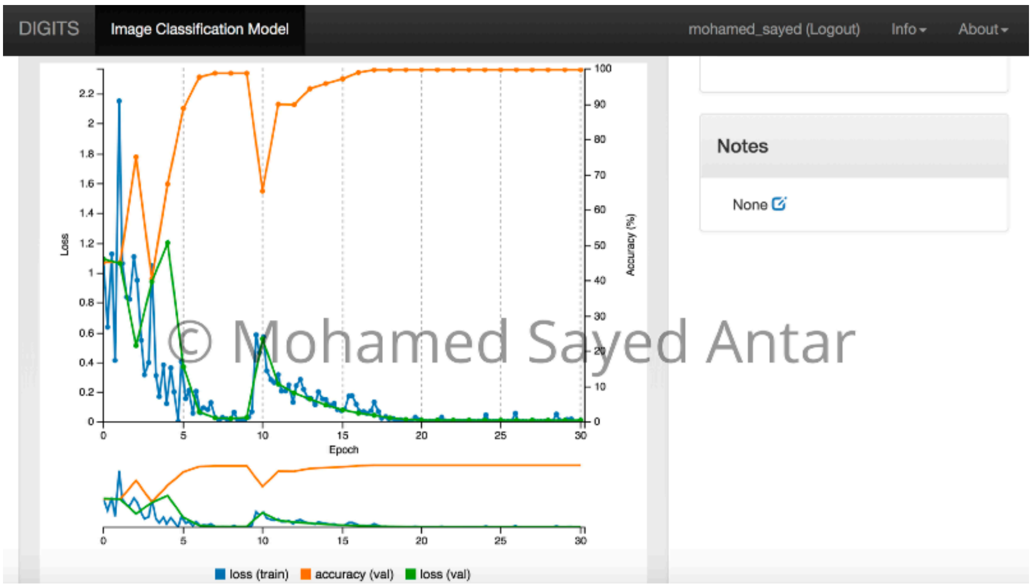
- by choosing `images` then `classification`
- now we can set the number of epochs, Snapshot interval, Base Learning Rate, Validation interval ....
- by choosing one of the 3 networks provided LeNet, AlexNet and GoogLeNet in this case GoogLeNet gave a better results than AlexNet.
- giving the model a name then create.



4 Results

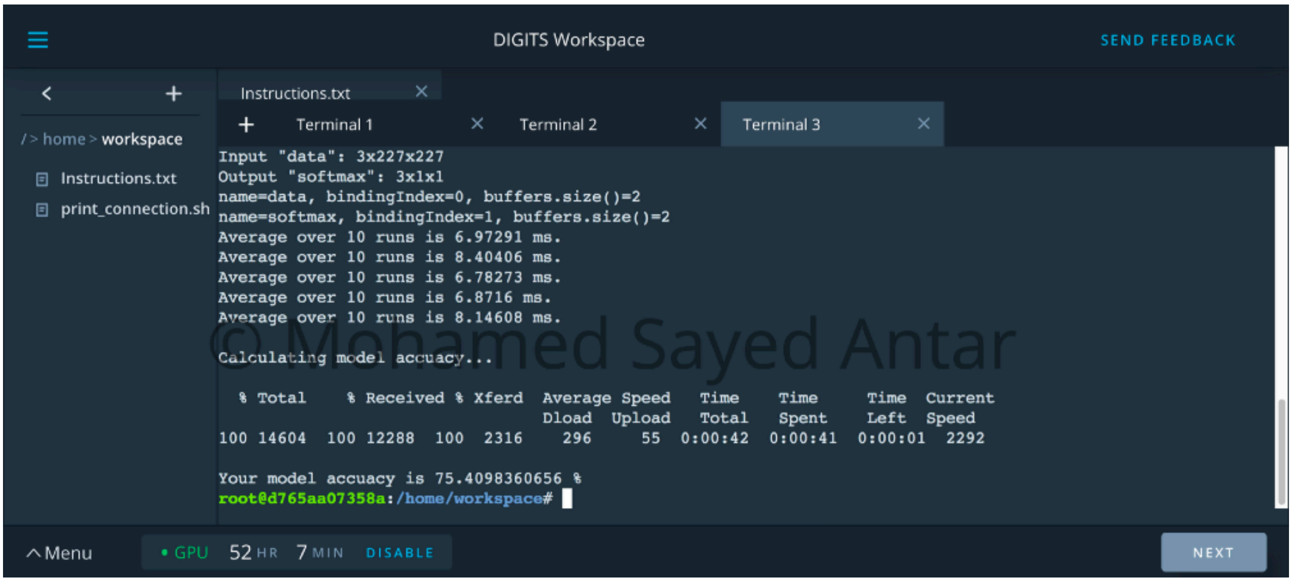
<u>first the Supplied Data</u>

the output model graph for the supplied data using AlexNet

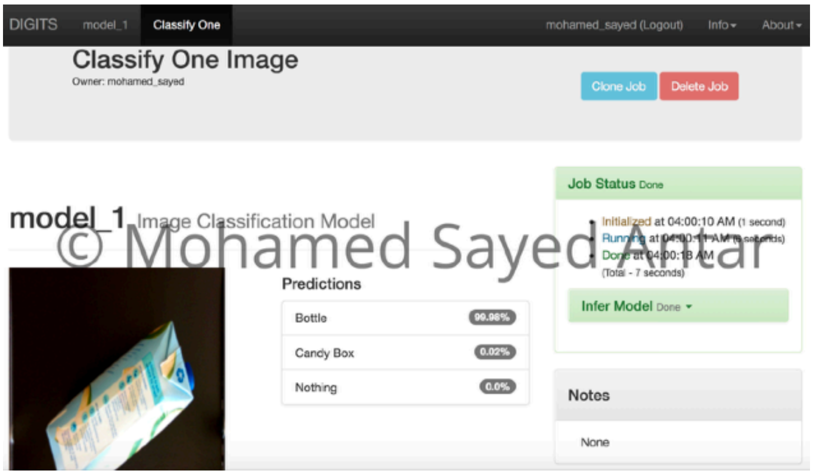


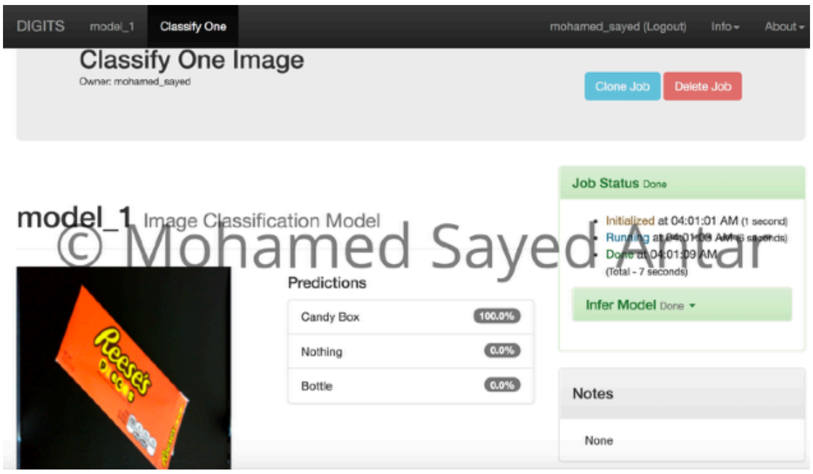
two ways to see the model results as below:

- by running the `evaluate` command in a new terminal in udacity workspace the results will be like so



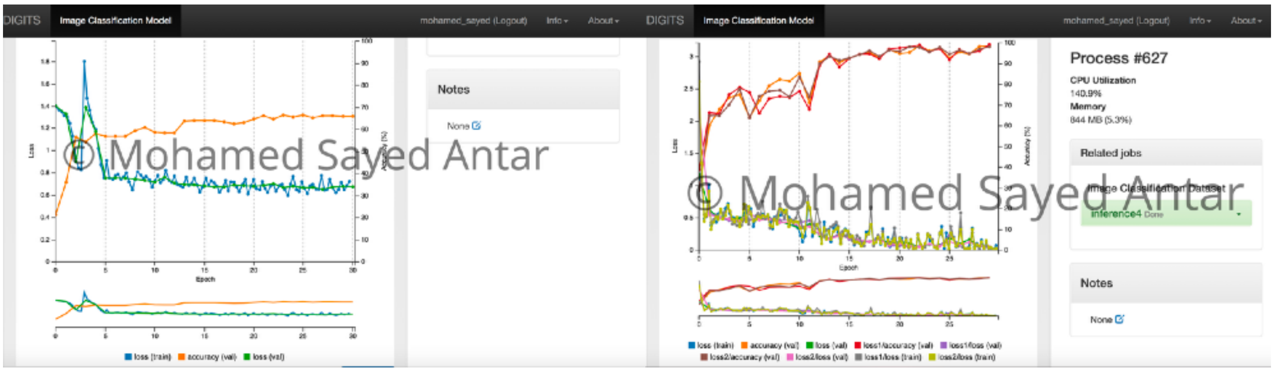
- by choosing images for test



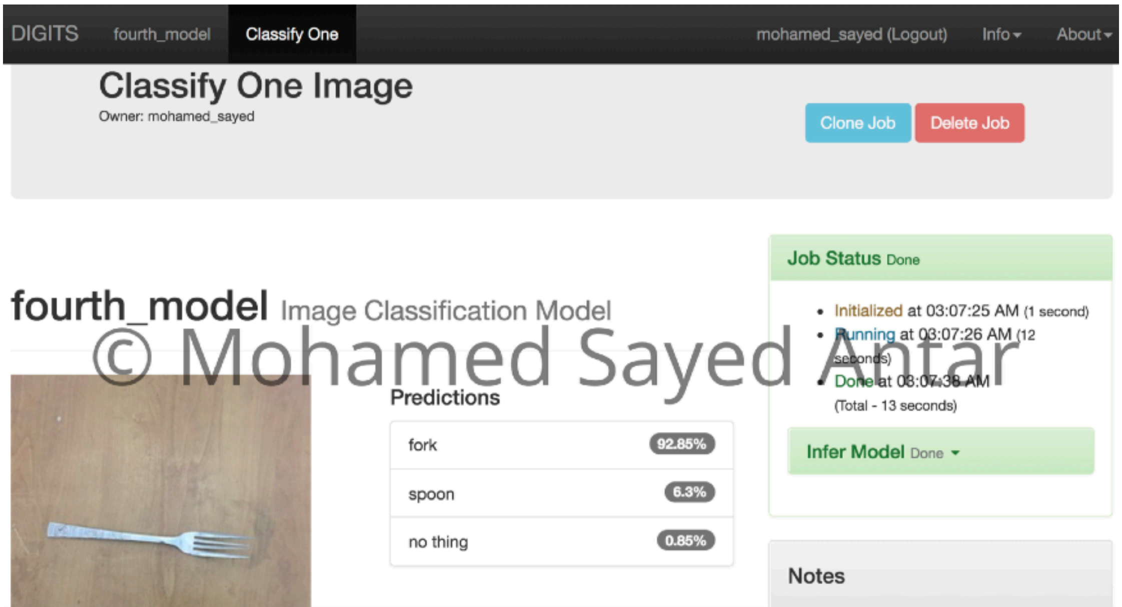


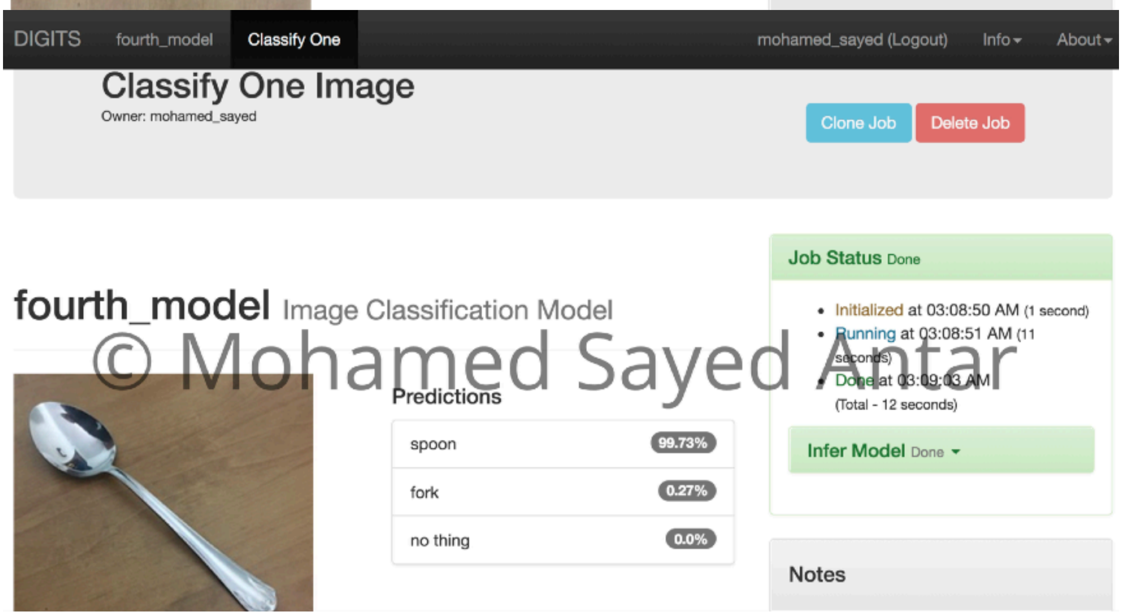
<u>second the Collected Data</u>

the 2 models results AlexNet and GoogLeNet



- to see this model result by choosing images for test





5 Discussion

huge difference in accuracy between AlexNet and GoogLeNet as shown previously and this difference appear more with small number of data given to the model like the previous case about 1070 images only for classification, while this difference become smaller with increasing the number of data, after many attempts the inference time for GoogLeNet is greater than AlexNet.

6 Conclusion / Future Work

for the inference time AlexNet is good, but developers in many cases look for the accuracy so GoogLeNet is the best choice, increasing the number of collected data will lead to increasing accuracy but in case too much data the change will be very small almost no thing.

for the robotic kitchen increasing the number of classes with almost every kitchenware known till now, will lead to more intelligent robots due to variety in classifying objects ability

7 references

https://www.asme.org/engineering-topics/articles/robotics/the-robotic-kitchen-is-cooking

https://www.iflscience.com/technology/robot-chef-home-could-arrive-2017/

https://meee-services.com/how-to-use-cooking-robots-in-your-kitchen/