Training miniVggNet by NVIDIA DIGITS (Xilinx UG1335)

Author: Alex He (ahe@xilinx.com)

Date: Mar-11-2019

Xilinx publish the Edge AI Tutorials to GitHub which show how to train NN model and then deploy the model by DNNDK. The project is at https://github.com/Xilinx/Edge-AI-Platform-Tutorials with scripts to help you. This doc give you another way to do the model training with NVIDIA/DIGITS which has a good visualization web interface. I create a docker image for easy deploy the NDVIDIA/DIGITS w/ Xilinx DNNDK.

The docker image is here.
https://hub.docker.com/r/alexhegit/dnndk

Please refer to this documentation to setup the docker container and start the DIGITS.
https://github.com/alexhegit/AlexTryMachineLearning/blob/master/CIFAR10_Caffe_Tutorial_UG1335/DNNDK_w_NvidiaDigits.md

Then I suppose your DIGITS service is running and just show how to train out a miniVggNet with CIFAR10.

1. Login the DIGITS
   Input http://localhost:5000 or replace the "localhost" with the IP of container if from remote machine in the web browser.

2. Create the CIFAR10 dataset in DIGITS

The CIFAR10 Caffe Tutorial (UG1335) have some scripts to download the CIFAR10 dataset by Keras. I suppose you have run the 1_write_cifar10_images.py which will download the original images in a structure tree.

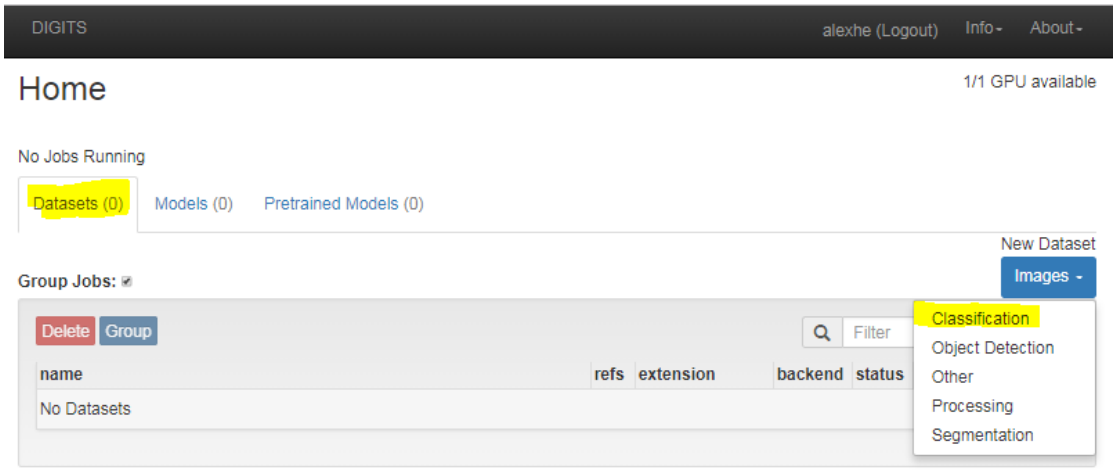
(https://github.com/Xilinx/Edge-AI-Platform-Tutorials/blob/master/docs/ML-CIFAR10-Caffe/caffe/code/1_write_cifar10_images.py)

The originals images root directory should be /root/ML/cifar10/input/cifar10_jpg/. There have 10 subclass folders each class in train/ and val/ like bellow.

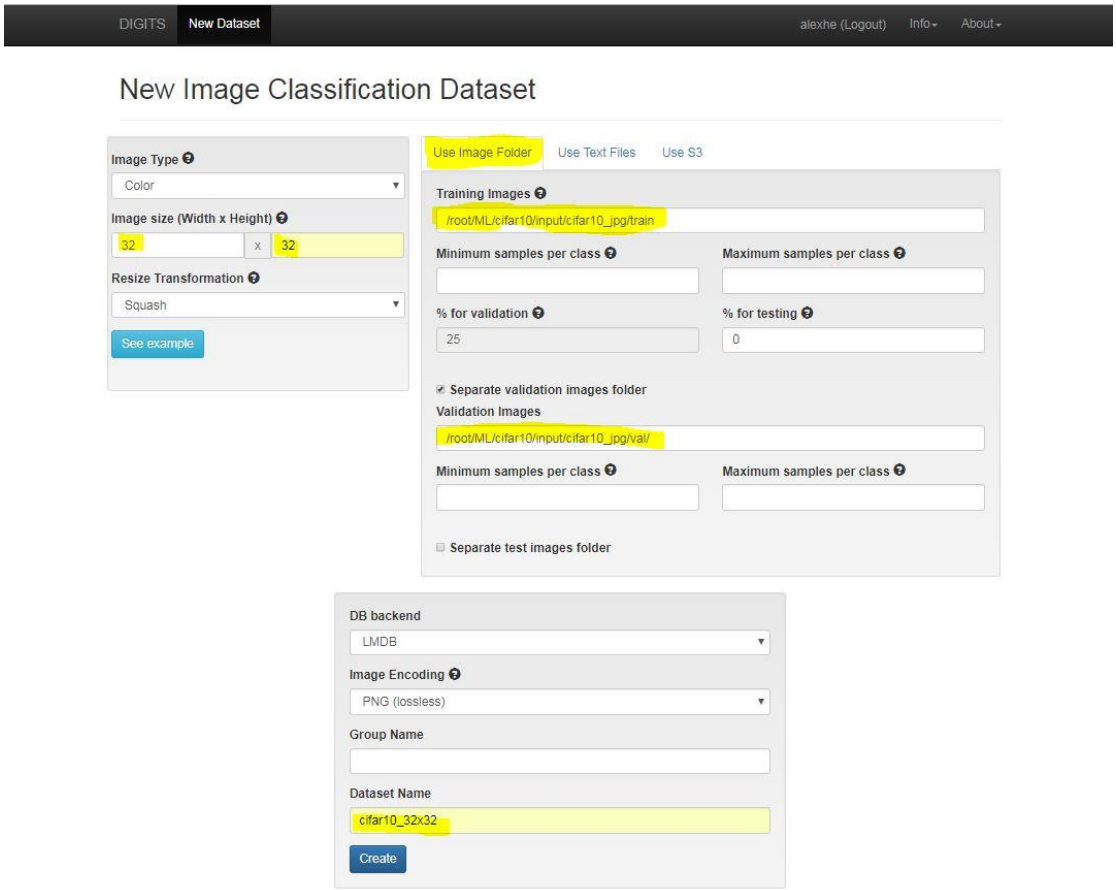
```
root@dnndk:~/ML/cifar10/input/cifar10_jpg# tree train/ -L 1
train/
|-- airplane
|-- automobile
|-- bird
|-- cat
|-- deer
|-- dog
|-- frog
|-- horse
|-- labels.txt
|-- ship
|-- train.txt
`-- truck

10 directories, 2 files
root@dnndk:~/ML/cifar10/input/cifar10_jpg# tree val/ -L 1
val/
|-- airplane
|-- automobile
|-- bird
|-- cat
|-- deer
|-- dog
|-- frog
|-- horse
|-- labels.txt
|-- ship
|-- test.txt
`-- truck

10 directories, 2 files
```

The DIGITS will use this structure image folder to create the training and valid dataset.

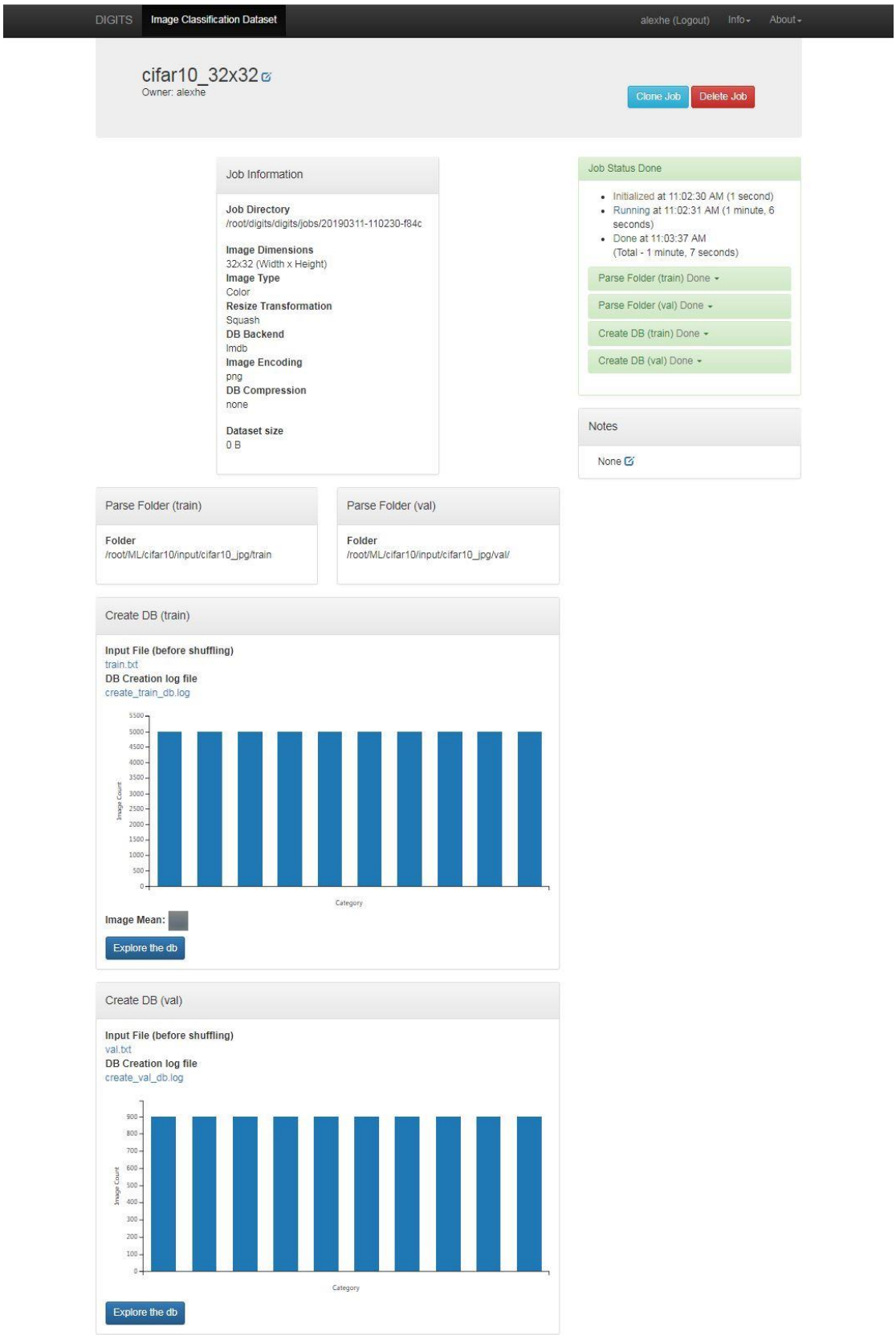
Click the “Dataset” label at top-left and Select the dataset type as “Classification”



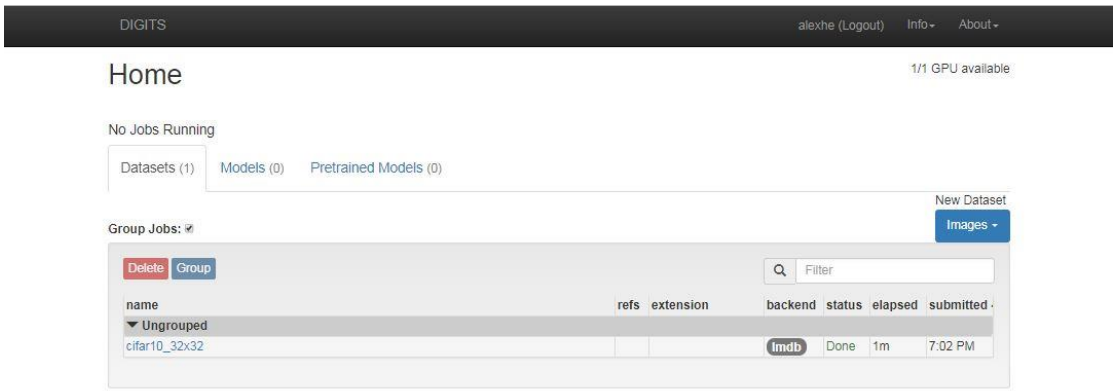
Input the contents highlight in the UI snapshot bellow. The PATHs of Training Images and Validation Images may changed as real of your enviroments. You just need to input the root directory with subclass folder for each class in it.



The dataset will be created as LMDB format.

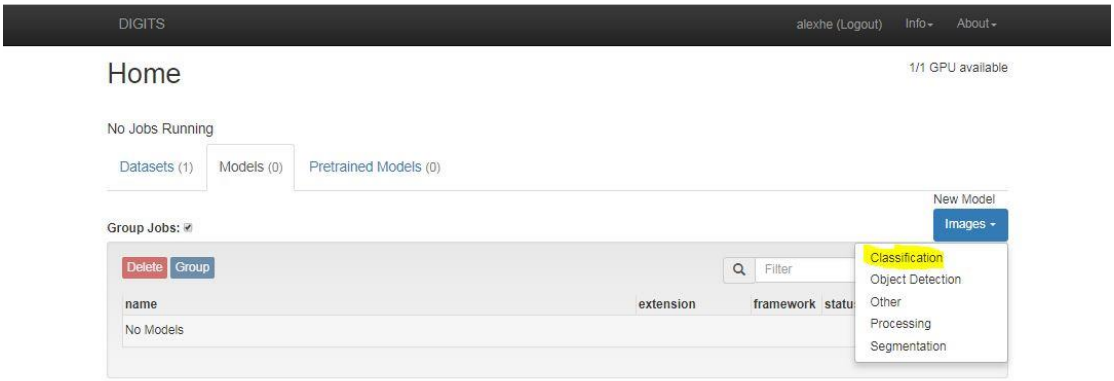


Now we have the cifar10_32x32 in the Datasets label.



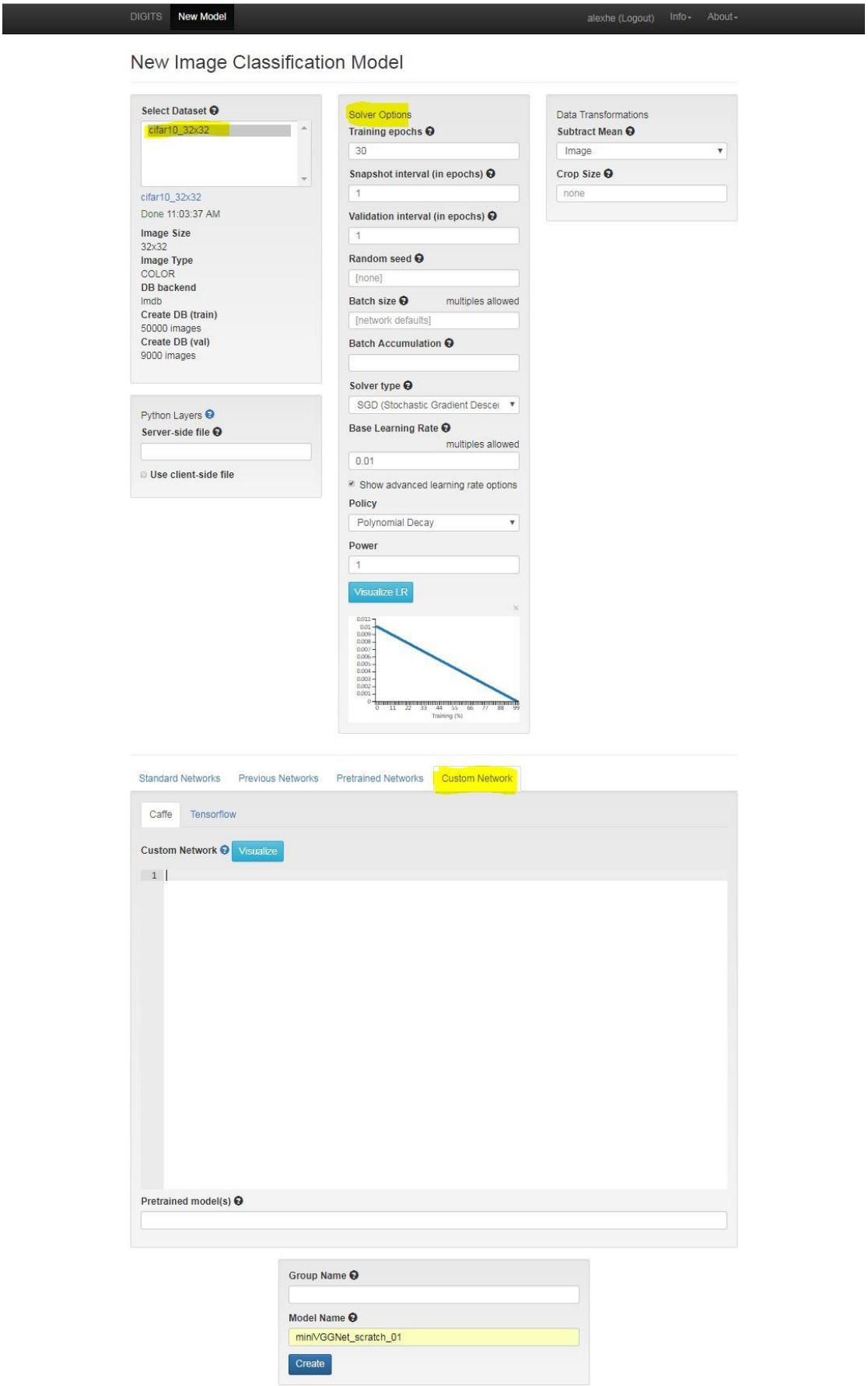
3. Create the miniVggNet model

   Click the “Models” label and chose the “Classification” type



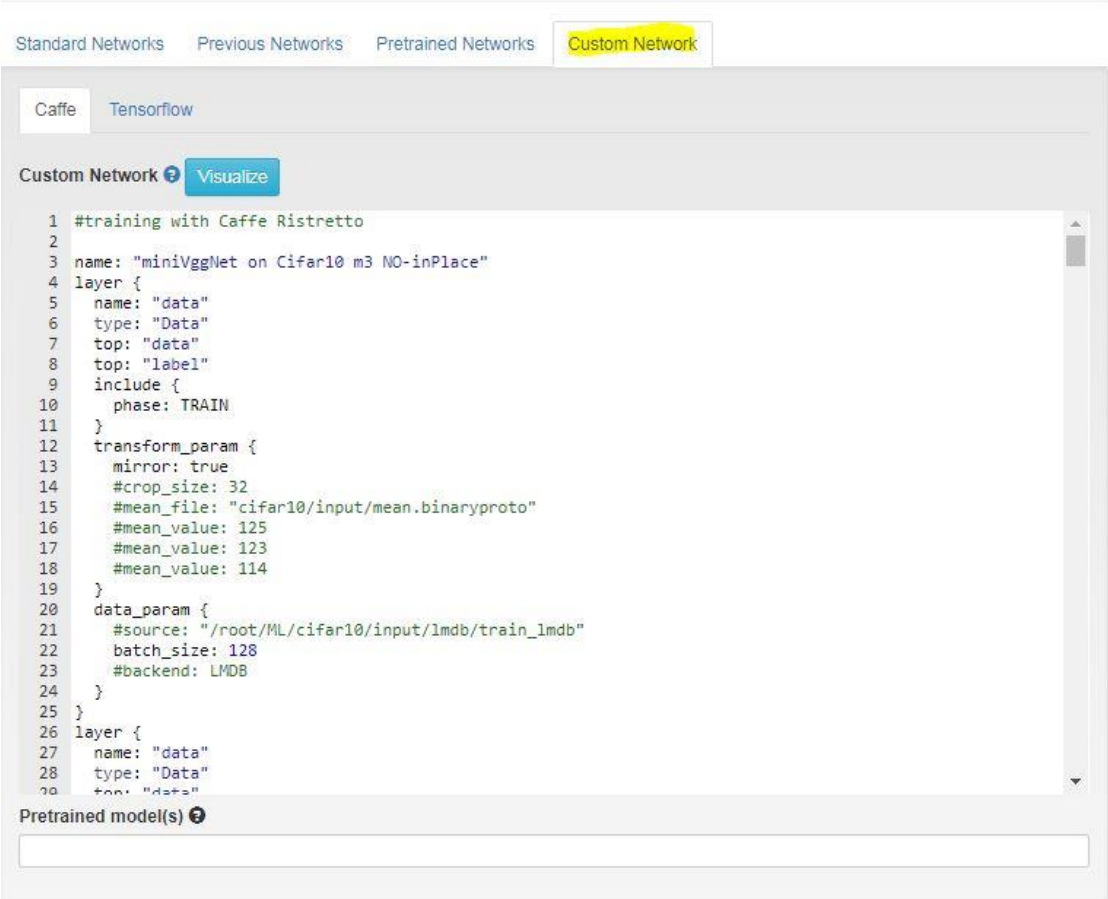
   Then

   - Select the cifar10_32x32 dataset we created at previous step.
   - Set the hyper-parameters of solver as you want
   - Chose Custom Network
   - Name the model



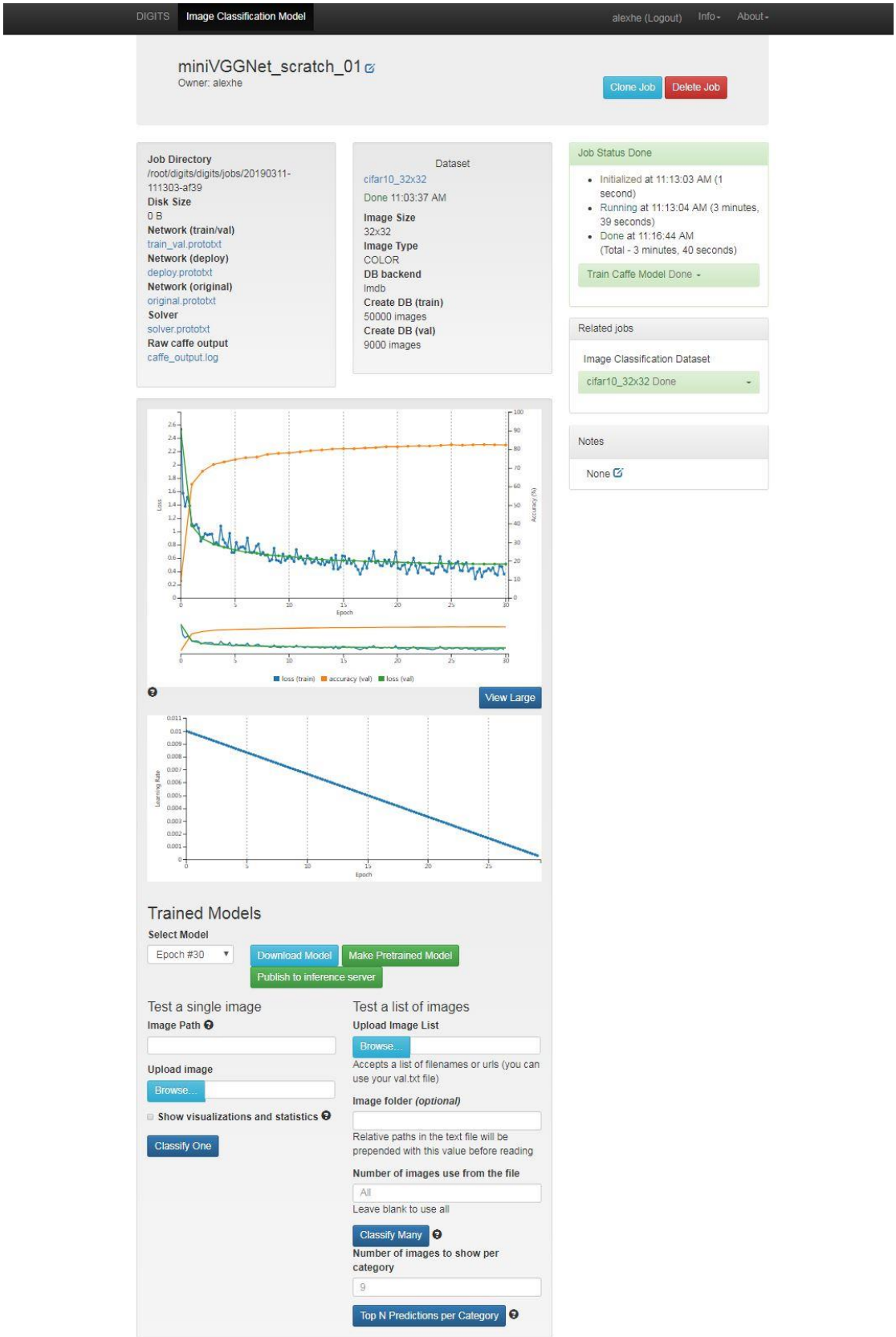
Define your network in prototxt format. Here is the sample of miniVggNet https://github.com/alexhegit/AlexTryMachineLearning/blob/master/CIFAR10_Caffe_Tutorial_UG1335/digits_train_val_3_miniVggNet.prototxt.

You can paste the content here.



4. Train your miniVggNet

   Click the “Create” button to start the training. Below is the snapshot of my training result. You can watch the accuracy/loss/learning rate/time/GPU statistics, etc of training process and results.

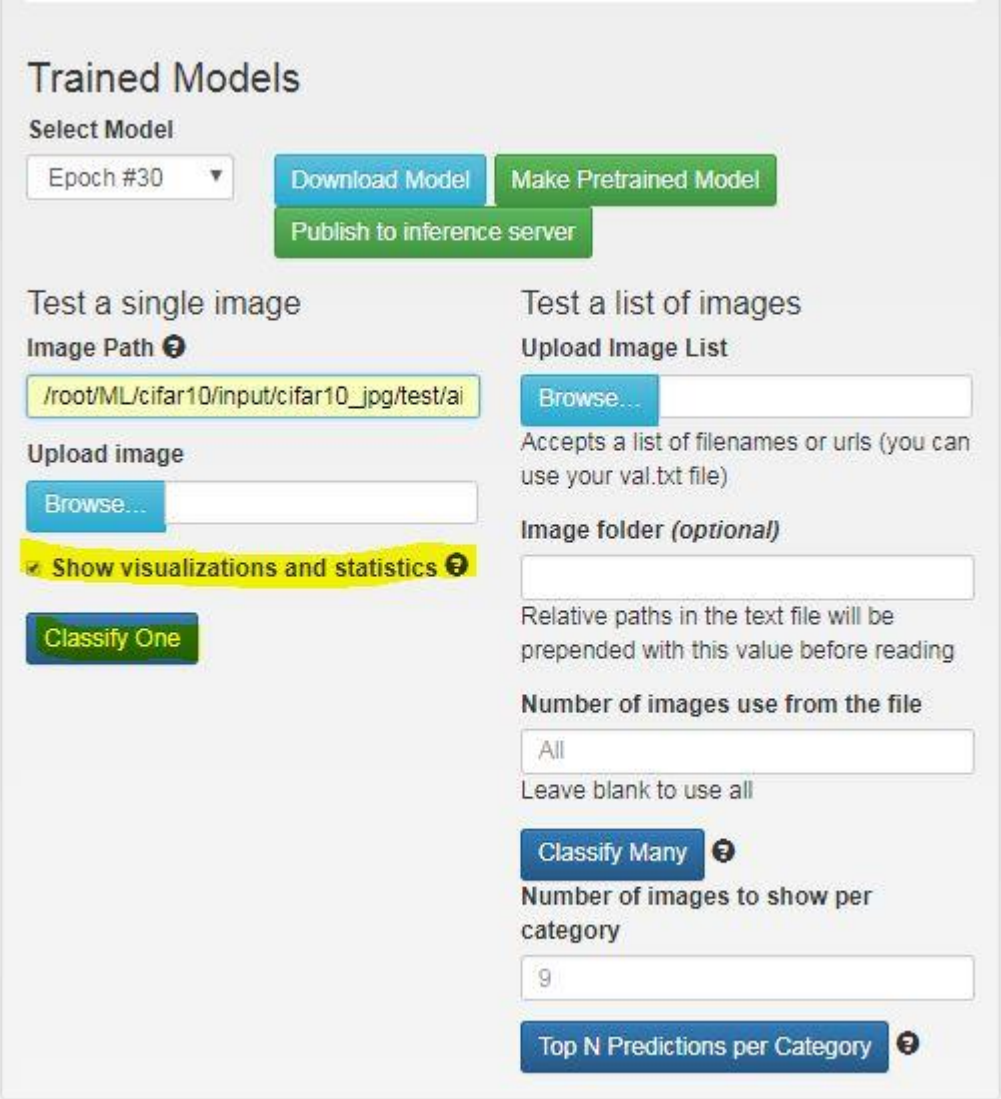


5. Test the model

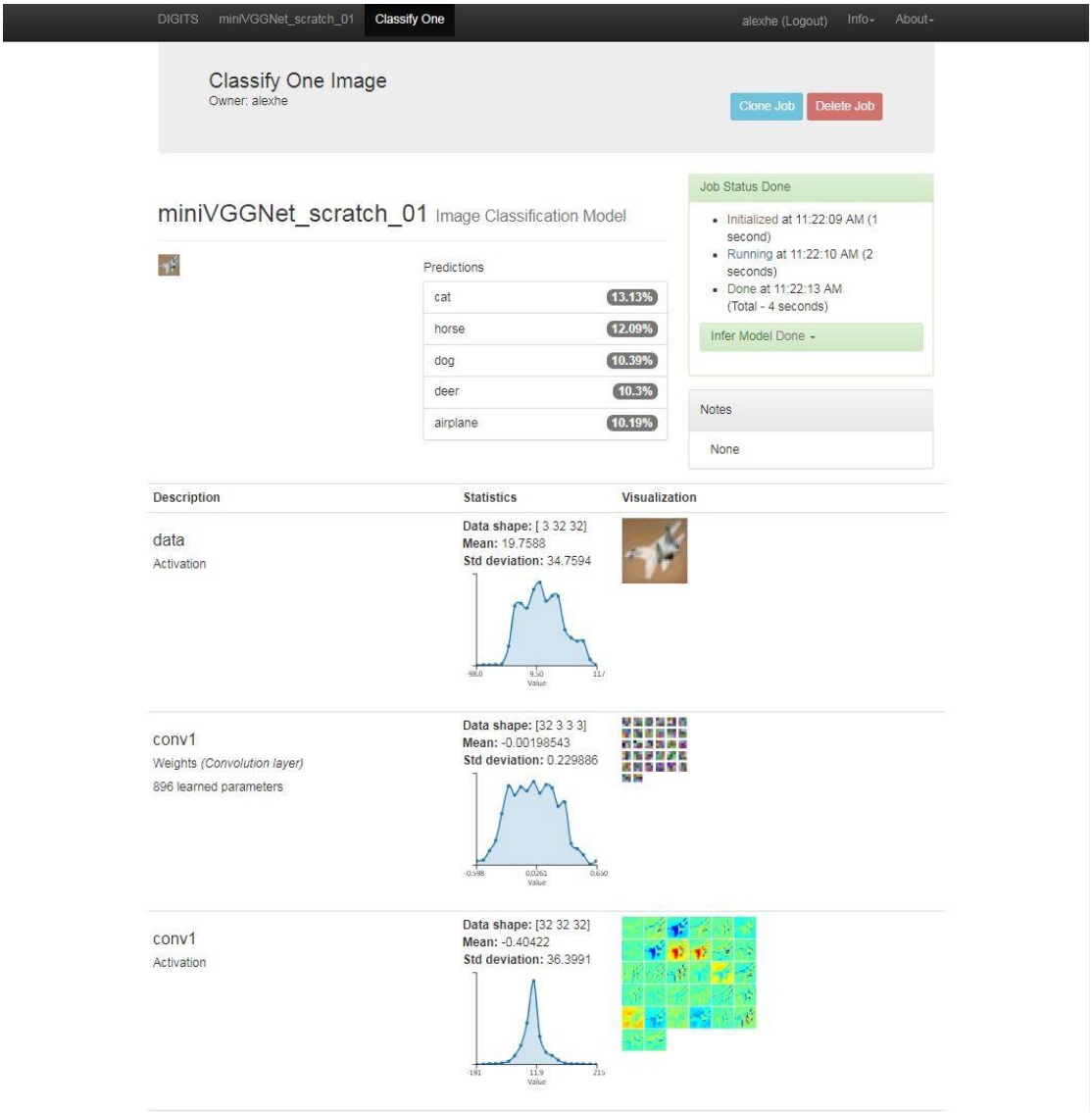
After the training fished, you can test it in the same UI in previous step. DIGITS provide two

type tests. Let me show you the “Test a single image”.

Input the Image Path and select “Show visualizations and statistics” if want.



You will see the details when image data pass through the Neural-Network as well as the predictions result.



You can chose the best model snapshot to be download for DNNDK deploy process as UG1335.