Ain Shams University
Faculty of Engineering
CESS – CHEP

Mohammed Ehab Elsaeed
16P8160

Selected Topics in Software Applications (Neural Networks)

CSE440
Project Documentation

Contents

1. Running Scripts

I have implemented two scripts located in the root directory of the project that would make running the project a one-click operation.

a. 16P8160_Docker.cmd

This script will:

1. Build the Docker image from the Docker file I developed (**dockerfiles/cpu.Dockerfile**)
2. Run the built Docker image as root and pass to it a bash script I developed (**code/16p8160_runscript.sh**).
3. The bash script passed will:
    a. Run Train.py.
    b. Run Inference.py using a preselected sample image of the digit 7.
    c. Run Inference.py again allowing you to run Inference.py using your own image. Please provide the full path (For an image in the running Docker container ex. "/home/trainingSet/3/img_7.jpg").

Please note that it is not possible to show an image in Docker. The only dependency needed to run the docker script is to have Docker installed on your machine.

b. 16P8160_Windows.cmd

This script will:

1. Run Train.py.
2. Run Inference.py using a preselected sample image of the digit 7.
3. Run Inference.py again allowing you to run Inference.py using your own image Please provide the full path (For an image in the file system).
4. Open your browser to localhost:6060 and run Tensorboard automatically.

To run this script, you have to setup your environment just like mine, please check chapter 5 of this document.

2. Implemented Bonuses

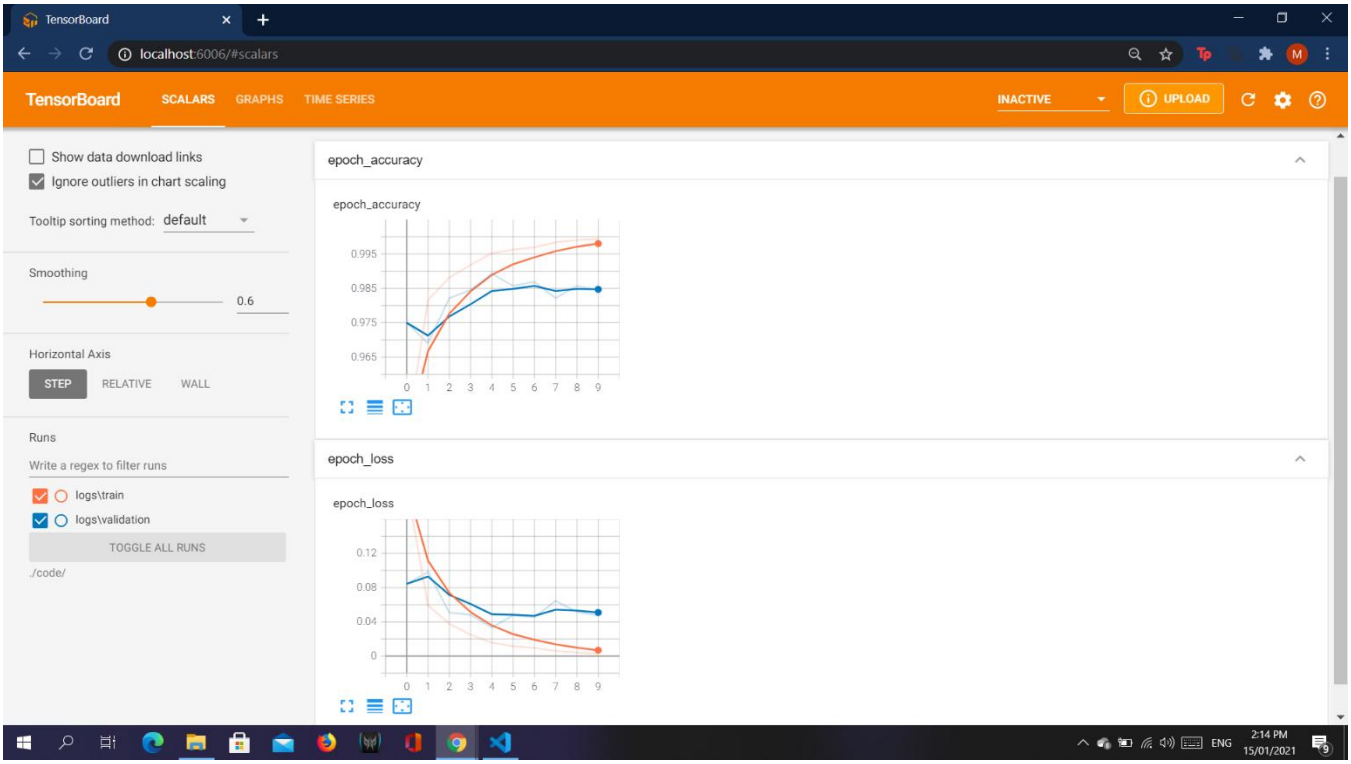
Input image of any size.

```
# Image resizing layer.Resize the batched image input to target height and width.
# The input should be a 4-D tensor
keras.layers.experimental.preprocessing.Resizing(
    28, 28, interpolation="bilinear"
),
# Output Shape: (n, 28, 28, 3), where n is the number of images fed to the model
```

Tensorboard.

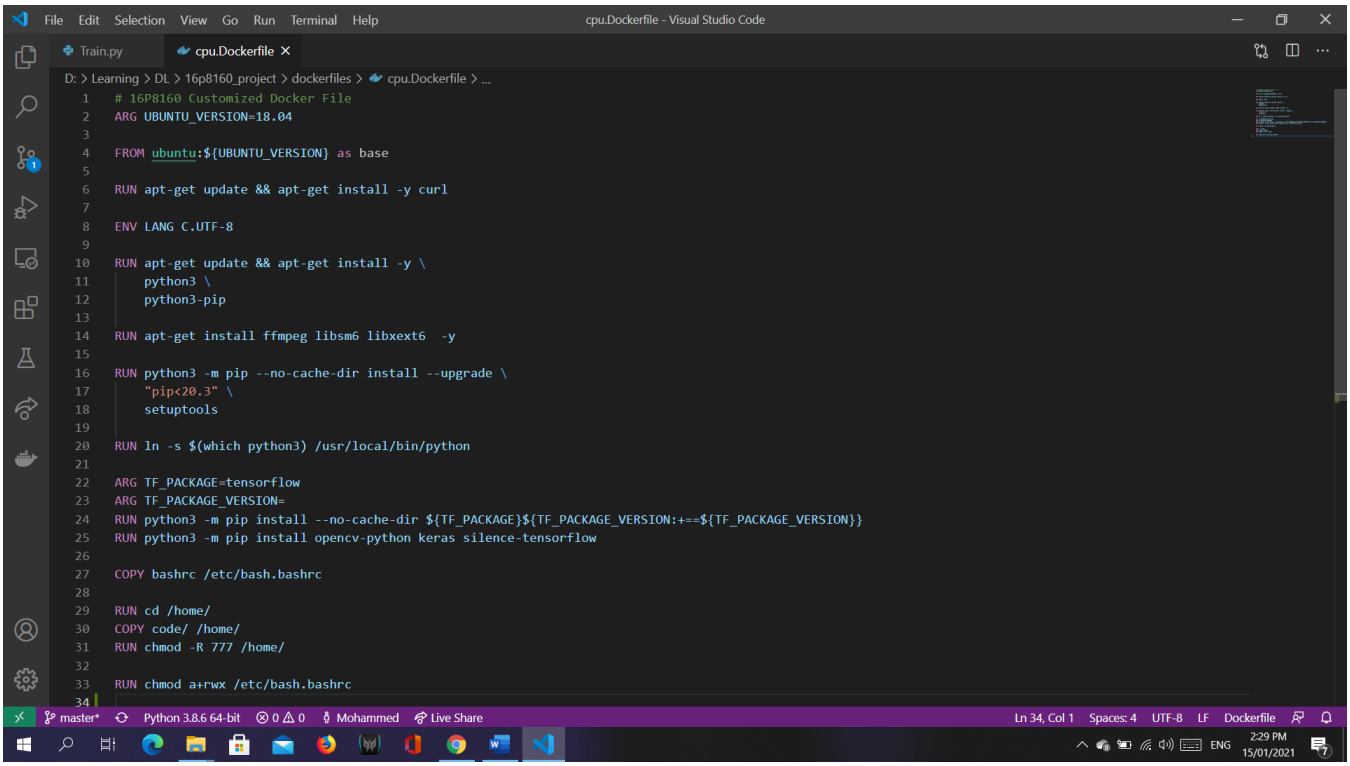
```
# BONUS1, tensorboard
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=f"{rootDir}/logs")
```

```python
# training the model
model.fit(
    x_train,
    y_train_one_hot,
    # To fit in memmory easily
    batch_size=128,
    # Through Trials, 10 epocs yielded acceptable accuracy and loss results
    epochs=10,
    # Using 98% of the data for training
    validation_split=0.02,
    # for BONUS1 Tensorboard
    callbacks=[tensorboard_callback]
)
```



Docker File for creating the environment and auto running the Training and Inference Scripts.

Docker file is at **dockerfiles/cpu.Dockerfile**



3. Model Summary

Layer	Input Shape	Output Shape	Number of Parameters
Resizing **Interpolation Method: Bilinear**	(n, any, any, 3)	(n, 28, 28, 3)	0
2D Convolution **Filters: 25** **Kernel Size: 3x3** **Padding: valid** **Activation Function: Relu**	(n, 28, 28, 3)	(n, 26, 26, 25)	700
2D Max pooling **Size: 2x2** **Strides: 2, 2** **Padding: valid**	(n, 26, 26, 25)	(n, 13, 13, 25)	0

Flattening	(n, 13, 13, 25)	(n, 4225)	0
Dense (Fully Connected) **Units (Neurons): 250** **Activation Function: Relu**	(n, 4225)	(n, 250)	1056500
Dense (Fully Connected) **Units (Neurons): 10** **Activation Function: SoftMax**	(n, 250)	(n, 10)	2510

4. Hyperparameters

Optimizer: ADAM

Learning Rate: 0.001

Epochs: 10

Batch Size: 128

Validation Split=0.02

5. Environment

OS: Windows 10 Version 10.0.18363 Build 18363

silence-tensorflow version 1.1.1

Tensorflow version 2.4

Keras version 2.4.3

Numpy version 1.19.4

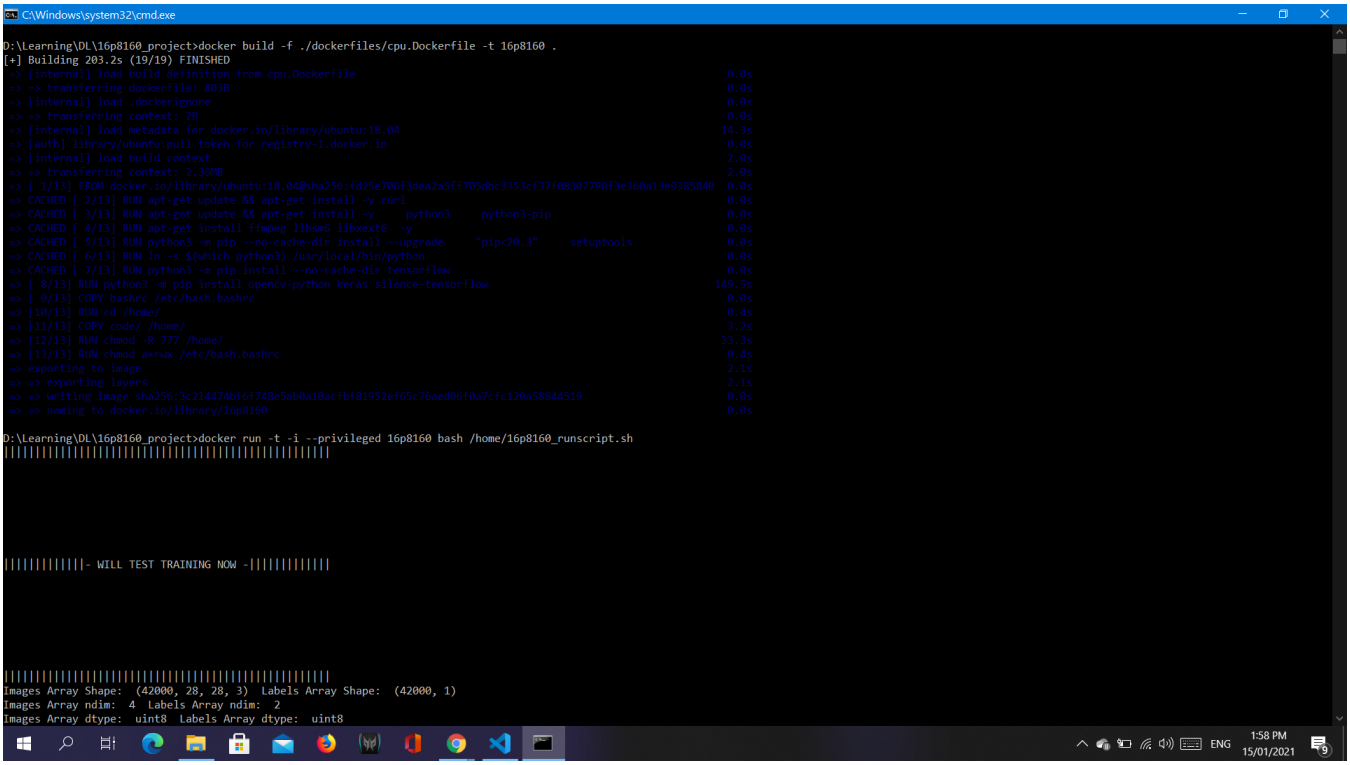
opencv-python version 4.4.0.46

NVIDIA CUDA version 11.2

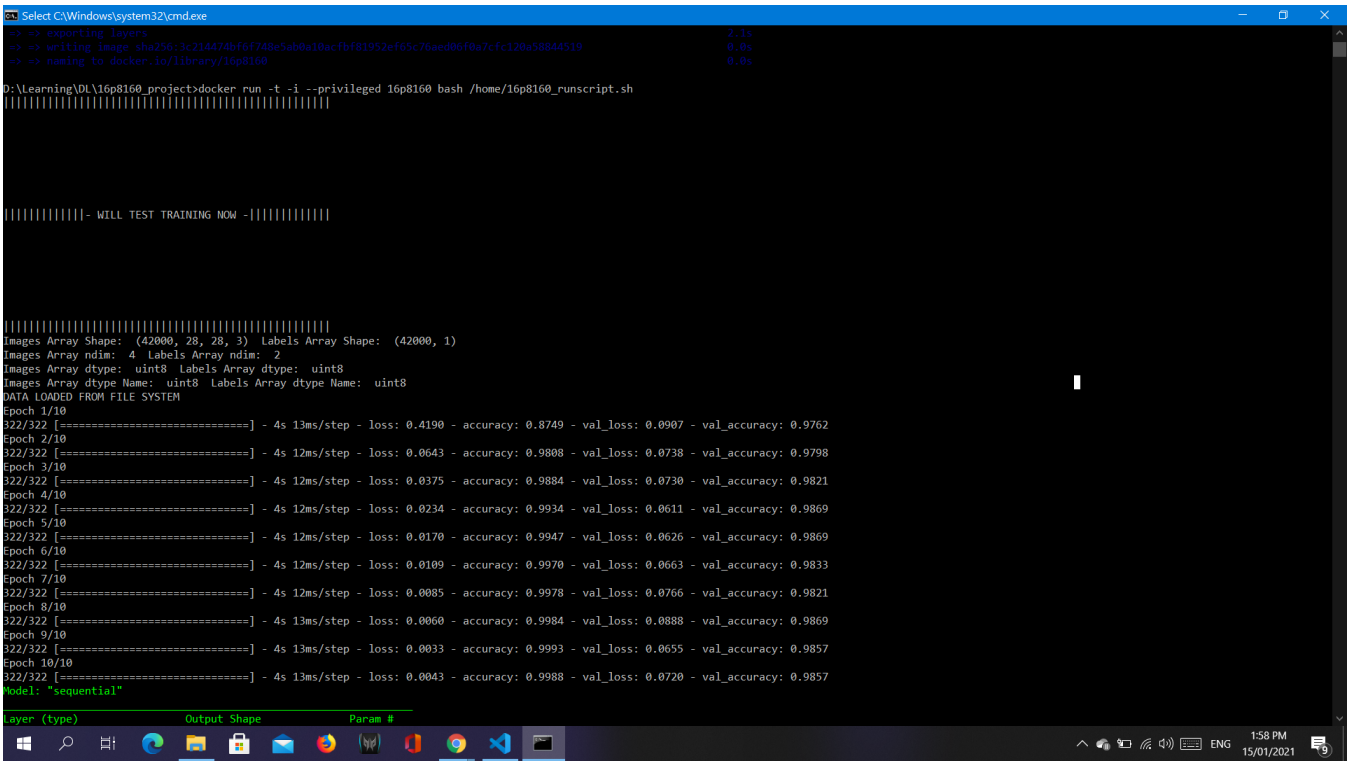
NVIDIA CUDNN version 8.0.5.39-windows-x64 for CUDA 11.1

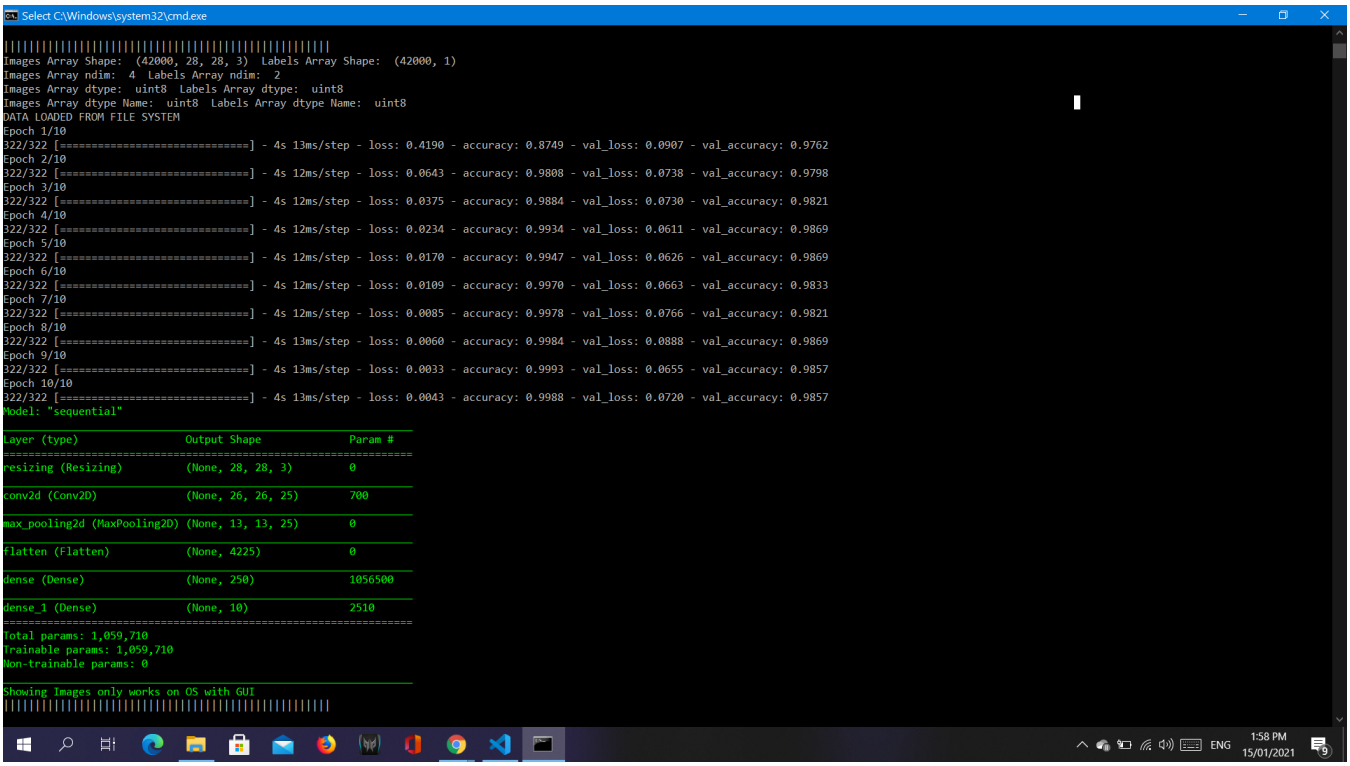
6. Screenshots

Docker Build & Run

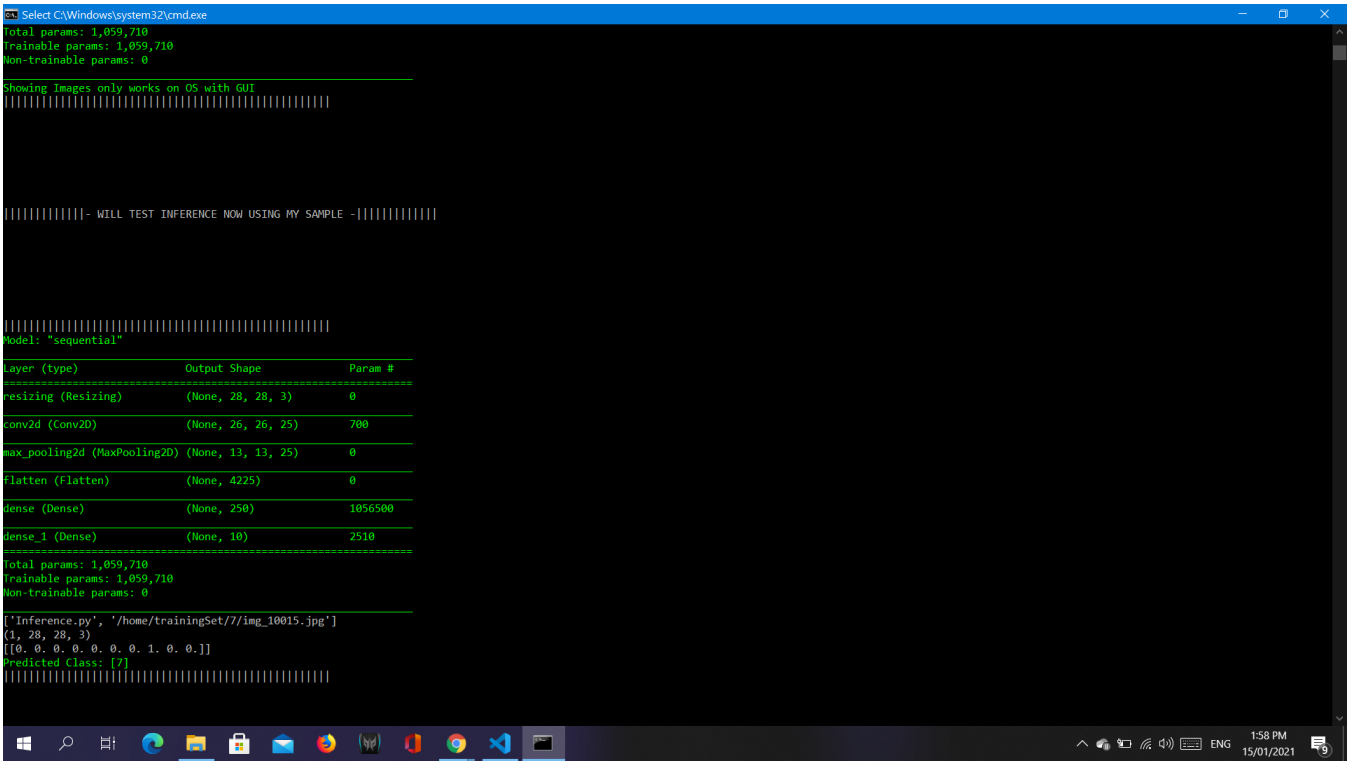


Docker Container Model Training

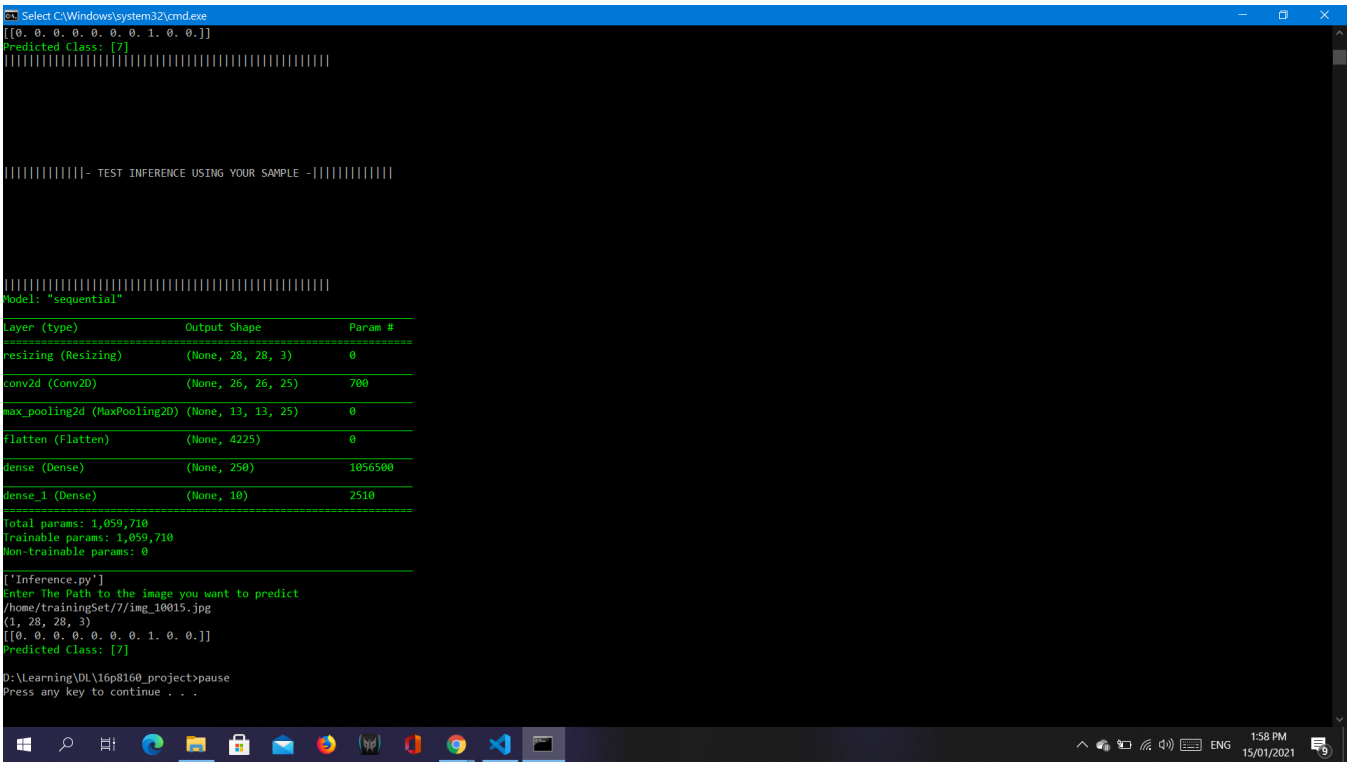




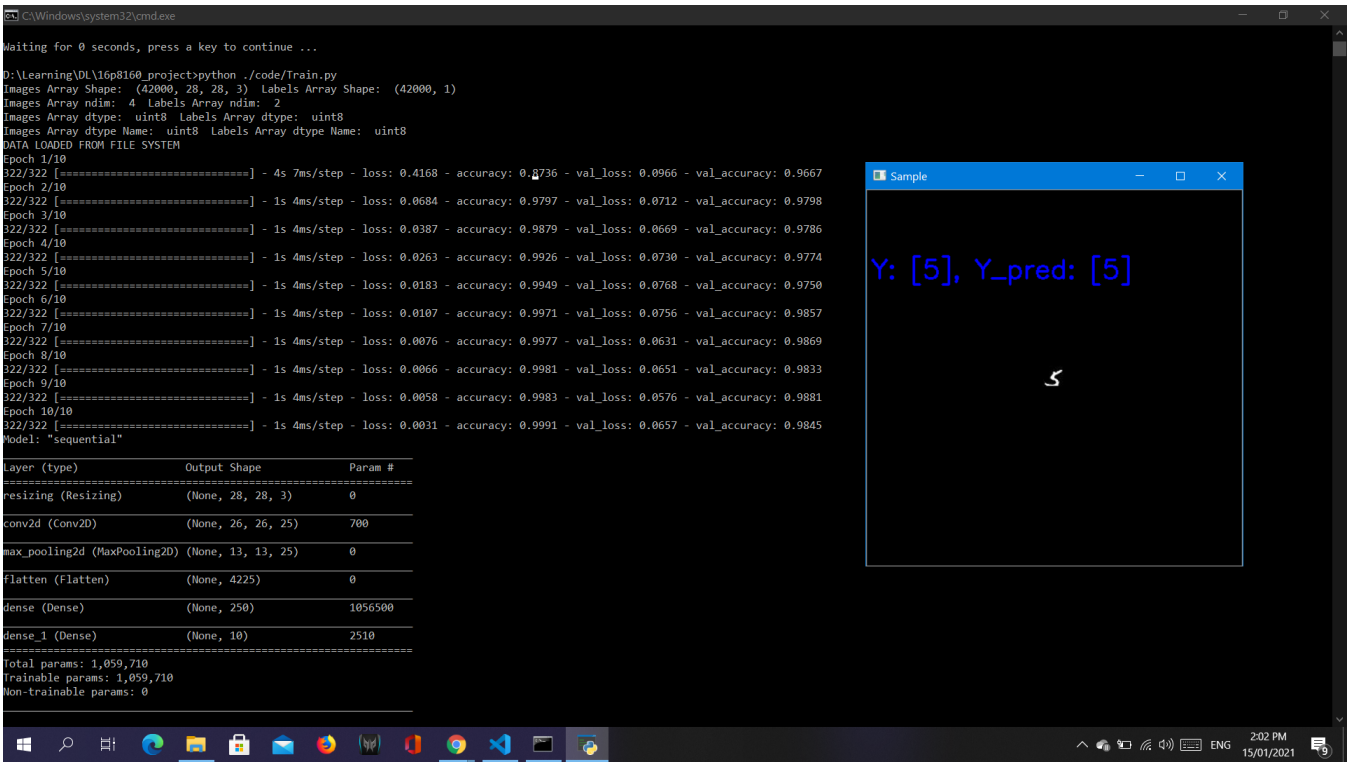
Docker Container Inference Using Preselected Sample

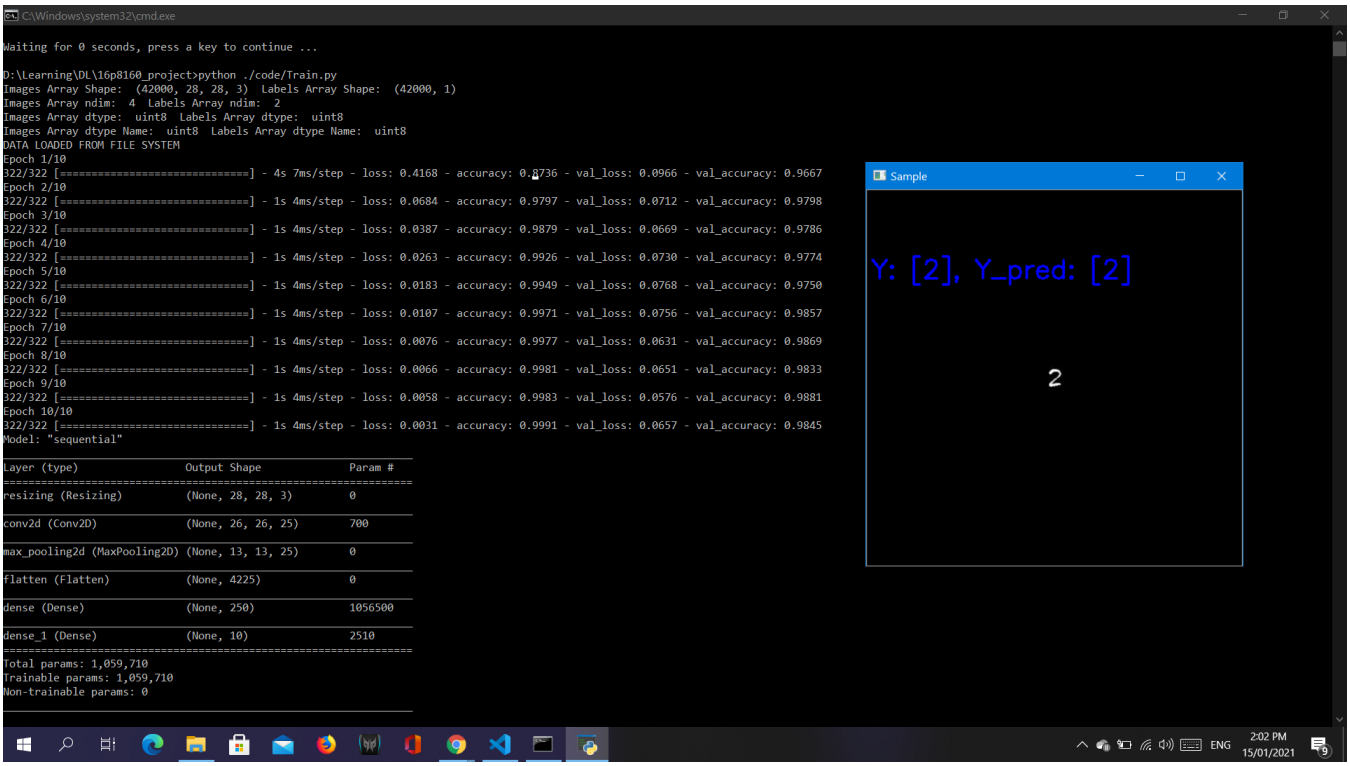


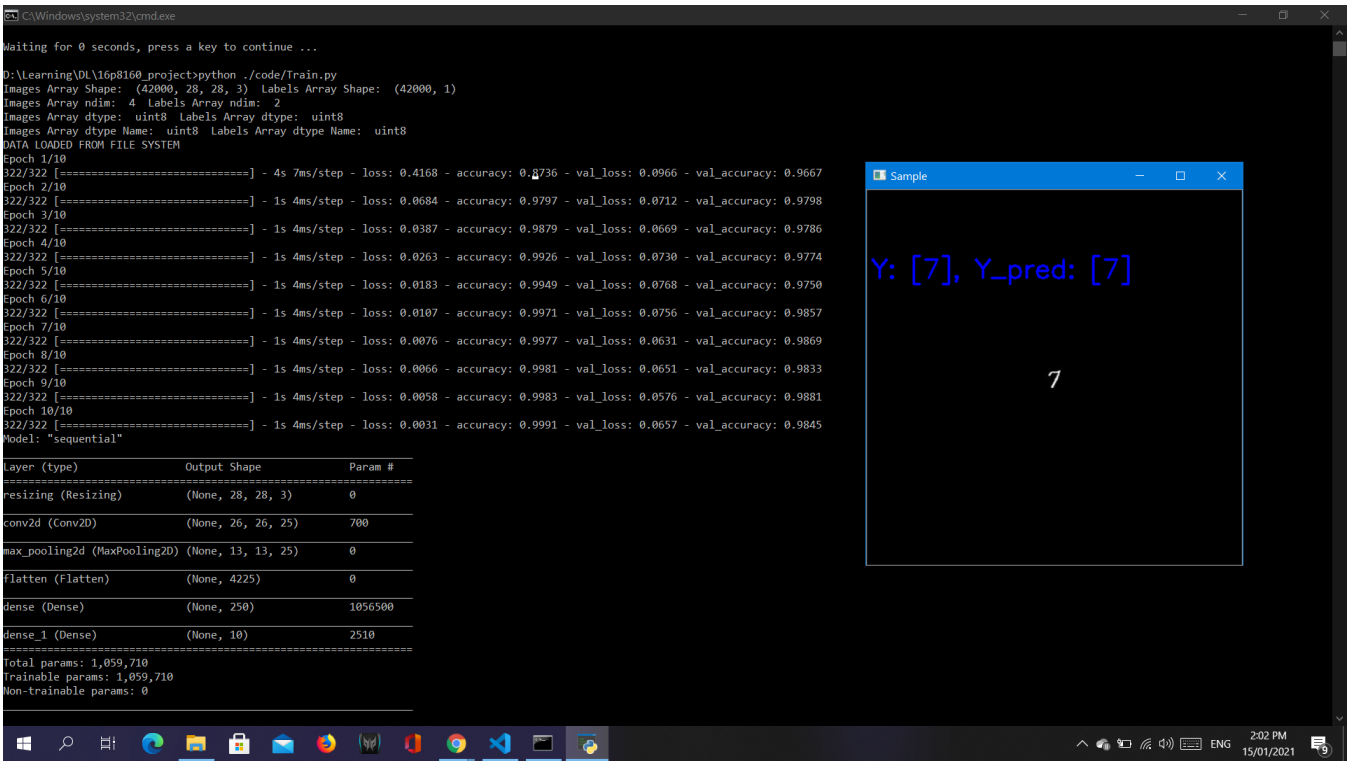
Docker Container Inference Using User-Input Image

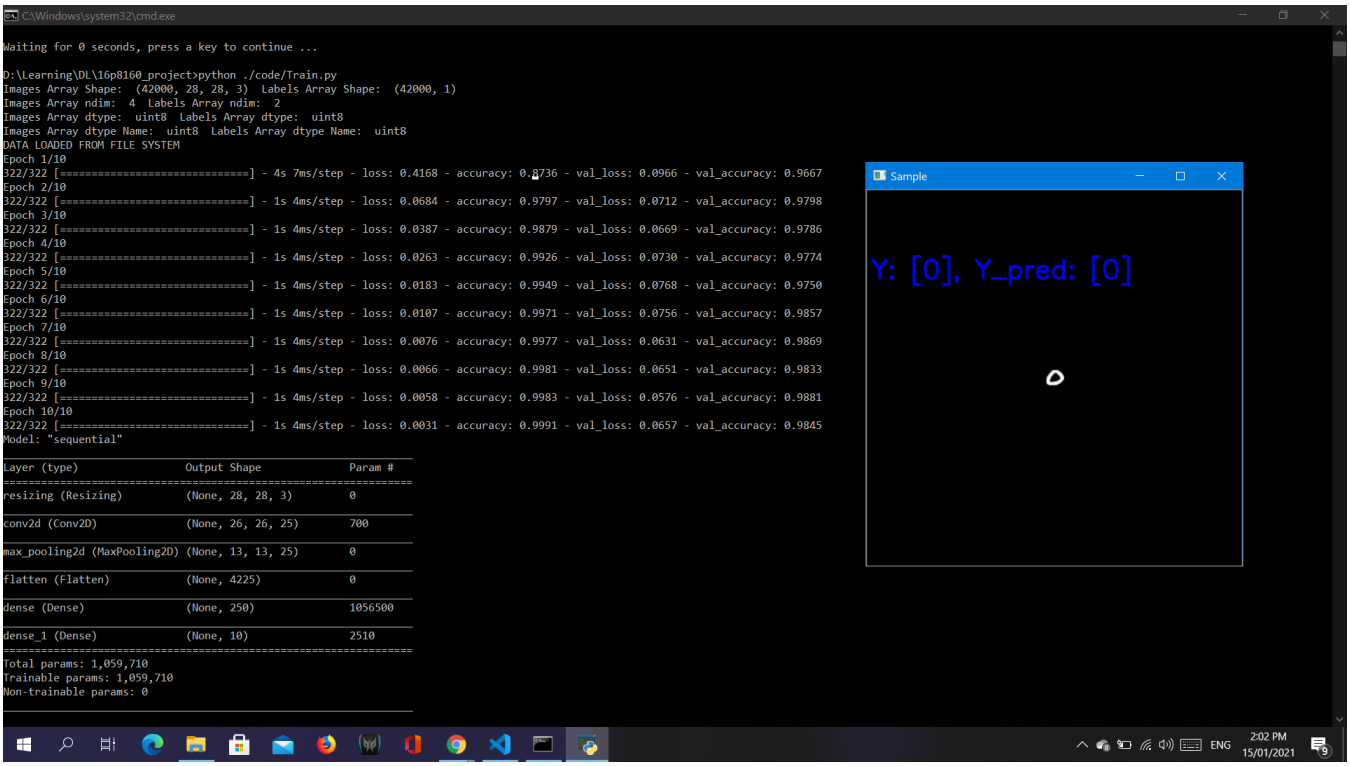


Windows Model Training and Showing Random Images Inference

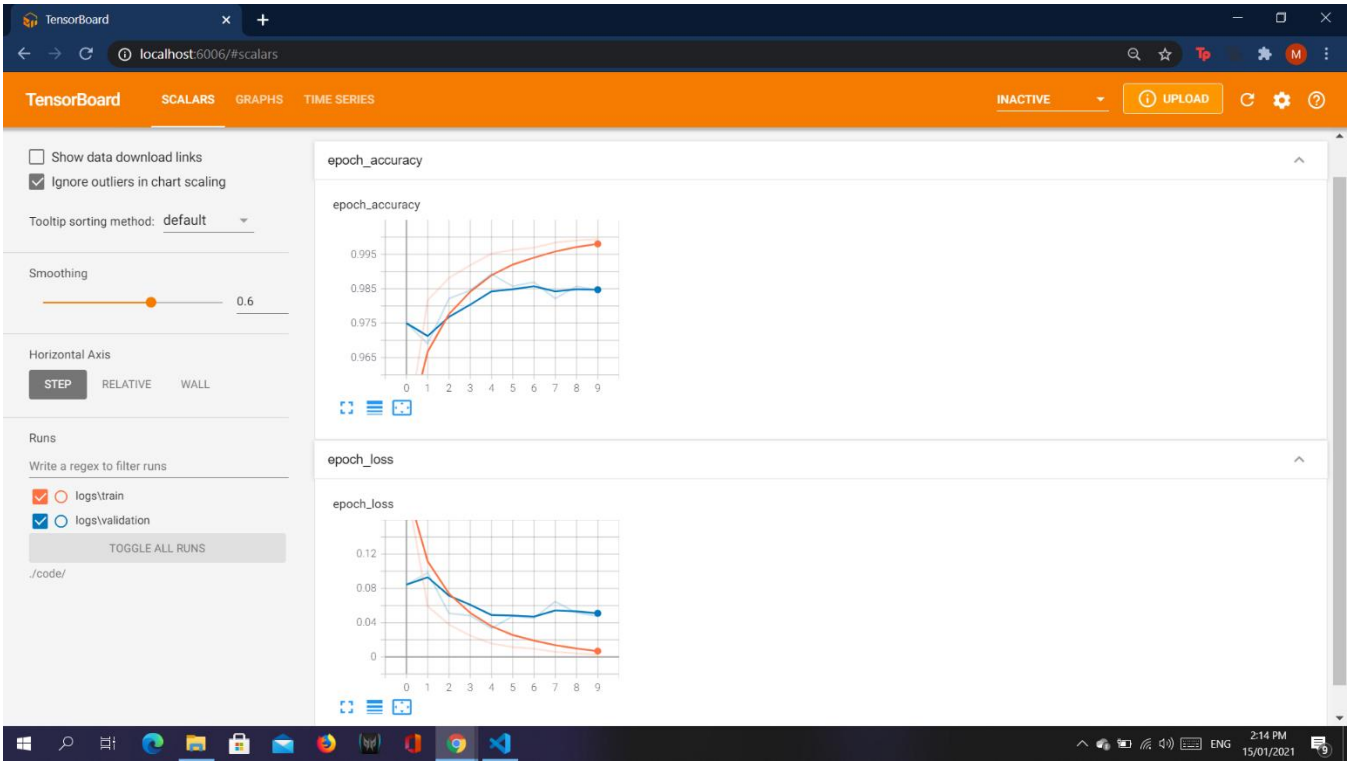


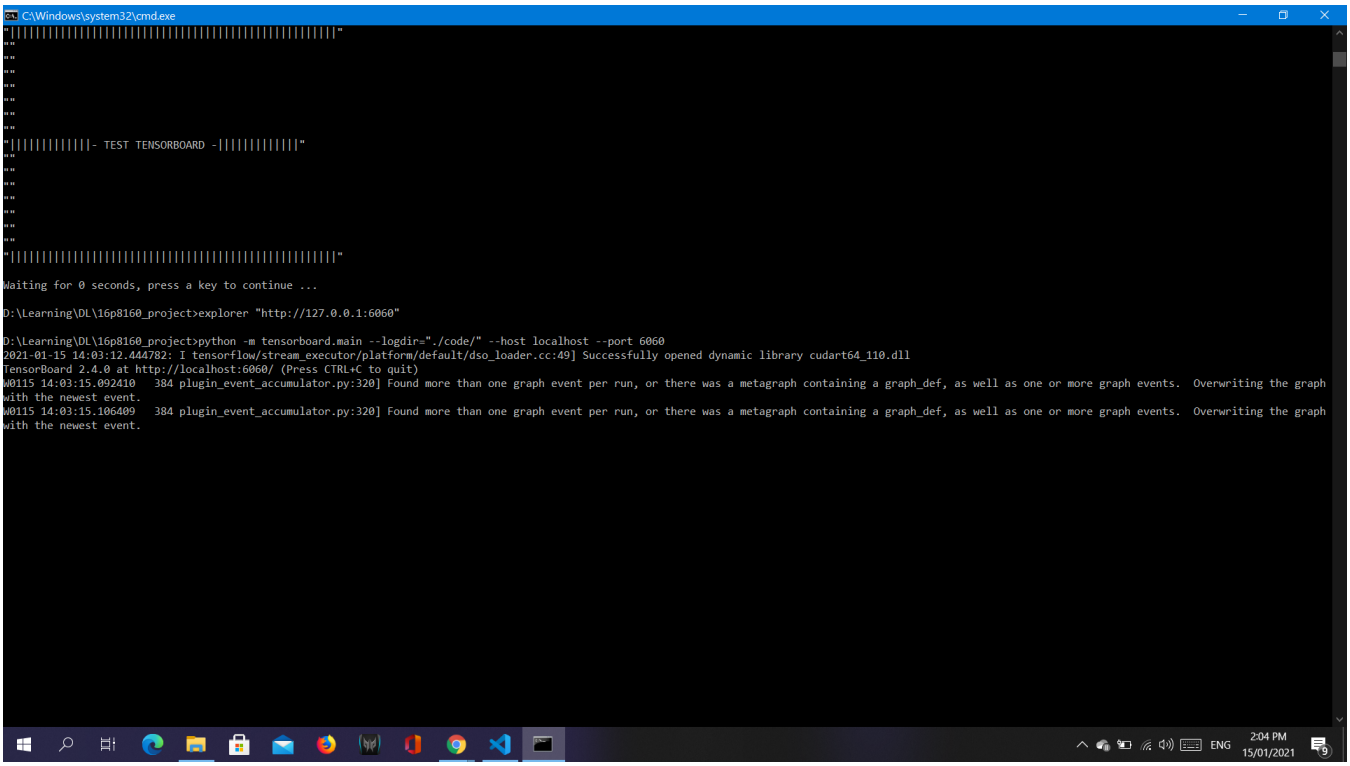

C:\Windows\system32\cmd.exe
Waiting for 0 seconds, press a key to continue ...
D:\Learning\DL\16p8160_project>python ./code/Train.py
Images Array Shape: (42000, 28, 28, 3) Labels Array Shape: (42000, 1)
Images Array ndim: 4 Labels Array ndim: 2
Images Array dtype: uint8 Labels Array dtype: uint8
Images Array dtype Name: uint8 Labels Array dtype Name: uint8
DATA LOADED FROM FILE SYSTEM
Epoch 1/10
322/322 [==============================] - 4s 7ms/step - loss: 0.4168 - accuracy: 0.8736 - val_loss: 0.0966 - val_accuracy: 0.9667
Epoch 2/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0684 - accuracy: 0.9797 - val_loss: 0.0712 - val_accuracy: 0.9798
Epoch 3/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0387 - accuracy: 0.9879 - val_loss: 0.0669 - val_accuracy: 0.9786
Epoch 4/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0263 - accuracy: 0.9926 - val_loss: 0.0730 - val_accuracy: 0.9774
Epoch 5/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0183 - accuracy: 0.9949 - val_loss: 0.0768 - val_accuracy: 0.9750
Epoch 6/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0107 - accuracy: 0.9971 - val_loss: 0.0756 - val_accuracy: 0.9857
Epoch 7/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0076 - accuracy: 0.9977 - val_loss: 0.0631 - val_accuracy: 0.9869
Epoch 8/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0066 - accuracy: 0.9981 - val_loss: 0.0651 - val_accuracy: 0.9833
Epoch 9/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0058 - accuracy: 0.9983 - val_loss: 0.0576 - val_accuracy: 0.9881
Epoch 10/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0031 - accuracy: 0.9991 - val_loss: 0.0657 - val_accuracy: 0.9845
Model: "sequential"
Layer (type) Output Shape Param #
resizing (Resizing) (None, 28, 28, 3) 0
conv2d (Conv2D) (None, 26, 26, 25) 700
max_pooling2d (MaxPooling2D) (None, 13, 13, 25) 0
flatten (Flatten) (None, 4225) 0
dense (Dense) (None, 250) 1056500
dense_1 (Dense) (None, 10) 2510
Total params: 1,059,710
Trainable params: 1,059,710
Non-trainable params: 0
Sample
Y: [2], Y_pred: [2]
2
ENG
2:02 PM
15/01/2021


C:\Windows\system32\cmd.exe
Waiting for 0 seconds, press a key to continue ...
D:\Learning\DL\16p8160_project>python ./code/Train.py
Images Array Shape: (42000, 28, 28, 3) Labels Array Shape: (42000, 1)
Images Array ndim: 4 Labels Array ndim: 2
Images Array dtype: uint8 Labels Array dtype: uint8
Images Array dtype Name: uint8 Labels Array dtype Name: uint8
DATA LOADED FROM FILE SYSTEM
Epoch 1/10
322/322 [==============================] - 4s 7ms/step - loss: 0.4168 - accuracy: 0.8736 - val_loss: 0.0966 - val_accuracy: 0.9667
Epoch 2/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0684 - accuracy: 0.9797 - val_loss: 0.0712 - val_accuracy: 0.9798
Epoch 3/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0387 - accuracy: 0.9879 - val_loss: 0.0669 - val_accuracy: 0.9786
Epoch 4/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0263 - accuracy: 0.9926 - val_loss: 0.0730 - val_accuracy: 0.9774
Epoch 5/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0183 - accuracy: 0.9949 - val_loss: 0.0768 - val_accuracy: 0.9750
Epoch 6/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0107 - accuracy: 0.9971 - val_loss: 0.0756 - val_accuracy: 0.9857
Epoch 7/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0076 - accuracy: 0.9977 - val_loss: 0.0631 - val_accuracy: 0.9869
Epoch 8/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0066 - accuracy: 0.9981 - val_loss: 0.0651 - val_accuracy: 0.9833
Epoch 9/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0058 - accuracy: 0.9983 - val_loss: 0.0576 - val_accuracy: 0.9881
Epoch 10/10
322/322 [==============================] - 1s 4ms/step - loss: 0.0031 - accuracy: 0.9991 - val_loss: 0.0657 - val_accuracy: 0.9845
Model: "sequential"
Layer (type) Output Shape Param #
resizing (Resizing) (None, 28, 28, 3) 0
conv2d (Conv2D) (None, 26, 26, 25) 700
max_pooling2d (MaxPooling2D) (None, 13, 13, 25) 0
flatten (Flatten) (None, 4225) 0
dense (Dense) (None, 250) 1056500
dense_1 (Dense) (None, 10) 2510
Total params: 1,059,710
Trainable params: 1,059,710
Non-trainable params: 0
Sample
Y: [7], Y_pred: [7]
7
ENG
2:02 PM
15/01/2021

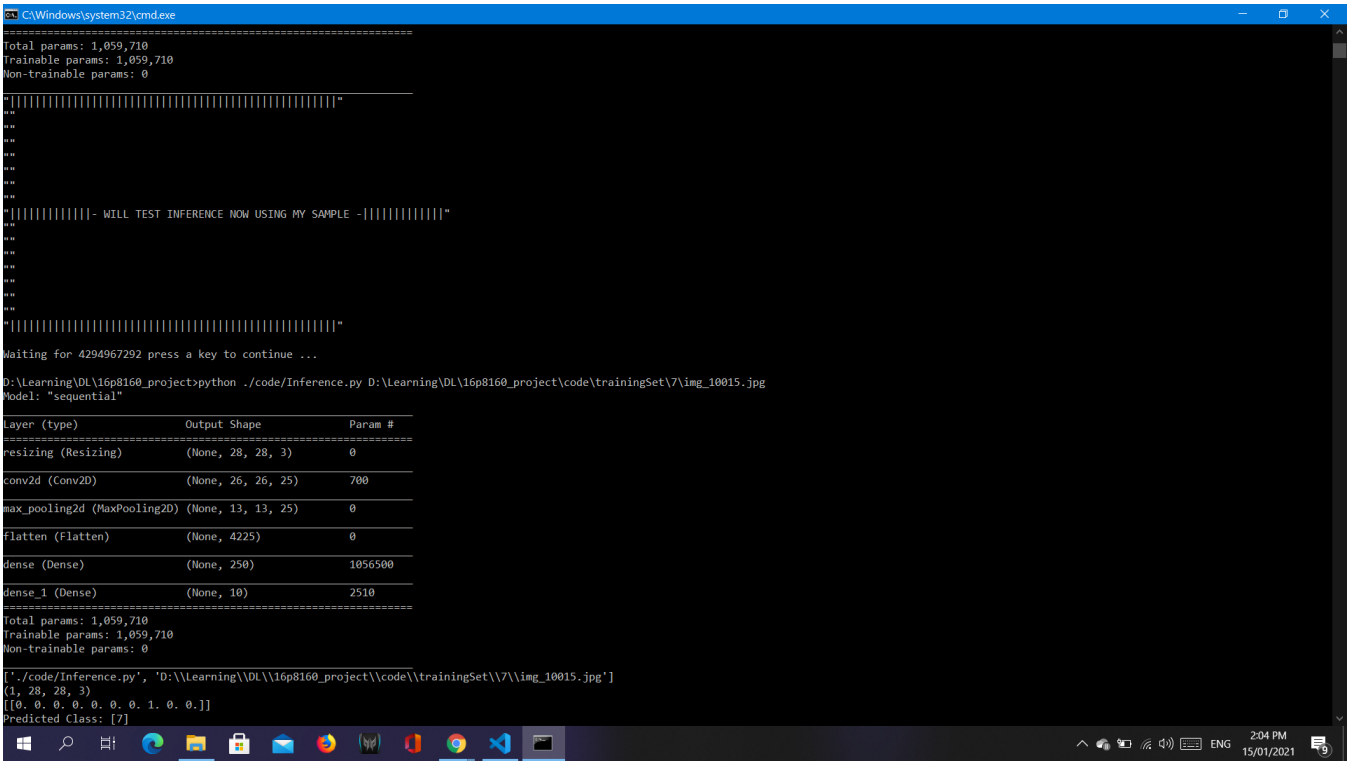


Windows Tensorboard





Windows Inference Using Preselected Sample



Windows Inference Using User-Input Image