B EE 525 Project #1 End – to – End ML-based Number Classifier
Due: 03/16/2025 11:59PM (Fixed Deadline)

(100 Points)

Q1. Write a code for an End –to – End number classifier using Raspberry Pi, Camera and 7-Segment Display. For this project you are expected to write the code in python for capturing an image through the camera, pre-processing it such that the ML model detects the image and sends it to 7-segment display.

a) Write a code to capture the image using the raspberry pi camera and preprocess the image. When you capture the image it is currently in the array [W, H] we need to convert the image to target size of [28, 28]

   a. We are using the cv2 library to load the image as a monochrome image in python. First install the cv2 library using the following script in the virtual-env

      - (tf-env) raspeberrypi@rasperrypi : pip install opencv

   b. Use the **imead()** function to load the image from a image_path

```
import cv2

path_to_image = r"path/to/image.jpg"

# Loading the grayscale image from a path to an array “image”
image = cv2.imread(path_to_image, cv2.IMREAD_GRAYSCALE)

# To print the shape of the image using shape() method
print(f” The shape of image array is {image.shape}”)

# The image shape is : (1944, 2592)
```

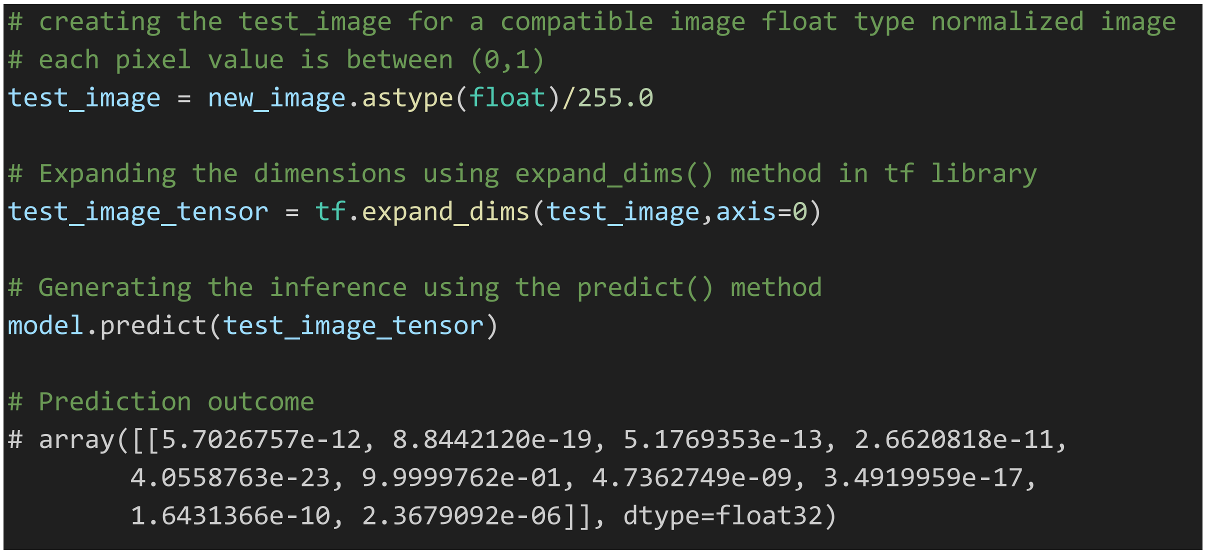
b) Option A: Preprocess the image using a resize() method. This method resizes the original image to a target shape

```
# Trying the resize() method
image_reshape = cv2.resize(image,(28,28))

# printing the outcome of reshape() method
print(f"The new image shape is : {image_reshape.shape}")

# The image shape is : (28, 28)
```

c) Option B: Preprocess the image using the slice technique. **Please make sure the cropped image that you are sending for inference generation has the target image within the crop**

```
# Trying the slice technique
# image[start_row:end_row, start_col:end_col]
image_cropped = image[50:78, 50:78]

# printing the outcome of reshape() method
print(f"The cropped image shape is : {image_cropped.shape}")

# The cropped image shape is : (28, 28)
```

d) Converting the new image to a compatible image shape for the CNN model and generating an inference using the predict() method.

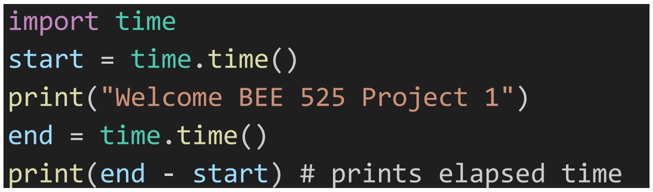


```
# creating the test_image for a compatible image float type normalized image
# each pixel value is between (0,1)
test_image = new_image.astype(float)/255.0

# Expanding the dimensions using expand_dims() method in tf library
test_image_tensor = tf.expand_dims(test_image,axis=0)

# Generating the inference using the predict() method
model.predict(test_image_tensor)

# Prediction outcome
# array([[5.7026757e-12, 8.8442120e-19, 5.1769353e-13, 2.6620818e-11,
        4.0558763e-23, 9.9999762e-01, 4.7362749e-09, 3.4919959e-17,
        1.6431366e-10, 2.3679092e-06]], dtype=float32)
```

e) Your code is expected to find which index has a maximum value, and display that using the 7-segment display.

f) After the execution, you need to print the time taken to execute the time taken to predict the outcome and display the image on the 7-segment display.

   a. You can use the following code to find the elapsed time between 2 points using the code below.



```
import time
start = time.time()
print("Welcome BEE 525 Project 1")
end = time.time()
print(end - start) # prints elapsed time
```

g) Write the code in such a way that prediction is done based on the input trigger from the keyboard. This objective is similar to the previous lab assignment, where the code is in a continuous loop state.

h) Please submit a report in PDF format based on the instructions below

   a. **Page limit: no more than 10 pages** including figures, tables, …
   b. A lab report should include:
   c. Abstract (e.g. a paragraph)
   d. List of Components used in the Design
   e. Flow chart for your code
   f. Code you have written with detailed comments
   g. Evaluate your implementation (e.g., processing speed)
   h. Conclusion
   i. State any problems to report or suggestions/hints
   j. References
   k. Video of the task at hand using 2 example scenarios