**CS3237 Introduction to IoT**

**Lab 4**

**ANSWER BOOK**

|  |  |
| --- | --- |
| Student ID:  A0242607J | Name:  Mitchell Kok Ming En |
| Student ID:  A0196650X | Name:  Jordan Yoong Jia En |

**Question 1 (2 MARKS)**

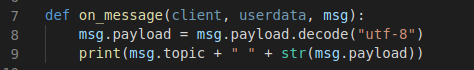
The ‘b’ means … it is present because …

‘b’ stands for ‘bytes’ - this means that what we are printing is the string decoding of a set of bytes.

**Question 2 (3 MARKS)**

The modified code is pasted below:

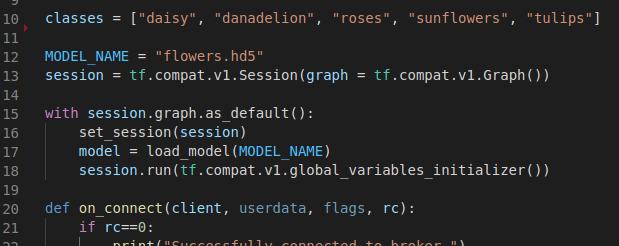
Extra line added to “on\_message” function to decode the string before printing



**Question 3 (2 marks)**

How I avoid long model load delays is:

We loaded the model as a global variable, so that it is loaded before the other functions run.



**Question 4 (2 mark)**

I ~~do~~/do not (cancel one) need to freeze weights because:

We are not training the model anymore – we are only using it to classify new inputs.

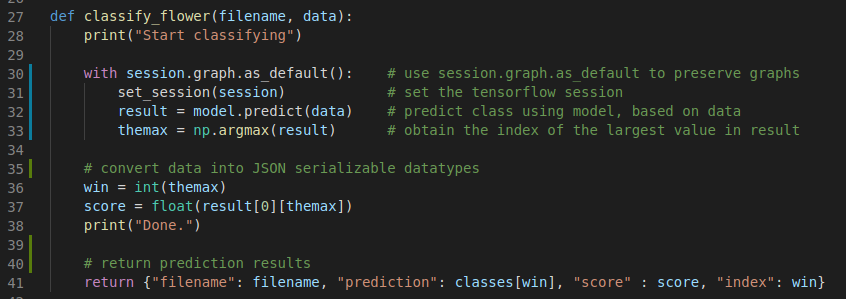
**Question 5 (4 marks)**

My predict function WITH explanation is:

The ‘classify\_flower’ function takes in filename and image data (Numpy array) as arguments when called from the ‘on\_message’ function.

Within the function, we set the Tensorflow session to the global session to preserve graph states correctly. We then call the ‘predict’ function from our pre-trained model to classify the image data. The largest value corresponds to the most probable class.

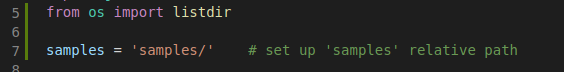
Finally, we convert the results into JSON serializable datatypes and return them as a dictionary to be published back to the sender.



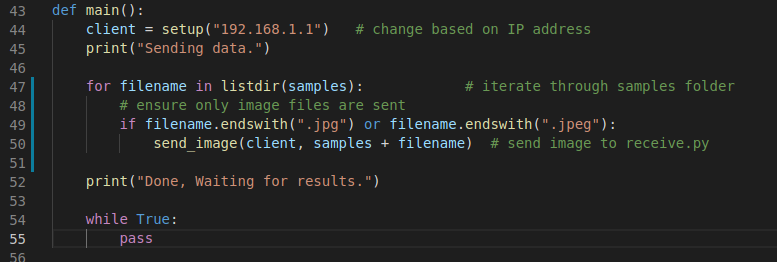
**Question 6 (4 marks)**

Relevant code with explanation:

First we import the ‘listdir’ function from the ‘os’ library so we can obtain a list of files. We also set up the relative path of the ‘samples’ directory.

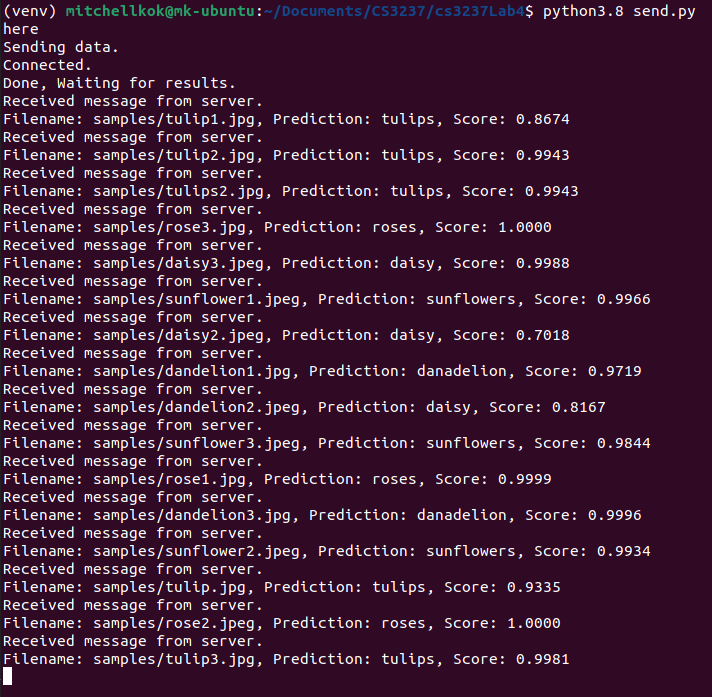


Next we setup the client and iterate through the list of files in the ‘samples’ directory. We determine that files are of ‘.jpg’ or ‘.jpeg’ format, and send them to the client if they are of the correct format.



**Question 7 (3 marks)**

My sample output:



The accuracy of my classifier is:

0.9531 on average

**TOTAL: \_\_\_\_\_\_\_ / 20**