Muhammad Ansari

COMP 469 Artificial Intelligence

March 3rd, 2018

Project #4: Image Classifier

**Step #4:**

Commands used for training:

py -m scripts.retrain \

--bottleneck\_dir=tf\_files/bottlenecks \

--model\_dir=tf\_files/models/ \

--summaries\_dir=tf\_files/training\_summaries/ "${ARCHITECTURE}"/project4step4 \

--output\_graph=tf\_files/retrained\_graph.pb \

--output\_labels=tf\_files/retrained\_labels.txt \

--architecture="${ARCHITECTURE}" \

--image\_dir=tf\_files/8rabbits

This command runs the scripts.py file, with the input parameters provided in the lines beneath it, that trains the image classifier with all of the default settings in tensorflow (no modifications to the number of learning steps, or learning rate).

Commands used for analysis (post-training):

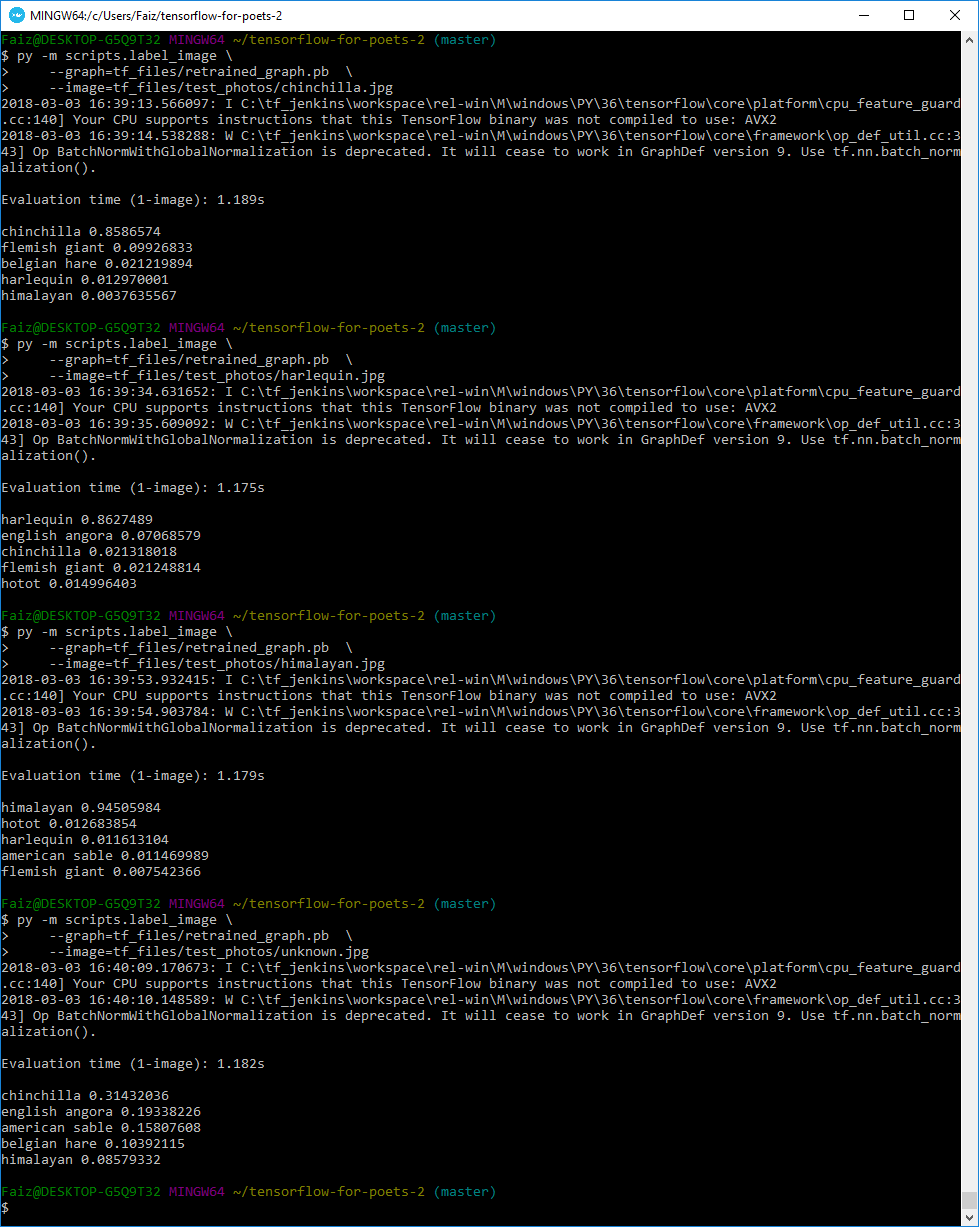
py -m scripts.label\_image \

--graph=tf\_files/retrained\_graph.pb \

--image=tf\_files/test\_photos/chinchilla.jpg

The only parameter that needed to be changed here was the ‘image’ parameter in order to analyze the different images I would provide to the classifier (chinchilla.jpg, harlequin.jpg, himalayan.jpg, unknown.jpg)

Results of step 4:



**Step #5:**

Commands used for training:

py -m scripts.retrain \

--bottleneck\_dir=tf\_files/bottlenecks \

--model\_dir=tf\_files/models/ \

--summaries\_dir=tf\_files/training\_summaries/ "${ARCHITECTURE}"/project4step5 \

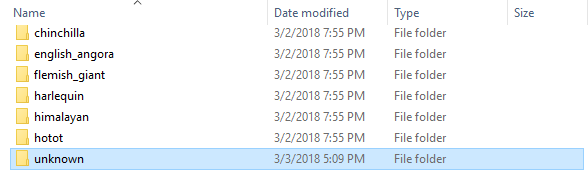
--output\_graph=tf\_files/retrained\_graph.pb \

--output\_labels=tf\_files/retrained\_labels.txt \

--architecture="${ARCHITECTURE}" \

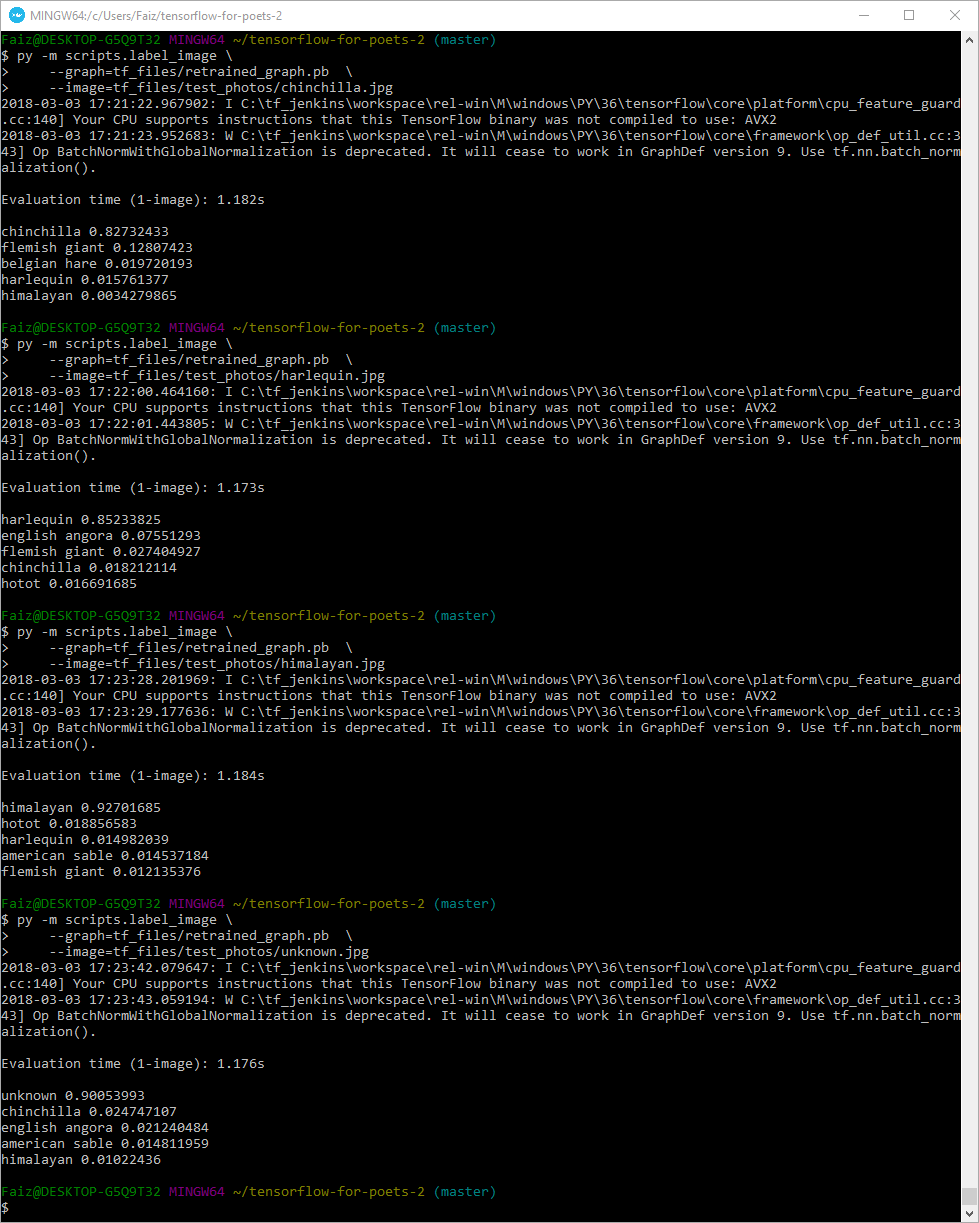
--image\_dir=tf\_files/8rabbits

The only difference in the commands here from step 4 is the summaries\_dir parameter which I just assigned to a new folder in order to store the training summary results for the updated training. I had to re-train the bot from step 4 because to fulfill the condition for step 5, I needed to essentially create a new category for the bot, called ‘unknown’, that would be used to label images that are clearly not images of rabbits. I did this by creating a new subdirectory within the ‘8rabbits’ folder called ‘unknown’ where I stored 323 images of various, random, non-rabbit, objects.



The commands I used for analysis were essentially the same as step 4. The retrained model was a success as the ‘unknown’ label successfully showed up when I inputted the unknown.jpg image to be classified.

Results of step 5:



**Step #6:**

Commands used for training:

py -m scripts.retrain \

--bottleneck\_dir=tf\_files/bottlenecks \

--model\_dir=tf\_files/models/ \

--summaries\_dir=tf\_files/training\_summaries/ "${ARCHITECTURE}"/project4step6b \

--output\_graph=tf\_files/retrained\_graph.pb \

--output\_labels=tf\_files/retrained\_labels.txt \

--learning\_rate=0.1 \

--architecture="${ARCHITECTURE}" \

--image\_dir=tf\_files/8rabbits

Like before, I changed the folder where the training summaries will be stored for the summaries\_dir parameter. But for this step, I also added a new parameter in order to complete the requirements of step 6. The ‘learning\_rate’ parameter which I added essentially controls the time allotted to each update to the final layer during the training of the classifier. I attempted this several times by providing different values for the ‘learning\_rate’ and found that by increasing the rate to 0.1 (from the default of 0.01), I was able to attain higher accuracies for the pictures of the rabbits as well as the unknown image. This does, however, contradict what was stated in the official documentation of TensorFlow For Poets where they claim that lessening the learning\_rate should ideally lead to higher accuracy predictions, versus increasing the rate which would lessen the accuracy. Despite this, my modifications resulted in higher accuracies for all of the images as compared to step 5. The new values can be seen in the output image below.

Results of step 6:

