‹ Return to "AI Programming with Python Nanodegree" in the classroom

Use a Pre-trained Image Classifier to Identify Dog Breeds

REVIEW | CODE REVIEW 6 | HISTORY

Meets Specifications

✨ Congratulations

Your submission was amazing. The efforts exerted are very appreciated and satisfies all the project requirements flawlessly. Please keep up the great work and keep learning and improving.

🍀 All my best wishes for your next project 🍀

Rate this project ☆☆☆☆☆

Timing Code

✓ **Student calls the time functions before the start of main code and after the main logic has been finished.**

You have taken the first step in measuring how the code is performing. Time is a great metric to see how the algorithms are performing.

There is a theoretical approach to measure how our program is working related to time, it's called Time Complexity. Follow these links for more information, it can be complex at the beginning but understanding these would help you down the line.
1 . https://en.wikipedia.org/wiki/Time_complexity
2 . https://www.geeksforgeeks.org/understanding-time-complexity-simple-examples/

Command Line arguments

✓ **adds command line argument for '--dir'**
uses default ='pet_images/'

Nice job using the command line arguments to pass various parameters to the code. That makes repeating a machine learning experiment in multiple settings much easier without touching the actual code.

✓ **adds command line argument for '--arch'**
default='vgg'

Experimenting with various architectures is as simple as one command that you pass during the execution.

✓ **adds command line argument for '--dogfile'**
default='dognames.txt'

Pet Image Labels

✓ **Makes sure files starting with '.' are ignored.**
Checks for '.' using a conditional statement.

✓ **Dictionary key and label are in the correct format and retrieves 40 key-value pairs.**
e.g:- {'Poodle_07956.jpg': ['poodle'], 'fox_squirrel_01.jpg': ['fox squirrel'] ... }

The labels dictionary is built correctly ✅ .

The values for keys are stored in a list instead of a string so that we can add more values to the list in the future.

✓ **'in_arg.dir' is passed as an argument inside check_images.py while calling the get_pet_labels function.**

Classifying Images

✓ **Appends images_dir to each value before making the function call.**

classifier(images_dir+users_key, model)

✓ **Convert the output to lower case and strip any whitespaces**

Converting to lowercase and truncating whitespaces is a common practice to maintain uniformity after reading strings.

✓ **Appends 1 to correct label, and 0 to falsely classified values**

Classifying Labels as Dogs

✓ **Check the displayed output and see if all matches are appropriately displayed.**

✓ **Check the displayed output and see if all non matches are appropriately displayed**

Results

✓ **All three models score as expected.**

Great job here, your results match the expected results.

Based on how the model is trained you might not see the same exact results if you rerun the model. However, your results would be close to the previous runs if you start to train the model every time.

DOWNLOAD PROJECT

6 CODE REVIEW COMMENTS ›

RETURN TO PATH