creating-dataset-1

April 2, 2024

0.1 Social Preformance Estimator // Creating the dataset... A walk-through

0.1.1 Author: Greta Perez-Haiek || As part of ML Engineering @ Rensselaer Polytechnic Institute

The purpose of this python file is to extract the feature vectors from the .jpg images extracted from my instagram profile.

In the Github, the "Images (jpg)" folder under the "Data" folder contains 185 .jpg images that needs extracting: if you would like to use your own images, feel free to replace the images in this file with your own with a numerical naming conjunction of "1.jpg, 2.jpg, 3.jpg ..."n".jpg" where "n" is the number of total images that you will train/test the model upon. The purpose of this python file is to extract the feature vectors from the .jpg images in "Images (jpg)".

Note: If you are interested in replacing the images with your own data, please make sure to also replace the data written in the "y_likes_data.txt" file with the number of likes corresponding to each image with the following conjunction:

1: [1.jpg's likes]

2: [2.jpg's likes]

...

"n": ["n".jpg's likes]

where "n" is the number of total images that you will train/test the model upon. Make sure that each conjunction is in it's own line in the .txt file! If you need an example, feel free to glance the "y_likes_data.txt" file that is currently present.

[22]:
```
'''Assuming that the "y_likes_data.txt" and "Images (jpg)" folder is␣
 ↪constructed correctly...
It's time to extract the features in each .jpg file, then store them in the␣
 ↪"X_feature_vector_data.txt" file!
This file, along with "y_likes_data.txt", will used in the "Training Model"␣
 ↪python file for machine learning purposes! '''

import matplotlib.pyplot as plt
import numpy as np
import cv2

#import "y" data
```

```
y = np.loadtxt("Data/y_likes_data.txt", comments='#', delimiter=":") #y[:, 0] =
 ↪jpg number, #y[:, 1] = likes!
X = [] #creates empty numpy array

#Extracts data from Images folder then appends to X variable
for i in range(y[:, 1].size):
    image = cv2.imread("Data/Images (jpg)/" + str(i+1) + ".jpg") #imports image
    target_size = (64, 64)
    image = cv2.resize(image, target_size) #resizes image
    image = np.array(image) #Converts image to an numpy array
    image = image.flatten() #converts numpy array to a 1D array
    X.append(image) #appends it to X
X = np.array(X) #converts X into numpy array
np.savetxt("Data/X_feature_vector_data.txt", X)#imports X variable into the
 ↪"X_feature_vector_data.txt" file
```

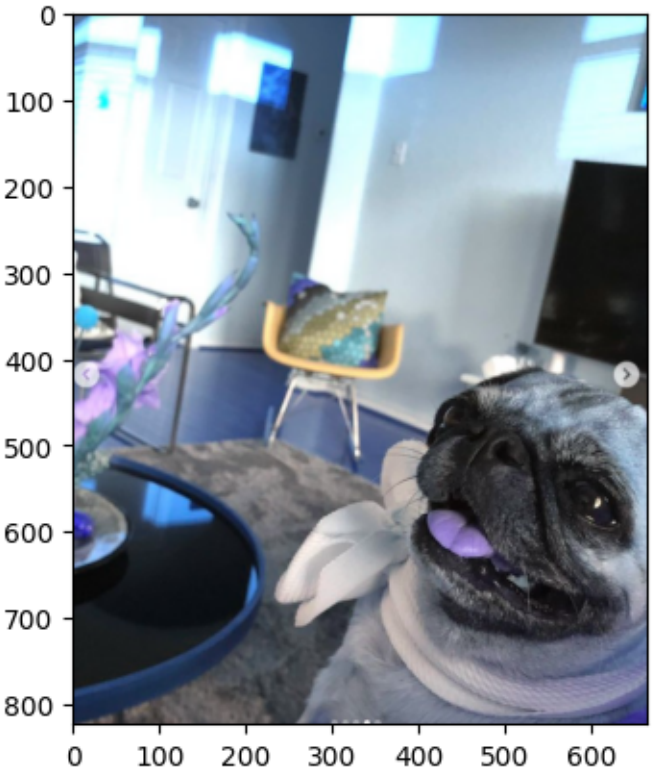
Run this python file (with the images in it's correct folder) to get y and X data! To help properly vizualize the data, consider the following code...

[23]:
```
print("Data Visualization!")
print()
print("The first couple of y values are...")
print(y[0, :], "likes")
print(y[1, :], "likes")
print(y[2, :], "likes")
print("...")
print(y[y[:, 1].size - 1, :], "likes")
print()
print("The 185th image looks like...")
image = cv2.imread("Data/Images (jpg)/185.jpg")
plt.imshow(image)
plt.show()
print()
print("The feature vector array of 185 is...")
image = np.array(image) #Converts image to an numpy array
image = image.flatten() #converts numpy array to a 1D array
print(image)
```

```
Data Visualization!

The first couple of y values are...
[ 1. 27.] likes
[ 2. 18.] likes
[ 3. 16.] likes
...
[185. 150.] likes

The 185th image looks like...
```




```
The feature vector array of 185 is…
[ 84 108 138 …  54  46 141]
```

Congrats! Now we're ready to design, train, and validate our future machine learning model!