

# Guided Project Report

## Hashing on face dataset

Name: Shruti Verma  
Course: AI and ML  
(Batch 4)  
Duration: 10 months

Problem Statement: Build a machine learning model using hashing on face dataset

### Prerequisites

What things you need to install the software and how to install them:

Python 3.8 or higher versions This setup requires that your machine has latest version of python. The following url <https://www.python.org/downloads/> can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/>. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/> You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.8 then run below commands in command prompt/terminal to install these packages `pip install -U scikit-learn` `pip install numpy` `pip install scipy` if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages `conda install -c scikit-learn` `conda install -c anaconda numpy` `conda install -c anaconda scipy`

### Dataset used

The data source is Yale face dataset. There are 11 images per subject, one for each of the following facial expressions or configurations:

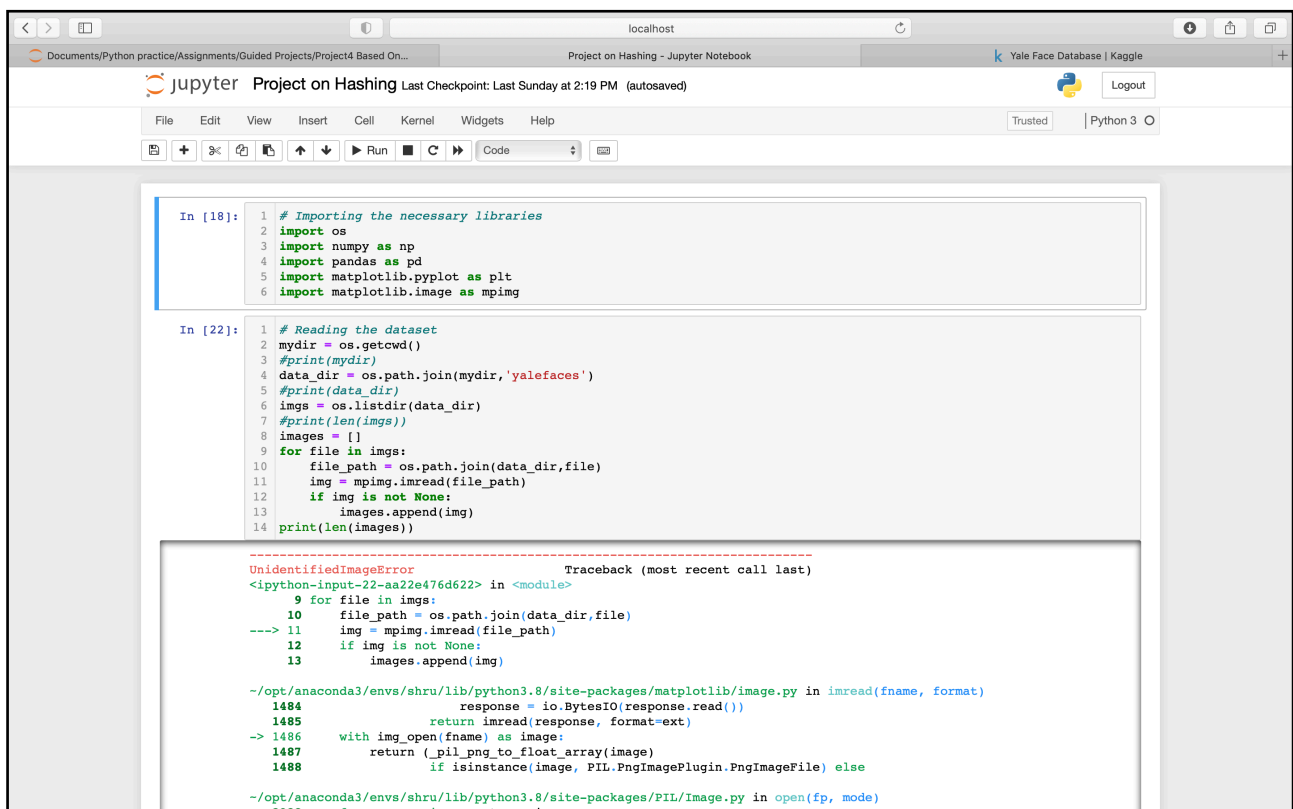
center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

## Method used for detection

Hashing

Reading data set > LSH > plotting images with same hash code

Importing the libraries and capturing images:



```
In [18]: 1 # Importing the necessary libraries
2 import os
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 import matplotlib.image as mpimg

In [22]: 1 # Reading the dataset
2 mydir = os.getcwd()
3 #print(mydir)
4 data_dir = os.path.join(mydir, 'yalefaces')
5 #print(data_dir)
6 imgs = os.listdir(data_dir)
7 #print(len(imgs))
8 images = []
9 for file in imgs:
10     file_path = os.path.join(data_dir, file)
11     img = mpimg.imread(file_path)
12     if img is not None:
13         images.append(img)
14 print(len(images))

-----
UnidentifiedImageError                                Traceback (most recent call last)
<ipython-input-22-aa22e476d622> in <module>
      9 for file in imgs:
     10     file_path = os.path.join(data_dir, file)
----> 11     img = mpimg.imread(file_path)
     12     if img is not None:
     13         images.append(img)

~/opt/anaconda3/envs/shru/lib/python3.8/site-packages/matplotlib/image.py in imread(fname, format)
    1484         response = io.BytesIO(response.read())
    1485         return imread(response, format=ext)
-> 1486     with img_open(fname) as image:
    1487         return (_pil_png_to_float_array(image)
    1488                 if isinstance(image, PIL.PngImagePlugin.PngImageFile) else

~/opt/anaconda3/envs/shru/lib/python3.8/site-packages/PIL/Image.py in open(fp, mode)
    2028         for message in accept_warnings:
```

Vectorising the images and storing in the list

```
localhost
Documents/Python practice/Assignments/Guided Projects/Project4 Based On... Project on Hashing - Jupyter Notebook Yale Face Database | Kaggle
jupyter Project on Hashing Last Checkpoint: Last Sunday at 2:19 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

12 if img is not None:
13     images.append(img)

~/opt/anaconda3/envs/shru/lib/python3.8/site-packages/matplotlib/image.py in imread(fname, format)
1484     response = io.BytesIO(response.read())
1485     return imread(response, format=ext)
-> 1486     with img_open(fname) as image:
1487         return _pil_png_to_float_array(image)
1488         if isinstance(image, PIL.PngImagePlugin.PngImageFile) else

~/opt/anaconda3/envs/shru/lib/python3.8/site-packages/PIL/Image.py in open(fp, mode)
2928     for message in accept_warnings:
2929         warnings.warn(message)

In [23]: 1 # Vectorizing the images and storing it in a list
2 imgs_vec = []
3 for image in images:
4     row,col = image.shape
5     img_vec = image.reshape(row*col)
6     img_vec_norm = img_vec/(np.linalg.norm(img_vec))
7     imgs_vec.append(img_vec_norm)
8     print(len(imgs_vec))
9     print(col*row)
10    print(img_vec.shape)

20
77760
(77760,)

In [24]: 1 # function to generate random unit vectors for hashing
2 def genRandomHashVec(m,length):
3     hash_vec = []
4     for i in range(m):
5         v = np.random.uniform(-1,1,length)
6         v_ = v/np.linalg.norm(v)
7         hash_vec.append(v_)
8     return hash_vec

In [25]: 1 # Function for Local Sensitive Hashing
2 def LSH(hash_vec,data_pt):
3     hash_code = []
4     for i in range(len(hash_vec)):
5         if np.dot(data_pt,hash_vec[i])>0:
6             hash_code.append('1')
7         else:
8             hash_code.append('0')
9     return hash_code
```

## Function for Local Sensitive Hashing

```
localhost
Documents/Python practice/Assignments/Guided Projects/Project4 Based On... Project on Hashing - Jupyter Notebook Yale Face Database | Kaggle
jupyter Project on Hashing Last Checkpoint: Last Sunday at 2:19 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

2 def genRandomHashVec(m,length):
3     hash_vec = []
4     for i in range(m):
5         v = np.random.uniform(-1,1,length)
6         v_ = v/np.linalg.norm(v)
7         hash_vec.append(v_)
8     return hash_vec

In [25]: 1 # Function for Local Sensitive Hashing
2 def LSH(hash_vec,data_pt):
3     hash_code = []
4     for i in range(len(hash_vec)):
5         if np.dot(data_pt,hash_vec[i])>0:
6             hash_code.append('1')
7         else:
8             hash_code.append('0')
9     return hash_code

In [26]: 1 # Generating 10 random vectors of the same size of the image vector
2 hash_vector = genRandomHashVec(10,len(imgs_vec[0]))
3 print(len(hash_vector))

10

In [27]: 1 # LSH function
2 LSH(hash_vector,imgs_vec[0])

Out[27]: ['0', '1', '0', '1', '0', '0', '0', '1', '1', '1']

In [31]: 1 # create image dictionary using the hash as the keys
2 image_dict = {}
3 for i in range(len(imgs_vec)):
4     hash_code = LSH(hash_vector,imgs_vec[i])
5     str_hash_code = ''.join(hash_code)
6     if str_hash_code not in image_dict.keys():
7         image_dict[str_hash_code] = [i]
8     else:
9         image_dict[str_hash_code].append(i)

In [32]: 1 # Displaying the hashes
2 cols_names = ['Hash codes','Image Index']
```

## Create image dictionary using the hash as the keys

```
localhost
Documents/Python practice/Assignments/Guided Projects/Project4 Based On... Project on Hashing - Jupyter Notebook Yale Face Database | Kaggle
jupyter Project on Hashing Last Checkpoint: Last Sunday at 2:19 PM (autosaved) Trusted Python 3

File Edit View Insert Cell Kernel Widgets Help

In [26]: 1 # Generating 10 random vectors of the same size of the image vector
2 hash_vector = genRandomHashVec(10, len(imgs_vec[0]))
3 print(len(hash_vector))
10

In [27]: 1 # LSH function
2 LSH(hash_vector, imgs_vec[0])
Out[27]: ['0', '1', '0', '1', '0', '0', '0', '1', '1', '1']

In [31]: 1 # create image dictionary using the hash as the keys
2 image_dict = {}
3 for i in range(len(imgs_vec)):
4     hash_code = LSH(hash_vector, imgs_vec[i])
5     str_hash_code = ''.join(hash_code)
6     if str_hash_code not in image_dict.keys():
7         image_dict[str_hash_code] = [i]
8     else:
9         image_dict[str_hash_code].append(i)

In [32]: 1 # Displaying the hashes
2 cols_names = ['Hash_codes', 'Image_Index']
3 df = pd.DataFrame(image_dict.items(), columns = cols_names)
4 df.head(60)
Out[32]:
Hash_codes Image_Index
0 0101000111 [0]
1 1100010111 [1]
2 1101110111 [2, 4, 7]
3 1001010111 [3, 15]
4 0101010111 [5, 18]
5 0001010111 [6]
6 0100010111 [8, 13]
7 1101010111 [9, 17, 19]
```

## Plotting the images with same hash code

```
localhost
Documents/Python practice/Assignments/Guided Projects/Project4 Based On... Project on Hashing - Jupyter Notebook Yale Face Database | Kaggle
jupyter Project on Hashing Last Checkpoint: Last Sunday at 2:19 PM (autosaved) Trusted Python 3

File Edit View Insert Cell Kernel Widgets Help

In [33]: 1 # Getting the keys and values of the Dictionary
2 keys = list(image_dict.keys())
3 values = list(image_dict.values())

In [34]: 1 # Plotting images with same hash code
2 igs = [imgs[i] for i in range(len(imgs)) if i in values[2]]
3 fig = plt.figure()
4 cols = 2
5 n_images = len(igs)
6 for n, image in zip(range(n_images), igs):
7     ax = fig.add_subplot(cols, np.ceil(n_images/float(cols)), n+1)
8     plt.gray()
9     plt.imshow(image)
10 fig.set_size_inches(np.array(fig.get_size_inches())*n_images)
11 plt.show()

<ipython-input-34-3c80e62a9c37>:7: MatplotlibDeprecationWarning: Passing non-integers as three-element position specification is deprecated since 3.3 and will be removed two minor releases later.
ax = fig.add_subplot(cols, np.ceil(n_images/float(cols)), n+1)

0
50
100
150
200
0 50 100 150 200 250 300

0
50
100
150
200
0 50 100 150 200 250 300
```

