**Name-Mahesh Desai**
**UBID- 50419649**

# PROJECT 3

Introduction:-

In this project there are two parts . In the first part we are detecting faces using a face detection algorithm and making bounding boxes around them and appending it to the result list.  In the 2nd part we are clustering the images which have the same features using the k-means algorithm.

Implementation:-

Part1:-

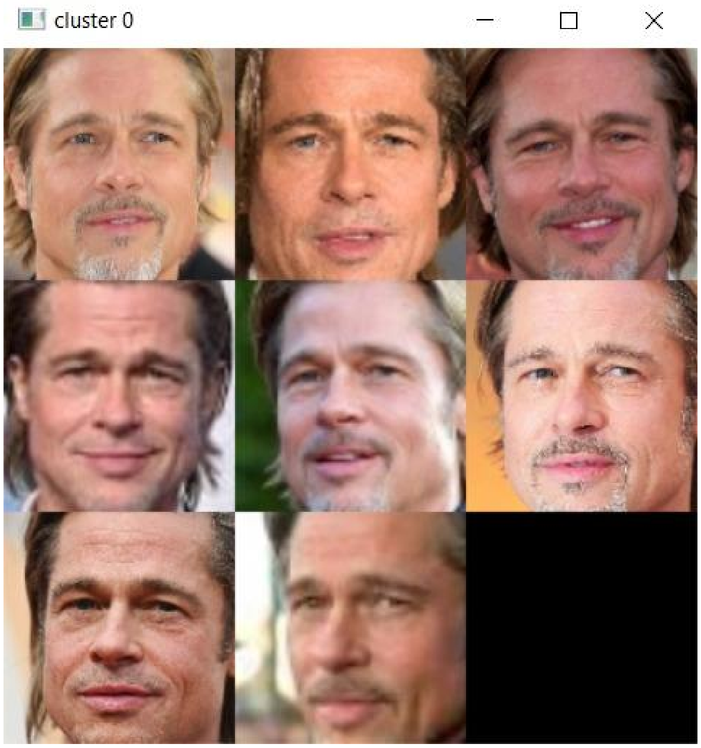
1) In this part we are implementing face detection on the given set of images. Here I am taking one image from the folder and reading it with the help of cv2.imread() , then I am converting that image from BGR to RGB color format.
2) Then I am using a face detection algorithm to detect faces in the images. Here I have tried various face detection libraries and then selected the library that gives the maximum result.
3) First I used cvlib face detection library to detect faces, which was detecting the faces properly but the bounding boxes were getting created near to the face. It was detecting the face not the entire head along with hair and ears. This gave an accuracy of only 15% .
4) Second, I used a face-recognition library. It was doing a bit better than the previous one and gave a result of 32%.
5) Lastly, Haar Cascade algorithm detected the faces properly and gave me an accuracy of 79.35%. I used the pre-trained '.xml' file for the face detection .
6) When the image was passed through the function, I got all the coordinates of the bounding box of the faces detected in the image.

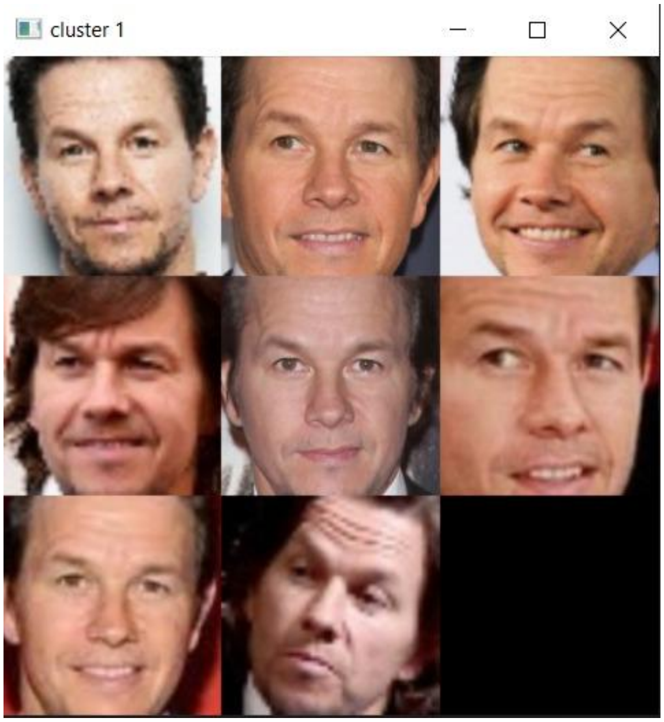
7) In the function I am passing an image , the scaling factor of the boxes and minimum neighbors to be considered while detecting faces.
8) Then after getting the coordinates of the bounding boxes, the image name along with the corresponding bounding box coordinate is put into the dictionary and then the dictionary is appended to the list. This entire list is returned and put into the json file.

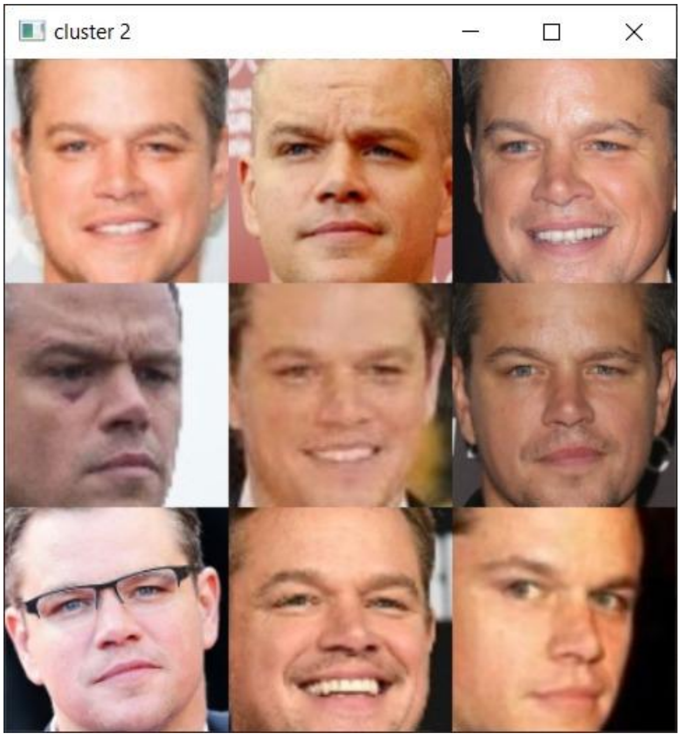
Part2:-

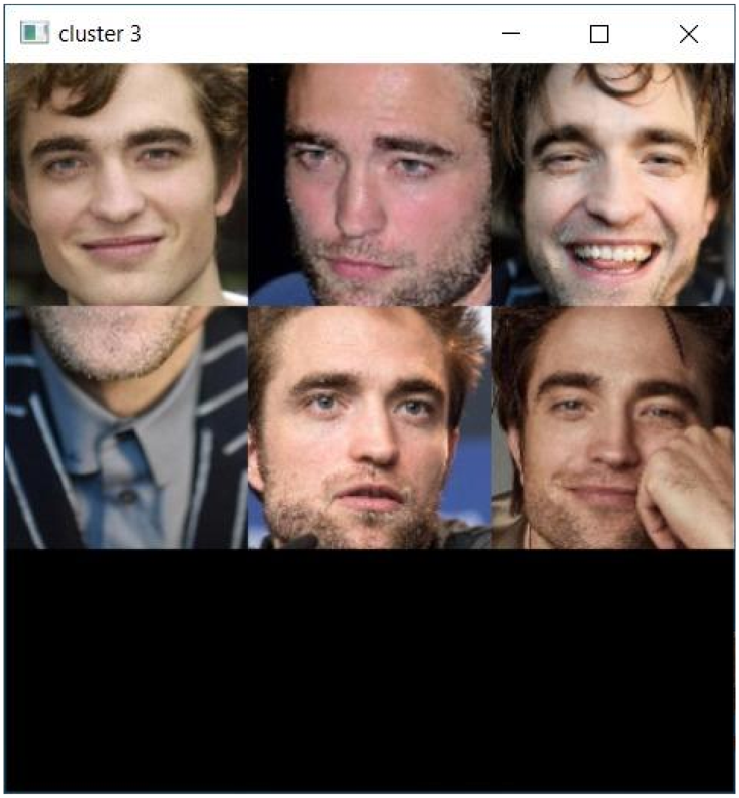
1) As in the 1st part we got the bounding boxes of the faces detected. Then with help of the coordinates that I have got from the face detection, I cropped that particular part of the image which has a face in it. This cropped image I am then sending to the encoding function.
2) After sending it to the encoding function , we are getting a 128 dimensional vector of the cropped image which has a face in it.
3) That vector is then appended in a dimensional list, which contains all the vectors for all the images in that folder.
4) After getting vectors of all the cropped images, we are sending the dimensional list to the k-means clustering algorithm. The Kmeans clustering algorithm, clusters the vectors on the basis of the similarity of the vectors and then gives us labels.
5) After getting the list of labels, I also took the list in which I had appended the image names from the folder.
6) First I iterated i upto K and then created a list to store images of same cluster. I already had created a list of labels which i got from k means and then used a second for loop to iterate labels list. If the value of i is equal to the values in the label list then that image is of a particular cluster and is appended to the image list.
7) Then the value of I and image list are put into the dictionary and is appended into the result list.

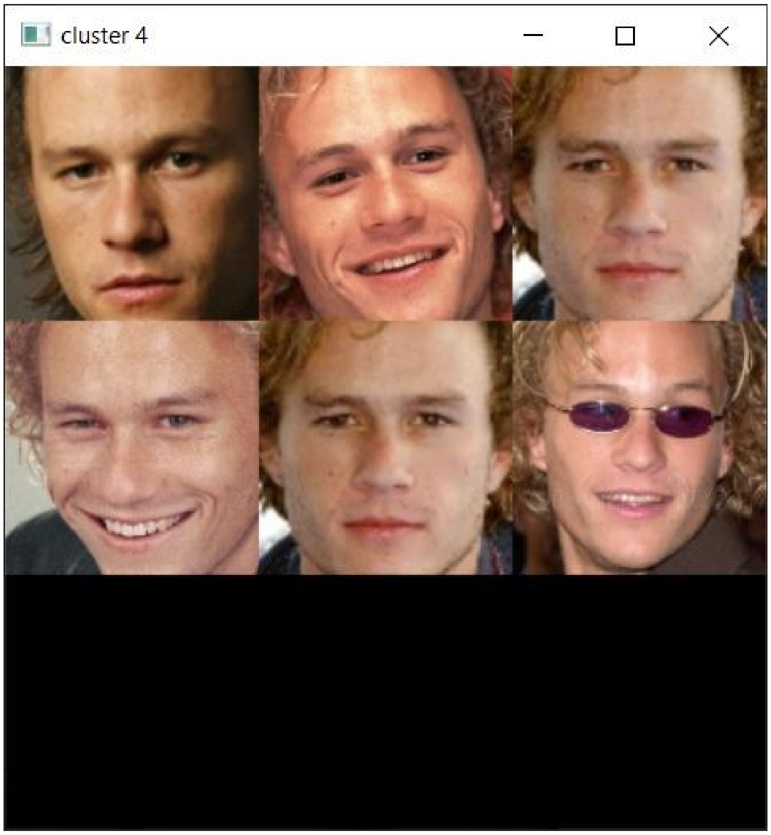
Images:




cluster 1


cluster 2


cluster 3


cluster 4

References:-

https://github.com/opencv/opencv/tree/master/data/haarcascades