CSE 473/573 - Computer Vision and Image Processing

Project 3

G. Laxmi Devi (50400730)

PART A

Algorithms tried and Ended up Using

- **HAAR Cascade Classifier:**
  Also known as the Viola/Jones Face Detector
  Key ideas:
  Rectangle features, Integral images for fast computation, Boosting for feature selection, and Attentional cascade for fast rejection of negative windows.
- This classifier is made of a series of stages wherein each stage weak learners are collected and then trained using boosting.
- **LBP Cascade Classifier (Local Binary Pattern):**
  In this classifier, the image is divided into blocks and each block considers its neighborhood pixels and updates the center pixel value with 1 if the neighborhood pixel value is greater or with 0 otherwise. Then these binary number is converted into a decimal number and histograms are plotted for each block to form a feature vector.
- HAAR classifier is computationally expensive than LBP but more accurate.
- Hence, I used the HAAR classifier for this project.

Results and Implementation challenges

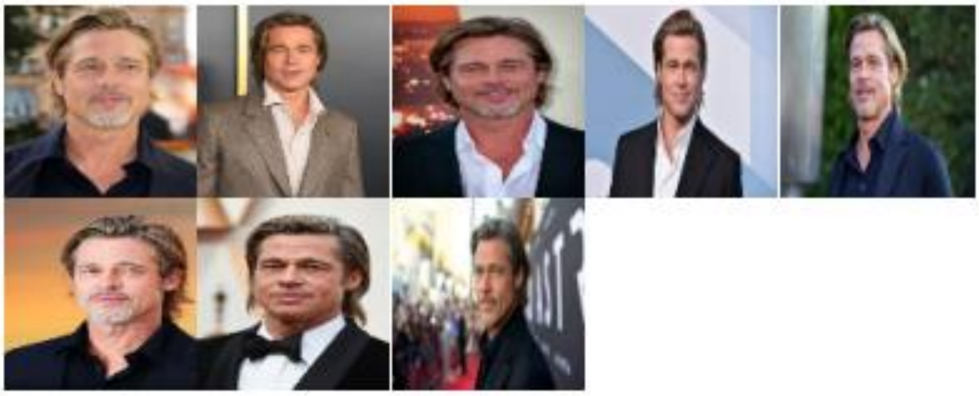
- With default scale factor and minimum neighbors lot of noise images have been detected.
- Tuning the parameters has decreased the number of noise images. Increasing a lot even the faces are not being detected.
- So, only a minimal increase in both the parameters has solved the problem.
- After tuning I have set the scale factor to 1.12 and minimum neighbors to 4.
- With the faces detected from the validation set the F1 score is **0.8184615384615385**

PART B

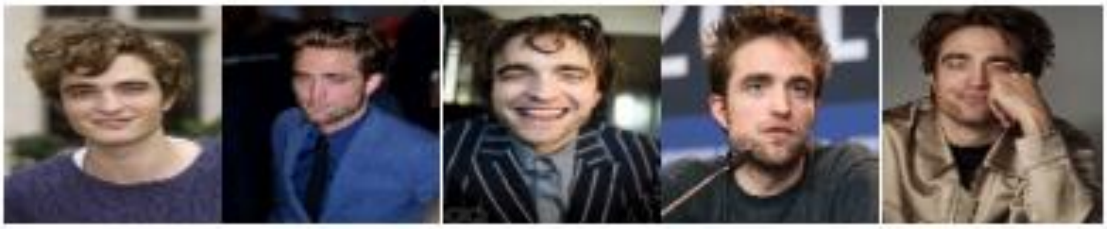
- **K-means**
  K-means clustering is the partitioning algorithm. It clusters all the objects even if the object is just noise.
- **DBSCAN**
  DBSCAN represents Density-Based Spatial Clustering. It discards objects defined as noise.

- **I have used DBSCAN for face clustering**.
- There are 36 faces in the given dataset. Using HAAR cascade for face detection it detected 39 faces out of which only 36 of them are faces and the rest are noise (facial features).
- Using K-means even the noise was allotted to a cluster, unlike DBSCAN.
- DBSCAN ignored these noise images and did not allocate them to any cluster.

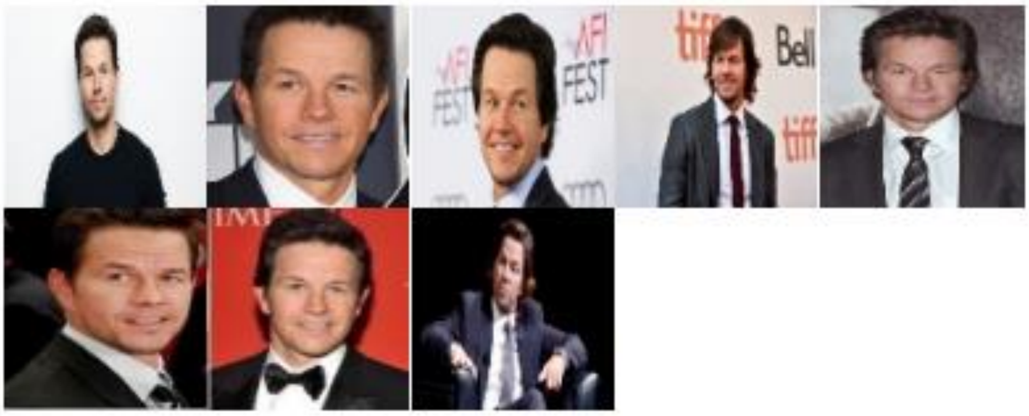
Cluster 0



Cluster 1



Cluster 2



Cluster 3

Cluster 4