UBID: 50417094
MRUNMAYEE VIJAY RANE

UNIVERSITY AT BUFFALO
CSE 573: COMPUTER VISION AND IMAGE PROCESSING
PROJECT 3

FACE DETECTION IN THE WILD

PART A: Face Detections

- Face Detections: Used Haar Cascade Classifier in OpenCV for face detection.
- Haar cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. The algorithm needs a lot of positive images i.e., images of faces and negative images i.e., images without faces to train the classifier. Then we need to extract features from it.

Face Detection was performed in following steps:

- Define Haar Cascade Classifier
- Found all jpg images in faceCluster_5 image folder and sorting them according to image file name numbers using glob and regex
- Read images in sorted order, Gray scaled them and used detect multiscale on Haar Cascade module to obtain x, y, width and height of detected faces.
- Converted detected face values in to numpy array and storing it in a list.
- Storing (x,y,w,h) values in a list of dictionaries.

Fbeta Score: 0.8113207547169811

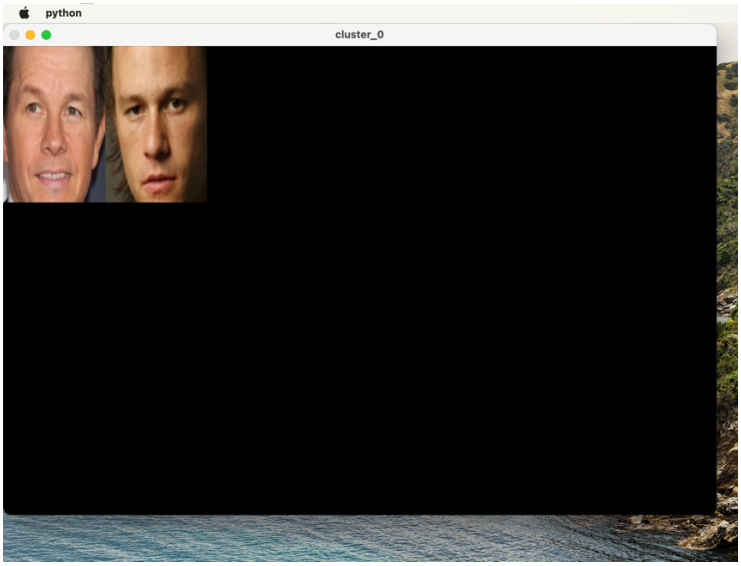
PART B: Face Clustering

- Face Detections: Used Haar Cascade Classifier in OpenCV and part A for face detection.
- The Kmeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields. ("K Means Clustering in Python. The KMeans algorithm clusters data by ...") Here, we use Kmeans algorithm to cluster the detected faces in the images.

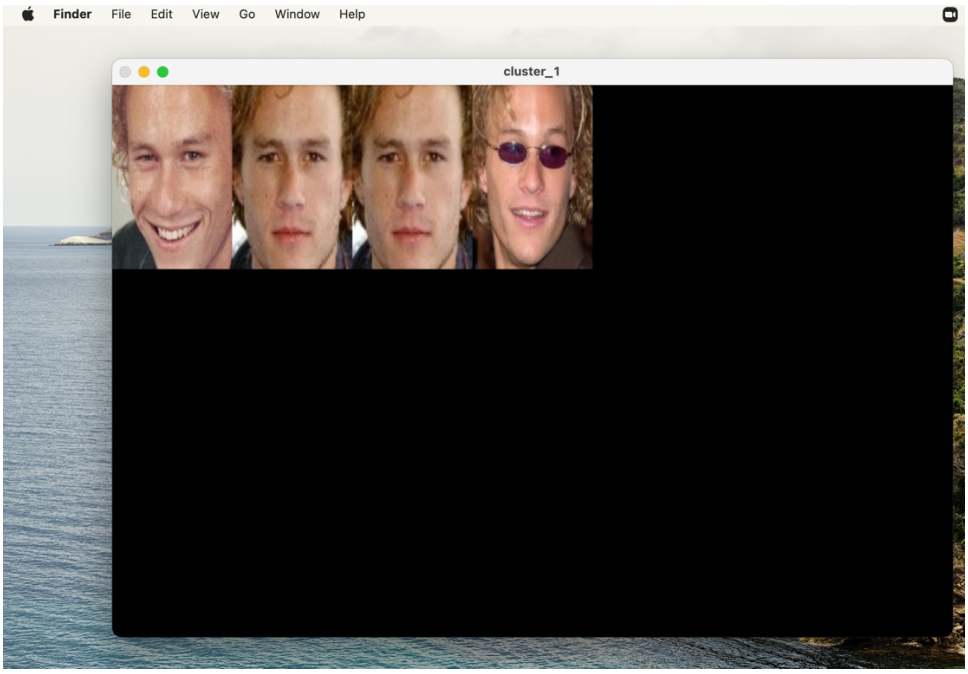
Face clustering was performed in following steps:

- Created separate crop images of the face detected and applying face_recognition module to get Face encodings on the crop images and there (x,y,w,h) values for obtaining features of the detected faces which is of an array of size 128.
- Reduced the dimension of encodings found on detected faces in order to fit training data for clustering.
- Applied K means cluster on face encodings obtained from crop images of detected faces using sklearns.
- Created a dictionary of values iname, bbox and cluster using the .labels_ function.
- Created dictionary of cluster number and its images in that cluster.
- Sorted the dictionary in order according to cluster number and storing it in a variable.
- Formatted sorted cluster and storing it in a list of dictionaries.

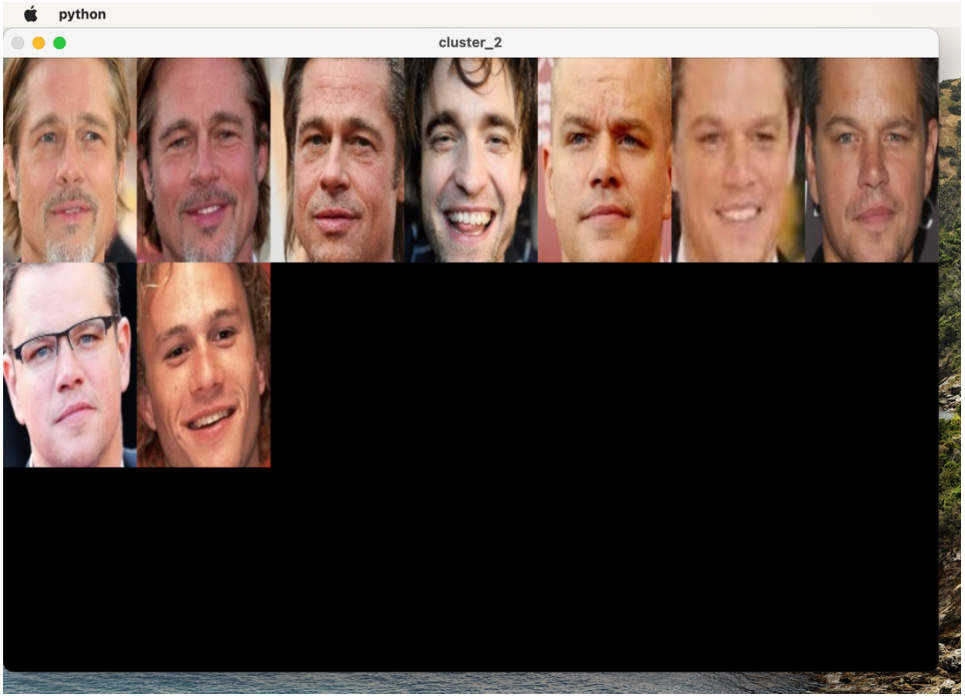
Face Cluster:
Cluster_0:



Cluster_1:

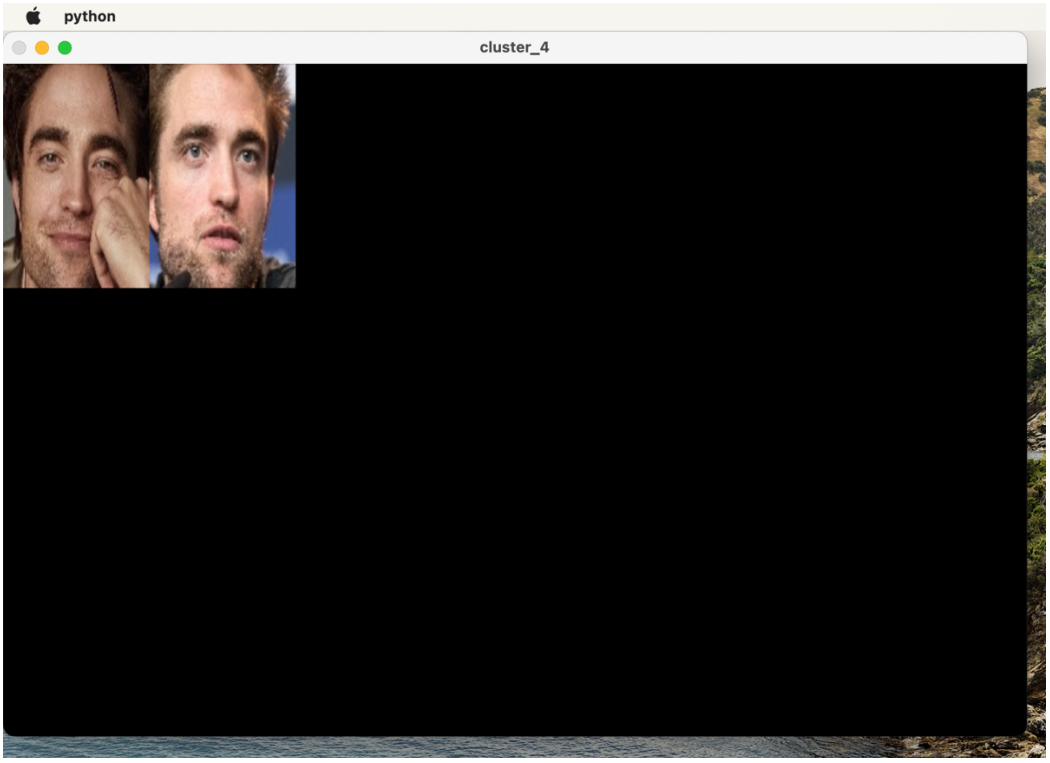


Cluster_2:



Cluster_3:

Cluster_4:



References:

1. https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf
2. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
3. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
4. https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html