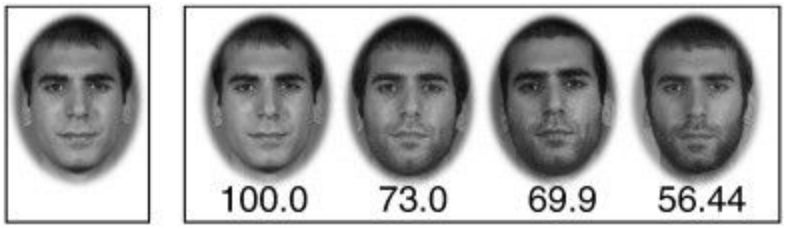
IMAGE PROCESSING PROJECT
-FACIAL HAIR DETECTION-

STUDENTS: PRATA MIRELA, SARATEAN TIMEEA
TEACHER: DIANA BORZA
GROUP: 30233

1. PROJECT DESCRIPTION

Facial hair analysis has recently received significant attention from forensic and biometric researchers due to three important observations as follows. Firstly, changing facial hairstyle can modify a person's appearance such that it effects facial recognition systems as illustrated in Fig. 1.



Therefore, detecting facial hair helps to distinguish male against female with high confidence in the gender classification problem.

The objective of this project is to implement an algorithm for face hair recognition. First, we take an image from the database and select multiple regions of the face such as cheekbones, because humans do not usually have hair there, beard, moustache, brows.

Then, we compute the Gabor histograms for these regions and compare them to find out if someone has facial hair. For preventing the case when the beard/moustache is incorrectly detected and segmented or the facial hair is not completely segmented from the facial image, an aggregately grouping will be implemented.

The dataset we chose is great for training and testing models for face detection, particularly for recognising facial attributes such as finding people with brown hair, are smiling, or wearing glasses. Images cover large pose variations, background clutter, diverse people, supported by a large quantity of images and rich annotations. This data was originally collected by researchers at MMLAB, The Chinese University of Hong Kong (specific reference in Acknowledgment section).

Overall

- 202,599 number of face images of various celebrities
- 10,177 unique identities, but names of identities are not given
- 40 binary attribute annotations per image
- 5 landmark locations

Data Files

- img_align_celeba.zip: All the face images, cropped and aligned
- list_eval_partition.csv: Recommended partitioning of images into training, validation, testing sets. Images 1-162770 are training, 162771-182637 are validation, 182638-202599 are testing
- list_bbox_celeba.csv: Bounding box information for each image. "x_1" and "y_1" represent the upper left point coordinate of bounding box. "width" and "height" represent the width and height of bounding box
- list_landmarks_align_celeba.csv: Image landmarks and their respective coordinates. There are 5 landmarks: left eye, right eye, nose, left mouth, right mouth
- list_attr_celeba.csv: Attribute labels for each image. There are 40 attributes. "1" represents positive while "-1" represents negative

2. BIBLIOGRAPHIC STUDY

2.1 BEARD AND MUSTACHE SEGMENTATION USING SPARSE CLASSIFIERS ON SELF-QUOTIENT IMAGES
[T. Hoang Ngan Le, Khoa Luu, Keshav Seshadri and Marios Savvides]

Another study that approaches the same problem first pre-processes images using the self-quotient algorithm and subsequently uses a sparse classifier to efficiently detect and segment regions containing facial hair. To the best of our knowledge, this is the first time a sparse classifier has been used for both detection and classification. We demonstrate the accuracy of beard and mustache segmentation by our method on images from the NIST Multiple Biometric Grand Challenge (MBGC) still face database and the color Facial Recognition Technology (FERET) database.

2.2 DETECTION AND ANALYSIS OF HAIR
[Yaser Yacoob and Larry Davis]

The authors describe an algorithm for automatic detection of hair. They assume that faces are in frontal view. The detection algorithm consists of the following steps:

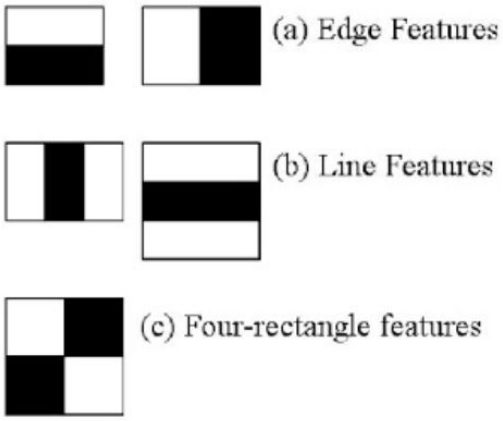
- Face detection: based on cascade of boosted classifiers

- Eye detection: cascade of boosted classifiers to train eye detectors to locate the eyes within a face region

3. PROPOSED METHOD

3.1 FACE DETECTION

Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.



Haar Cascade Detection:

- load the required XML classifiers for face
- load our input image (or video) in grayscale mode
- find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h). Once we get these locations, we can create a ROI for the face

3.2 SELECT REGION OF INTEREST

A region of interest is a region of the image where we are interested and the processing is done in that region. We can easily take the small portion of the image and do the processing in that part instead of processing the whole image.
Usually, a ROI is specified as a rectangle. In OpenCV, a rectangular ROI is specified using a rectstructure.

For this project, we choose the beard and the cheeks as regions of interest because we have to compare those two areas to see if a person has beard.

3.3 HISTOGRAM OF GABOR

Histogram of Gabor (HoG), is defined as the histogram of a 2D Gabor feature constructed using a Gabor filter.

Given a testing image, its corresponding HoG feature is computed in four steps as follows:

1. Apply the 2D Gabor filter at different frequencies and orientations to obtain a set of Gabor filtering images
   - ➔ For each image we computed 32 feature maps using convolution
2. Compute the histogram of each filtering image
   - ➔ For each feature map we computed the histogram
3. Concatenate the histograms of all Gabor filtering images
   - ➔ Then, we concatenated the 32 histograms in one large histogram of size 32*256
4. Normalize the set of histograms using the *l*2-norm normalization.
   - ➔ In the final step we normalized the histogram using l2 normalization
   - ➔ The l2-norm is calculated as the square root of the sum of the squared vector values.

We applied this algorithm for each area of interest: beard and cheeks and we obtained two normalized histograms.

3.4 COMPARE HISTOGRAMS

In the final step we compare the two histograms: beard histogram and cheeks histogram. To do this, we compute the distance between the two histograms using different methods. If the distance is close to 0, the person does not have beard. Otherwise, the person has beard.

The metrics we used to compare the two histograms:

1. **Correlation (CV_COMP_CORREL)**

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}}$$

where

$$\bar{H_k} = \frac{1}{N} \sum_J H_k(J)$$

and N is the total number of histogram bins.

2. Chi-Square (CV_COMP_CHISQR)

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

3. Intersection (method=CV_COMP_INTERSECT)

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

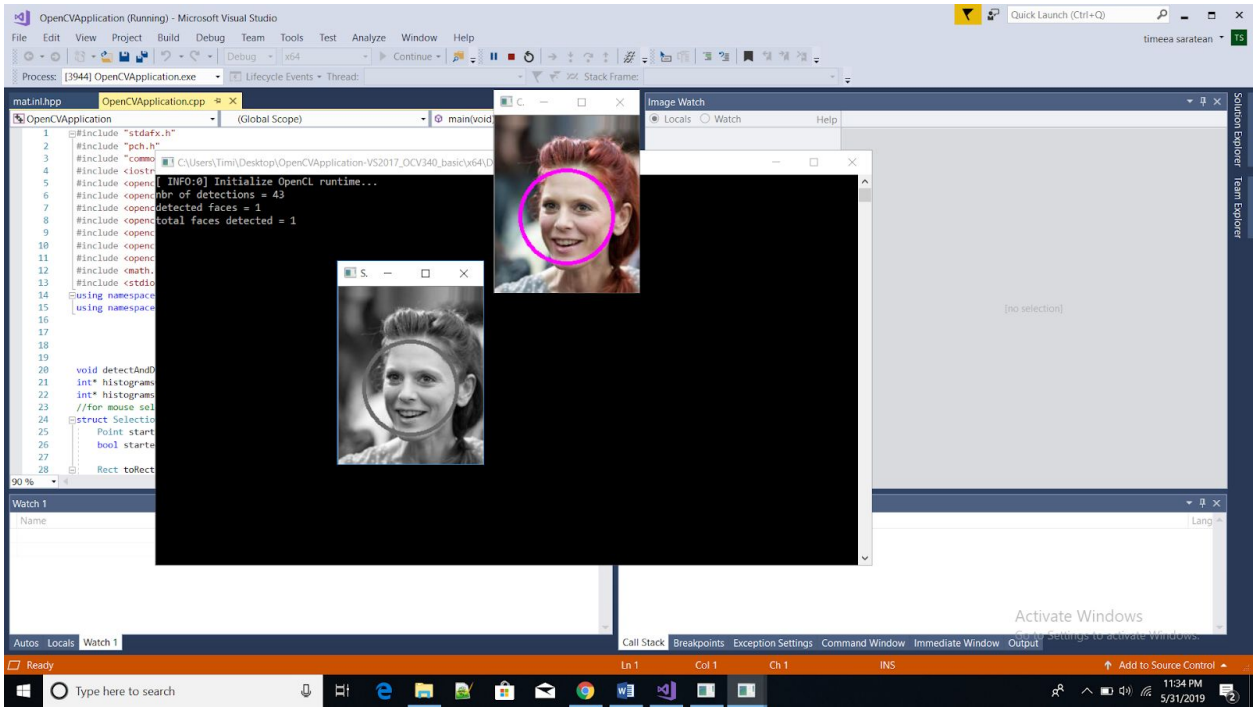
4. Bhattacharyya distance (CV_COMP_BHATTACHARYYA)

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H_1}\bar{H_2}N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

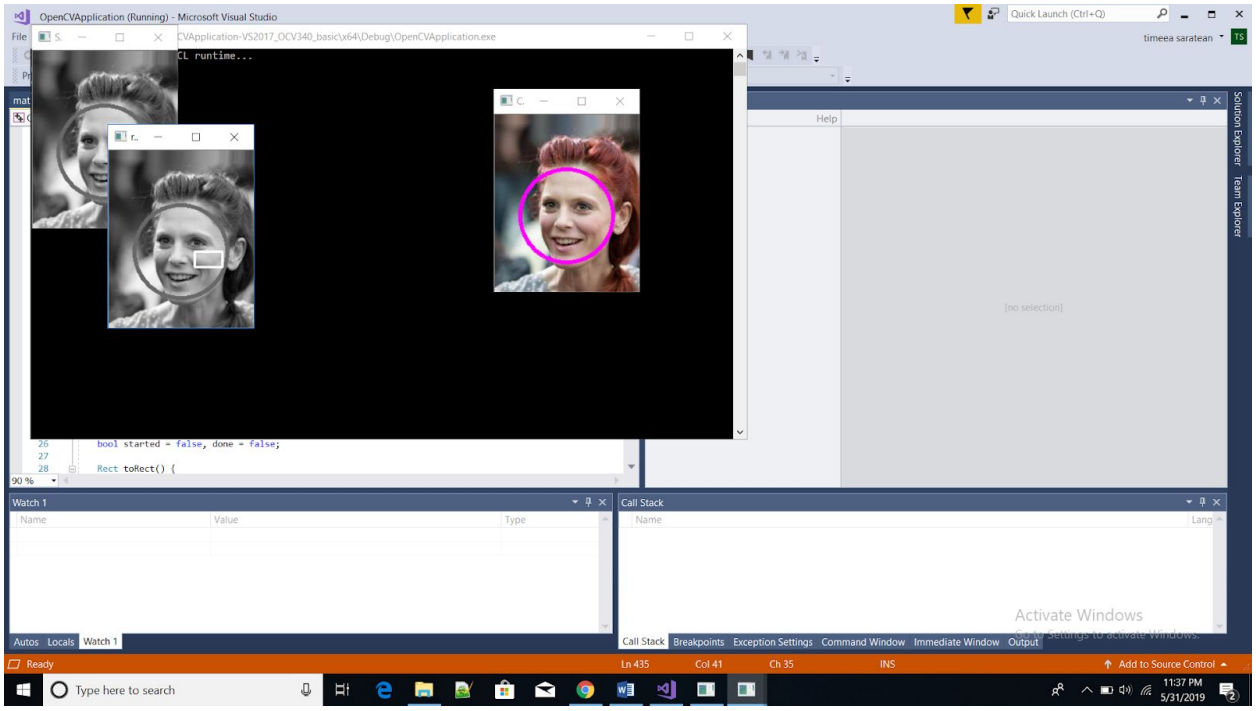
4. EXPERIMENTAL RESULTS

We conduct the experiments on a large database to evaluate our proposed self-detecting and segmenting facial hair algorithm. The images used for our experimental results are from the Celeb Dataset which contains 202,599 number of face images of various celebrities. This data was originally collected by researchers at MMLAB, The Chinese University of Hong Kong (specific reference in Acknowledgment section). Each database is divided into three categories: having facial hair (either moustache or beard), male without facial hair and female without facial hair.

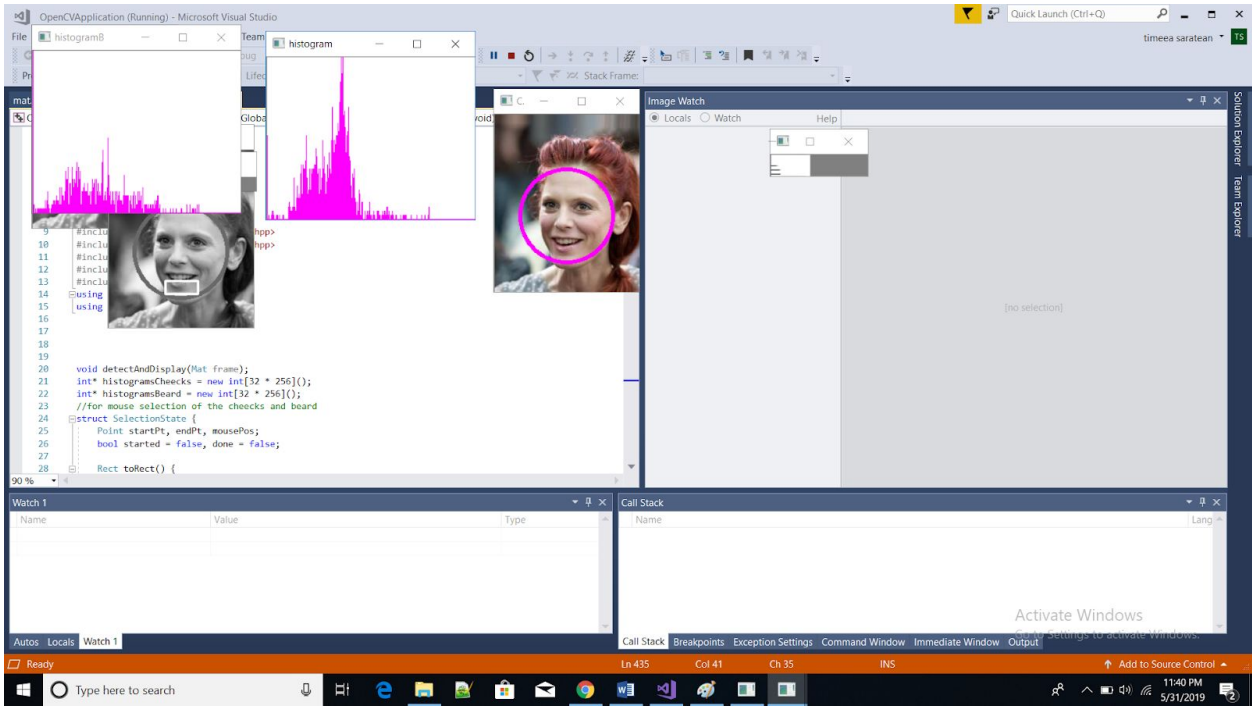
Face detection looks like this:



Then, we select our areas of interest by drawing a rectangle with the mouse on them:



After this step, the program computes the histograms for our selected areas: cheeks and beard:



5. CONCLUSION AND FURTHER DEVELOPMENTS

We have proposed a fast, robust and simple beard detection and segmentation algorithm. Our algorithm consists of four stages, namely, face detection, select regions of interest, histogram of Gabor and computing the distance. First, you select an image and the algorithm detects the faces in that image. After that, you select a cheek area and a beard area. The algorithm computes the distance from the histograms in that area and tells you if the person in that image has beard.

For further developments, we can make the algorithm to recognize all facial hair: moustache, eyebrows and beard. For that, we have to choose more than two regions of interest and compute the histogram of Gabor for each region.

BIBLIOGRAPHY

https://www.sciencedirect.com/science/article/pii/S0262885616301500

https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html

https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_comparison/histogram_comparison.html

https://www.kaggle.com/jessicali9530/celeba-dataset

http://users.utcluj.ro/~rdanescu/teaching_pi.html