



Haar Cascade Vs Facial Landmarks Technique for Face Recognition Uniqueness

Norazlin Ibrahim¹, Yusman Yusof², Muhammad Ghazali Bin Ithnin³

^{1,2}Industrial Automation Section, Universiti Kuala Lumpur Malaysia France Institute, Malaysia
{norazlin, yusman,}@unikl.edu.my,
Universiti Kuala Lumpur Malaysia France Institute, Malaysia
ghazali6013@gmail.com

ABSTRACT

This paper presents Haar Cascade vs Facial Landmarks Technique for Face Recognition Uniqueness that developed the Face detection module using ROI and Selective Features of Distances and angles between facial feature's centroid using Haar Cascade Technique and distance contour using Facial Landmarks technique. The feature's data was tested by the Neural Network module for verification purpose. When the Haar Cascade Technique is implemented, the angle is unreliable to be used as a feature. The System has achieved accuracy of 60% Haar cascade, 85% Face Landmark using the distance features.

Key words: Selective Features, Haar Cascade, Facial Landmarks.

1. INTRODUCTION

Facial recognition is one of the methods to identify a person's face with a single digital image or a video frame [1]. Face features is unique [2, 13]. Hence, it is possible to extract different facial features from different people through an image processing algorithm. Instead of Facial Recognition, text can also be used to know the emotion of an individual even in the event of writing [3].

In image processing, it is not an easy task to perform a system that identifies a person due to complexity of image data. In order to acquire the data, suitable features for example colour and texture [12] must be considered to enable data acquisition. Real time implementation is crucial when it comes to extracting features [1]. Issues such as screen freezes and window crashes may arise while executing the system. In this study, we are required to develop a good technique to acquire image data, the best method to identify unique features of different people and implement a real time execution of the system for a smooth performance [4].

1.1 Research Background

Portions of image which are highlighted or selected to perform operations on are referred as a Region of Interest (ROI). ROI is defined by creating a binary mask where the size of the image is the same with the binary image. The pixels that define the ROI are set to 1 and other pixels are all set to 0 when masking the image. ROI is not necessarily focused on one portion only; it can be defined more depending on the task requirement.

In Image Processing, the important aspect to differentiate the object is by using feature extraction. Since machine learning has its own language to interpret data, it also has its own vision to be able to view an object, to differentiate locations and characteristics of an object. Features that are considered in visual system are shape, colour and texture which will be extracted to illustrate the images. In terms of facial extraction, the face, eyes, nose, mouth and jaw are the common features that are extracted. dlib facial Landmark detection method extracts feature of human's face which are eyes, nose, eyebrows, mouth and jaw. (x, y) coordinate is associated with the facial landmarks and will be mapped to specific regions of the face [5].

Measuring distance between points in an image is an important factor when analysing images to be implemented in image processing and computer vision. When a measurement is done, a distance transform occurs where the binary data of image specifies from one pixel to the closest pixel available. Euclidean Distance and Manhattan Distance are among the commonly used methods to measure distance. [6]

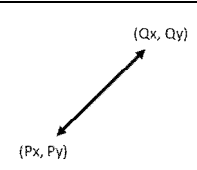
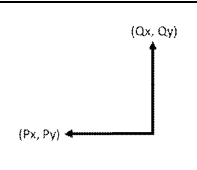
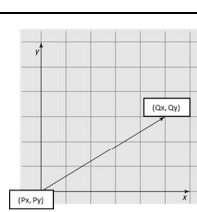
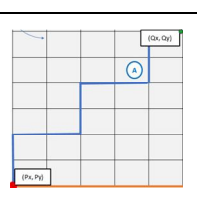
The Euclidean Distance is a linear vector between two points of pixel which is measured in Euclidean space. For an example, the distance point of P and Q are to be measured in an image. Their coordinates will be $P = (P_x, P_y)$ and $Q = (Q_x, Q_y)$. End points of the line segment for P and Q will create hypotenuse of a right-angle triangle. In order to define the distance between the two points, the square root of the sum of the squares of differences between these two coordinates are calculated [7]. Equation 1 shows the definition of distance between two points in Euclidean Geometry.

$$d(P,Q) = \sqrt{(Qx - Px)^2 + (Qy - Py)^2} \tag{1}$$

This distance measuring method follows strictly on horizontal and vertical axis. To compute diagonal distance, Pythagorean Theorem must be applied, which will become Euclidean distance method because Euclidean applies Pythagoras Theorem Equation to obtain its distance. If Manhattan distance method is applied on an image, the distance will be calculated following grid by grid Comparison between Euclidean and Manhattan Distance which can be seen in Table 1. The Equation of Manhattan Distance is shown in equation (2).

$$d(P,Q) = |Qx-Px|+|Qy-Py| \tag{2}$$

Table 1: Comparison of Euclidean vs Manhattan Distance

Method	Euclidean	Manhattan
Direction of Measurement		
Distance Measured in Cartesian Plane		

The comparison shows different paths taken by two different methods to define the distance of the 1st point and the 2nd point. The Euclidean method takes linear path on the Cartesian plane while Manhattan method strictly follows the grid line of the Cartesian plane. From this observation, Euclidean method are used to calculate distance between facial features.

2. METHODOLOGY

The face detection system is a part of the Face Recognition System module. The system will capture and acquire images from live video by using a webcam. Then the images will be processed by Python using OpenCV library. The Python program will execute a window that has ROI box for the users to position their face at the right point in front the camera. The user will press a button to capture his or her image. Then Haar cascades will detect the facial features on the image inside the ROI box. The distance between facial features is computed which is used as facial feature’s data. After that, the extracted data will be sent to the next module which is Neural Network for training and validation. Figure 1 highlights the module of Face Detection in this project. The face detection is implemented using two different techniques which are Haar

Cascade and dlib Facial Landmark [6].

2.1 Setting up ROI

The Bounding Box is formed with the implementation of the calculations in Figure 2 to get the coordinate (x1, y1) and (x2, y2). These two coordinates are used to form Bounding Box [7].

2.2 Face Detection using Haar Cascades Technique Implementation

Transparency was set up before the facial function loop. The transparency overlay’s purpose is to remove the Bounding Box that is formed by Haar Cascades [8]. The rectangle formed became transparent after the implementation of the overlay.

The centroid for eyes is calculated with the purpose to find the centroid in order to figure out the exact coordinate of a certain facial feature. Referring to Figure 3, the lines are drawn between centroids for each facial feature to verify the distance to be measured based on legend below

- v1 = Distance between Centroid of Right Eye to Left Eye
- v2 = Distance between Centroid of Right Eye to Nose
- v3 = Distance between Centroid of Left Eye to Nose
- v4 = Distance between Centroid of Nose to Mouth
- v5 = Distance between Centroid of Right Eye to Mouth
- v6 = Distance between Centroid of Left Eye to Mouth
- a = angles of each side of triangles formed by all 6 vectors

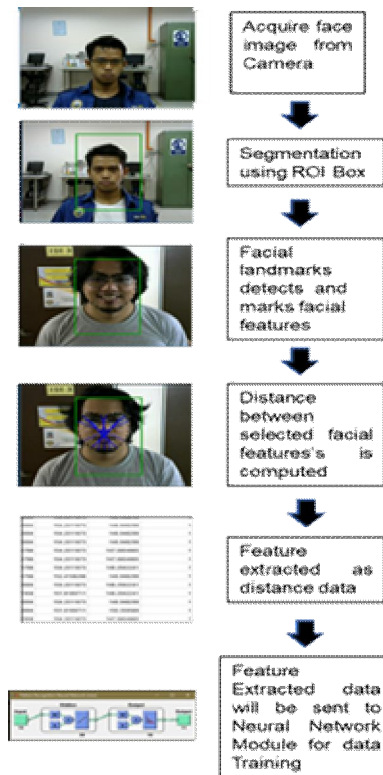


Figure 1 :Workflow of the face detection module

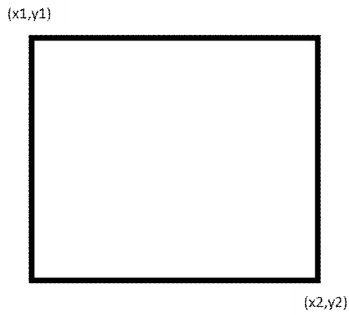


Figure 2 :Positioning ROI

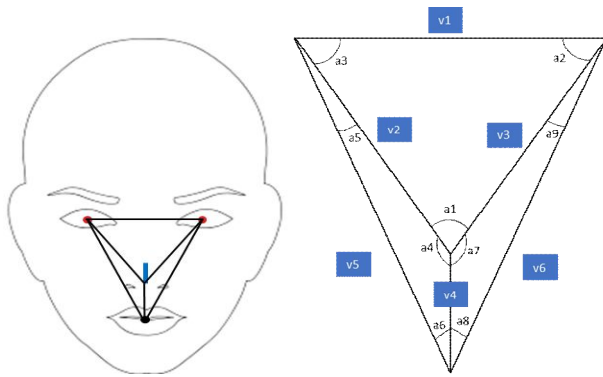


Figure 3: The distances and angles computed

To find the distance between facial feature’s centroids’ coordinate can be referred to equation 3 R_x, R_y amd L_x, L_y refer to left and Right. Angles are calculated following the Law of Cosine formula [7] in equation 3 and 4.

$$v1 = \sqrt{(ceRx - ceLx)^2 + (ceRy - ceLy)^2} \tag{3}$$

$$Angle\ v1 = \frac{(v2)^2 + (v3^2) - (v1)^2}{2(v2)(v3)} \tag{4}$$

2.3 Face Detection using dlib Facial Landmark Technique Implementation

There are two functions that are resized for smoother animation and function shapes array [9]. Figure 4 is the computation to find distance on selected points from facial landmark to be used as feature which are represented as v1 to v9. Table 2 is the range of points that are referred to parts of face.

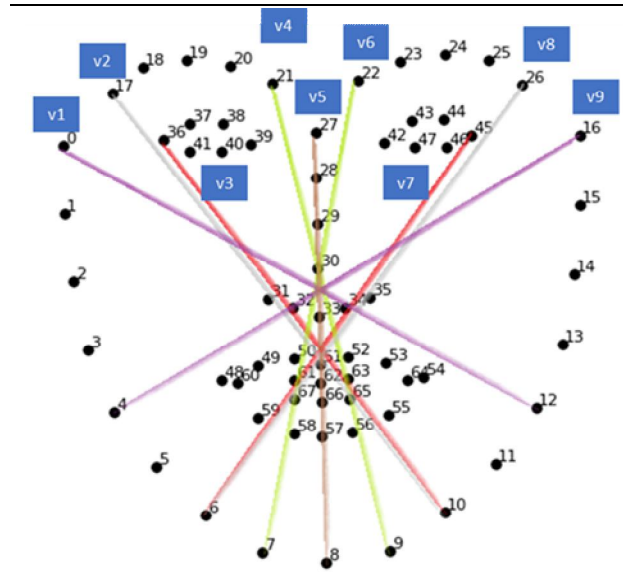


Figure 4: Drawn lines and calculated distances

Table 2: Range of points referred to parts of face

Points (0-16) = Jaw	Points (31-35) = Lower Nose
Points (17-21) = Right Eye Brow	Points (36-41) = Right Eye
Points (22-26) = Left Eye Brow	Points (42-45) = Left Eye
Points (27-30) = Upper Nose	Points (46-51) = Mouth

The calculation for Facial Landmarks technique shown in equation 3

$$v1 = \sqrt{((p1x2 - p1x1)^2) + ((p1y2 - p1y1)^2)} \tag{3}$$

2.4 Implementation of Saving and Transferring Feature’s Data

The features of distances and angles are saved to csv file to be transferred to Neural Network Module, trained, and tested.

2.5 Socket Programming

Socket Programming uses the UDP (User Datagram Protocol) in order to setup the Neural Network module’s on PC’s IP (Internet Protocol) address and port to enable the data to be sent. Another setup is done to receive feedback from Graphic User Interface (GUI). When a face is detected, its facial

feature data are sent to the Neural Network module to test and compare the newly acquired data with the pre-trained data set. This is the part where the face recognition occurs. When the Neural Network verifies the data, it will send the data to Graphic Use Interface (GUI) module and the GUI will send back the data to the socket informing the access is granted.

3. RESULTS AND DISCUSSION

3.1 Face Detection Implementation

Figure 5 shows summary of the results of implementation of two different techniques. From image acquisition, ROI Segmentation, Face Detection to Neural Network Classification.

In this module, the face detected condition requires the user's face to be in the ROI to control the environment. Figure 6 shows the condition where the face is partially out of the ROI box disabling the face detection for Haar Cascade Technique. The triangular lines did not form at this condition even though the left eye of the person below is detected. The Facial Landmark Technique is implemented with the same ROI.

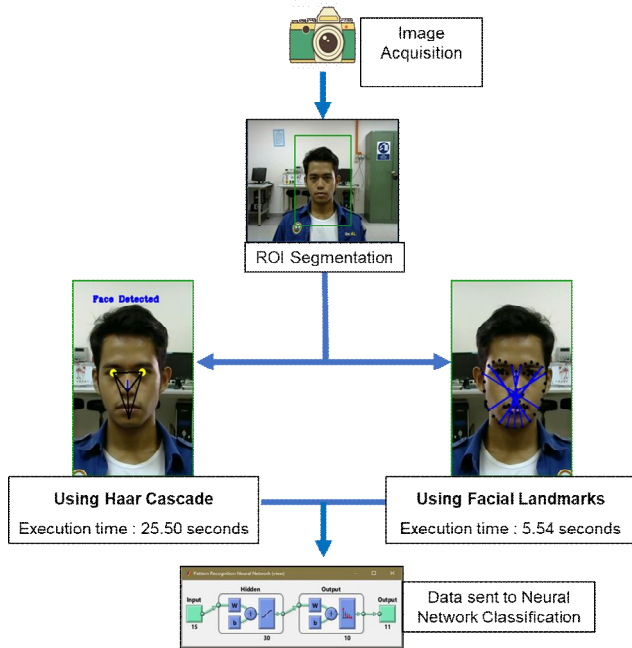


Figure 5: Implementation of two technique

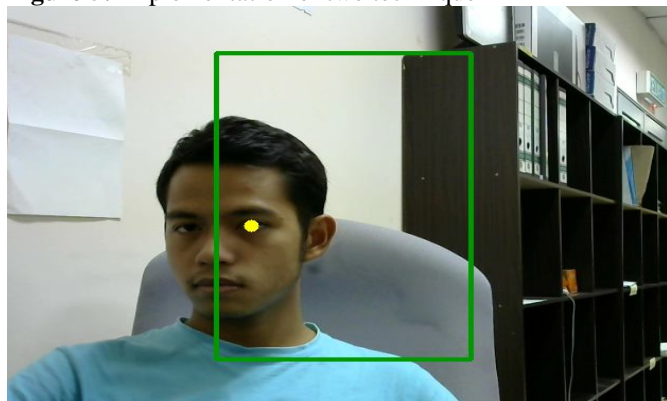


Figure 6 :Face not detected outside ROI box

1) Haar Cascade Technique's Result

Figure 7 shows the condition where all the facial features are inside the ROI box thus triggers the Face Detection function. The triangular lines are drawn according to the feature's distance data and extracted for training. The green box at the top left corner of the frame shows the value of each facial feature.

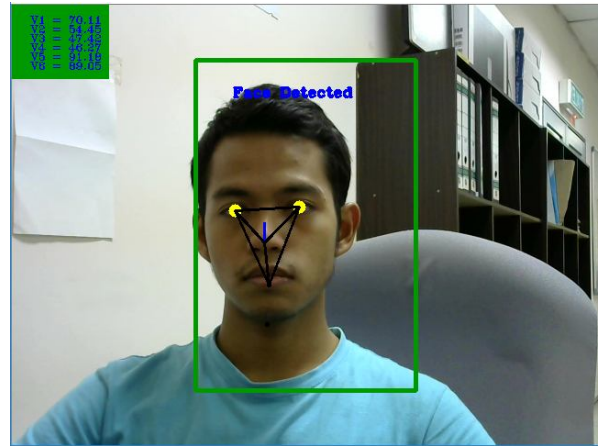


Figure 7: Face detection using Haar Cascade

Figure 8 shows the execution of face detection using Facial Landmark technique. The circles are shaped according to the lining of the face feature referring to the 68 points of (x,y) coordinates from dlib library as mentioned

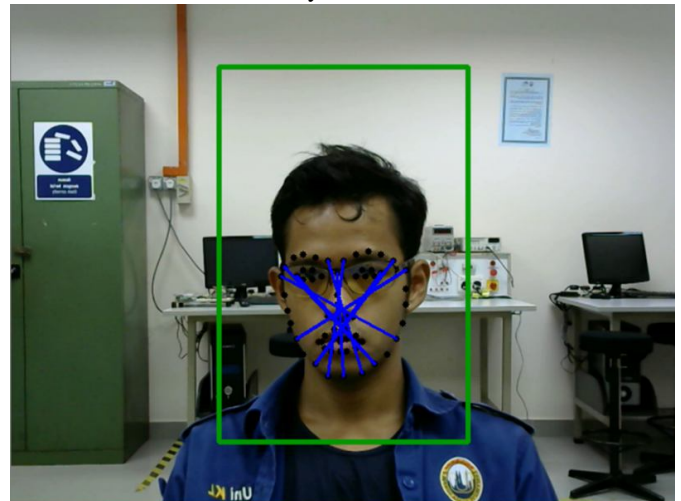


Figure 8:Face detection using Facial Landmark technique

2) Socket Programming Result

In Figure 9, the sequence event occurs from the implementation of socket programming. When a face is detected, its facial feature data are sent to the Neural Network module to be tested and the newly acquired data are compared with the pre-trained data set where this is the part where the face recognition occurs. Once the Neural Network verifies the data, it will send the data to Graphic Use Interface (GUI)

module and the GUI will send back data to the socket informing the room access is granted[10,11].

```

Face Detected
v1 = 81.05
v2 = 61.07
v3 = 61.59
v4 = 46.69
v5 = 101.24
v6 = 101.14
av1 = 82.71
av2 = 48.37
av3 = 48.92
av4 = 139.55
av5 = 17.41
av6 = 23.04
av7 = 137.73
av8 = 24.18
av9 = 18.09

Waiting To Retrieve Data

ACCESS GRANTED
>>>
    
```

Figure 9: Socket Sent Data and Received Feedback

The graph from Figure 10 shows the average values P1 until P10 which represent classes of different people, v1 to a9 are the features of distance and angle. Only features v5 and v6 which are the distance between eyes and mouth have obvious differences and the angles data are tightly closed to each other classes. This shows that the angles are not a unique feature. The persons 2, 3 and 6 have good distance value of differences compared to other classes.

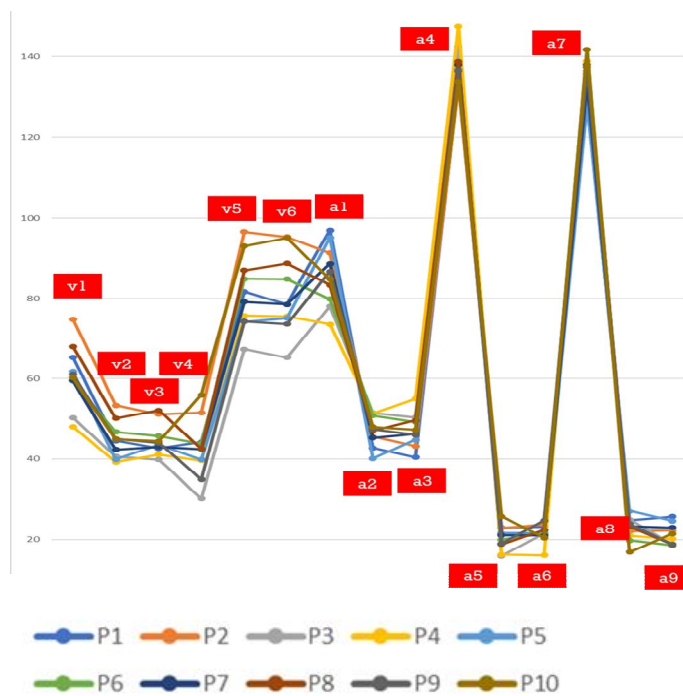


Figure 10: Graph of Average Distance and angles using Feature for Haar Cascade

Figure 11 shows more obvious difference of distance data among the classes using Facial Landmark compared to Haar Cascade

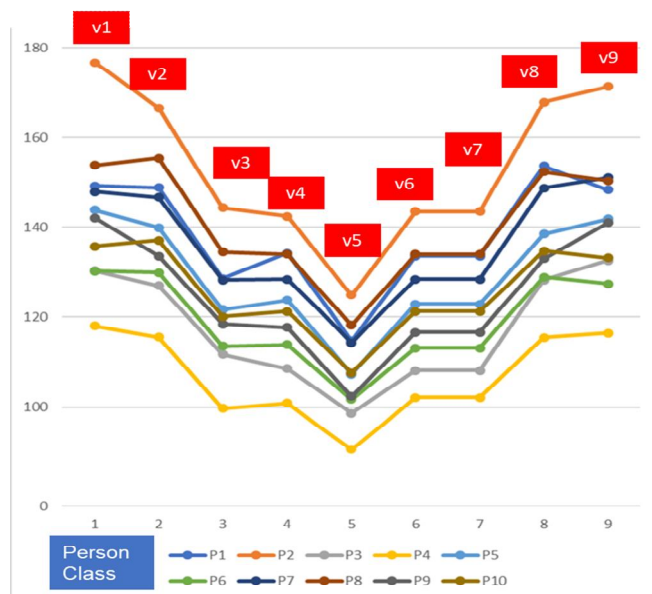


Figure 11: Average Distance for Facial Landmark

4. CONCLUSION

The Face detection module is developed using ROI and Selective Features of Distances and angles between facial feature’s centroid. Verification is needed when the feature’s data is sent to the Neural Network module. When the Haar Cascade Technique is implemented, it is found that the angles are not suitable to be used as a feature. Haar Cascade and Facial Landmark use distance to measure the feature extraction.

The System achieves accuracy of 60% Haar Cascade, 85% face landmark due to conflicts of facial feature’s measurement among different humans which are not that unique. The system is finalised by using Facial Landmark Technique into the Face Recognition System.

The system ROI is only able capture persons of specific heights only because the camera is fixed to one angle. A person who is taller or shorter is unable to fit into the ROI box. The project also has calibration issues in terms of defining the best distance between person’s face to camera. Static frontal face view gives less data variety among classes because it is 2 Dimensional (2D).

REFERENCES

1. Ranjana Dahake, M. U. Kharat , Priti Lahane, **Challenges and Advances in Human Face Recognition from Real Time Video**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol 5, No.6, November - December 2016

2. Shwetank Arya, Neeraj Pratap and Karamjit Bhatia **Future of Face Recognition: A Review**, *Science Direct Elsevier. Procedur Computer Science*, Vol. 58, pp. 578-585, 2015.
<https://doi.org/10.1016/j.procs.2015.08.076>
3. Priyanka Thakur , Rajiv Shrivastava ,**A Review on Text Based Emotion Recognition System**, Volume 7, No.5, September - October 2018 International Journal of Advanced Trends in Computer Science and Engineering
<https://doi.org/10.30534/ijatcse/2018/01752018>
4. Norazlin Ibrahim, Mohd. Marzuki Mustafa, Siti Salasiah Mokri, Lee Yee Siong, Aini Hussain. **Detection of snatch theft based on temporal differences in motion flow field orientation histograms**, *International Journal of Advancements in Computing Technology* vol. 4, pp. 308-317 Jun. 2012.
5. **Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python**.
[Online].Available:
<https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-pythonC>. [Accessed: 10 April 2019].
6. Balu, R. **Design and development of automatic appendicitis detection system using sonographic image mining**. Shodhganga : A Reservoir of Indian Theses @ INFLIBNET,167. Retrieved from[Online].Available
<http://shodhganga.inflibnet.ac.in/handle/10603/33597>
7. Mohd Saifulnizam Zaharin, Norazlin Ibrahim, Tengku Azahar Tuan Dir. **Motion Detector Using High Level Feature Extraction** International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 14580-14584.
8. Anis Hazirah Rodzi, Zalian Mohd Zin, Norazlin Ibrahim. **Vision based Eye Closeness Classification for Driver's Distraction and Drowsiness Using PERCLOS and Support Vector Machines: Comparative Study between RGB and Grayscale Images** , Journal of Physics: Conference Series, 2019
<https://doi.org/10.1088/1742-6596/1235/1/012036>
9. Nguyễn Hữu Tuân, Trịnh Thị Ngọc Hương. **Distance Metrics for Face Recognition By 2D PCA**. pp 2019-2022.2015.
10. OpenCV_ Drawing Functions in OpenCV.
[Online].Available
https://docs.opencv.org/3.1.0/dc/da5/tutorial_py_drawimg_functions.html, 2015
11. OpenCV. (2016b). OpenCV: Face Detection using Haar Cascades. Doxygen. [Online].Available
http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
12. D. Sarala , Tejas Kanikdaley , Sharad Jogi , Rahul K. Chaurasiya ,**Content-Based Image Retrieval Using Hierarchical Color and Texture Similarity Calculation**, Volume 7, No.2, March - April 2018 International Journal of Advanced Trends in Computer Science and Engineering.
<https://doi.org/10.30534/ijatcse/2018/02722018>
13. SVRK Rao, Srujana.G, B Priyanka, Rupalin Nanda, **Effective Analysis of Human Facial Appearance using JAFFE Images**, Volume 7, No.6, November - December 2018 International Journal of Advanced Trends in Computer Science and Engineering
<https://doi.org/10.30534/ijatcse/2018/17762018>