IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Static Hand Gesture and Face Recognition System

Prof. Kalpana Joshi, Akshata Salunkhe, Renuka Jadhav, Divya Sakhare

Abstract

This paper presents a recognition system, which improves human-computer interaction. In this paper, we've designed a face recognition system and hand gesture recognition system to perform different tasks. We've implemented detection and recognition using Haar-cascade classifiers and LBPH recognizers in the face recognition system. Our hand gesture recognition system detects the gesture and maps it with different tasks like launching applications such as notepads, paint, and opening websites. Gesture recognition is performed using a convex hull. Both the systems have been implemented with the help of OpenCV. Our results show that HCI can be improved with minimum hardware requirements.

Keywords

HCI(Human-Computer Interaction), Haar-cascade classifier, LBPH, Convex Hull

1. Introduction

Human-computer interaction studies the way humans(users) interact with computers and focuses on improving this communication. HCI initially was concerned with computers only, but now it has expanded to various forms of information technology. With the continuously evolving response to new technology and new consumer demands, it has become essential to make human-computer interaction more natural.

This system consists of two main modules, face detection, and recognition system, and another module is the hand gesture recognition system. The face recognition system performs two major tasks of identifying and verifying a person in front of the camera. This system uses a webcam to capture an image of the person, process it, and display the person's name if the face is there in the dataset. The system is independent of different parameters like hairstyle, facial expressions, etc. If the image matches with the photo in the database, the face is labeled with the person's name, and if it does not match, the image displays unknown. The dataset is created with multiple faces in different conditions, and all the images are labeled. Input images are checked across all these images. The system also displays the percentage, which tells the similarity between input images and images in the dataset.

Now consider the second module, which is a hand gesture recognition system. Generally, two approaches used for hand gesture recognition are Non-vision based and vision-based. An example of non-vision-based gesture recognition recognizes hand movement using censored gloves. In a vision-based approach, we don't use any hand devices such as data gloves. Another way to classify hand gesture recognition is static and dynamic gesture recognition. In static gesture recognition, the position and orientation of the hand in the frame do not change for a duration of time. Suppose there are changes within a given period, then the gesture is considered a dynamic gesture. Waving of the hand, sliding finger, etc., comes the face, under dynamic gestures while showing thumbs up comes under static gestures. The system discussed in this paper is a vision-based, static hand gesture system that recognizes the gesture and maps the gesture with an appropriate task assigned for the particular gesture.

2. LITERATURE SURVEY

The conducted literature survey gives insights about different methods that can be implemented to achieve face recognition and hand gesture recognition. It also helps to understand the advantages and disadvantages of various methods. The literature survey is divided into face detection and recognition modules, and the other is the hand gesture recognition module.

A more prioritized part of face recognition is face detection. Mainly face detection is divided into two parts: one containing faces and the other representing non-face regions. Face detection takes an image or video taken by a webcam as an input and the output as the detected face within these images using background subtraction. For face detection, [1] gives many methods such as Skin color-based Face detection Methods, Viola-Jones Face Detection system, Cascading, Face Detection system based on retinal connected neural network (RCNN). To avoid time complexity and get accurate results [2] uses HAAR cascade face detection is used. [3] explains these features on the image makes it easy to find out the edges of the lines in the image or pick areas where there is a sudden change in the intensities of the pixels.

Face detection has two modes one is face verification, and another is face identification or recognition. Face verification is one on one matching as the input image is compared with a specific test image. It is used for authentication purposes. Face recognition is one of many matching as input images are compared with dataset images. This becomes challenging when two faces are similar. Eigenfaces, Fisherfaces, and Local Binary Pattern Histogram(LBPH) are currently practiced algorithms for face recognition. Each method has a different approach for feature extraction from the image. One of the oldest and most simple face recognition algorithms is eigenface, as it eliminates false detection. Still, a change in luminosity can affect results and have to retrain data after adding a large number of new faces to the dataset. Fisher's face is an advanced version of eigenface as different lighting conditions have limited effect on the classification process. Also, it takes less space and is the fastest algorithm compared to eigenface and LBPH but is more complex in finding projection of face space than eigenface.LBPH is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and encodes the result as a binary number. It can represent local features in the image and is also robust against monotonic grayscale transformation, but the structural information captured by it is limited; only pixel difference is used, magnitude information is ignored. After analyzing the advantages and shortcomings of all three algorithms, the LBPH algorithm is shortlisted for the proposed system design.

The commonly used methods for taking input from the user in the form of capturing hand gestures that have been observed are data gloves and cameras. The approach used for gesture recognition in [5] and [8] uses data gloves equipped with accelerator sensors and resistance sensors. Cost-efficient models like [11], [1], and [10] have implemented their system using a simple web camera. [11] have used YCbCr color space for skin color detection. Inbuilt web cameras require less computational cost and do not provide unnecessary additional information. Hence, we have used the laptop's inbuilt webcam in our model instead of using additional cameras or data gloves. Several methods have been used for image pre-processing, which includes noise removal, thresholding, edge detection. [10] uses spatial filtering and linear filtering for noise removal, [11] uses the Gaussian blur technique. To perform edge detection, Canny edge detection has been used in [2] and [12]. The approach [2] uses for gesture detection is CPHD(Contour Point Distribution Histogram). Gestures are compared with the CPDH database of hand gestures, then get the closest one by using Euclidean distance and determine, after that, the type of hand gesture. [12] uses convex hull for gesture recognition while [3] uses convex hull technique combined with haar cascade, which helps to detect more gestures. The system proposed in this paper does not use any data markers such as data gloves; instead, it makes use of an inbuilt webcam which makes the system cost-efficient and reliable to use in day-to-day life.

3. Research Methodology

The proposed recognition system is divided into two parts-

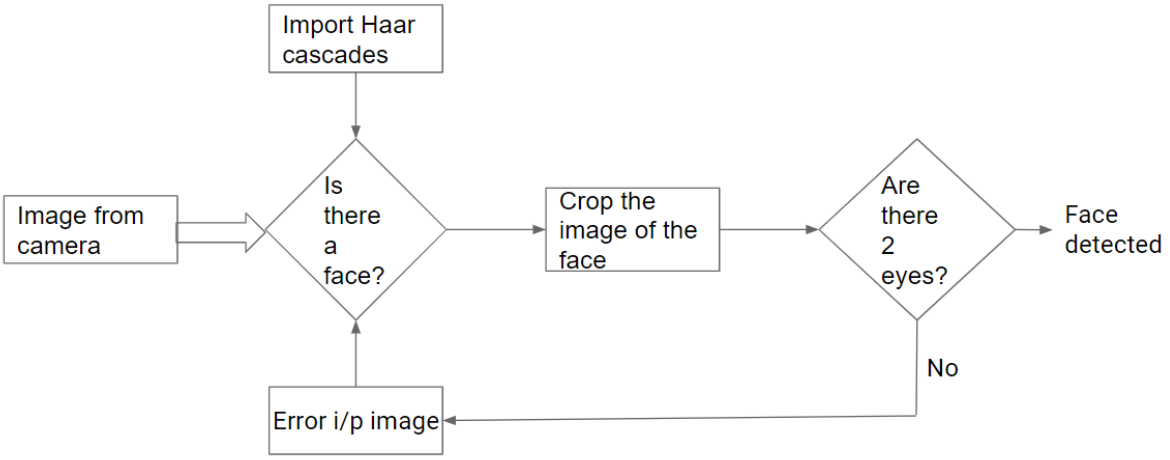
1. Face detection and recognition system
2. Hand gesture recognition system

3.1 Face Detection and Recognition

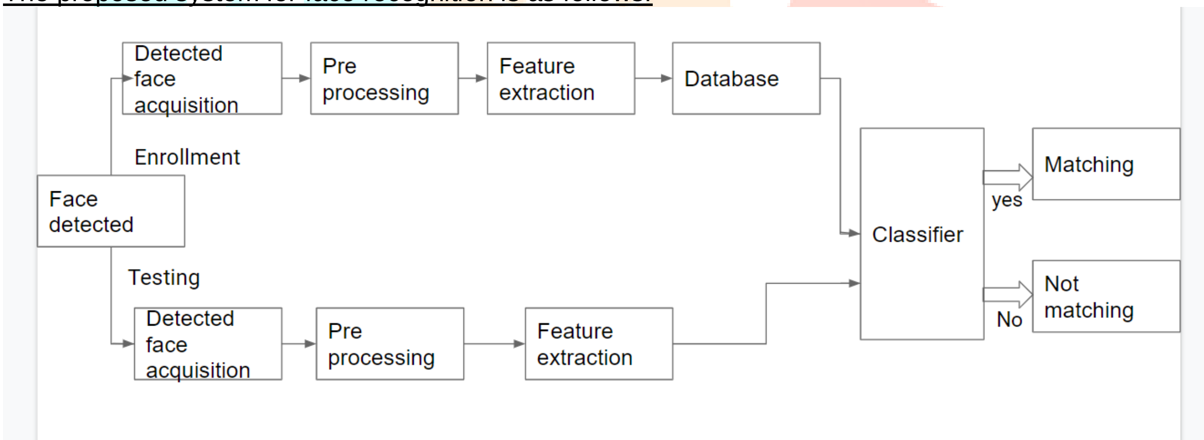
Face recognition system encompasses three main phases which are

- Face detection
- Feature extraction
- Face recognition

The proposed system for face detection is as follows:



The proposed system for face recognition is as follows:



3.1.1 Face Detection

As said above, for face detection, HAAR cascade features are applied to images. HAAR features on the image make it easy to find out the edges of the lines in the image or pick areas where there is a sudden change in the intensities of the pixels.

Initially, the algorithm needs many images of faces (positive image)and images without faces(negative images) to train the classifier. From that, we need to extract features from it. For this, Haar features shown in the image below are used.

HAAR Features:

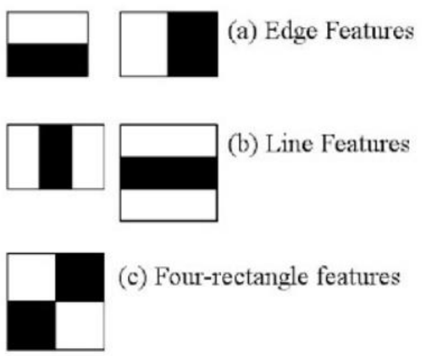


Fig 1: HAAR Feature Fig 2: HAAR features applied on the face

Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the aggregate under the black rectangle. Now all possible sizes and locations of each kernel are used to calculate plenty of features. We need to find the sum of the pixels under the white and black rectangles for each feature calculation.

The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature chosen relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant.

Software Implementation:

A face detection system using Haar-cascades training is required to create new Haar-cascades; OpenCV has a robust set of Haar-cascades used for the project. Using face-cascades alone caused random objects to be identified, and eye cascades were incorporated to obtain stable face detection. Classier objects are created using the classier class in OpenCV through the cv2.CascadeClassier() and loading the respective XML files. A camera object is created using the cv2.VideoCapture() to capture images.
By using the CascadeClassier.detectMultiScale(), objects of various sizes are matched, and location is returned. Using the location data, the face is cropped for further verification. Eye cascade is used to verify there are two eyes in the cropped face. If satisfied, a marker is placed around the face to illustrate a face is detected in the location.

3.1.2 Feature Extraction

From the detected face, we are extracting the features through a Local Binary pattern Histogram. In LBPH, first, we compute the local binary pattern images, and then histograms are created.

Local Binary Pattern Histogram (LBPH):

The detected face image and reduced the image dimension by applying the LBPH method, extracting the texture of the feature of the picture by dividing the image into local images and extracting the binary pattern for each local region. The original LBP operator, which works on eight neighbors of a pixel. Image is divided into small regions called cells. Each pixel in the cell is compared with each of its eight neighbors. The center pixel value will be used as the threshold value. The eight-neighbors-pixel will be set to one if its value is equal to or greater than the center pixel; otherwise, the value is set to zero. Accordingly, the LBP code for the center pixel is generated by concatenating the eight neighbor pixel values (ones or zeroes) into a binary code, which is converted to a 256-dimensional decimal for convenience as a texture descriptor of the center pixel.

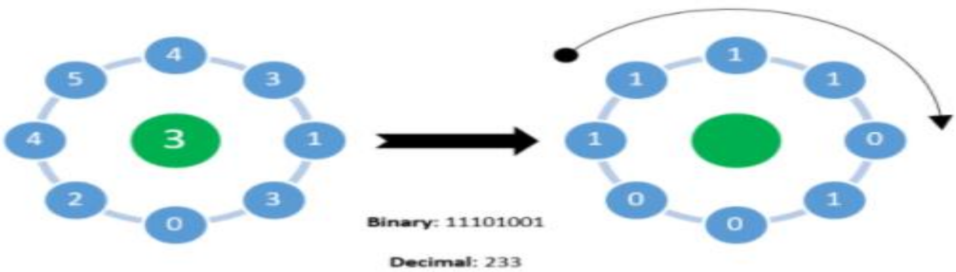


Fig 3: Original LBP operator

The mathematical formulation of the LBP operator is given by :

$$LBP\,(x) = \sum_{i=1}^{8} s(G(x^i) - G(x))2^{i-1}$$

$$s\,(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

For dimension reduction, we used the histogram to reduce the image features from a 256-dimensional decimal to a 59- dimensional histogram, which contains information about the local patterns. A separate bin for each uniform way and one particular bin for all non-uniform patterns used by histogram. In the 8-bit binary number, we have 58 uniform patterns; therefore, we used 58 bins for them and one bin for all non-uniform patterns. The global description of the face image is obtained by concatenating all regional histograms. The overall value of LBPH can be represented in a histogram as

$$H\,(k) = \sum_{i=0}^{n}\sum_{j=1}^{m} f\left(LBP_{P,R}\,(i,j)\,,k\right), k \in [0,k]$$

Fig 4: LBPH

3.1.3 Face Recognition:

The extracted features are fed to the classifier which recognizes or classifies by using an algorithm. The classifier compares the test image with the images saved in the database and can be done with a supervised classifier.

3.2 Hand Gesture Recognition:

We have proposed a markerless hand gesture recognition system, that follows a very efficient methodology as shown below:

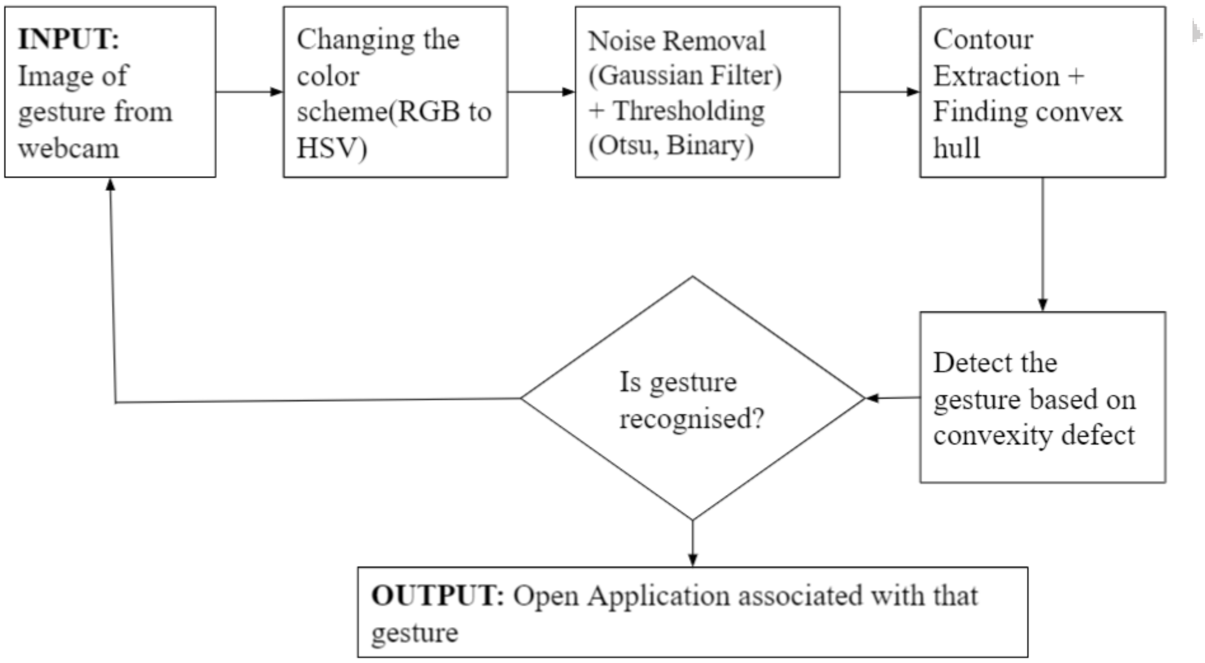


Fig 5: Methodology for hand gesture recognition system

3.2.1 Noise Removal and Image Smoothening

The input image is taken from the webcam, is in RGB color space, is cropped to a size of 200 * 200 pixels. It is then converted to HSV color space. This process is shown in the figure below:

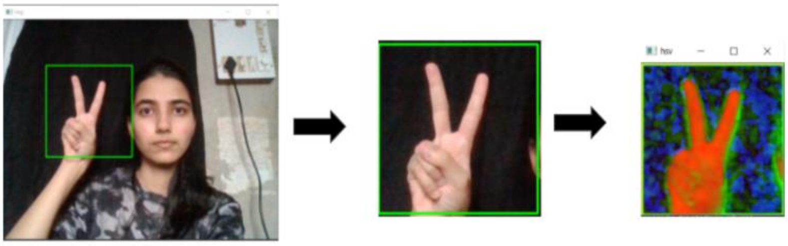


Fig 6: Process of cropping and converting RGB input image to HSV scale image

Random variation in the brightness produced while taking input images from a webcam can be considered as noise. Noise is an undesirable component in the image and needs to be removed. To perform this, a Gaussian filter is used. A Gaussian kernel of size (5,5) is used, which convolves with each point in an input array. These are then added to produce the output array. A 2-D Gaussian kernel is mathematically represented as shown in eq.

$$G_0(x,y) = Ae^{\frac{-(x-\mu_x)^2}{2\sigma_x^2} + \frac{-(x-\mu_y)^2}{2\sigma_y^2}}$$

3.2.2 Thresholding

Thresholding is a type of segmentation that generates a binary image from grayscale images to make it easier to analyze. The binary image is an image consisting of exactly two types of pixels, white and black. In thresholding, each pixel in the image with intensity, say I, is compared with Threshold intensity value say T. If I\textgreater T, then it is replaced by white pixel. In our system, we have used binary threshold and Otsu's thresholding.

Expression for binary threshold can be given as shown in the equation below:

$$\text{dst(x,y)} = \begin{cases} maxVal & src(x,y) > threshold \\ 0 & otherwise \end{cases}$$

So if the intensity of pixel src(x,y) is greater than threshold T then it'll be initialized to maxVal(i.e. 255), else it'll be replaced by 0. Otsu's thresholding was named after Nobuyuki Otsu[10]. Otsu's thresholding is a type of clustering-based image thresholding used to automatically perform thresholding to obtain a binary image. Before performing thresholding, this method assumes that the image has a bimodal histogram, i.e., the image has only two classes of pixels - foreground and background. The key idea here is to iterate through all the possible values of threshold and measure the spread of background and foreground pixels. Then find the threshold where the spread is least. The algorithm iteratively searches for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes (background and foreground), so that their inter-class conflict is maximum. The formula for finding intra-class conflict at any threshold is given as shown in the equation below:

$$\sigma^2(t) = \omega_{bg}(t)\sigma_{bg}^2(t) + \omega_{fg}(t)\sigma_{fg}^2(t)$$

Where $\omega_{bg}(t)$ and $\omega_{fg}(t)$ represents the probability of the number of pixels for each class at threshold t and σ^2 represents the variance of color values.

Fig 7: Front end window that shows the thresholded version of the input gesture

Fig 8: Contour extraction

3.2.3 Contour Extraction

Contour detection is an efficient tool for object detection in image processing. Curves that link continuous points which are of the same color or same intensity are called contours. In this paper, contour detection is used to recognize the hand, which will help recognize the gesture with the help of contour analysis. Before finding contours, a threshold has been applied to the binary image to achieve higher accuracy because in OpenCV, finding contour is like finding a white object on black background. After this next step is to draw the contours, which can be used to remove any shape provided the boundary points are known. The image below shows the front-end window that portrays the contour of the user's gesture input.

3.2.4 Convex Hull

Mathematically, the convex hull of a set X of points in an affine space is defined as the smallest convex set that contains X. To calculate the convex hull of the given contour, the palm is considered. The convex hull will be the convex polygon, where the tip of the fingers will be the vertices of the polygon. This will simplify the complex gesture structure. Convex hull can be represented mathematically as shown in the equation below:

$$Convex(S) = \left\{ \sum_{i=1}^{|S|} \alpha_i x_i | (\forall_i : \alpha_i \geq 0 \wedge \sum_{i=1}^{|S|} \alpha_i = 1 \right\}$$

The image below shows the front end window that portrays the convex hull along with the contour of the user's gesture input

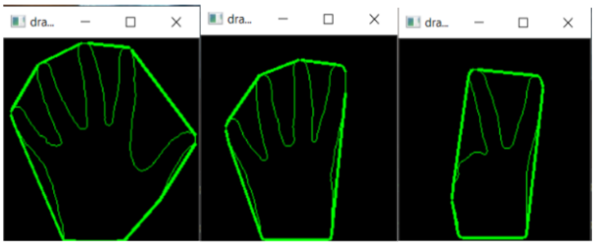


Fig 9: Convex Hull

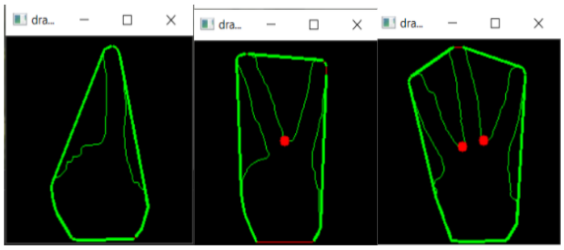


Fig 10: Convexity Defects

3.2.5 Convexity Defect

The divergence of contour from the convex hull of gesture is defined as a convexity defect. Points farthest from the convex points are considered defects. So, if the fingertips are considered as convex points, then through between the fingers will be defects in this case. A defect is counted if the angle between the two fingers is within 90 degrees, so anything greater than 90 degrees is not considered. By counting these defects, gestures will be recognized.

Fig. 6 shows the front end window that portrays the convexity in the user's gesture input (Convexity defects are presented by red points)

4. RESULTS AND DISCUSSION

4.1 Development of GUI:

The GUI is developed using the python library "Tkinter" for GUI creation. The first GUI figure contains a "Welcome" push button to start the system. After this, a new GUI window will appear, which includes three pushbuttons. The pushbuttons are designed for face recognition, dataset creation, and training. Any authorized user can make use of the "Detection" push button, but only Admin can make use of the "Create" and "Train" push buttons. GUI has been developed to improve the user experience.

4.2 Dataset:

In our system, we have made a provision that the data of a new person can be added only in the presence of "Admin" to avoid malicious data addition. If someone other than the admin tries to add their data to the system, the system will not allow them to do so, and a "Warning Box" will appear.
First Admin will unlock the system then the dataset of the person will be created. One hundred images of a person will be taken with different expressions. The following image shows the dataset created.

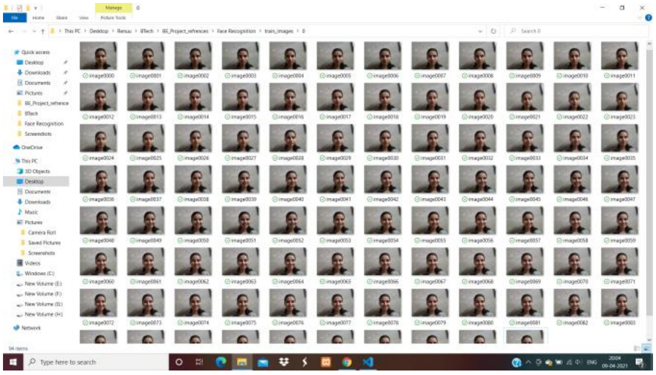


Fig 11: Dataset and training model

Fig. 12 GUI Window-1

Fig. 13 GUI Window-2

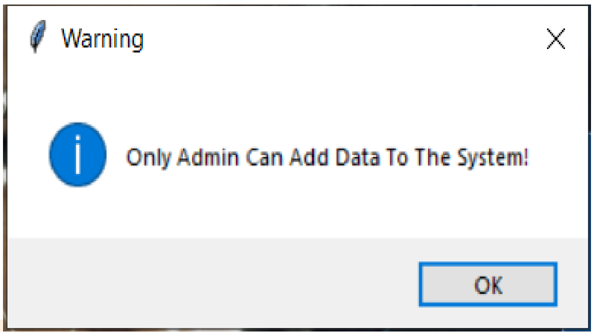


Fig. 14 Warning box -1

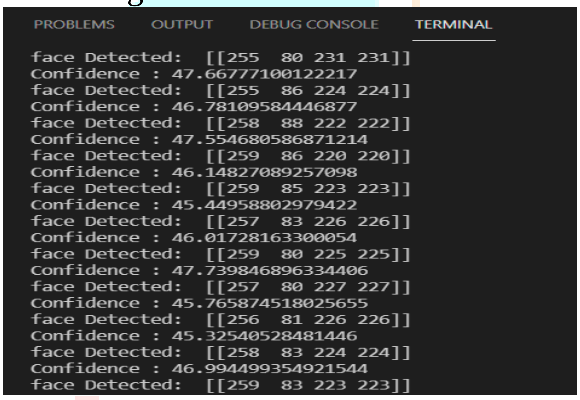


Fig 15: Confidence for recognized face

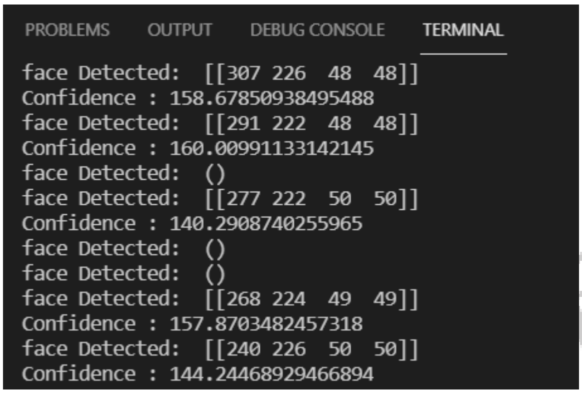


Fig 16: Confidence for unknown face

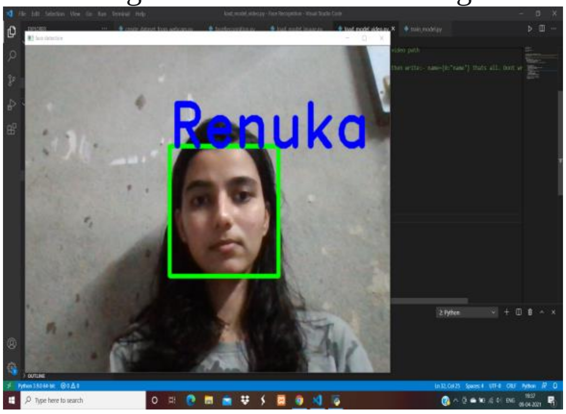


Fig 17: Face recognized under good brightness

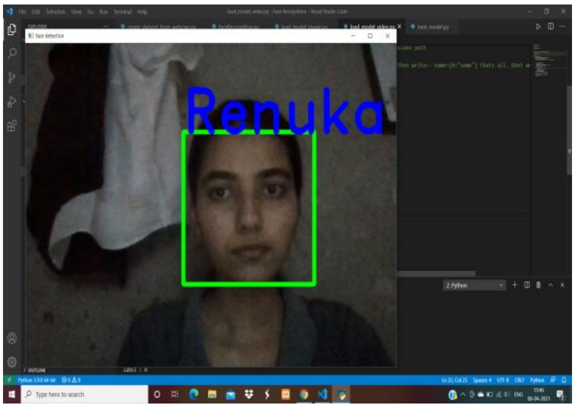


Fig 18: Face recognized under bad brightness condition condition

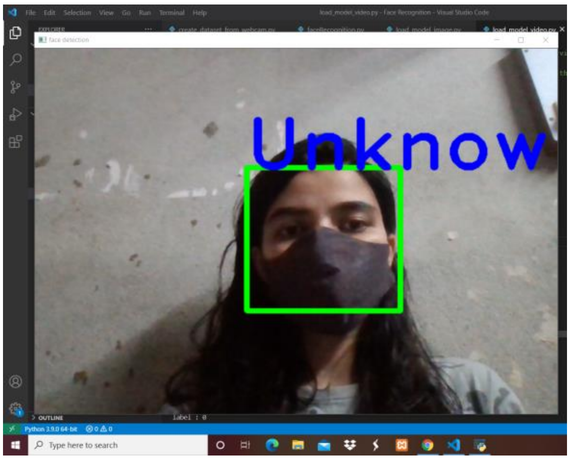


Fig 19: Person not recognized -1

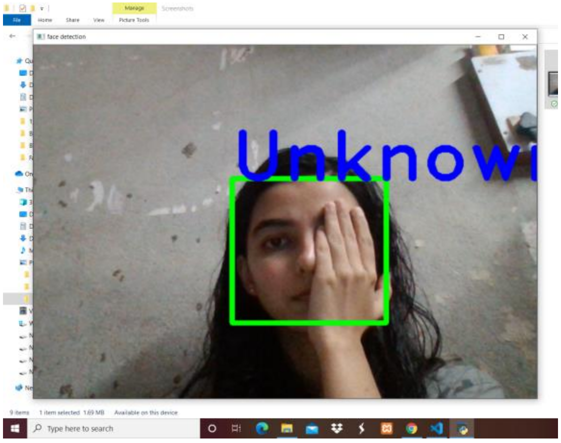
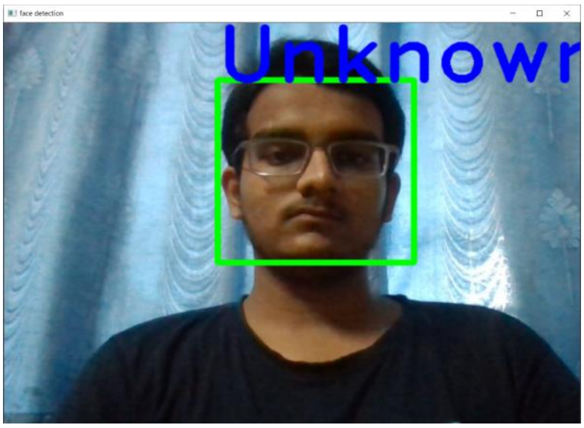


Fig 20: Person not recognised - 2



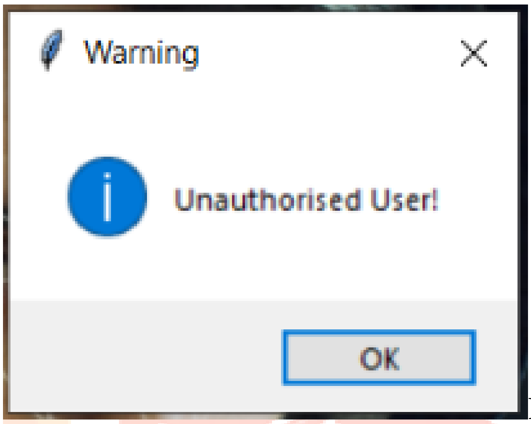
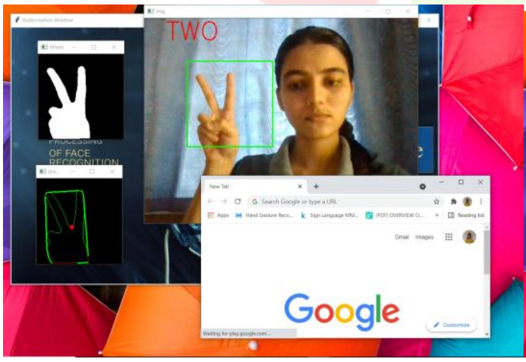


Fig 21: Unknown Person

Fig 22: Warning Box - 2

Fig 23: The static gestures used in the gesture recognition system



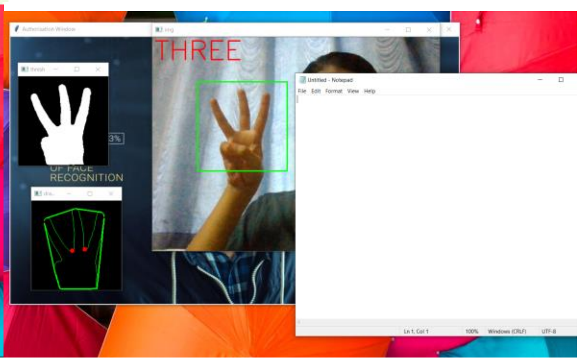


Fig 24 (a): Gesture “2” launching Google home page in the browser

Fig 25 (b): Gesture “3” launching notepad

5 Checking accuracy of the system:

Sr.No	Condition	0	1	2	3	4	5	Face Recognition Accuracy
1.	Good Background	100%	100%	100%	100%	100%	100%	100%
2.	Noisy Background	90%	100%	100%	90%	90%	90%	100%
3.	Camouflage Background	20%	40%	30%	20%	30%	20%	93.33%
4.	Light effect							
	4.1 Normal	100%	100%	100%	100%	100%	100%	100%
	4.2 Bright	100%	100%	100%	100%	100%	100%	100%
	4.3 Low	100%	100%	100%	100%	100%	100%	90%
5.	Almost similar faces	-	-	-	-	-	-	90%
6.	More than one face in a frame	-	-	-	-	-	-	80%
7.	Face in different angles	-	-	-	-	-	-	60%
8.	Face covered with mask or scarf	-	-	-	-	-	-	90%
9.	Different looks(with specs, different hairstyle)	-	-	-	-	-	-	95%

5.1 Condition Variation:

5.2 Parameter Variation:

5.2.1 Varying area_ratio:

Area ratio is calculated to differentiate between gestures "0" and "1" as both of the gestures have convexity defects equal to zero. To decide the efficient value of area_ratio several trials are performed and then the value of area_ratio is decided. Following table shows the results for different values of area_ratio:

Value of area_ratio	0	1
2	No	Yes
3	No	Yes
4	No	Yes
5	No	Yes
6	Yes	Yes
7	Yes	Yes
8	Yes	Yes
9	Yes	Yes
10	Yes	Yes
11	Yes	Yes
12	Yes	Yes

After taking several trials, the finalized value for area_ratio is 10.

6. Applications:

Application is limited only by our imagination. Both face recognition and Hand Gesture recognition have a wide range of applications.

1. Face recognition systems are highly used in security.
2. In the pandemic, there is a need for lesser interaction at public places so that the face recognition and Hand gesture recognition systems can be used to perform activities in public places.
3. Not only blind people can use it, but also several industries can use this technology.
4. Hand gestures can be applied in telemedicine, where doctors can help the patients, and other doctors remotely can also operate different machines without actually touching them during the surgeries.
5. Industrial robots can also take advantage of hand gestures and face recognition.
6. Facial recognition systems are widely used in law enforcement to find the person of interest.

7. Conclusion:

This project will help detect and recognize the faces in different environmental conditions(like brightness variation) very efficiently. It thoroughly explains the methods and technologies used in the project. Finally, it shows the results of the proposed system. Considering all factors, LBPH combined with Haar-cascade classifiers can be implemented as cost-efficient face detection and recognition solution. Making use of the inbuilt webcam of the laptop also contributes to making this system more cost-efficient. The hand gesture recognition system makes use of Convex hull and Convexity defects to recognize the hand gestures. With minimum hardware, the system works very efficiently. Total six static hand gestures are recognized successfully, and different activities are performed based on those gestures. This improves the HCI and is also very helpful for visually disabled people. Future work can be carried on to detect and recognize the moving faces(dynamic recognition). A liveliness check can be added to the face recognition system to avoid face spoofing to improve accuracy. We also aim to extend our domain scenarios and apply our tracking mechanism to various hardware, including digital TV and mobile devices for a range of users.

8. References :

1.Aashni Hariaa , Archanasri Subramaniana , Nivedhitha Asokkumara , Shristi Poddara , Jyothi S Nayak (2017) 'Hand Gesture Recognition for Human Computer Interaction'

2. Bousaaid Mourad (2014) ' Real-Time System of Hand Detection And Gesture Recognition In Cyber Presence Interactive System For E-Learning.'

3. Deepak Gurung, Cansen Jiang, Jeremie Deray, Désiré Sidibé (2013) 'Hand Gestures Recognition and Tracking'

4 Dr.Vipulsangram.K.Kadam , Deepali G. Ganakwar (2017) 'Face Detection: A Literature Review'

5. Faizan Ahmad1, Aaima Najam2 and Zeeshan Ahmed3 'Image-based Face Detection and Recognition:"State of the Art"'

6. Granit Luzhnica, Jorg Simon, Elisabeth Lex, Viktoria Pammer (2016) 'A Sliding Window Approach to Natural Hand Gesture Recognition using a Custom Data Glove'

7. LAHIRU DINALANKARA(2017) 'Face detection and face recognition using open computer vision classifiers'

8. Ms.K.Priyadharshini, Ms.Tripty Singh (2012) 'Research And Analysis On Segmentation And Thresholding Techniques'

9. MOHANNAD A. ABUZNEID , (Member, IEEE), AND AUSIF MAHMOOD, (Senior Member, IEEE) (2018) 'Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network'

10. Saransh Sharma1, Samyak Jain2, Khushboo3 (2019) 'A Static Hand Gesture and Face Recognition System For Blind People'

11. Smitha1, Mr. Vishnu Siju2, Ms.Ashwini3(2020)'A survey on machine learning algorithms for face recognition.'

12. Suma S L1, Sarika Raga (2018) ' Real Time Face Recognition of Human Faces by using LBPH and Viola Jones Algorithm'

13. Tahir Khan (2015) ' Hand Gesture Recognition based on Digital Image Processing using MATLAB'

14. V. K. Bairagi1 'Gloves Based Hand Gesture Recognition Using Indian Sign Language'

15. Yanan Xu1 , Dong-Won Park1 and GouChol Pok2 (2017) 'Hand Gesture Recognition Based on Convex Defect Detection'