International Journal of Innovative Research in Engineering
Special Issue, Volume 4, Issue 3(May-June 2023), PP: 69-76.
Innovations in Computing, Materials and Communication Technologies (ICICMCT'23) **ISSN No: 2582-8746**

Face and Gesture Based Human - Computer Interaction System

Niya K.S.[1], Anu Augustin[2]
[1]*P.G.Student,Department of Computer Science and Engineering, IES College of Engineering, Chittilappilly, Kerala, India.*
[2]*Assistant Professor, Department of Computer Science and Engineering, IES College of Engineering, Chittilappilly, Kerala, India.*

How to cite this paper:
Niya K.S.[1], Anu Augustin[2], "Face and Gesture Based Human - Computer Interaction System", IJIRE-V4I03-69-76.

Abstract: *Computers and people interact in a number of ways, and these interactions are made possible by the interface between the two. To make interactions between people and computers as simple and efficient several methods has been proposed. The vision-based systems enabling facial and hand gesture identification are the most promising ones for contactless human-computer interaction (HCI). This project proposed a human-computer interaction (HCI) system by combining a face and gesture recognition system. Face recognition based authentication system is more popular than any other biometric features like fingerprint and eye iris recognition. The face recognition system performs identifying and verifying a person in front of the camera. If the image matches with the photo in the database, the face is labelled with the person's name and a successful login is done within system. If it does not match, the image displays unknown and an unauthorized access is detected. Here face detection and recognition is implemented using HAAR-cascade classifiers and LBPH recognizers. The hand gesture recognition system recognises the gesture and links it with a variety of actions, including launching programmes like windows media player, MS word, PowerPoint, Notepad, screenshot capturing, opening Google and some of the mouse control actions. Here hand tracking performs with the help of Media Pipe library provided by Google which is open-source and it is a well-trained model to achieve high performance. The Media Pipe library can be used to analyse hand gestures using a variety of technologies. Media Pipe hands library will employ two models. 1) A palm detector model that generates a bounding box of hand and 2) A hand landmark model that predicts the hand skeleton. Gestures can be used by users to effortlessly communicate with computers that have RGB cameras.*

Key Word : *Human-Computer Interaction, Face and hand gesture recognition, HAAR-cascade classifiers, LBPH recognizers, Media Pipe library.*

I.INTRODUCTION

Human-computer interaction is the study of how people (users) communicate with computers. Groups lacking formal training and knowledge on interfacing with certain computing systems might benefit from human-computer interaction. Users do not have to think about the complexity and intricacies of using the computing system with efficient HCI designs. Face recognition is the most flexible and promising biometric authentication method because the object being authenticated is unaware that it is being scanned. When recognizing an individual, facial characteristics including nose width, eye distance, and jaw line are taken into consideration. It is used in a variety of application areas, including human-computer interaction, face tracking, facial expression identification, tiredness monitoring, visual surveillance, gesture recognition, and others. Generally face detection falls under two categories: region with faces and non-face regions. There are numerous methods available for detecting faces, including skin color-based systems, the Viola-Jones face detection system, cascading, and a face detection system based on the retinal connected neural network (RCNN). HAAR cascade face detection is used to reduce time complexity and obtain accurate results. For face recognition, the algorithms Eigen faces, Fisher faces, and Local Binary Pattern Histogram (LBPH) are now in use. Eigen face, one of the simple and oldest face recognition systems, prevents erroneous detection. However, a change in brightness can impact the outcomes. Fisher's face is a more sophisticated variant of eigen face because the classification process is not greatly affected by changes in lighting. The LBPH texture operator labels the pixels in an image by thresholding the neighborhood of each pixel and encoding the outcome as a binary integer.

The goal of gesture recognition is to create a method that can recognize certain human gestures and use them to transfer data. In the gesture recognition method, a camera captures the movements of the user's body and transfers the information to a computer, which then uses the gestures as input to operate devices or software. Typically, there are two methods used to recognize hand gestures: vision-based and non-vision-based. The detection of hand motion while wearing censored gloves is an illustration of non-vision-based gesture recognition. We don't employ any hand equipment, such as data gloves, in a vision-based method. Static and dynamic gesture recognition are further classifications for hand gesture recognition. The hand's alignment and position in the frame do not vary over the course of static gesture identification. The gesture is regarded as dynamic if changes occur within a specific time frame. Static motions include giving the thumbs up whereas dynamic gestures include waving the hand, sliding the finger, etc. Wearable gloves are used in methods without computer vision, but since they require sensors and other hardware, they are too expensive. KNN (K Nearest Neighbour), ANN, and convex hull are only a few of the hand gesture recognition methods available, although most of these techniques require a large amount of training and recognition data. Google offers the Media Pipe library, an open source, cross-platform framework for processing perceptual input from many modalities like audio and video. Media Pipe Hands is a high-quality hand and finger tracking system. It is a well-trained model to deliver outstanding performance. The price of the hardware becomes a significant factor when using these techniques. Recently, significant attempts have been made to create intuitive and natural user interfaces for computer-based systems that are based on human gestures. The main benefit is that it eliminates the need for users to wear supplemental tools, which can be uncomfortable. This motivates the touch-less human computer interactive system.

The paper is organized as follows. The related work in the Section II offers a critical evaluation of all the earlier papers and studies. The design of the suggested system is presented in the Section III. It provides a detailed introduction to the proposed system architecture. All the methodologies that were used in this project are described in the Section IV. The Section V gives information on the system's implementation. The Section VI discusses and explains the result and analysis. The project has concluded and future works are discussed in the last chapter. Supporting papers and publications are cited in the references.

II.RELATED WORK

A. Face Authentication

A method based on information theory was proposed by V.P. Kshirsagar et al. [1] which decomposes facial images into a linear combination of eigen faces. A new image is projected onto the subspace spanned by the eigen faces to perform recognition. The faces are then classified by comparing their position in the face space to those of the known people. Eigen faces are eigenvectors that describe each of the face space's dimensions and can be defined as different facial features. Any face can be expressed as linear combinations of the eigenvectors of the covariance matrices. It has been proven that eigen faces are capable of delivering important features and minimising the input size for neural networks. On face images in the FERET database, M.S. Bartlett et al. [2] presented two ICA architectures: one that considered the images as random variables and the pixels as outcomes, and the other that considered the pixels as random variables and the images as outcomes. For the faces, the initial architecture discovered spatially local basis images. A factorial face code was produced by the second architecture. The best result came from a classifier that integrated the two ICA representations. A linear transformation is in search to represent a collection of random variables as a linear combination of statistically independent source variables. Therefore, ICA offers a more effective data representation than PCA. For real-time applications, this approach is a little challenging. By maximising the between-class scatter and minimising the within-class scatter, Suman Kumar Bhattacharyya et. Al[3] introduced LDA (Linear Discriminant Analysis) to face recognition. By employing the linear discriminant criterion, the LDA approach overcomes the drawback of the Principle Component Analysis method. The ratio of determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix of the projected samples will be increased by this criterion. The projected test image is compared to each projected training image to identify the test image. The test image has chosen the closest training image. However, the main issue with applying LDA is that it might run into the issue of small sample size.

A system called Elastic Bunch Graph Matching (EBGM) is presented by L. Wiskott et al. [4] for identifying human faces from large database that contains one image for each individual. Based on a Gabor wavelet transform, labelled graphs are used to represent faces. An elastic graph matching approach extracts image graphs of new faces, and can be compared by using a simple similarity function for the accurate positioning of nodes. Variations in facial positions are not particularly taken into account by this procedure. Due to its high light sensitivity and lengthy matching process, EBGM is unsuitable for use in real situations. To improve performance, the graphs placed on the face should be compared, however these graphs demand a significant amount of convolution image storage. A face recognition application using the fisher face approach was proposed by MustaminAnggo et al. [5] which uses databases and GUI applications that are implemented in the Papuan facial image form. These methods are based on the Principal Component Analysis (PCA) for dimensionality reduction and then Fisher's Linear Discriminant (FDL) method for obtaining features of image characteristic. Fisher faces algorithm is applied for image recognition, and minimal euclidean is used for the identification or matching of face images. Fisher face and Eigen face are comparable, but Fisher face has a higher ability to classify different classes image. This method can categories the training set with various people and face expressions. Compared to the Eigen face technique, our facial expression accuracy could be higher. Additionally, Fisher face is more invariant to changes in light intensity because it removes the first three principal components. It takes a long time to compute the ratio of between-class scatter to within-class scatter. It requires large storage for the face and more time for perform recognition.

B. Hand Gesture Recognition

A method for real-time and gesture recognition has been put out by Yikaet al. [6]. This system include mainly three steps. The first step uses Adaboost to initiate tracking and recognition. Adaptive hand segmentation is used in the second step during detection and tracking using motion and colour cues. Finally, palm-like and finger like structures are extracted using scale-space features detection. The palm-finger configuration determines the types of hand gestures. Static hand gesture recognition for human-computer interaction was proposed by Hamid et al. [7]. The aim of this work was to present a vision-based hand gesture identification system that uses both supervised feed-forward neural networks with back training and wavelet networks for feature extraction from images. Three fundamental phases constitute the proposed system for recognising hand gestures: preprocessing, feature extraction and classification. By using two methods it perform gesture recognition. First, wavelet network is used for feature extraction. After this a neural network is employed for gesture recognition. The main drawback of this technology is that it takes a long time for neural networks to classify the input, which causes a delay while classifying different gestures. Moment-based static gesture recognition was suggested by Padam et al. [8]. Through the identification of skin color, this system identifies the hand region and derives the binary silhouette. These photos have been scaled and rotated to a normalised form. Using a minimum distance classifier, the moment features of the normalized hand gestures are classified.

A hand gesture detection system using the Kinect sensor and HOG characteristics was proposed by Hui Li et al. [9]. By determining the features of hands characteristics, this method chooses HOG features that adapted to the light transform. Light and rotation should not affect the extraction of hand features. To extract hand features, it creates histograms of oriented gradient (HOG), which has wide application on target detection in recent years. To train the hand gesture models an Adaboost training algorithm is used. Finally, it creates an effective system for static hand gesture recognition. But still some situations such as hands covered in front of the body or objects that resemble hands are lead to high missing and false rate. Gesture recognition based on the fuzzy c-means clustering algorithm has been proposed by Xingyan Li et al. [10]. The Fuzzy C-Means method offers sufficient reliability and speed for achieving the specified task. Preprocessing is used to detect hands in images. The HSV channel is used to find regions of an image that resemble skin. The segmented hand shape is then transformed into a feature vector that includes contours, edges, and other details. The key drawback of this approach is that wrong object extraction occurs if the objects are larger than the hand.

III.SYSTEM DESIGN

Proposed system develops a vision-based static hand gesture and face recognition human interactive system and maps the gesture with an appropriate task assigned for the particular gesture such as launching applications like word, PowerPoint, notepad, windows media player, screenshot capture, some of mouse actions. The proposed recognition system architecture is shown in Fig.1 System is divided into two parts, Face authentication system and Hand gesture recognition system.

A. Face Authentication System

Face authentication system encompasses three main phases which are data collection, training phase and testing phase. In data collection phase faces of authorized persons has to be stored within the dataset. At first we convert the image from color to gray scale mode. For detecting the faces system has used a Haar cascade classifier where a cascade function is used to extract the features from the images. To make this possible, system uses Haar features like edge, line, and four-rectangle. Finally, the detected faces are store in the dataset with unique IDs for further process. In training phase, the faces in the dataset can be trained and create a trained model. At first, system loads the dataset and HAAR cascade classifier. LBPH algorithm is used for training the faces and finally system creates a training model. Authentication begins with the testing phase of the user. Here dataset image is compared with the real time image of the person who desires to get an access to the system. If a match is detected then a successful login is observed and goes to the hand gesture recognition system otherwise an illegitimate person is detected and the system will show a message about this unauthorized access.

B. Hand Gesture Recognition System

Using a single webcam, this system can predict a human's hand skeleton. Using Google's open-source Media Pipe library, which is a well-trained model to achieve high performance, it is possible to identify hand gestures. The palm detection model and the hand landmark model are the two models that make up the ML pipeline at the backend of the hand tracking system. The system primarily comprises two phases.

1) Hand landmark detection
2) Hand gesture recognition

Phase 1 begins with the palm detection model, which outputs an appropriately cropped palm image that is then forwarded to the subsequent landmark model. The hand skeleton is predicted using the hand landmark model. Inside the detected hand region this model can detect 21 3D hand-knuckle coordinates x,y and z axis. By using these landmarks we can detect the hand gesture. In phase 2, ten different gestures such as numbers from 1 to 10 are used. Each gestures are mapped to actions like mouse control, press click, press up, and press down, screenshot capture, launching applications like word, PowerPoint, notepad and open Google browser.

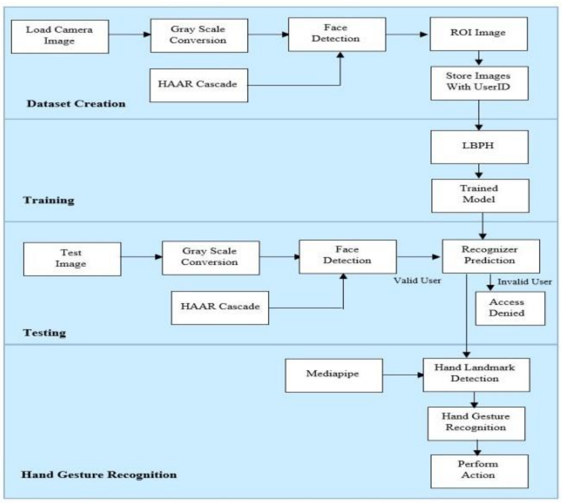


Fig. 1 Proposed System Architecture

IV.METHODOLOGY

The main two parts of this system are the facial authentication and hand gesture recognition. The two key methods of a face authentication system are face detection and face recognition. Face detection and face recognition are performed using

HAAR-cascade classifiers and LBPH recognizers, respectively. The hand gestures can be recognised with Media Pipe Hand. Based on this the gestures are mapped to the corresponding actions.

A. Haar Cascade Classifier

Object detection deals with detection of instances of objects in videos and images. It is related to computer vision, image processing, and deep learning. Object detection is efficiently accomplished with HAAR Cascade classifiers. Paul Viola and Michael Jones suggested this approach in their study titled 'Rapid Object Detection using a Boosted Cascade of Simple Features'. It is simple to identify the edges of the lines in the image or select regions where there is a sharp change in the pixel intensity by applying HAAR features on the image. To train the classifier, the algorithm initially requires a large number of photos with faces (positive images) and images without faces (negative images). For this, HAAR features shown in the Fig.2 is used.

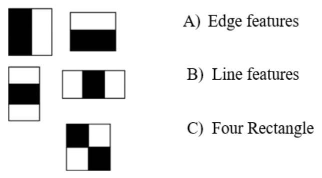


Fig.2 HAAR Features

The aggregate of pixels under the white rectangle is subtracted from the sum of pixels under the black rectangle to provide a single value for each feature. The fact that the region around the nose and cheeks is brighter than region around the eyes is chosen as first feature. It is very difficult to calculate a single value for each feature. For quickly calculating the sum of pixel values in an image we use integral image concept. Nearly 16,000+ features are present in a 24x24 size detector window. In these features few of them are most important. To identify best features we use an Adaboost algorithm which check the performance of all classifiers. Sub regions which do not contain a face produces a weak response is classified into negative images. Sub region which has strong response contain a human face.

B. Lbph (Local Binary Pattern Histogram)

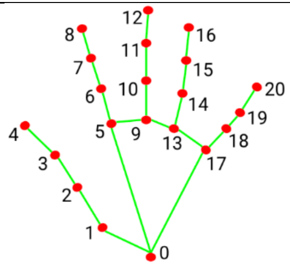
Face recognition is accomplished using the Local Binary Pattern Histogram (LBPH) method. It is based on a local binary operator made to recognise both the front and side faces of a human. The following procedures are done in order to identify the face in each new image. Four parameters are commonly used by the LBPH algorithm: In order to create a circular local binary pattern, radius is used. To build the circular local binary pattern the number of sample points is provided by the neighbours parameter. Grid X and grid Y represents the number of cells in a horizontal direction and vertical direction respectively. To recognize the faces it utilizes a dataset which contains images of the people. To enable the algorithm to recognise each image and export the output, each image is given a distinct ID that can either be a number or a name. The same person's images are always listed under the same ID.

A sliding window approach has been used to construct an intermediate image that more accurately represents the original image while considering two parameters: the neighbour and the radius. By comparing the 8 neighbour values to the threshold value, new binary values are formed**.** The value is set to 1 for any neighbour value larger than the threshold value and to 0 for any neighbour value less than the threshold value. A binary number matrix is created by this, but not include the threshold. The binary number is converted to a decimal value that represents the pixels of the original image to produce the centre value of the matrix.The image generated in step is divided into several grids using the grid parameters X and Y. Each histogram on each grid in this grayscale image represents the intensity of the occurrences of each pixel. A new histogram that represents the characteristics of the original image is then produced by combining each histogram. For each image from the training data histograms will be created. To determine which image best represents the input image's histogram, two histograms are compared. This output contains the image's ID or name. A confidence measurement, which is the calculated distance, is also returned by this procedure. How accurately the system recognises the image is automatically estimated by the confidence and threshold. A confidence value below the specified threshold indicates the correctness.

C. Media Pipe

An open source, cross-platform framework called Media Pipe is used to process perceptual information from a variety of modalities, including audio and video. Face recognition and posture estimation are two examples of the solutions employed in Media Pipe. Here, the system tracks hands using Media Pipe Hands. From a single frame obtained from a webcam we can deduce 21 3D coordinates in X, Y, and Z axes. This approach allows for more efficient advanced hand and finger tracking, and it can be used in low-performance settings like mobile situations. Media Pipe frame work provide an initial palm detector called Blaze Palm. For detecting the hand first we train the palm instead of the hand detector. Following that, the non-maximum suppression algorithm is used on the palm, where it is modelled using square bounding boxes to prevent other aspect ratios and 3-5 times fewer anchors. Next, encoder-decoder of feature extraction is used and lastly reduce the focus loss during training by using a lot of anchors. The following hand landmark model uses regression to precisely localise 21 3D hand-knuckle coordinates inside the observed hand regions after the palm has been recognised. The hand landmark model is resistant to self-occlusions, hazy hands, and even partially visible hands. The system manually employed 30K real-world images with

21 3D coordinates to gather ground truth information for the model, as shown in Fig. 3. The method additionally renders a hand model with high quality and map it to the corresponding 3D coordinates. It covers every possible hand positions and provide extra supervision on hand geometry.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig. 3 Hand Landmark in Media Pipe

V. IMPLEMENTATION

The main two parts of this system are the facial authentication and hand gesture recognition. The two key methods of a face authentication system are face detection and face recognition. Face detection and face recognition are performed using HAAR-cascade classifiers and LBPH recognizers, respectively. The hand gestures can be recognised with Media Pipe Hand. Based on this the gestures are mapped to the corresponding actions.

A. Face Authentication System

For dataset creation system first import libraries such as Open CV and os. Open CV provides features for object detection, face recognition, and tracking. The module import name for Open CV is cv2. The images are stored in a folder namely dataset. By importing OS module we can access path to this folder. Capture video from a webcam. Open CV provides a function cv2.Video Capture() which helps to create a video capture object. By using this object we can perform desired operations on the video which is captured by a webcam. With the help of this object we can set up an infinite while loop and read() function call for reading the frames. For showing the frames in the video we apply cv2. imshow() method. This loop will execute infinitely until the user clicks a specific key, here we use the key ‘q’.

Next step is the face detection. haar cascade_frontal face_default.xml module is used for face detection. By using the cv2. Cascade Classifier() method the cascade is loaded. The path to the cascade xml file is passed as a parameter to this method. cv2.COLOR_BGR2GRAY() method is used for the gray scale conversion of the image. cv2. Cascade Classifier. detect MultiScale() is used to detect the face object from the image. Image is the first argument to this method, the second is the scalefactor, and the minNeighbors is the third one. By default we take values of 1.3 and 5 which provides the best result. When faces are found Rect(x,y,w,h) returned as result which is the positions of the detected face. Draw a rectangle to the cropped image by calling cv2. rectangle() method. Color of the frame in RGB, thickness of the rectangle, horizontal and vertical initial position, width and height are the parameters to this method. Next we create a ROI of the face here value of patch will be our region of interest. Last we stores image at dataset folder with unique ID by calling cv2. imwrite() method. The system captures takes 500 images of the authenticated person within 2-3 minutes. Using the statement cv2. waitKey(100)&0xFF==ord('q'): break; the system will stop the program by entering ‘q’ key. Finally, when the programme is finished, we need to close the window and turn off the camera.

Next we perform the face training. By using a labelled image dataset we can train our recognizer. Face detection is performed by using Haar Cascade and recognizer helps to extract the features from the images. Import packages such as Open CV, os, numpy, Python Image Library (PIL) and import Image from PIL. PIL supports the image editing operations. Define a function get Images And Labels() with path to the dataset as parameter. This function will be return face Samples and Ids as a tuple. The detected faces and their corresponding labels will be included in face Samples and Ids respectively. Images are opened from the specified path and convert into gray scale form with the help of Image module. We can convert a PIL image into NumPy arrays by calling np. array() method. Then we need to detect the face object in the images by detect Multi Scale() with NumPy array as parameter. ROI image of this face is used to training the recognizer. Train function train() has two parameters, the images of faces and the corresponding labels assigned to these faces. After the recognizer has been trained, the data is saved into the file train.yml in the designated location.

Next we perform face testing. At first import all the packages Open CV, numpy and os. For face recognition, create Local Binary Pattern Histograms (LBPH). Load the trained model trainer.yml and prebuilt model haarcascade_frontalface_default.xml. Use cv2. Cascade Classifier() function to create classifier with path to the cascade xml file as its parameter. Read the video frame by using object of cv2. Video Capture() method. Convert the image into gray scale form by calling cv2.COLOR_BGR2GRAY() method. Detect Multi Scale(gray, 1.2,5) is used to get all face from the video frame. Draw rectangle around the face by using cv2. rectangle(). To predict the image belong to which ID system calls the function recognizer. predict(). If the returned ID is existing then the system displays the name above the rectangle using cv2. Put Text()method. When we press key ‘q’ the system will stop the program and finally we turn off the camera and close the window.

B. Hand Gesture Recognition System

First import all the necessary packages such as cv2, media pipe, pyautogui, os, sub process, PyQt5, loadUi, QDialog, QApplication, QLabel, QtGui, QPixmap, time and Image Grab. Next is the Hands model initialization. System uses the Hands model from mediapipe solutions to detect hands. Initialize the Hands class using mp. solutions. hands with a variable mp_hands. Configure the hands function to hold the landmark points. Parameters of this functions include model_complexity, min_detection_confidence, min_tracking_confidence. Model_complexity is set to 1 by default. The minimum confidence level required to consider a detection from a person-detection model as successful is specified using the min_detection_confidence variable and 0.5 is the default value. The minimum confidence level required to consider a detection from the landmark-tracking model as successful is specified using the min_tracking_confidence and 0.5 is the default value. The detected key points will automatically be drawn using Mp.solutions. drawing_utils, so we won't have to manually draw them. So initialize drawing utils for drawing landmarks on the image.

The image is then processed by the hands model, which detects the hands. Initially, we must use Open CV to continually capture camera frames. To do this, we create an object cap that captures video and takes a webcam's image as its input. It then converts the image from BGR to RGB using cv2. COLOR_BGR2RGB() method. This is due to the fact that MediaPipe only supports RGB, not BGR format Using cv2.flip(image, flip code), flip the image around the y-axis. Then processes this image to identify the hands in the image. We are now processing all of the model's results and illustrating and draw on image. Convert the image from RGB to BGR using cv2.COLOR_RGB2BGR() method. We initialize a variable fingerCount that will hold the fingers count. The multi_hand_landmarks field contains the estimated landmarks for each identified hand. If the values are present in this field, we will loop through each one and draw the points that were detect on the image. The following actions are taken for each hand found:

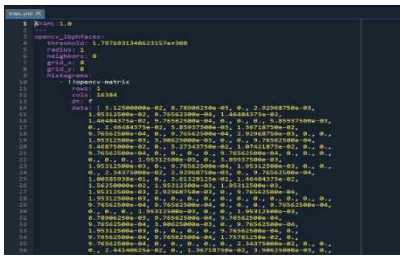
1) Check the hand label that was discovered
2) Save each landmark's x and y coordinates.
3) Check the coordinates of each finger to see if it has been raised to increase the count of fingers.
4) Use the draw_landmarks method to draw hand landmarks.

Check the label with your hand index (left or right). The left hand is labelled "right" and the right hand is labelled "left" because we are using the webcam input capture. To maintain the positions (x and y) of landmarks, set the list variable hand Landmarks. List each landmark's x and y coordinates in this list. There are two methods for determining whether a finger is raised: first the system will examine the hand label and the values of the THUMB_TIP and THUMB_IP x positions for the thumb. If the _TIP is to the right of the _IP for the left hand and the opposite for the right hand, the thumb is seen as being raised. Second check the values of the _TIP and _PIP y coordinates for the other fingers. If the _TIP is higher than the _PIP, the finger is regarded as raised. Now that we have the hand landmarks from the previous pre processing, we need to put the points on the image so that we can evaluate the performance of our hand landmarks recognition model. We will draw the landmarks on the image using the mp_drawing.draw_landmarks method from the drawing_utils package. We pass the image on which we want to create the landmarks as our first input. The list of hand landmarks is the second input that we pass. Then display the finger count using cv2. Put Text() method. In final stage we map these hand gestures to certain actions. The corresponding action is taken and performed based on these gestures mapping.

1) If the figure Count is 1, then mouse control action is performed. Cartesian coordinates in the form of X and Y are used to identify locations on your screen. On the left, the X coordinate is 0, and as it advances to the right, it increases. The Y coordinate goes from 0 at the top to increasing values as it descends. When you provide X and Y integer coordinates to the move To() function, it will move the mouse pointer to those locations.
2) If the figure Count is 2, then single click of the mouse is performed. A single left-button mouse click is simulated by the click() method at the mouse's current location. The definition of a "click" is to push the button down and then release it.
3) If the figure Count is 3, then selection in downward is performed. Call the press() function and supply a string from the pyautogui. KEYBOARD_KEYS, such as enter, esc, or f1, to accomplish this. Press down arrow key using Pyautogui.press('down').
4) If the figure Count is 4, then selection in upward is performed. pyautogui.press('up') press the up arrow key.
5) If the figure Count is 5, then we can take screenshots and automatically save it into the Screenshot folder. The Python Imaging Library, or PIL, gives the image editing features. The clipboard or screen contents can be copied to a PIL image memory using the Image Grab module. The screen is captured by the PIL. Image Grab.grab() method.
6) If the figure Count is 6, then we can open the google chrome browser and search data in web. Web browser Python module will be used to open a URL in a browser eg. Web browser.open(url).
7) If the figure Count is 7, then we can open Microsoft word application and work with a word document. The os module in Python allows us to interface with the operating system. This OS module gives us a function called start file() that opens a file by taking the file name as a string parameter.
8) If the figure Count is 8, then we can open the windows media player and play the music as our wish. First, we must import the VLC library. By calling sub process. call() function we can launch media player application and listen the songs.
9) If the fingureCount is 9, then we can open PowerPoint application and work with a powerpoint presentation. Similar to MS word application we can also launch a power point application by calling os.startfile() method.
10) If the figure Count is 10, then we can open Notepad. Application and work with a text document. Similar to MS word application we can also launch a Notepad application by calling os.startfile() method.

VI.RESULTS AND DISCUSSION

A unique identifier will be used to define the user's face while creating the dataset. From a real-time video, the system extracted 500 images of the user, grouping them all together under that unique ID. Images are scaled down to various pixel sizes and then converted to gray scale. Finally, with that specific userID, all of the images are kept in a directory called dataset. Fig. 4(a) shows a portion of the dataset collection. Images in the dataset are trained with LBPH algorithm and created a train.yml. Fig. 4(b) shows the portion of trainer.yml file.

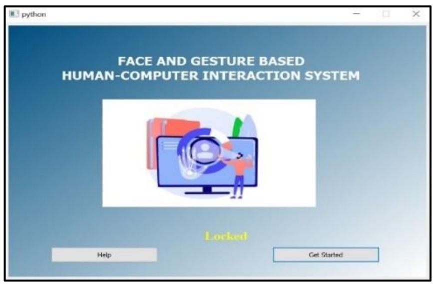


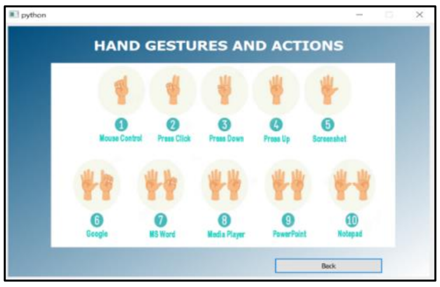
```
%YAML:1.0
---
opencv_lbphfaces:
   threshold: 1.7976931348623157e+308
   radius: 1
   neighbors: 8
   grid_x: 8
   grid_y: 8
   histograms:
      - !!opencv-matrix
         rows: 1
         cols: 16384
         dt: f
         data: [ 3.12500000e-02, 8.78906250e-03, 0., 2.92968750e-03,
             1.95312500e-02, 9.76562500e-04, 1.46484375e-02,
             1.46484375e-02, 9.76562500e-04, 0., 0., 0., 5.85937500e-03,
             0., 1.46484375e-02, 5.85937500e-03, 1.36718750e-02,
             9.76562500e-04, 0., 9.76562500e-04, 2.92968750e-03, 0., 0.,
             1.95312500e-03, 3.90625000e-03, 0., 0., 9.76562500e-04,
             5.46875000e-02, 0., 5.27343750e-02, 1.07421875e-02, 0., 0.,
             9.76562500e-04, 0., 0., 0., 0., 9.76562500e-04, 0., 0., 0.,
             0., 0., 0., 1.95312500e-03, 0., 5.85937500e-03,
             1.95312500e-03, 0., 9.76562500e-04, 1.95312500e-03, 0., 0.,
             0., 2.34375000e-02, 2.92968750e-03, 0., 9.76562500e-04,
             1.00585938e-01, 0., 3.61328125e-02, 1.46484375e-02,
             1.56250000e-02, 1.95312500e-03, 1.95312500e-03,
             1.95312500e-03, 2.92968750e-03, 0., 9.76562500e-04,
             1.95312500e-03, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
             9.76562500e-04, 9.76562500e-04, 0., 0., 0., 9.76562500e-04,
             0., 0., 0., 1.95312500e-03, 0., 0., 1.95312500e-03,
             8.78906250e-03, 9.76562500e-04, 9.76562500e-04,
             9.76562500e-04, 3.90625000e-03, 0., 9.76562500e-04,
             1.95312500e-03, 0., 0., 0., 0., 9.76562500e-04, 0.,
             9.76562500e-04, 9.76562500e-04, 1.75781250e-02, 0.,
             9.76562500e-04, 0., 0., 0., 0., 0., 2.34375000e-02, 0., 0.,
             0., 2.44140625e-02, 0., 1.36718750e-02, 3.90625000e-03, 0.,
```

(a) (b)

Fig. 4 Dataset Collection& Training (a) Portion of the dataset collection (b) Portion oftrainer.yml file

After training phase testing phase begins. For this we create a GUI using PyQt. Fig.5 (a) shows the home page of our systemwhich contains two options: One option for helping the user to know about the hand gestures and their actions mapping. Fig. 5 (b) shows the help page. Another option is for entering into our system.





(a) (b)

Fig. 5GUI using PyQt (a) Home page (b) Help page

When the button is pressed authentication begins. Here dataset image is compared with the real time image of the person who desires to get an access to the system. If a match is detected then a successful login is observed and goes to the hand gesture recognition system. If it is an unmatched then an illegitimate person is detected. 10 different gestures can detect by the system. Fig. 6 shows screenshot capturing using hand gesture recognition system.

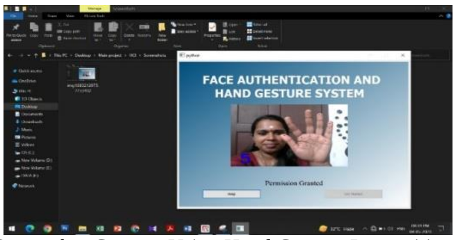


Fig. 6 Screenshot Capture Using Hand Gesture Recognition System

VII.CONCLUSION

HCI is crucial in designing intuitive interfaces that people with different abilities and expertise usually access. Most importantly, human-computer interaction is helpful for communities lacking knowledge and formal training on interacting with specific computing systems. To make interactions between people and computers as simple and efficient several methods has been proposed. When implementing these methods, hardware cost becomes an important issue. Gestures are useful for computer interaction since they are the most primary and expressive forms of human communication. This system develops a vision-based static hand gesture and face recognition human interactive system and maps the gesture with an appropriate task assigned for the particular gesture such as launching applications such as word, power point, notepad, media player, Google, some of the mouse actions, screenshot capturing. Face recognition access control allows a person to use their face to unlock the computer system.Here face detection and recognition are performed by using Haar-cascade classifiers and LBPH recognizers. The proposed HCI system not only can detect facial features can recognize hand gestures correctly anywhere in the whole image. System can efficiently extract hand regions and detect hand gestures using a Media Pipe library which is an open source cross-platform framework provided by Google for processing perceptual data from different modalities such as video and audio. A successful recognition of gesture with an accuracy of 98.2% and face recognition with an accuracy of 97% is expected through this system.

References

[1] *V.P. Kshirsagar, M.R. Baviskar and M.E. Gaikwad,"Face recognition using Eigenfaces", 3rd International Conference on Computer Research and Development,2011*

[2] *M.S. Bartlett, J.R. Movellan and T.J. Sejnowski,"Face recognition by independent component analysis", IEEE Transactions on Neural Networks ,Volume-13, Issue-6, 2002*

[3] *Suman Kumar Bhattacharyya and Kumar Rahul,"Face recognition by linear discriminant analysis", International Journal of Communication Network Security, ISSN: 2231 – 1882, Volume-2, Issue-2, 2013*

[4] *L. Wiskott, J.M. Fellous, N. Kruger and C. von der Malsburg,"Face Recognition by Elastic Bunch Graph Matching", IEEE Trans. On Pattern Analysis and Machine Intelligence, 19(7):775-779, 1996*

[5] *M. Anggo and La Arapu,"Face Recognition Using Fisherface Method",* Journal of Physics: Conference Series, *vol. 1028, no. 1, pp. 12119, 2018*

[6] *Priyal, S. Padam and Prabin K. Bora, "A study on static hand gesture recognition using moments", International Conference on Signal Processing and Communications (SPCOM). IEEE, 2010*

[7] *Li, Hui, Lei Yang, Xiaoyu Wu, ShengmiaoXu and Youwen Wang,"Static hand gesture recognition based on HOG with Kinect", 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, vol. 1, pp. 271-273. IEEE, 2012*

[8] *Li and Xingyan,"Gesture recognition based on fuzzy cmeans clustering al-gorithm", Department Of Computer Science The University Of Tennessee Knoxville, 2003*

[9] *Hasan, Mokhtar M. and Pramod K. Mishra,"HSV brightness factor matching for gesture recognition system", International Journal of Image Processing (IJIP) 4, no. 5 (2010): 456-467, 2010*

[10] *Nguyen, T.N., Vo, D.H., Huynh, H.H. and Meunier,"Geometry-based static hand gesture recognition using support vector machine", 13th International Conference on Control Automation Robotics and Vision (ICARCV), (pp. 769-774). IEEE, 2014*