Novel Illumination-Invariant Face Recognition Approach via Reflectance-Luminanceand Local Matching Model with Weighted Voting System

ABSTRACT

In this study, a novel approach has been introduced for face recognition that is unaffected by changes in illumination. This method is based on the reflectance-luminance model and incorporates local matching using a weighted voting technique to eliminate any artifacts present in the retina images. A total of 37 different linear and nonlinear filters were tested, including high pass and low pass filters, to achieve the reflectance component of the image, which remains invariant to changes in illumination. Among these filters, the maximum filter, which is a simple filter with low computational complexity, yielded the best results in extracting the illumination invariants. It was observed that the illumination invariants obtained through this method outperformed other methods such as QI, SQI, and image enhancement techniques in terms of recognition accuracy. Importantly, the proposed method does not require any prior knowledge about the facial shape or illumination conditions and can be applied to individual images independently. Unlike many existing methods, this approach does not rely on multiple images during the training stage and does not require any parameter selection to generate the illumination invariants. To further enhance the robustness of illumination, a weighted voting system was introduced. Certain regions of the image, which may adversely affect the recognition outcome due to poor illumination, occlusion, noise, or lack of distinctive information, were identified using predefined factors such as grayscale mean, image entropy, and mutual information. The proposed method was also compared to other face recognition methods in the presence of occlusions, and it demonstrated promising results outperforming existing methods. The Python algorithm successfully detects obstructed faces, genders, and ages in videos with a face matching accuracy between 80.9% to 96.9% based on proximity.

Keywords: Face Recognition, Haar Feature,Viola-Jones Method,Raspberry Pi.

1. INTRODUCTION

In today's networked world, the need to maintain the security of information or physical property is becoming increasingly important and challenging. With the rise of cybercrimes such as credit card fraud and computer break-ins, traditional access control systems based on ID cards, keys, and passwords are proving to be vulnerable to exploitation by criminals. One way of mitigating this fraudulent activity is to introduce the biometric verification technology, where it controls an automated method of verifying or recognizing the identity of a person based on physiological characteristics such as fingerprints, facial features, or behavioral patterns like handwriting or keystroke patterns. Unlike traditional access control systems that rely on external identity means, biometric systems identify individuals based on unique biological traits that are difficult to forge. This makes them highly effective in preventing unauthorized access to data or personal property.

Among the various biometric features, facial recognition stands out as one of the most important and widely used. Humans have a natural ability to recognize different faces with ease, a task that has proven to be challenging for automated face recognition systems.

Facial recognition technology relies on unique facial features to identify individuals accurately. It uses recognition technology which boasts exceptional accuracy rates compared to other biometric technologies such as fingerprint or iris scanners. The human face offers a vast amount of data points that can be used for identification, making it more precise and reliable. Additionally, facial recognition can work even if the person's face has undergone minor changes due to aging, facial hair, or different lighting conditions.One of the key advantages of facial recognition technology is its non-intrusive nature. Users can be identified simply by looking at a camera, eliminating the need for physical contact or manual input. This makes the process more user-friendly and convenient, especially in high-traffic areas where quick identification is essential.Implementing facial recognition technology can result in long-term cost savings for businesses and organizations. Due to its accuracy and efficiency, fewer resources are needed to manage security processes, reducing operational costs. Additionally, the scalability of facial recognition technology makes it a cost-effective solution for businesses of all sizes.

2. LITERATURE REVIEW

Automated facial recognition, a technology that emerged in the 1960s, was first developed by a group of researchers including Woody Bledsoe, Helen Chan Wolf, and Charles Bisson. Their pioneering work involved utilizing computers to identify human faces, marking the inception of early facial recognition endeavors. Referred to as the "man-machine" project, this initiative required a human operator to establish the coordinates of various facial features in a photograph before the computer could proceed with facial recognition processes. “Tasks such as pinpointing the pupil centers, the inner and outer corners of the eyes, and even identifying the widow's peak on the hairline were conducted on a graphics tablet to determine these coordinates accurately. Subsequently, these coordinates were utilized to compute a total of 20 distances, encompassing measurements like the mouth's width and the distance between the eyes. Approximately 40 images could be processed per hour by a human in this meticulous manner, enabling the creation of a comprehensive database containing the calculated distances. Upon completion of this manual input, a computer would autonomously compare the distances for each photograph, calculating the disparities between them and presenting the closest matches as potential identifications” [1-2].

“Purely feature-based methodologies for facial recognition were surpassed in the later years of the 1990s by the introduction of the Bochum system, a pioneering approach that leveraged the utilization of the Gabor filter for capturing facial features” [3-5]. “This innovative system also involved the computation of a structured grid mapping the intricate features of the face. The inception of the Bochum system marked a significant advancement in the field of facial recognition technology. Christoph von der Malsburg and his esteemed research team based at the University of Bochum made noteworthy strides in the mid-1990s with the development of Elastic Bunch Graph Matching, a sophisticated technique designed to extract facial features from images through the process of skin segmentation. By the year 1997, the groundbreaking face detection methodology devised by Malsburg had surpassed the capabilities of most other facial detection systems available in the market at that time" [6-7]. The renowned "Bochum system" for face detection was subsequently commercialized under the name ZN-Face, catering primarily to operators of airports and other high-traffic locations.

Review paper [23] demonstrates" a survey on challenging facial aspects and recent face detection methods.The advent of real-time face detection within video footage was made achievable in 2001 through the Viola-Jones object detection framework tailored for facial recognition and gesture detection". "Pioneered by Paul Viola and Michael Jones, this approach seamlessly integrated their proprietary face detection method with the Haar-like feature concept in image recognition, leading to the inception of AdaBoost - a revolutionary real-time frontal-view face detection system. Fast forward to the year 2015, the Viola-Jones algorithm had been successfully deployed on compact, low-power detectors embedded within handheld devices and various other systems. As a result, the Viola-Jones algorithm not only expanded the practical applications of facial recognition systems but also facilitated the integration of novel functionalities within user interfaces and teleconferencing platforms" [8-11].

Paper [22] presents a modified Viola-Jones algorithm for face recognition, focusing on handling face orientation variations and improving algorithm stability, with 3D images and comparative performance analysis. In [21], the research based on Viola-jones method outperformed four other machine learning algorithms in accuracy comparison.In this paper, the main objective is to design a real time face recognition system using reflectance luminance with Viola-Jones classified image.

3. METHODOLOGY

"Viola-Jones classified the image of the value of simple features and uses three types of features, which were square, three-square features, and a four-square feature. The value of these features was the difference between black and white regions. In each sub-window image, the total number of Feature Haar was very large, much larger if compared to the number of pixels. To ensure that classification could be done quickly, the learning process should eliminate most features available and focus on a small set of necessary features. AdaBoost aimed to form face templates" [8]. "The facial object was searched using Viola-Jones which refers to the plot as shown in Figure 2, where the grayscale image would be scanned per sub-window to look for positive features with AdaBoost and Cascade Classifier. If a face was detected, a rectangular image would be drawn on the face. The object's detection group casts an image based on the value of a simple feature. The basic operation of a feature was much faster than pixel processing. Some Haar Features represent the rectangular region of the image and add up all the pixels in the area. A classification method that was used for multiple levels of selection. At each level, it performs the selection using the AdaBoost algorithm that has been trained by using the Haar Feature. The selection was useful for separating between sub-windows containing positive objects (images that are detected to have the desired object) with negative objects (the detected images do not have the desired object)" [12]. The Viola-Jones method combined the general keys: Haar Like Feature, Integral Image, Adaboost learning, and Cascade classifier. Haar Like Feature was the difference in the number of pixels from the area inside the rectangle. An example of Haar Like Feature was presented in Figure 1.

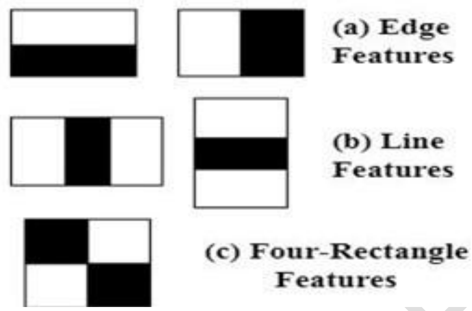


Fig. 1. Haar Like Feature

Haar Like Feature values were derived from the difference in the number of dark area pixel values by the number of bright area pixels:

$$F(Haar) = \sum F_{white} - \sum F_{black}$$

F(Haar) was the total feature value, ∑White was the feature value on the brightest area and ∑Fblack was the feature value on the dark area. Haar features are composed of either two or three rectangles. Face candidates are scanned and searched for Haar features of the current stage. The weight and size of each feature and the features themselves are generated using a machine learning algorithm from AdaBoost [9]. The integral image was a technique for calculating the feature value quickly by changing the value of each pixel into a new image representation, as shown in Figure 2.

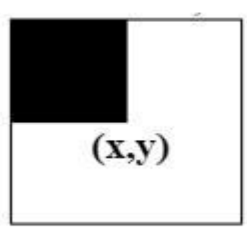


Fig. 2. Integral image (x,y).

The Adaboost learning algorithm is used to improve classification performance with simple learning to combine many weak classifiers into one powerful classifier. The weak classifier was a correct answer with a less accurate truth level. A weak classifier was stated below [14]:

$$h_j(x) = \begin{cases} 1, & if\, p_j f_j < p_j \theta_j(x) \\ 0, & other \end{cases}$$

where h j(x) was a weak classification, Pj is parity to j, θj was the threshold to j and x is a sub-image dimension such as 24 x 24. The steps to get a strong classifier were expressed in an algorithm as follows,

$$h(x) = \begin{cases} 1, \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 \qquad other \end{cases}, \text{ where } \alpha_t = \log \frac{1}{\beta_t}.$$

The Cascade classifier was a method to combine complex classifiers in a multilevel structure that could increase the speed of object detection by focusing on only possible imagery areas. The cascade classifier structure was presented in Figure 3.

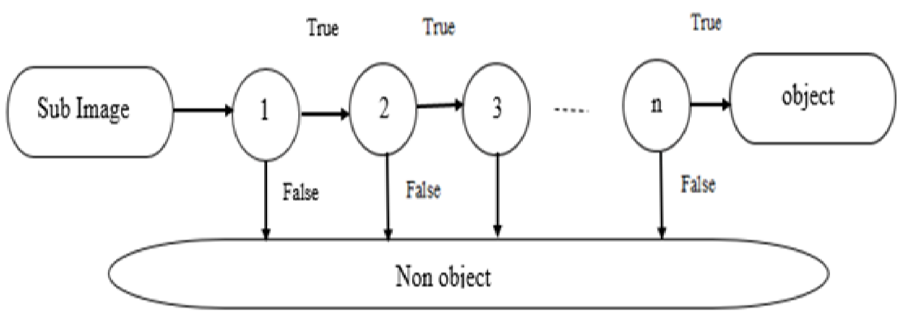


Fig. 3. Cascade classifier.

The methodology employed for face detection involved the application of the Viola-Jones algorithm, as illustrated in Figure 4. This diagram depicted the sequential process of face detection, starting from the initial stages to the final outcomes. Initially, the system would analyze a sample of the facial image captured by the camera. Subsequently, the Haar features of the image were computed by segmenting it into boxes to determine the contrast between dark and bright regions based on predefined thresholds. If the contrast exceeded the set threshold, the presence of a feature was affirmed. The integration of hundreds of Haar features across various scales was efficiently performed using the Integral Image technique. This approach involved the cumulative addition of pixel values in a systematic manner, commencing from the top left to the bottom right of the image. The selection of specific Haar features and establishment of threshold values were facilitated by a machine learning technique known as AdaBoost, which amalgamates multiple weak classifiers to form a robust classifier. These AdaBoost classifiers functioned as effective filter circuits for image classification purposes. Each filter, comprising a weak classifier or Haar filter, operated independently, with the inability of any filter to classify an image area resulting in its immediate exclusion as a non-face region. Conversely, successful passage through all filter processes within the circuit led to the classification of the image area as a face. After the filter stage was the cascade classifier, wherein the order of filters was determined by the weights assigned through AdaBoost. Priority was given to filters with higher weights, with the most significant weighted filter positioned at the forefront to expedite the elimination of non-face regions. The final stage involved the display of the detected object in the sample image, indicating whether a face was present or absent.

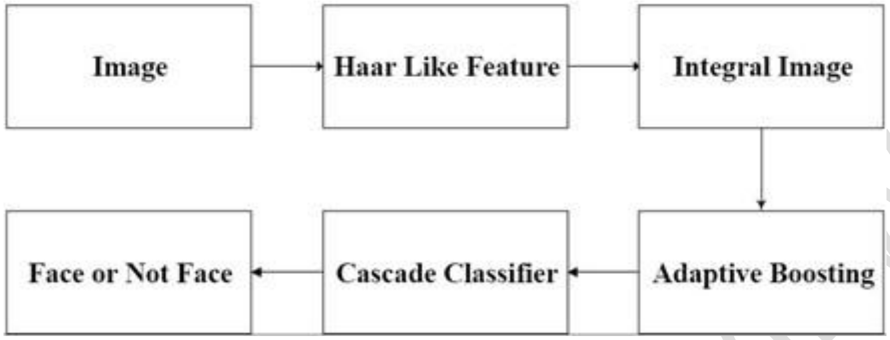


Fig. 4. The Cascade scheme classifier face detection process with Viola-Jones method.

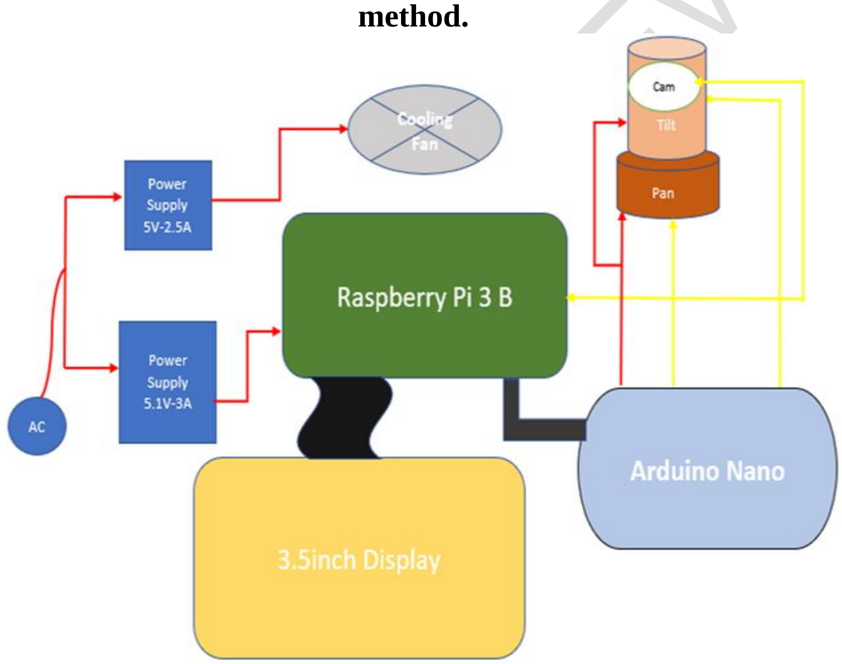


Fig. 5. Overall architecture of the project

The block diagram in figure 5 illustrates the design with two power supplies: one for the Raspberry Pi module and the other for the cooling fan, while the remaining components are linked to the processor module.

4. COMPONENT DESCRIPTION

Thefollowingcomponentswereusedduringtheexperimentalprocedure.

4.1 HARDWARE DESCRIPTION

i) Raspberry Pi 3 model B: The Raspberry Pi 3 Model B is the third generation of the Raspberry Pi series, featuring a 1.2GHz quad-core processor, 1GB of RAM, built-in Wi-Fi, and Bluetooth connectivity. [14-15]

Fig. 6. Raspberry Pi 3 model B.

ii) Arduino Nano: The Arduino Nano is a compact and versatile microcontroller board built around the ATmega328 (for the Arduino Nano 3.x version) or ATmega4806 (for the Arduino Nano Every version). With its small size and compatibility with breadboards, it offers similar capabilities to the Arduino Uno but in a different form factor. Notably, it doesn't include a DC power jack and utilizes a Mini-B USB cable for connectivity [13].

iii) Raspberry Pi Display: A Raspberry Pi display is a small screen that can be connected to your Raspberry Pi board to provide a visual output for your projects. It features a 3.5" display with 480x320 16-bit color pixels and a resistive touch overlay [16].

Fig. 7. Raspberry Pi display.

iv) Power Supply: This project requires two sources of power supply. One power module board is used for Raspberry Pi 3 module with 5V-3A rating. The other one is used for Pan Tilt camera platform with 5V-2.5A rating [17].

v) Pan Tilt camera platform: A pan tilt camera platform is a type of camera mount that allows for both horizontal (pan) and vertical (tilt) movement. These platforms are typically equipped with motors that allow for smooth and precise movement. The camera is mounted on top of the platform, and the motors can be controlled remotely. [18]

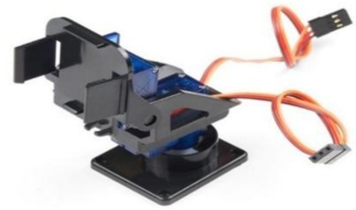


Fig. 8. Pan tilt for moving camera.

vii) Raspberry Pi Camera Module V2: The Camera v2 is the new official camera board released by the Raspberry Pi Foundation. The Raspberry Pi Camera Module v2 is a high-quality 8-megapixel Sony IMX219 image sensor custom-designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464-pixel static images, and supports 1080p30, 720p60 and 640x480p60/90 video[18].

Fig. 9. Pan tilt for moving camera.

viii) DC-5V 40mm Cooling Fan: A DC-5V 40mm cooling fan is a small but efficient fan that is designed to provide cooling solutions for the processor. It operates on a 5V power supply and has a compact 40mm size, making it ideal for tight spaces where traditional cooling solutions may not fit [19].

Fig. 10. DC-5V 40mm Cooling Fan.

ix) Memory Card: A memory card is a type of storage device that is used for storing media and data files. It provides a volatile and non-volatile medium to store data and files from the attached device.

x) Heatsink Kit: This is an aluminum heatsink kit used for cooling the chips on Raspberry Pi. Compared to other heat sinks, it has been attached with thermal conductive adhesive tape, allowing the Raspberry Pi to cool more efficiently. Comes with 3 plates, of which 2 small and 1 large. Dimensions: Large plate-14 x 14 x 6mm, Small plate- 8 x 8 x 4mm [20].

Fig. 11. Aluminum heatsink kit.

4.2OPERATING SYSTEM AND SOFTWARE DESCRIPTION

i) Raspbian: Raspbian is a lightweight operating system based on Debian Linux that is specifically designed for Raspberry Pi devices. It is known for being fast, reliable, and user-friendly, making it a popular choice among hobbyists and professionals alike.

ii) OpenCV: OpenCV is a powerful open-source computer vision library that allows developers to perform various image processing tasks with ease. One common task is smoothing images, which helps to reduce noise and improve the overall quality of an image.

iii) Python: Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

iv) Arduino IDE: The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus

4.3 IMPLEMENTATION

Thewholesetupcanbesegmented intotwoparts; theyarehardwareimplementationsandsoftwareimplementations.

4.3.1 <u>HARDWARE IMPLEMENTATION</u>

The power supply for the setup which is acquired from the AC source is divided into dual voltage switching power module as can be seen in figure no. The 5V-2.5A rated power supply module is used for the cooling fan and pan tilt camera platform; where the 5.1V-3A rated power supply module is connected to the Raspberry Pi 3 module using a USB port. The Arduino nano is connected to the Raspberry Pi. Then the pan tilt module is connected to the Arduino nano. The camera is connected to the Raspberry Pi module. The whole setup can be observed from the following block diagram.

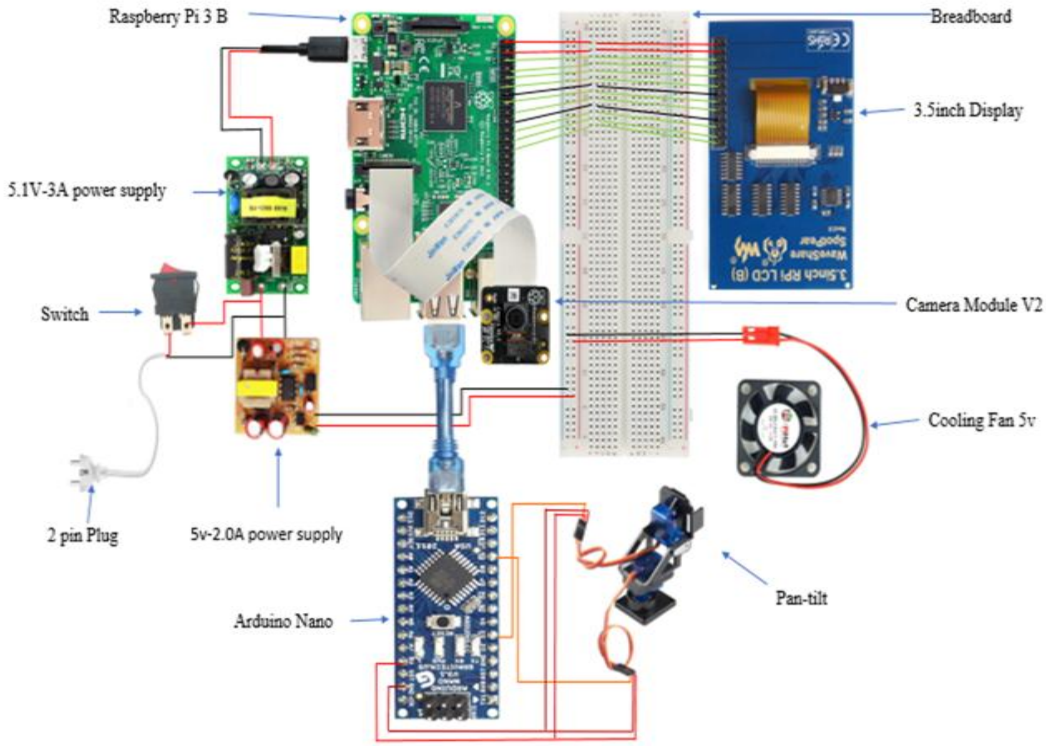
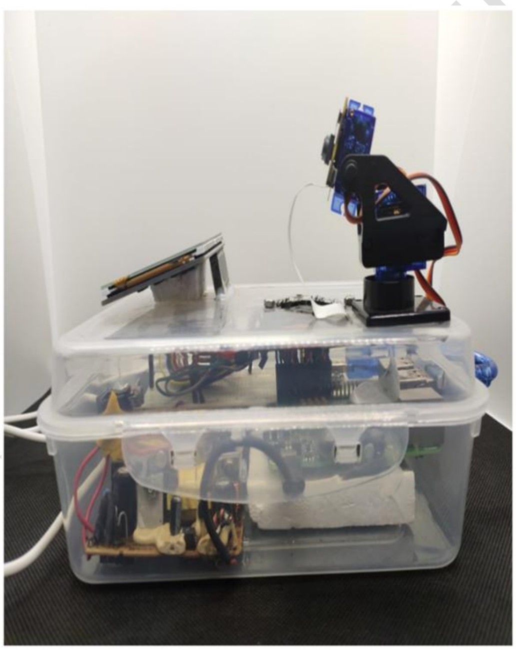


Fig. 12. Circuit component connection.



a. Side View of the project **b. Inside structure of the project**

Fig. 13. Full circuit component connection.

4.3.1 <u>SOFTWARE IMPLEMENTATION</u>

To make a face recognition program, at first, the recognizer has been trained with a dataset of previously captured faces along with its ID. For instance, there exist two individuals, whereby the initial individual is assigned the identification number 1, and the subsequent individual is assigned the identification number 2. As a result, all images depicting the first individual within the dataset will possess the identification number 1, while all images depicting the second individual within the dataset will possess the identification number 2. Subsequently, the dataset images are employed to train the recognizer, enabling it to accurately predict the identity of a newly presented face from a live video frame. Therefore, the program can be divided into four primary components: Dataset generator, Trainer, Face recognition, and Face tracking (specifically for pan-tilt functionality).

4.3.1.1 Dataset Generation

First a dataset is created by importing important library files. The dataset will acquire a limited number of facial samples from a singular individual within the live video frame. Subsequently, an identification code will be assigned to each sample, and they will be stored in a newly created folder. The folder will bear the name of the dataset. The dataset is programmed to take 30 samples of an individual from a live video feed from different angles.

```
while(True):
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceDetector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        sampleNum=sampleNum+1;
        cv2.imwrite('dataset/User.'+str(id)+"."+str(sampleNum)+".jpg",gray[y:y+h,x:x+w])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
        cv2.waitKey(100);

    cv2.imshow('frame',img);
    cv2.waitKey(1);
    if (sampleNum>30):
        break
```

Fig. 14. Collecting 30 samples of Face.

4.3.1.2 Training the Dataset

To perform face recognition, it is needed to train a face recognizer, using a pre-labeled dataset using the OpenCV.A python "trainner.py" file must be generated initially in the same directory where the dataset script was previously saved. Subsequently, a folder titled "trainer" is established in the identical location. This designated folder serves as the storage location for the recognizer after its training process.Now, inside this function, the following steps are done.

- Load the training images from the dataset folder.
- Capture the faces and Id from the training images.
- Put them In a List of Ids and ‘FaceSamples’ and return it.

Now, there is a requirement to iterate through the images by utilizing the image path. Consequently, the images along with their respective IDs will be loaded and incorporated into lists.

```python
#now looping through all the image paths and loading the Ids and the images
for imagePath in imagePaths:
    #loading the image and converting it to gray scale
    pilImage=Image.open(imagePath).convert('L')
    #Now we are converting the PIL image into numpy array
    imageNp=np.array(pilImage,'uint8')
    #getting the Id from the image
    ID=int(os.path.split(imagePath)[-1].split(".")[1])
    # extract the face from the training image sample
    faces.append(imageNp)
```

Fig. 15. The loop for converting image to gray scale and numpy array

4.3.1.3 Face Recognition

It is necessary to instantiate a recognizer object by utilizing the OpenCV library and importing the training data. Preceding this operation, it is obligatory to store the script in the identical location as the "trainer" folder.
To start the main loop, it is needed to do the following basic steps:

- First to start capturing frames from the camera object.
- Next convert it to Gray Scale.
- Then to detect and extract faces from the images.
- Next use the recognizer to recognize the ID of the user.
- Last to put predicted Id/Name and Rectangle on a detected face.

```python
while True:
    ret, im =cam.read()
    gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray, 1.2,5)
    for(x,y,w,h) in faces:
        cv2.rectangle(im,(x-20,y-20),(x+w+20,y+h+20),(0,255,0),2)
        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
        if(conf<65):
```

Fig. 16. Main loop for detecting and recognizing the face.

So, it's quite like the dataset code. The only difference there is add some new conditions. In the above two lines, the recognizer is predicting the user ID and confidence of the prediction respectively. In the next line, write the User ID on the screen below the face, which is (x, y+h) coordinate. Now with this, pretty much done adding some more finishing touches like it is showing user ID instead of the name and information.

4.3.1.3 Face Tracking with Pan-tilt Module

Face Tracking is started by importing the necessary packages. Now it is needed to create a cascade classifier using haar cascade for face detection and set the cascade file in the same folder. Now, the video capture object has been created. The data is transmitted to the Arduino Nano and the amount of data to be transmitted per second is set. When a person's face is recognized, the following command is printed and if it is not recognized, 'unknown' command is printed.

```
if(conf<80):

    if(Id==1):
        Id='Ahmed Hossain Siddiqui'
        Id2='EEE06207444'
        Id4= 'Project Member'
        Id3='Stamford University Bangladesh'
        text = "P={:.2f}%".format(conf)
        cv2.putText(im,str(Id),(x,y+h+40),font,1,(255,255,255),2,cv2.LINE_8)
        cv2.putText(im,str(Id2),(x,y+h+65),font,1,(255,255,255),2,cv2.LINE_8)
        cv2.putText(im,str(Id4),(440,25),font3,1,(255,255,255),2,cv2.LINE_8)
        cv2.putText(im,str(Id3),(x,y+h+85),font3,1,(255,255,255),2,cv2.LINE_8)
        cv2.putText(im,str(text),(13,30),font3,1,(0,255,0),2,cv2.LINE_8)
        cv2.putText(im,str(G),(13,50),font3,1,(0,255,0),2,cv2.LINE_8)
        cv2.putText(im,str(A1),(13,70),font3,1,(0,255,0),2,cv2.LINE_8)
```

Fig. 17. Person's information and print command.

5. RESULT AND DISCUSSION

5.1EXPERIMENTAL RESULT

At first, automatic the camera is turned on to capture that person's face and a maximum of 30 image samples are collected. After taking samples, whenever a person comes in front of the camera, it shows the identity from the dataset.

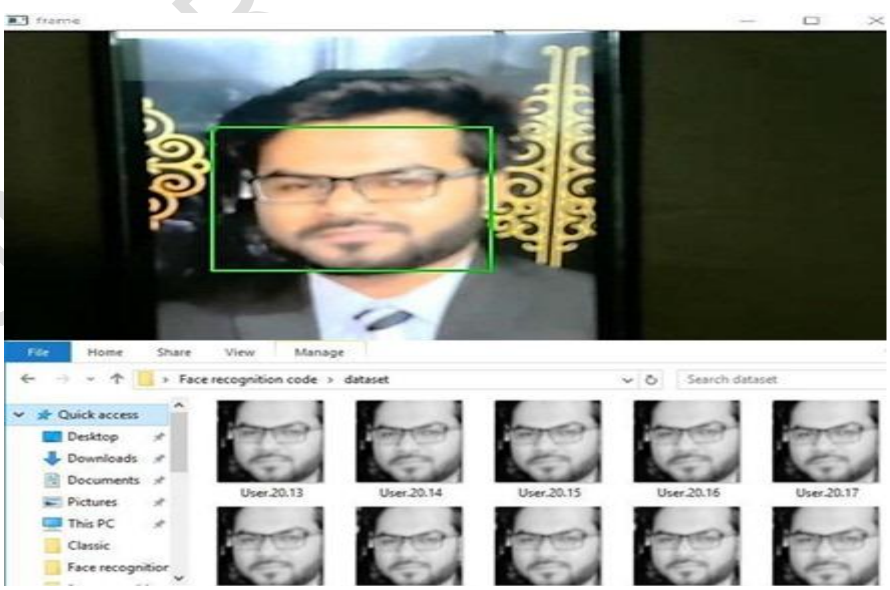


Fig. 18. Collecting Face Image from video feed and taking 30 samples.

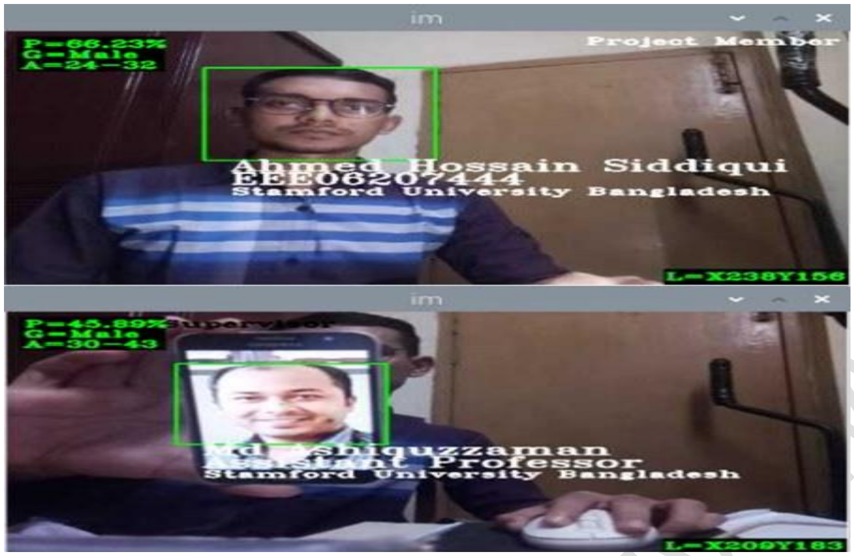


Fig. 19. Showing the identity of the project member.

5.2 RESULT COMPARISON

A higher rate of recognition was achieved by maximizing the number of images utilized as a reference for face matching and controlling the conditions under which the facial image retrieval image and test face image shared the same conditions. Moreover, when conducting face recognition using a webcam or pi cam, it is important to minimize interference from the user's surroundings. However, the accuracy of the Face Recognition system using the Viola-Jones method may decrease if the training images of the face have significant similarities. In comparison to other methods that achieve a tolerance angle of 0 - 5 degrees for face recognition, this project demonstrates an angle tolerance of up to 70 degrees, resulting in a 1:14 improvement ratio. In this project, the distance between the face and the camera was only 20-65 cm, but with the installation of a high megapixel camera, this range could be extended to 20 - 200 cm. The Python code developed for this project is also capable of recognizing obstructed faces, such as those wearing eyeglasses, and faces in low lighting conditions.

Table 1. Result Comparison

Picture	Example	Result	Comments
	Slant	Detected Lost	The front-facing position of the webcam with accuracy is 100% with a detection time of less than 1 second, while the maximum slope of ± 70 °.

	Light	Detected	In dark or bright lights: Still able to detect the presenceof objects.
	Distance	Detected	The closest distance ± 20cm, at a distance of ± 65cm could still be detected
	Stunted	Detected Lost	Able to detect facial objectseven if obstructed ifthe contour was the same asthe facial contour. Butifanything covers the face,itdoesn't recognize the face.

5.3 LIMITATION OF THE PROJECT

This project aims to determine a person's gender and age. Identifying gender and age from a video is challenging for both humans and algorithms due to reliance on appearance. The accuracy of the algorithm used is subpar as it fluctuates when the subject moves their face and is dependent on their appearance. This project achieves a face matching accuracy of 80.9%-96.9%, which can be adjusted by changing the distance. The script utilizes the haar cascade file for face detection to improve accuracy. At 63cm, a perfect accuracy of 96.9% is achieved. Increasing the distance reduces accuracy and may result in face loss. Using a high megapixel camera can minimize this error and provide clear long-distance face and object visualization.

5.4 FUTURE WORK

In future, security and ethical concerns can be included in the work, as of now no security system has not been integrated into this work. Paper [24] discussed ethical issues and privacy concerns in face recognition which can be implemented into this project work.

5.4 COST OF THE PROJECT

Table 2 illustrates the comprehensive expenditure of the research endeavor.It is evident that the overall cost incurred for the design of the prototype amounts to BDT 13,000.00 (equivalent to only thirteen thousand Bangladeshi takas), which is approximately us$115.

Table 2. Cost of Components

Equipment	Quantity (pcs)	Price (BDT)
Raspberry Pi 3 Model B	1	4575
3.5-inch RPi display	1	2400
Power Supply	1	500
Raspberry Pi Camera Module	1	3,200
Pan/tilt	1	700
DC-5V 40mm Cooling Fan	1	120
Breadboard	1	90
Memory Card	1	450
Jumper Wire	-	120
Heatsink kit for Raspberry Pi	1	90
AC 2 pin Plug	1	15
Switch	1	10
Box	1	200
Arduino Nano	1	500
Total Amount		13000

6. CONCLUSION

By utilizing a large number of reference images and controlling shared conditions, a high recognition rate was attained in face matching. Interference from user surroundings should be minimized when employing webcams or pi cams for face recognition. The Viola-Jones method's accuracy may decline with similar training face images. This project achieved an impressive 1:14 improvement ratio with an angle tolerance of up to 70 degrees, surpassing the 0 - 5 degrees tolerance of other methods. The developed Python code can identify obstructed faces, genders, and ages from videos, with a face matching accuracy ranging from 80.9% to 96.9% depending on distance and utilizing a haar cascade file for detection.

REFERENCES

1. Nilsson, Nils J. (2009). The Quest for Artificial Intelligence. Cambridge University Press.
2. De Leeuw, Karl; Bergstra, Jan (2007). The History of Information Security: A Comprehensive Handbook. Elsevier. p. 266.
3. Gates, Kelly (2011). The Biometric Future: Facial Recognition Technology and the Culture of Surveillance. NYU Press. pp. 48–49. ISBN 9780814732090. https://eholgersson.files.wordpress.com/2019/08/our-biometric-future-facial-recognition-technology-and-the-culture-of-surveillance.pdf

4. Malay K. Kundu; Sushmita Mitra; Debasis Mazumdar; Sankar K. Pal, eds. (2012). Perception and Machine Intelligence: First Indo-Japan Conference, PerMIn 2012,

Kolkata, India, January 12-13, 2011, Proceedings. Springer Science & Business Media. p. 29. ISBN 9783642273865

5. Wechsler, Harry (2009). Malay K. Kundu; Sushmita Mitra (eds.). Reliable Face Recognition Methods: System Design, Implementation and Evaluation. Springer Science & Business Media. pp. 11–12. ISBN 9780387384641.

6. Jun Wang; Laiwan Chan; DeLiang Wang, eds. (2012). Neural Information Processing: 13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006, Proceedings, Part II. Springer Science & Business Media. p. 198. ISBN 9783540464822

7. Li, Stan Z.; Jain, Anil K. (2005). Handbook of Face Recognition. Springer Science & Business Media. pp. 14–15. ISBN 9780387405957.

8. An application of viola jones method for face recognition for absence process efficiency (Journal of Physics) article: RudolfoRizkiDamanik et al 2018 J. Phys.: Conf. Ser. 1007 012013. https://www.researchgate.net/publication/324854259_An_application_of_viola_jones_method_for_face_recognition_for_absence_process_efficiency

9. A. HendroTriatmoko, S. HadiPramono, H. S. Dachlan, 2014, “Penggunaanmetode viola-jones danalgoritmaeigen eyes dalamsistemkehadiranpegawai”, Jurnal EECCIS Vol. 8, No. 1, Juni 2014 (use English version) https://jurnaleeccis.ub.ac.id/index.php/eeccis/article/view/234/207

10. Chen, S.K; Chang, Y.H (2014). 2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014). DEStech Publications, Inc.

11. Alam, S. S., Islam, N. N. A. J., & Ahammad, K. T. (2017). Hand gesture detection using haar classifier with appropriate skin color, kernel sizing & auto thresholding. *International Journal of Scientific & Engineering Research*, *8*(3), 98.

12. D. Ari Prasetya and I. Nurviyanto, 2012, “Deteksiwajahmetode viola jones padaopencvmenggunakanpemrograman python”, Simposium Nasional RAPI XI FT UMS.

13. Sheikh, Protik Parvez, et al. "Analysis of Patient Health Using Arduino and Monitoring System." Journal of Engineering Research and Reports 26.3 (2024): 25-33.

14. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

15. "Raspberry Pi Foundation - About Us". Raspberry Pi. Retrieved 23 August 2020.

16. https://www.elecrow.com/wiki/index.php?title=3.5_Inch_480x320_TFT_Display_with_ Touch_Screen_for_Raspberry_Pi#Feature

17. https://github.com/raspberrypi/documentation/blob/master/hardware/raspberrypi/power/ README.md

18. https://www.electronics.com.bd/rasberry-pi-camera-module-board-rev-28mpwebcam-video-1080p-720p-also-for-raspberry
19. https://bdspeedytech.com/index.php?route=product/product&product_id=214
20. https://bdspeedytech.com/index.php?route=product/product&product_id=675
21. Sheikh, Tasfia., Saha, Reno. (2022). Face Mask Detection Using Viola-Jones and Cascade Classifier. Available from: 10.1109/ICAISS55157.2022.10011114
22. Ж., Е., Амиргалиева. (2023). Development of a modified viola-jones algorithm for face recognition. Habaršy - Abajatyndaġy Almaty universitetì. Fizika-matematikaseriâsy, Available from: 10.51889/2022-1.1728-7901.08
23. Rana, Manish. (2022). Review on various face recognition databases. i-manager's journal on pattern recognition, Available from: 10.26634/jpr.9.2.19189
24. (2023). A Review Paper on Face Recognition Techniques. Available from: 10.1109/icaccs57279.2023.10112907