ISSN 2278-3091
Volume 12, No.4, July - August 2023
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse011242023.pdf
https://doi.org/10.30534/ijatcse/2023/011242023

Arduino Based Real-Time Face Recognition And Tracking System

Muneera Altayeb[1], Amani Al-Ghraibah[2]
[1] Al-Ahliyya Amman University, Amman, Jordan, m.altayeb@ammanu.edu.jo
[2] Al-Ahliyya Amman University, Amman, Jordan, a.ghraibah@ammanu.edu.jo

Received Date June 11, 2023 Accepted Date: July 22, 2023 Published Date: August 06, 2023

ABSTRACT

People must remember more passwords and carry more cards these days, as security becomes more important every day. However, such implementations are becoming less safe and feasible, resulting in a surge in interest in biometric systems techniques. Biometrics systems are those that record the physical characteristics of persons in an electronic environment and use that information to recognize them when needed. It works on the idea of identifying a person's physical traits that he cannot change and are distinguishable from others. The designed system aimed to successfully achieve face recognition through real-time detection, comparing the result with pre-recorded face images and tracking the human's face using a webcam-based system. The system accuracy has been tested in different cases; based on the detection distance, person's appearance variation, multi-angle posing and multi-people. The system successfully implements face recognition, detection, and tracking with a satisfactory accuracy of 93%.

Key words: Artificial Intelligence, Image processing**,** Face recognition, OpenCV.

1. INTRODUCTION

The majority of us use smartphones, which frequently use face recognition technology to unlock the device. This technology provides a powerful approach to protect personal data and ensure that sensitive data stays unavailable to the criminal even if the phone is stolen. Face recognition technology is used in many areas, such as safety, security, and payments. Face recognition refers to the difficulty of recognizing or validating a person in a digital image or video frame based on the facial biometric pattern. To authenticate a person, the system acquires a set of unique biometric data points related to facial expressions. face recognition is a type of personal identification system that uses a person's personal traits to identify them. The process of human face recognition is divided into two phases: face detection, which occurs relatively quickly in humans unless in situations when the item is positioned at a short distance away, and the second phase recognizes a face as an individual (Identifying the individual).

Biometrics is used in a facial recognition system to map facial features from a photograph or video. Facial recognition is a technology-based method for human face recognition and matching identification in which the system compares information to a database of known faces and can help verify a person's identity [1],[2]. Near Field Communication (NFC) technology and a mobile application are used in the attendance system as presented by Bhise, Abhilash, et al. Each student is given an NFC tag with a unique ID during their enrolment into the college. The lecturer's cell phone will then be used to track attendance for each lesson by clicking on or re-positioning these tags. The phone's built-in camera will then capture the student's face, allowing all data to be sent to the server for confirmation and verification. The advantages of this technology are the ease of use of NFC as it creates a fast connection between colleges and significantly reduces the time it takes for students to attend. The professor also found the convenience of a system that uses the lecturer's phone as an NFC reader, inconvenient. [3].

Priambodo, et. al introduced Face Tracking for Flying Robot which is created to follow a face object. The system is based on 2 stages: the face recognition stage and the position control system. Haar cascade approach is used as a face detection algorithm which is implemented using Python and OpenCV computer languages. A camera is used and installed on DJI Tello mini drone. and A Laptop is connected via Wi-Fi; the camera transmits the captured image of the detected person's face to the laptop to be processed using a computer vision algorithm. There are two basic network streams; It is a two-way duplex communication for sending and receiving requests between the laptop and the drone, and a one-way, half-duplex connection for streaming video from the drone to the laptop connection. The face detection system is written in Python and uses the Open-CV and djitellopy libraries. The location of the drone is then compared to the fixed point in the center of the picture to determine inaccuracies, allowing control signals for up/down, forward/backward, and right/left motions to be determined [4].

Bhaganagare, Bhakti, and Avinash D. Harale suggested in their research a security system that uses iris recognition to identify the person. Iris recognition is a biometric access control system that leverages the most unique feature of the human body, and it is used in automated border crossings, national ID systems, and other applications. The suggested technology displays information about recognized people

on an LCD, allowing users to enter the restricted area while remaining anonymous. The iris recognition data of the people are stored on personal computers and sent to the microcontroller through serial interface RS232, and used for serial communication between the PC and the microcontroller. The DC power supply required for the system and the Liquid Crystal Display (LCD) is used to display the person's status [5].

Setyawan, et. al proposes a system to make facial tracking resistant to occlusion and background characteristics. This face-tracking system is based on an enhanced Mean Shift (MS) algorithm which is used to monitor the face as a target and enhance the toughness of occlusion. Face detection and tracking are part of this algorithm which is Viola-Jones. The MS tracking technique is used to decrease occlusion, while corrected background-weighted histogram (CBWH) is utilized to reduce background interference; CBWH is a method used to increase MS tracking capability in object tracking location mode due to background noise [6].

Zarkasi, Ahmad, et al presented in their paper a technique for face recognition based on auto-extracted facial marks. It's based on extracting the face landmarks and a triplet of regions and their related geometric invariance. The points on the outer corners of the eyes and nose will subsequently be joined with lines to form a triangle then, the triangle's area is calculated. The gathered data is then trained to detect faces using the Support Vector Machine technique. This technology is implanted into a mobile robot using Raspberry Pi. The robot camera is mounted to a Pan Tilt servo, which serves as the robot's neck. If the camera detects a face, the robot calculates the area of the detected face rectangle and the center point of the rectangle to determine where it will go. The faces captured on camera are used as input, then, the picture is processed for face detection using the Haar Cascade approach. Then, the feature extraction using the Facial landmarks detection technique is performed, and the feature values acquired are compared to a feature database previously trained using the Support Vector Machine approach. The recognized face is the result of the software development [7].

Lian, Zhichao, et al introduced in their work a real time face tracking system based on multiple information fusion, which uses a multiple object tracking. Face detection is handled using a multi-task convolutional neural network (MTCNN). The appearance, motion, and shape features are used for tracking to deal with tracking failures that caused by object occlusion or fast object movement. The weights of different characteristics for feature fusion are adjusted depending on the tracking condition The approach of modifying feature weights based on scenes may handle problems like continuous tracking, interruptible tracking, and object interaction [8].

The novelty of this work lies in the mechanism of building and operating the real-time face recognition and tracking system that is guided by a new combination of Android application, Arduino, and digital image processing, The designed system aimed to achieve face recognition successfully by detecting a human face in real-time, comparing the result with pre-recorded face images and tracking the human's face using a webcam-based system. This system consists of Arduino Nano as a controller, a webcam, two servo motors to move the webcam, and a PC. The PC is used to create a database for facial images. This database is implemented using Python and PyCharm Integrated Development Environment (IDE).

2. HARDWARE REQUIREMENTS

In this project, the Arduino is the microcontroller of the system that performs all the decisions such as determining the direction of movement of the camera. The system is provided with two servo motors connected to the Arduino, one of which is responsible for horizontal movement and the other for vertical movement. Figure 1 describes the system block diagram while Figure 2 shows the servo motors connection. The basic elements required for this project are listed below

- Arduino Nano
- USB Webcam
- Servo motor
- Computer
- System chassis

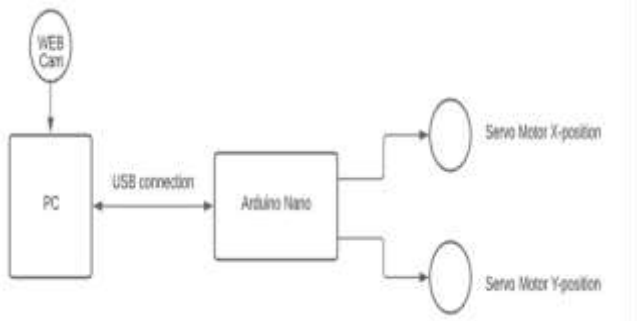

Figure 1: The system block diagram

The servo motors need 5 Volt to operate. Due to this, 220 Volts (AC) adapter is used which converts the AC power from the source and into DC power by decreasing the 220 Volts level and provides 12 Volts DC. Which then stepped down to 5 Volts by a Buck converter (LM2596) to provide the Arduino and the servo motors with the needed voltage. The buck converter is a DC-DC converter that provides an output voltage that is lower than the input voltage; it converts the uncontrolled DC supply to a lower output voltage that is stabilized. The power supply circuit shown in Figure 3.[9]-[11]

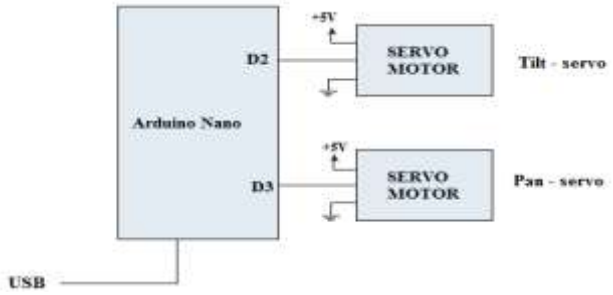

Figure 2: Schematic of the servo motors connection

The LM2596 buck converter is used in the project to stabilize the output voltage to 5V, it has excellent load and line regulation. Which has a high-precision potentiometer, and is capable of efficiently driving a load of up to 3A as shown in Figure 3. .

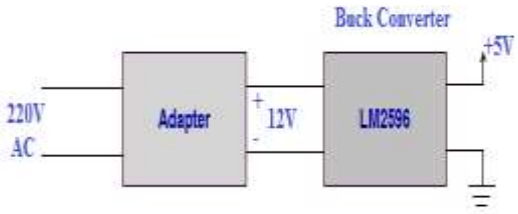

Figure 3: The power supply implemented circuit

3. SOFTWARE FRAMEWORK

In this part, the system is configured so that the Arduino is connected to the computer by an integrated development environment (IDE). Figure 4 shows the Arduino connections, so that the commands are sent from the computer to the Arduino, the steps for configuration the Arduino Nano are listed as follows[12]-[18]:

- Step 1: Installing Python on the Computer
- Step 2: Install PySerial
  PySerial is a Python API module which is used to read and write serial data to Arduino or any other Microcontroller.
- Step 3: Python Code
  First up, to get the Python sending data over the serial port, a simple program is needed.
- Step 4: Arduino Code
  To initiate a connection with the Arduino from Python, user should first figure out which COM Port the Arduino is on.

```
# Importing Libraries
import serial
import time
arduino = serial.Serial(port='COM4', baudrate=115200, timeout=.1)
def write_read(x):
    arduino.write(bytes(x, 'utf-8'))
    time.sleep(0.05)
    data = arduino.readline()
    return data
while True:
    num = input("Enter a number: ") # Taking input from user
    value = write_read(num)
    print(value) # printing the value
```

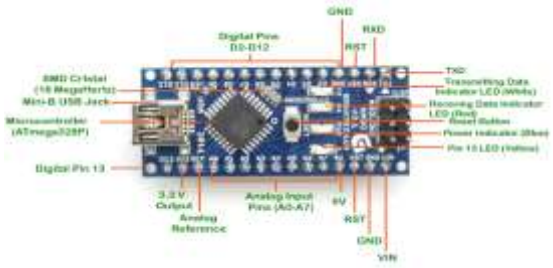

Figure 4: Arduino connection

3.1 System flow chart

The flow chart in Figure 5. describes the function of the system. When the Arduino is started, the system will receive commands from the computer via a USB connection, and then the camera will start capturing clips as the system detects faces in the captured images, then the system applies an algorithm to control the movement of camera to track faces according to the images stored in the database in the system.

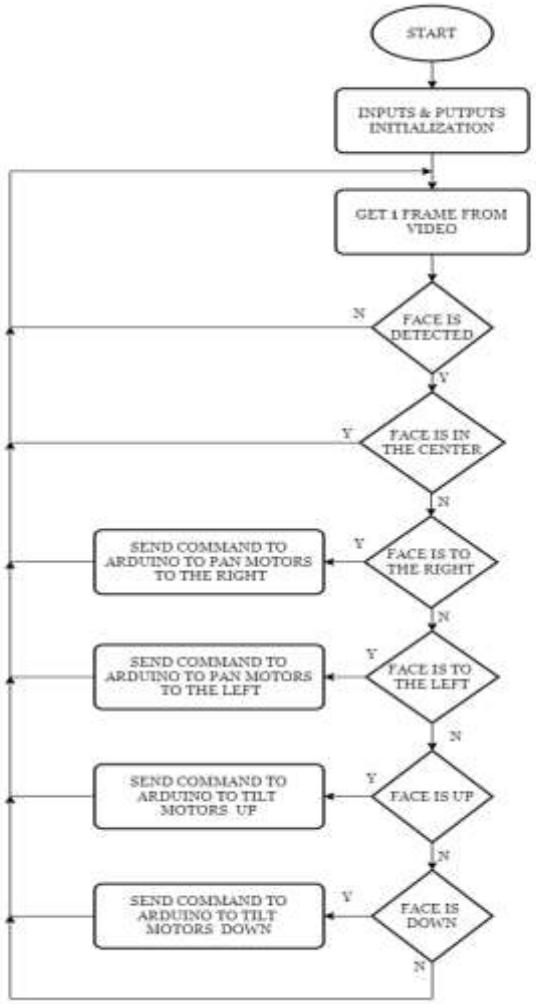

Figure 5: System general working flowchart.

The flowchart of controlling the servo motors by the Arduino is shown in Figure 6 and it's described as follow:

- If the Arduino receives "+" command, then it will pan the "pan- servo motor" 1 degree to the right.

- If the Arduino receives "-" command, then it will pan the "pan- servo motor" 1 degree to the left.

- If the Arduino receives "U" command, then it will tilt the "tilt- servo motor" up with 1 degree.

- If the Arduino receives "D" command, then it will tilt the "tilt-servo motor" down with 1 degree.

- If the Arduino receives "C" command, then it will return both the servos to the origin.

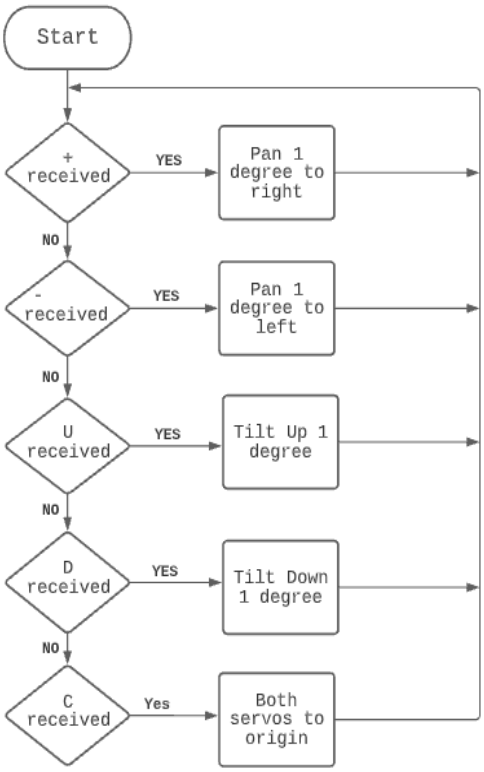

Figure 6: Servo motors control Flowchart

3.2 Image Processing

Image processing is accomplished via program PyCharm on the PC using Python. PyCharm is used in this project as an IDE to create Python-based database. This is the last part of implementing the project. It starts with downloading PyCharm program on the PC [19]-[22]. Then the images' database has been created, To create the facial images database, we should first create a folder that would include the images, then this folder's images will be encoded into the system[23] –[25]. Feed the frame to the system where the system runs an algorithm to detect the location of the face, to obtain the coordinates of the face in the frame and then the system applies the recognition steps as shown in Figure 8-b/c/d. After that a rectangle appears around the face and the name of the detected person as shown in figure 7-(e) or or "unknown" If the detected person is unknown, then the system checks if the image location is larger than the center, it would move to the right. otherwise, it would move to the left by sending serial commands (up, down, left, or right).

4. RESULT AND DISCUSSION

This system is really special as computer vision and artificial intelligence are used to detect and track faces. The accuracy of the system has been tested with a number of conditions such as distance, appearance variation, and disguise. The final System prototype of this project is shown in Figure 8.

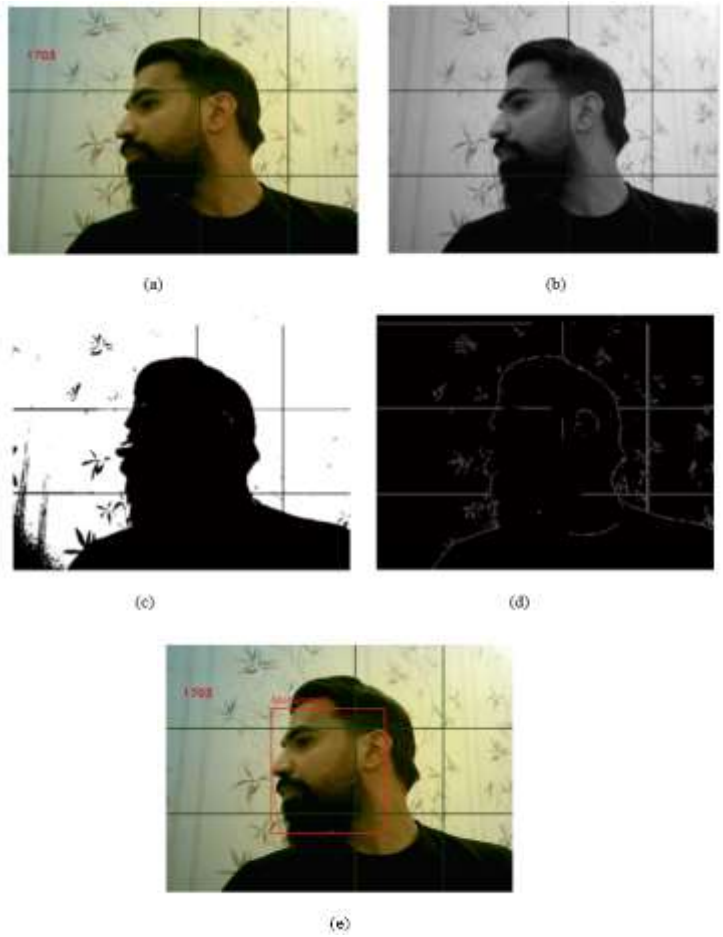

Figure 7: Face detection and labeling

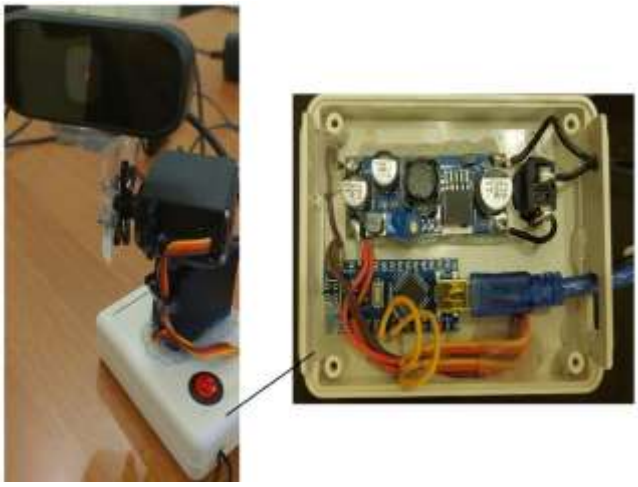

Figure 8: System prototype.

4.1 Face recognition accuracy based on detection distance

The aim of the system is to match a name to a specific face. First, the system has been tested 10 times to determine the algorithm accuracy at recognizing one person from a data set that contains multiple images of many people based on the detection distance. The result of each trial is expressed in Table 1. The face recognition accuracy can be calculated as the percentage of the recognized faces per the total number of tested faces of the same person/persons. In the

face recognition test based on the detection distance, the face recognition accuracy can be measured using eq. 1.:

$$\text{Accuracy} = \frac{no.of\ correctly\ recognized\ face}{total\ no.of\ face\ tested} * 100\% \quad (1)$$

Table 1: Accuracy obtained based on detection distance

Trial #	Detection Distance	Result
1	0.5 meter	succeeded in recognizing
2	1.5 meters	succeeded in recognizing
3	2.5 meters	succeeded in recognizing
Face recognition success result		10/10

When a person is in close proximity to the Cam (0.5 , 1.5, 2.5) meter away, the system succeeds in detecting his face and recognizing it as shown in Figure 9 .

Figure 9: (a) Person is 0.5 meter away ,(b) Person is 1.5 meter away, (c) Person is 2.5 meter away

4.2 Face recognition accuracy based on a person's appearance variation

The system has been tested based on a person's appearance variation, each change has been tested 10 times to determine the algorithm accuracy at recognizing one person in the image. The result of this test is expressed in Table 2. Figure 10 shows different cases tested according to a person's appearance variation.

Table 2: Accuracy obtained based on person's appearance variation

Appearance variation	Result	Face Recognition Accuracy
Without any additions	10/10	100%
With Eyeglasses & cap	9/10	90%
With Eyeglasses only	10/10	100%
With cap only	10/10	100%
With sunglasses only	9/10	90%
Accuracy average		90%

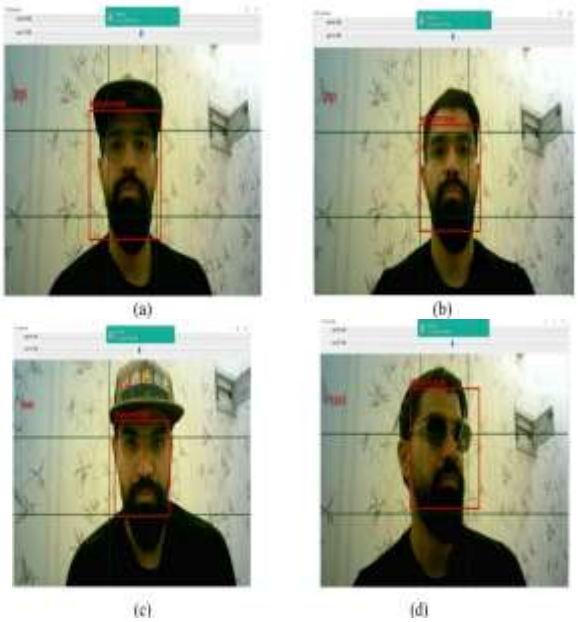

Figure10: (a)With Eyeglasses & cap, (b) With Eyeglasses only, (c) With cap only (d) With sunglasses only

.

4.3 Face Recognition Accuracy Based on Multi-Angle Pose

The system has been tested based on multi-angle pose as shown in Figure 11, each pose has been tested 10 times to determine the algorithm accuracy at recognizing one person in the image. The result of this test is expressed in Table 3.

Table 3: Accuracy obtained based on multi-angle pose

Angle Pose	Result	Face Recognition Accuracy
90˚	10/10	100%
135˚	10/10	100%
180˚	6/10	60%
Accuracy average		86.67%

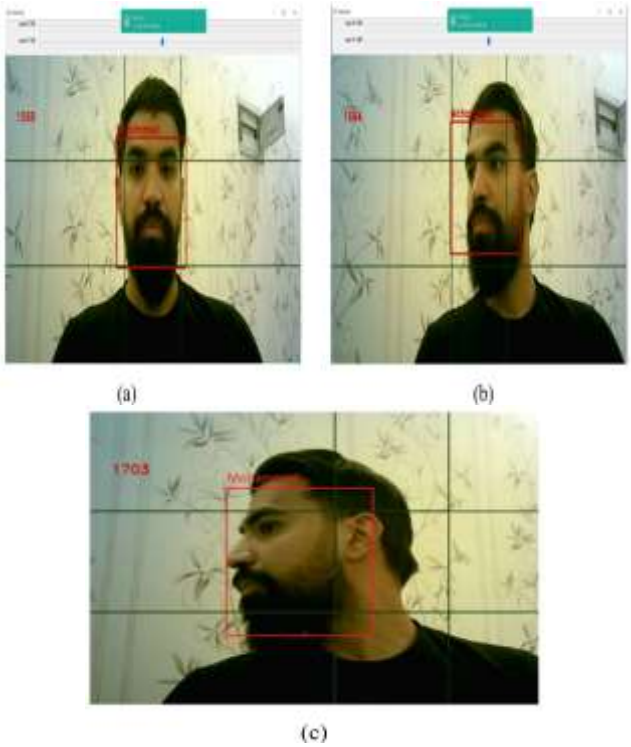

Figure 11:Face recognition test when the posing (a) Angle 90° (b) Angle is 135° ,(c) angle 180°.

4.4 Face Recognition Accuracy of Multi-People in One Image

In this case, the system has been tested when multi people are in one image, each case has been tested 10 times to determine the algorithm accuracy at recognizing faces in multi-people images. Figure 12 shows the system can successfully recognize the faces of multiple people in one image. The two people are known; they are included in the facial image dataset. The results of this test are expressed.

Table 4: Accuracy obtained based on in case of multi-people

Number of persons	Result	Face Recognition Accuracy
2 known persons	10/10	100%
1 known person with 1 unknown person	10/10	100%
3 persons: 2 known, 1 unknown	8/10	80%
Accuracy average		93.3%

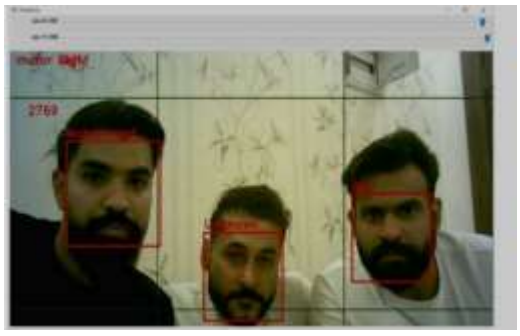

Figure 12: 2 people known, 1 unknown person

5. CONCLUSION

This work presented the process of designing Real Time Face Recognition and Tracking System. As this system works on comparing the result with pre-recorded face images and tracking the human's face using a webcam-based system. The novelty of this work lies in the idea of building and operating a different and unique system for real-time face recognition and tracking that is guided by a new combination of Android applications, Arduino and digital image processing, as well as system is fully functional and built from readily available components. It is worth noting that this system can be used in various applications such as surveillance systems at airports and the classroom attendance system, as well as many security and military uses. The system has been implemented and tested in different cases with a number of conditions such as distance, appearance variation and disguise. It proved that it has high accuracy in detecting individuals' faces recognize and tracking them; the performance was about 93%.

6. ACKNOWLEDGMENT

Special thanks to our students Mohammad Abulhasan ,Jassem Ibrahim Alsairafi and Ali Taleb Alblushi for contributing to this work by performing experiments to validate the accuracy of the system.

REFERENCES

1. Bhise, A., Khichi, R., Korde, A., & Lokare, P. D. (2015). **Attendance system using NFC technology with embedded camera on mobile device.** International Journal of Advanced Research in Computer and Communication Engineering, 4(2), 350-353.
2. Minaee, S., Luo, P., Lin, Z., & Bowyer, K. (2021). **Going deeper into face detection**: A survey. arXiv preprint arXiv:2103.14983.
3. Selvi, K. Senthamil, P. Chitrakala, and A. Antony Jenitha. "**Face recognition based attendance marking system**." International Journal of Computer Science and Mobile Computing 3.2 (2014): 337-342
4. Abioye, David & Omotehinwa, Temidayo Oluwatosin. (2018). **Design and Implementation of a Fingerprint Based Identity Management System.** 7. 75-80.
5. Devi, N. S., & Hemachandran, K. (2017). **Content based feature combination method for face image retrieval using neural network and SVM classifier for face recognition**. Indian Journal of Science and Technology, 10(24), 1-11.
6. Priambodo, A. S., Arifin, F., Nasuha, A., & Winursito, A. (2021, November). **Face Tracking for Flying Robot Quadcopter based on Haar Cascade Classifier and PID Controller**. In Journal of Physics: Conference Series (Vol. 2111, No. 1, p. 012046). IOP Publishing.

7. Ghoshal, A. M., Aspat, A., & Lemos, E. (2021). **OpenCV Image Processing for AI Pet Robot**. International Journal of Applied Sciences and Smart Technologies, 3(1), 65-82.
8. Zarkasi, A., Abdau, F., Anda, A. J., Nurmaini, S., Stiawan, D., Suprapto, B. Y., ... & Kurniati, R. (2022). **Implementation of Facial Landmarks Detection Method for Face Follower Mobile Robot**. Generic, 14(1), 19-24.
9. Altayeb, M., & Al-Ghraibah, A. (2022). **Voice controlled Camera Assisted Pick and Place Robot Using Raspberry Pi**. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 10(1), 51-59.
10. Renuka, B., Sivaranjani, B., Lakshmi, A. M., & Muthukumaran, D. N. (2018). **Automatic Enemy Detecting Defense Robot by using Face Detection Technique'**. Asian Journal of Applied Science and Technology, 2(2), 495-501.
11. Reddy, M. R., & SreelekhaAssistant, M. **IOT BASED FACE RECOGNITION AND DOOR LOCK SYSTEM USING ARDUINO**.
12. Hughes, J. M. (2016). **Arduino: a technical reference: a handbook for technicians, engineers, and makers**. " O'Reilly Media, Inc.".
13. Rathnala, P., Patnaik, M. P. K., Sura, S. R., Prasad, B., Siva, N., Rao, M., & Rayan, D. N. **Design of an Efficient Face Recognition system using Deep Learning Technique**.
14. Almufti, S. M., Marqas, R. B., Nayef, Z. A., & Mohamed, T. S. (2021). **Real time face-mask detection with arduino to prevent covid-19 Spreading**. Qubahan Academic Journal, 1(2), 39-46.
15. Joshi, P. (2015). **OpenCV with Python by example.** Packt Publishing Ltd.
16. Hasan, R. T., & Sallow, A. B. (2021). **Face Detection and Recognition Using OpenCV.** Journal of Soft Computing and Data Mining, 2(2), 86-97.
17. Dinalankara, L. (2017). **Face detection & face recognition using open computer vision classifies**. ResearchGate.
18. Ravishankar, V., Vinod, V., Kumar, T., & Bhalla, K. (2022). **Sensor integration and facial recognition deployment in a smart home system.** In Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2021 (pp. 759-771). Springer Singapore.
19. Reitz, K., & Schlusser, T. (2016). **The Hitchhiker's guide to Python: best practices for development**. " O'Reilly Media, Inc.".
20. El Mrabet, A., Tber, A., Benaly, M., Hlou, L., & El Gouri, R. (2023). **Unlocking Doors: A TinyML-based Approach for real-time Face Mask Detection in Door Lock Systems**. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 11(2), 515-527.
21. ben Abdel Ouahab, I., Elaachak, L., Bouhorma, M., & Alluhaidan, Y. A. (2021, June). **Real-time facemask detector using deep learning and raspberry pi**. In 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA) (pp. 23-30). IEEE.
22. Ismael, K. D., & Irina, S. (2020). **Face recognition using Viola-Jones depending on Python**. Indonesian Journal of Electrical Engineering and Computer Science, 20(3), 1513-1521.
23. Su, X., Gao, M., Ren, J., Li, Y., Dong, M., & Liu, X. (2022). **Face mask detection and classification via deep transfer learning.** Multimedia Tools and Applications, 1-20.
24. Zheng, G., & Xu, Y. (2021). **Efficient face detection and tracking in video sequences based on deep learning**. Information Sciences, 568, 265-285.
25. Mahbub, U., Sarkar, S., & Chellappa, R. (2019). **Partial face detection in the mobile domain. Image and Vision Computing**, 82, 1-17.