



## Development of a Face Tracker Using OpenCV and Arduino

Daniel Aliu<sup>1</sup>, Kumater Peter Ter<sup>2</sup>

<sup>1</sup>Department of Computer Engineering,  
Edo State University Uzairue, Nigeria

<sup>2</sup>Faculty of Air Engineering,

Air Force Institute of Technology Kaduna, Nigeria

### ABSTRACT

*This paper introduces a novel application that harnesses the capabilities of OpenCV's Haar cascade classifiers to achieve precise human face detection. This is complemented by the integration of a pan/tilt servomechanism, under the control of an Arduino UNO, to enable real-time face tracking. The proposed approach employs a multi-stage classification process that efficiently identifies human faces while effectively filtering out background elements. The coordinates of the detected faces, provided by OpenCV, are seamlessly transmitted to the Arduino UNO using pyserial, facilitating the necessary adjustments through servo motors. The collaborative pan/tilt mechanism exhibits dynamic responsiveness, allowing the camera to dynamically adjust by up to 2 degrees, ensuring a consistent alignment with the identified face. This amalgamation of computer vision and embedded systems exemplifies the synergy between the two domains, opening avenues for various applications such as surveillance and human-computer interaction. This convergence promises pioneering advancements in the realm of responsive automation and innovative technological development.*

### ARTICLE INFO

Article History

Received: March, 2023

Received in revised form: May, 2023

Accepted: May, 2023

Published online: June, 2023

### KEYWORDS

Haar Cascade, Face Detection, OpenCV,  
Arduino, Tracking

### INTRODUCTION

Biometric access control refers to automated techniques of validating or recognizing the identification of a living individual based on physiological traits such as fingerprints, facial features, or behavioural characteristics such as handwriting style or keyboard patterns (Pato & Millett, 2010). Biometric systems are difficult to counterfeit since they identify a person based on biological traits. Biometric technology provides various advantages over traditional identifying systems.

Face recognition and tracking have received a lot of attention in recent decades as the demand for security applications such as video surveillance and biometric surveillance has increased (Prasad & Kiran, 2021). The presence of a human face in a digital video frame is detected and tracked using face tracking technology. To find a human face, the computer must first capture an image with a camera and a frame-grabber, then process the image, scan for essential traits, and utilize these elements to calculate the location of the face (Bhanshe & Jaybhaye, 2018). To follow a human

face, the machine must not only locate a face but also find the same face across multiple photos. This requires the capability of the system to estimate motion while finding the face.

Furthermore, the system must control the camera, such as panning, tilting, and zooming, in order to monitor faces outside of a specific range. This technology is adaptable to computers and other devices. Despite its current success, there is still continuous research in this subject to improve the speed and accuracy of facial recognition systems (Adedokun et al., 2019; Salama AbdELminaam et al., 2020). Any face tracking system's accuracy is heavily dependent on the face detection system.

Biometric technologies are quickly becoming an essential component (gateway) for information security systems. As a result, biometric systems must meet stringent security criteria. Because biometrics provide the technology foundation for large-scale and highly sensitive identifying systems (e.g., passports, identification cards), the question of assessing

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved

the security of biometric technologies is an important one (Husztí-Orbán & Aoláin, 2020; Abdullahi et al., 2021). Other concerns with face detection and identification systems include individuals with identical faces, such as identical twins and others, in which case the system may make a mistake or error in processing the person's image in order to provide access to the rightful user (Paone et al., 2014).

### Feature Extraction

Feature extraction in machine learning, pattern recognition, and image processing begins with a set of initial measurement data and constructs derived values (features) designed to provide information and non-redundancy, thereby facilitating subsequent learning steps and generalization and, in some cases, leading to

better human results (Yu et al., 2023). Dimensionality reduction is related to feature extraction. When the algorithm's input data is too extensive to handle and is suspected of being redundant (for example, the same measurement of feet and meters, or the repeatability of an image reported in pixels), it can be turned into a smaller set of features (also known as feature vector) (Zebari et al., 2020). Feature selection refers to the process of determining the initial feature subset. The chosen features must include relevant information from the input data so that the required task can be completed using a simplified representation rather than the whole initial data (Huang et al., 2023). The algorithm in feature extraction uses training data to best identify features that it can consider a face (Figure 2.1).

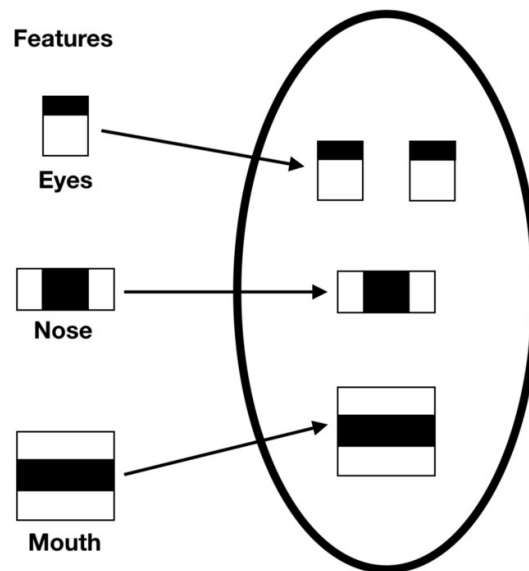


Figure 2.1: Theoretical face model (Alankar et al., 2021)

### Facial Detection

Face detection is one of the most difficult problems in image processing since it seeks to apply to all possible appearance variations caused by changes in illumination, occlusions, facial features, and so on (Zhang, 2023).

Face detection employs classifiers, which are algorithms that determine if an image contains a face (1) or does not contain a face (0). In order to improve accuracy, classifiers

have been trained to recognise faces using hundreds to millions of photos. Some of the factors that can affect the result of face detection algorithms are head pose (Chen et al., 2021), facial expression (Anwarul & Dahiya, 2020), image orientation (Li et al., 2020), occlusion (Thazheena & Aswathy Devi, 2017), and illumination (Rajeshwari et al., 2014).

### OpenCV

\*Corresponding author: Daniel Aliu  
[aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved

OpenCV is a large open-source library for computer vision, machine learning, and image processing, and it currently plays a significant part in real-time operation, which is critical in today's systems. It can process photos and movies to recognize items, faces, and even human handwriting. It is, therefore, feasible to analyze the OpenCV array structure for analysis such as face detection when it is combined with libraries such as Python's Numpy (TH. Hasan & Bibo Sallow, 2021). This is the approach used for this work.

### Haar Cascade

Haar Cascade is an object detection algorithm that is used to recognize faces in

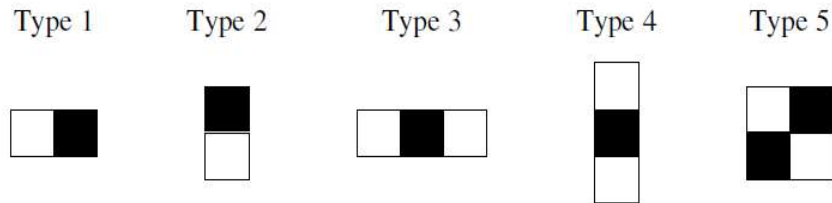


Figure 2.2: Types of Haar Features (Behera, 2020)

These characteristics can be difficult to identify in a huge image. This is where integral images come into play, because the integral image reduces the number of processes (Momoh et al., 2020).

### Calculating Integral Images

Calculating Integral Images is the second stage of applying the Haar Cascade. In an integral image the value at pixel (x,y) is the sum of pixels above and to the left of (x,y). Integral image allows for the calculation of sum of all pixels inside any given rectangle using only four values at the corners of the rectangle (Shetty et al., 2021).

### Using Adaboost

Adaboost selects the best features and trains classifiers to use them. It combines "weak classifiers" to produce a "strong classifier" that the algorithm may use to detect items.

Weak learners are generated by dragging a window over the input image and computing Haar features for each subsection. This difference is compared to a learned threshold that distinguishes between non-objects and

images or real-time video (Behera, 2020). The detection method is based on the wavelet template concept, which specifies the shape of an item in terms of a subset of wavelet coefficients. The algorithm can be explained in four stages.

### Calculating Haar Features

The first step is to collect the Haar features. A Haar feature (Figure 2.2) is essentially a set of calculations performed on consecutive rectangular sections in a detection window. The calculation is adding the pixel intensities in each location and then subtracting the amounts.

objects. Because these are "weak classifiers," it takes a huge number of Haar characteristics to construct a strong classifier (Wei et al., 2021).

### Cascading Classifiers

The final stage in implementing Haar Cascade is to use cascading classifiers to integrate the weak learners into a strong learner. The cascade classifier is composed of several stages, each of which contains a collection of weak learners. Boosting is used to train weak learners, resulting in a highly accurate classifier based on the mean prediction of all weak learners. The classifier determines whether to report that an object was found (positive) or to go on to the next region (negative) based on this prediction. Because the majority of the windows contain nothing of interest, stages are designed to reject negative samples as quickly as feasible.

### METHODOLOGY

The suggested system is designed to recognize human faces in real-time video utilizing OpenCV's Harr cascade classifiers and an Arduino UNO pan/tilt servomechanism to monitor the user's face. The flowchart in Figure

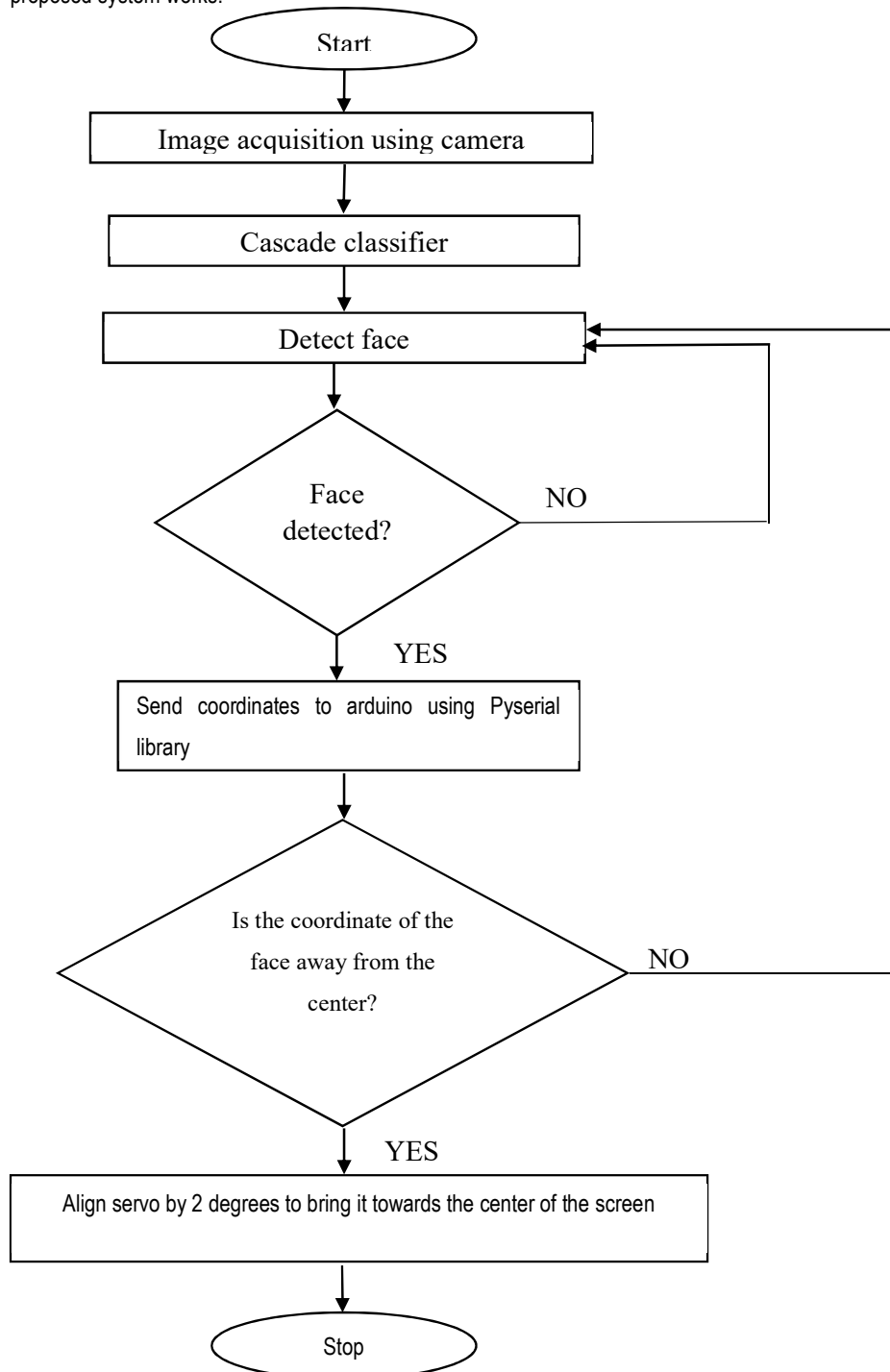
\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved

3.1 shows the step-by-step process in which the proposed system works.



**Figure 3.1:** Flow diagram for the proposed methodology

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved



### Face Detection and Feature Extraction

In detecting the face, "Haar cascades", a pre-trained model for detecting human faces was used. Pyserial and OpenCV libraries were also applied to the project, both of which were acquired and implemented using Python.

### Image Transformation

The image was loaded and transformed into grayscale. It was noted that images commonly observed are presented in the RGB format, consisting of Red, Green, and Blue channels. But for OpenCV to process the images, they would have to be stored in the BGR sequence, signifying Blue, Green, and Red channels. To suit the requirements of image recognition, a conversion from the BGR configuration to grayscale was deemed necessary. The primary motivation for adopting grayscale was its enhanced process ability and reduced computational intensity, stemming from its singular black-and-white channel.

### Feature Location

Following the successful conversion of the image from RGB to grayscale, the emphasis switched to the accurate recognition of facial features. The approach to achieve this included using an instantiated object known as the "face\_classifier," which was preloaded with the "haarcascade\_frontalface\_default.xml" data. To do this, an internal function called "detectMultiscale" was called. This function was used to locate distinguishing features or specific regions inside the freshly processed image. Its technique made extensive use of all of the features inherent in the "face\_classifier" object. This extensive set of features was then used to detect similar attributes within the image under consideration.

The following parameters were passed to the function:

1. The grayscale variable, sometimes known as "gray" in this context.
2. The option "scaleFactor," which determined how much the image was shrunk at each image scale. This scale factor was crucial in the development of the scale pyramid. A scaleFactor of 1:1 was used and it found to function properly for real-time face detection.

3. "minNeighbors," a parameter indicating how many neighbors each candidate rectangle should keep. This parameter was found to have an effect on the quality of the discovered faces. Choosing a higher number resulted in fewer detections but of higher quality. A number between 3 and 6 was deemed appropriate for this parameter. A minNeighbors value of 6 was chosen in this experiment, and it proved to be successful for real-time face detection.

### Calculating the Coordinates

OpenCV was discovered to offer face coordinates in the form of pixel values. The video resolution was initially set to 640 x 480 pixels, as observed. These coordinates included the top-left pixel values (x and y), as well as the detected face's height and breadth. The face's center coordinates were an important reference point, and determining them required a calculation:  $x + \text{width}/2$  and  $y + \text{height}/2$ . Notably, these estimated positions were represented visually as a green dot. These coordinates had an important role, as they were sent to the Arduino to allow camera angle modifications.

In addition, a distinguishing characteristic was included in the form of a white square in the center of the frame. This square indicated a precise place where the center of the face (the green dot) was intended to be. Any deviation of the green dot beyond the bounds of this square was meticulously watched during operation. If the green dot drifted outside the defined square zone during facial movement, the servo mechanism quickly intervened to realign the camera, ensuring the dot was reinstated within the designated area.

### Face Tracking

After the features were detected and extracted, the coordinate information was then used for tracking.

### Arduino and Python Interface

The Pyserial library was used to create a communication channel with the Arduino UNO. Python was in charge of transmitting the center coordinates as a single string during this operation. To illustrate, a string like "X100Y200" was formed, where 100 after

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved

the 'X' signified the center x-coordinate and 200 after the 'Y' denoted the center y-coordinate.

### Servo Configuration

The face tracking system included two rotary actuators, one for the X-Axis (pan) and one for the Y-Axis (tilt). The "pan" servo-controlled horizontal camera movement by

regulating the azimuth angle, while the "tilt" servo-controlled vertical camera repositioning by manipulating the elevation angle. Notably, the camera was mounted atop the vertically moveable servo (Figure 3.4), forming a critical link between both mechanical components (Figure 3.2).

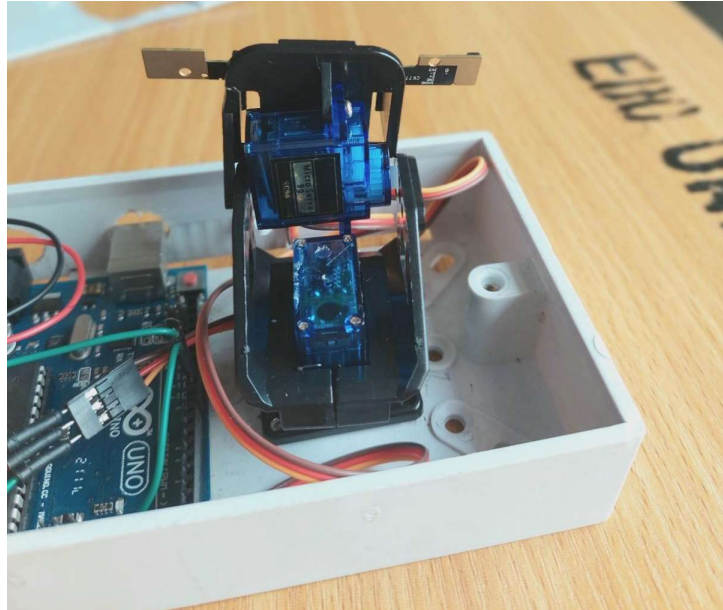


Figure 3.2: Servo Arrangement

### Servo to Arduino Connection

When connecting the servo motors to the Arduino UNO board, the procedure began with breadboard testing as shown in Figure 3.3, which was then transferred to a Vero board for greater reliability.

Furthermore, the Arduino UNO served a second purpose in this setup. It not

only facilitated servo operation, but it also acted as a power source. This entailed attaching the servo's positive terminal to the '5v port' on the Arduino UNO board, while the negative terminal was connected to the 'GND port,' ensuring a consistent power supply to the servos.

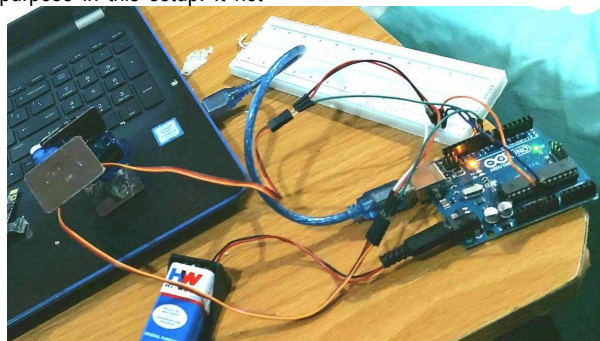


Figure 3.3: Arduino Servo Setup

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved

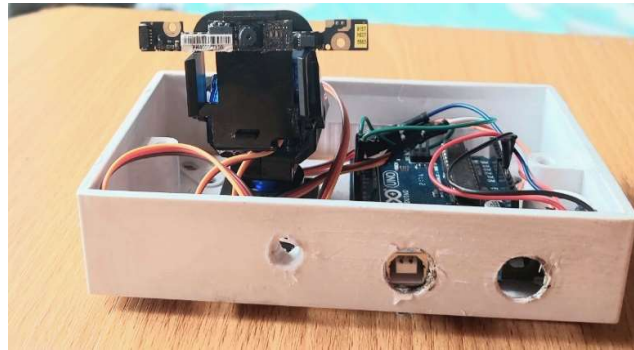


Figure 3.4: Camera Setup

## RESULTS

The given experimental results show that the system was capable of detecting frontal faces. The real-time face detection technology was put through extensive testing to determine its performance under various scenarios. The scenarios covered in the testing included:

1. Face Recognition: The technology was tested to determine when a face was correctly detected.
2. No Face Detection: In turn, the system was tested to see how it would react if no faces were recognized.

Figure 4.1 depicts a case in which the subject's face was effectively recognized, as evidenced by the proper alignment of the facial features. Figure 4.2, on the other hand, shows an instance where the subject's face was not properly aligned, resulting in partial capture of facial features. As a result, the system failed to detect the face in this case.

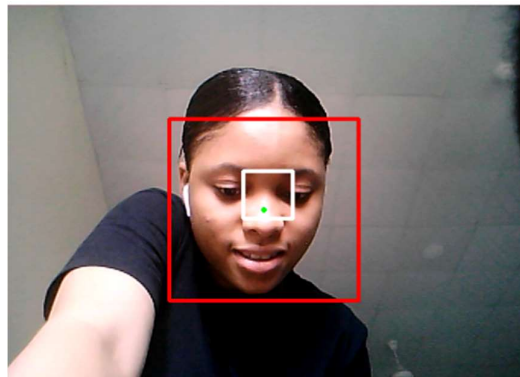


Figure 4.1: Image of a detected face

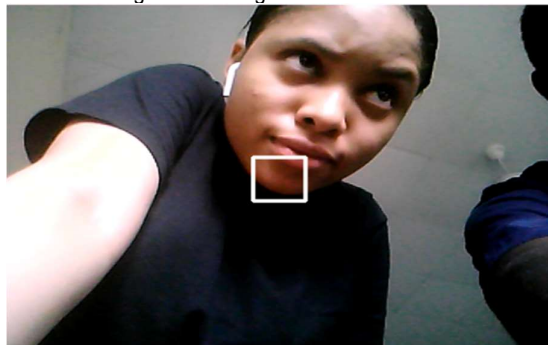


Figure 4.2: Image of a face that was not detected

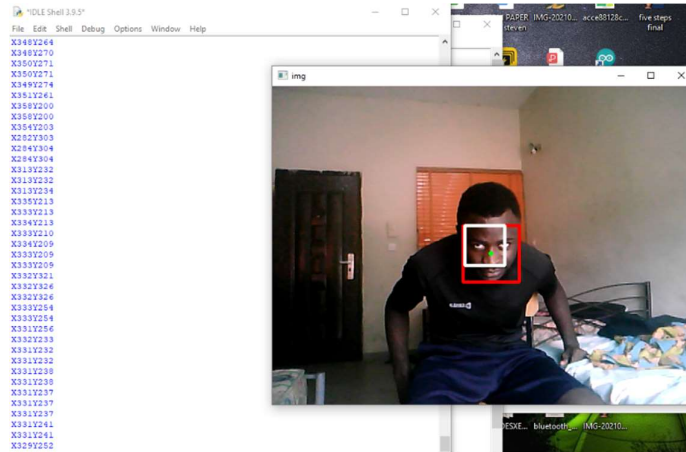
\*Corresponding author: Daniel Aliu  
[aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

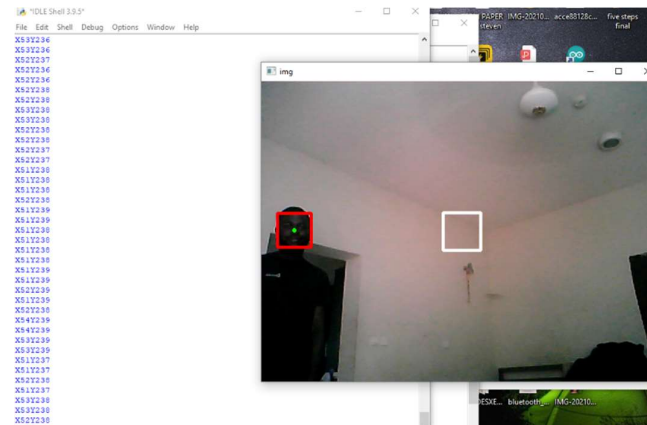
© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved

The real-time face tracking system was also tested to show when the face is being tracked as well as the coordinates. The test was conducted in a well-lit room. Figures 4.3 and 4.4

show the subject's face being tracked at various angles, as well as the accompanying X and Y-axis coordinates.



**Figure 4.3:** Image of a tracked face when the subject is sitting and the subject's face is pointing downward, along with the corresponding coordinates



**Figure 4.4:** Image of a tracked face when the subject is standing, along with the corresponding coordinates

### Test for Occlusion

This test was carried out under low-light conditions. The subject was wearing dark sunglasses in the first photograph in Figure 4.5a, and the Haar Cascade classifier could distinguish his face. The subject in Figure 4.5b had his hand over his mouth, yet the algorithm was still able to detect his face. The algorithm

could not detect the subject's face in Figure 4.5d, when he was wearing a facemask. There was no face detection in Figure 4.5f, which showed the subject completely hiding his face with his hands. It was discovered that the system could recognise a face mostly when the mouth or eyes were visible.

\*Corresponding author: Daniel Aliu  
[aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved



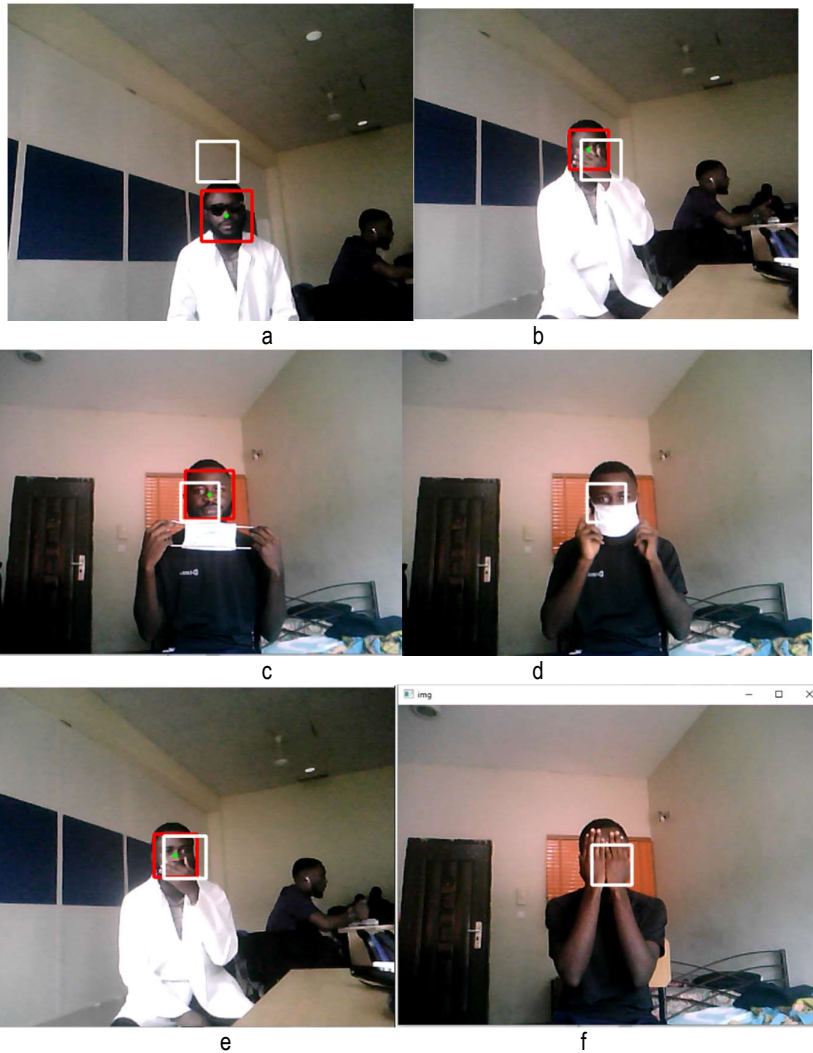


Figure 4.16: Test for occlusion

**Test for detection time and speed**

Table 4.1 shows the time and speed at which the face was detected for five different tests. The time was measured using a stopwatch. The average time was then calculated using:

$$\text{Average} = \frac{\text{sum of the values}}{\text{number of values}} = \frac{0.77+1.06+0.88+0.79+0.69}{5} = 0.84\text{s}$$

Table 4.1 also shows the speed at which the face was detected. The speed was calculated using:

$$\text{Speed} = \frac{\text{distance}}{\text{time}}$$

The distance used while conducting the test is the distance of the camera from the face was kept constant at 1.5m. The average speed in detecting the face is 1.806m/s.

**Table 4.1:** computational time and speed

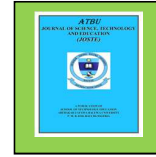
S/N	TEST	TIME(s)	SPEED(m/s)
1	1	0.77	1.75
2	2	1.06	1.41
3	3	0.88	1.70

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved



S/N	TEST	TIME(s)	SPEED(m/s)
4	4	0.79	2.0
5	5	0.69	2.17
	<b>Average</b>	<b>0.84</b>	<b>1.806</b>

## CONCLUSION

In this research work, the authors proposed a real-time face detection and tracking system. Face detection and tracking were mainly based on facial features. They used the Haar cascade algorithm for face detection and tracking. This algorithm is efficient and fast when compared to other algorithms. It is able to detect the frontal face(s) with a high detection rate and track the face successfully. However, the system is best used with plain backgrounds because of the rate of false detection when used on multicoloured backgrounds. The system was implemented using Python code, where the camera captured the face, the Haar cascade classifiers detected the face, and tracking was accomplished by moving the camera mounted on a servo pan tilt kit using an ARDUINO Microcontroller. The system can be implemented in various places where security and video surveillance is highly essential.

The authors concluded that the proposed system is a feasible and efficient solution for real-time face detection and tracking. The system is able to detect and track faces with high accuracy, even in challenging conditions. The authors also suggested that the system could be further improved by using more advanced algorithms and by incorporating additional features, such as eye detection and tracking.

## REFERENCES

- Abdullahi, I. M., Isah, O. R., Aliu, D., & Muyideen, O. M. (2021). A Smart Real-Time Standalone Route Recognition System for Visually Impaired Persons.
- Adedokun, A. E., Abdulrazak, M. B., Omuya, M. M., Bello-Salau, H., & Sadiq, B. O. (2019). A spatio-temporal based frame indexing algorithm for qos improvement in live low-motion video streaming. *ATBU Journal of Science, Technology and Education*, 7(3), 305-315.
- Alankar, B., Ammar, M. S., & Kaur, H. (2021). *Facial Emotion Detection Using Deep Learning and Haar Cascade Face Identification Algorithm* (pp. 163–180). [https://doi.org/10.1007/978-981-16-0695-3\\_17](https://doi.org/10.1007/978-981-16-0695-3_17)
- Anwarul, S., & Dahiya, S. (2020). A comprehensive review on face recognition methods and factors affecting facial recognition accuracy. *Proceedings of ICRIC 2019: Recent Innovations in Computing*, 495–514.
- Behera, G. S. (2020, December). *Face Detection with Haar Cascade*. Towards Data Science.
- Bhanshe, V. K., & Jaybhaye, M. D. (2018). Face Detection and tracking using Image processing on Raspberry Pi. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1099–1103.
- Chen, S., Zhang, Y., Yin, B., & Wang, B. (2021). TRFH: towards real-time face detection and head pose estimation. *Pattern Analysis and Applications*, 24, 1745–1755.
- Huang, X., Dong, R., Wu, B., Sato, K., Ikuno, S., Wang, Z., & Nishimura, S. (2023). A real-time recognition gait framework for personal authentication via image-based neural network: accelerated by feature reduction in time and frequency domains. *Journal of Real-Time Image Processing*, 20(5), 92. <https://doi.org/10.1007/s11554-023-01349-w>
- Husztí-Orbán, K., & Aoláin, F. N. (2020). *Use of Biometric Data to Identify Terrorists: Best Practice or Risky Business?* Human Rights Center, University of Minnesota Minneapolis.
- Li, L., Mu, X., Li, S., & Peng, H. (2020). A review of face recognition technology. *IEEE Access*, 8, 139110–139120.
- Momoh, M. O., Ndunagu, J. N., Abanihi, V. K., Yusuf, S. M., & Abdullah, M. I. (2020). A Comparative Analysis of Video Compression Standard and Algorithms: State of the Art. *ATBU Journal of*

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved



- Science, Technology and Education*, 8(2), 181-188.
- Paone, J. R., Flynn, P. J., Philips, P. J., Bowyer, K. W., Bruegge, R. W. V., Grother, P. J., Quinn, G. W., Pruitt, M. T., & Grant, J. M. (2014). Double Trouble: Differentiating Identical Twins by Face Recognition. *IEEE Transactions on Information Forensics and Security*, 9(2), 285–295. <https://doi.org/10.1109/TIFS.2013.2296373>
- Pato, J., & Millett, L. I. (2010). *Biometric Recognition*. National Academies Press. <https://doi.org/10.17226/12720>
- Prasad, M. V. D., & Kiran, N. S. (2021). Video surveillance-based security system using OpenCV and Arduino uno. *NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal* | NVEO, 1522–1528.
- Rajeshwari, J., Karibasappa, K., & GopalKrishna, M. T. (2014). Survey on skin based face detection on different illumination, poses and occlusion. *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, 728–733.
- Salama AbdELminaam, D., Almansori, A. M., Taha, M., & Badr, E. (2020). A deep facial recognition system using computational intelligent algorithms. *PLOS ONE*, 15(12), e0242269. <https://doi.org/10.1371/journal.pone.0242269>
- Shetty, A. B., Bhoomika, Deeksha, Rebeiro, J., & Ramyashree. (2021). Facial recognition using Haar cascade and LBP classifiers. *Global Transitions Proceedings*, 2(2), 330–335. <https://doi.org/10.1016/j.gltp.2021.08.044>
- TH. Hasan, R., & Bibo Sallow, A. (2021). Face Detection and Recognition Using OpenCV. *Journal of Soft Computing and Data Mining*, 2(2). <https://doi.org/10.30880/jscdm.2021.02.02.008>
- Thazheena, T., & Aswathy Devi, T. (2017). A review on face detection under occlusion by facial accessories. *International Research Journal of Engineering and Technology (IRJET)*, 4, 672–674.
- Wei, S., Zhu, L., Chen, L., & Lin, Q. (2021). An AdaBoost-Based Intelligent Driving Algorithm for Heavy-Haul Trains. *Actuators*, 10(8), 188. <https://doi.org/10.3390/act10080188>
- Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., & Gao, M. (2023). Techniques and challenges of image segmentation: A review. *Electronics*, 12(5), 1199.
- Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, 1(2), 56–70. <https://doi.org/10.38094/jastt1224>
- Zhang, N. (2023). A Study on the Impact of Face Image Quality on Face Recognition in the Wild. *ArXiv Preprint ArXiv: 2307.02679*.

\*Corresponding author: Daniel Aliu

✉ [aliu.daniel@edouniversity.edu.ng](mailto:aliu.daniel@edouniversity.edu.ng)

Department of Computer Engineering Department, Edo State University Uzairue, Nigeria.

© 2022 Faculty of Technology Education, ATBU Bauchi. All rights reserved