



Attendance Based Systems for Face Recognition using a Camera Rail Approach

Ubong Essien^{1*}, Godwin Ansa¹

Received: January 14, 2023 /Accepted: February 15, 2023

© REJOST 2023

Abstract

Many research works on face recognition-based attendance systems have employed either a single camera or multi camera architecture in an attempt to capture and produce images that cover the entire participants of a class. Single stationary camera architectures have shown to have blind spots. This paper proposes a cost-effective single camera rail system that can be adaptive and implemented with any face recognition system designed to greatly improve audience coverage. It can be deployed in various audience locations and managed centrally. It also can be implemented in either wired or wireless modes. The system is a microcontroller-based (Embedded Systems Programming (ESP) microcontroller ESP 8266) system that consists of a single camera placed on a rail to allow for capturing of images of different sections of a class. The microcontroller determines each stop position of the camera by calculating the stop positions based

on the configuration data (field of view angle, the maximum object distance and the rail length) stored in the MYSQL database. The stop positions are each passed to the stepper motor to pull the camera on the rails with the aid of a beaded string. The camera feed is captured programmatically using the Open-Source Computer Vision Library (Open CV). The frame at each stop position is automatically saved within a folder created for the class. Test results show that the entire participants of a class were covered and captured with stop positions calculated by the system and depending on the size of the class, the system can comfortably calculate stop positions that can cover the class attendees and the images captured was successfully transferred to the face recognition system for further processing.

Keywords: Angular field of view; microcontroller; camera architectures; object distance; stop positions

✉ Ubong Essien: ubongessien@aksu.edu.ng

¹ Department of Computer Science, Akwa Ibom State University, Mkpato Enin, Akwa Ibom State, Nigeria.

Introduction

A meta-analysis conducted by Credé *et al.* (2010) reveal that attendance has a positive

effect on both course grades and Grade Point Average (GPA) and is the single strongest predictor of college grades. According to Gomes *et al.* (2020), face recognition-based attendance systems are rapidly outperforming other forms of biometrics (Fingerprints, Radio frequency identification, etc.) because they use a set of features unique to a specific individual. Most face recognition-based attendance systems have employed basically two architectures for capturing attendees which either produces a video feed or still images. The architectures are:

- i. Single camera architecture
- ii. Multi camera architecture

A single camera architecture system is a system that uses one camera to capture the audience while a multi camera architecture system is a system, which uses more than one camera. This paper proposes a single camera rail system for use with face recognition-based attendance systems. This system is a microcontroller-based system that can be implemented both in wired or wireless modes. This system is based on the camera's field of view and works by moving the camera on a rail to different positions to capture different sections of a class attendees.

The single camera architecture is limited by the field of view of the camera used. This makes some of the designs implemented in some research to be suited only for small class space with numerous assumptions regarding the positions of the students. A multiple camera architecture is expensive and the systems must account for camera collaboration, information fusion, and object matching (Mothwa and Mapati, 2018).

Related Literature

Tribedi *et al.* (2022) and Kalel *et al.* (2022) proposed an Automated Attendance Management System where images are captured from a webcam and detected using the Viola-Jones technique, the detected face is resized to the desired size. The face recognition process is carried out using a basic Local Binary Patterns Histogram algorithm. After the

recognition is completed, the attendance is immediately updated in an SQLite database with the relevant information. The major issue with the design of this system was that it can only capture one student at a time and time taken to process the whole images captured from different webcams grows as the number of students grows.

Baig *et al.* (2022) designed an effective and secure technique for personal authentication using facial recognition. In this research, Open CV was used to capture the images from an android phone camera (single camera) and sent to the face recognition for processing using Convolutional Neural Network (CNN). The attendance was marked on an excel sheet. This system is personal, will only capture and account for one student and will not be fool proof as students may not be physically available in the class but can logon to the face recognition server and attendance will be marked.

Fuzail *et al.* (2020) proposed integrating real-time face detection algorithms into an existing Learning Management System (LMS). This system automatically detects and registers students' attendance during a lecture. As shown in Figure 1, this face recognition-based system scans the class and captures frontal images using Eigen faces and a single camera architecture with rotating capability.

Satpute *et al.* (2020) used Open-source Computer Vision Library (Open CV) in their project. Python was used to create the application. A single camera architecture was used for this project as camera module was installed on a Raspberry Pi. The Raspberry Pi serves as the project's central processing unit. The Open CV and the camera are both installed on the Raspberry Pi.

Gomes *et al.* (2020) proposed a system that uses a single camera architecture approach to capture frontal faces from video feeds. It uses the Haar Cascade algorithm applied to video to capture individual faces of students and obtain distinct features of the face by using line

features and edge features. The Haar cascade algorithm basically works by giving the parts of the face that are most needed for detection, i.e., the Region of Interest (ROI).

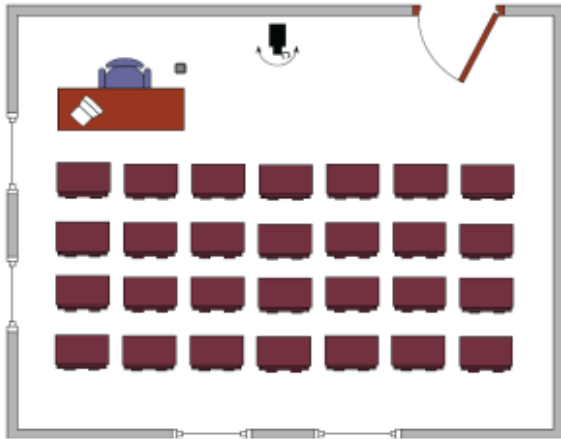


Fig. 1: Class room setup (Fuzail *et al.*, 2020)

It also processes and crops out other regions of the face that do not play a role in the image processing. Faces are extracted and saved once they are detected. Based on the Local Binary Pattern Histogram (LBPH) algorithm, each image stored in the database has histogram value calculated and is crosschecked against the calculated Histogram value of the images extracted from the captured video feed.

For working with frontal and full-view faces, Harguess *et al.* (2009) suggested a technique that can address the aforementioned restrictions. Half-face or side-view faces were discovered and identified by the suggested multi-camera model. The Cylinder Head Model (CHM), which tracks faces across several cameras, is used to create the system. Figure 2 illustrates how the system identified a common face by merging the faces' half-parts.

This system requires enough disk space to store the amount of trained data (full and half view images). The system can also be improved by clearly modelling the camera positioning architecture.

According to Wang *et al.* (2014), multi-camera technology is essential for reducing detection and recognition challenges. In object tracking

and multi-view scenes, multi-camera architecture provides more comprehensive data than a single camera. It can help with occlusion, chaotic scenes, and dealing with drastic changes in light.

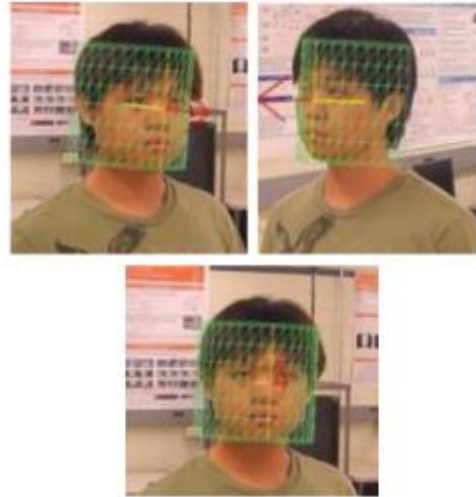


Fig. 2: Side-face detection fusion using CHM (Harguess *et al.*, 2009)

A multi-camera system was proposed by An *et al.* (2013). According to the scientists, the intrinsic limitations of camera technology and the picture collecting method make face recognition in surveillance recordings challenging. Non-cooperative subjects are captured with noise and blur in arbitrary positions and resolutions under different lighting conditions. Additionally, because the items are moving and multiple cameras are often dispersed throughout a camera network, various cameras frequently catch the same object from various angles. This study suggests a probabilistic face recognition method suitable for a network of video security cameras with multiple cameras. Data from various cameras and temporal cues from successive frames are combined using a Dynamic Bayesian Network (DBN).

The idea of constant k-coverage of an area is put up by Chaoyang *et al.* (2016) (k is an integer greater than 1). Identifying faces from various angles was the main concept. No matter where it was or which way it turned, the face was always in the field of vision. According to

the author, camera sensors meet the criteria for vector sensors since they can capture the same target from several angles. The angle between the target's looking direction and the camera's viewing direction also influences the camera

sensors' ability to detect objects. Full-view coverage is a crucial idea since it ensures that a target will be both detected and recognized. Figure 3 displays the estimated camera angle view.

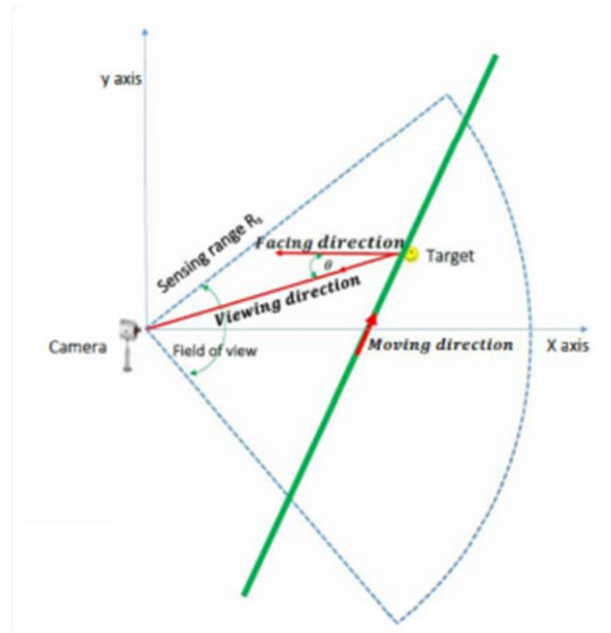


Fig. 3: Computation of the target-point angle (Chaoyang *et al.*, 2016)

Face recognition is recognized as an excellent technique for person identification at a distance because of its covert and remote sensing capabilities, according to Park *et al.* (2013). However, when faces are photographed from a distance (more than 5 meters), the resolution is low, which reduces the accuracy of face matching. They suggested an imaging system with a static and Pan, Tilt, and Zoom (PTZ) camera to capture high-definition facial images up to a distance of 12 meters to solve this issue. The distance invariance property is achieved in the study by proposing a novel coaxial concentric camera configuration between static and PTZ cameras. This enables it to use a straightforward calibration approach. A linear prediction model and camera motion control is also utilized by the system.

Lim *et al.* (2017) propose an IoT framework based on three analysis modules: face recognition, motion analysis, and behavior understanding, to effectively perform classroom

monitoring tasks such as taking attendance, identifying entering and leaving activities, and analyzing students' concentration levels. This system captures the audience for processing using a single camera architecture.

Methodology

This paper proposes a single camera architecture. The system is a microcontroller-based system that consists of a single camera placed on a rail to allow for capturing of images of different sections of a class. The microcontroller determines each stop position of the camera for the next face recognition process. The operation parameters like the rail length, camera's angular field of view (AFOV) are fetched by the from a MYSQL database. This system is a wireless-based system that requires all the components of the system to be connected to the same network for seamless operation. The prototyping model is used and a prototype of the camera rail system is created.

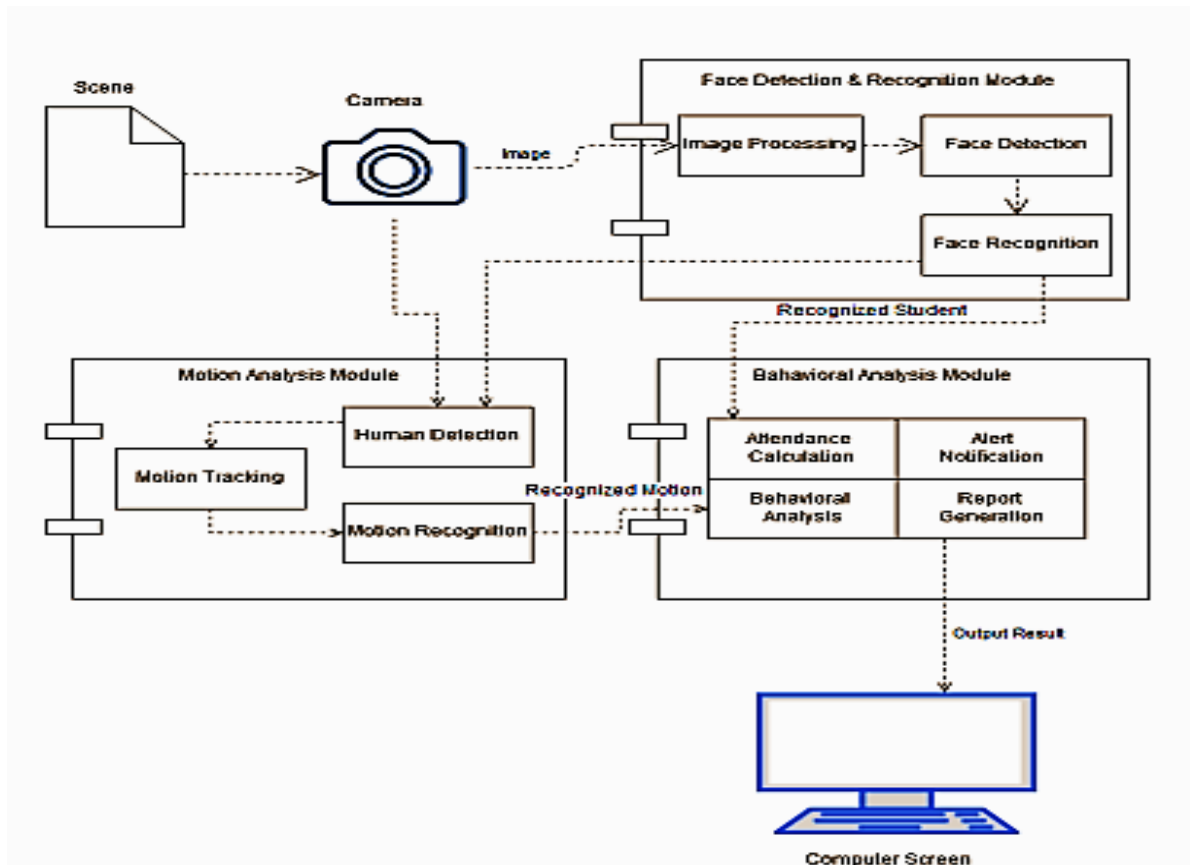


Fig. 4: Diagram of the IoT framework for automated classroom management system (Lim *et al.*, 2017)

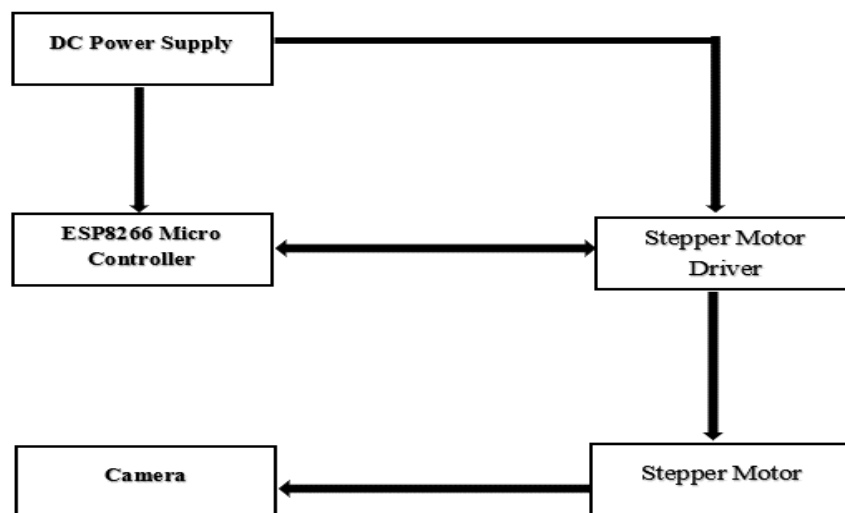


Fig. 5: The proposed camera system architecture

Operation of the Camera System

The camera system consists of rolling rails, a microcontroller to handle the automation of the movement of the camera on the rails. The microcontroller is connected to a database where it can fetch the system configuration. The following configuration parameters will be

fetches from the configuration table of the face recognition database:

- i. The Field of View of the camera
- ii. The Rail Length
- iii. The Maximum Object Distance

These parameters will be used to determine the stop positions of the camera. The movement of

the camera is facilitated by the stepper motor and is controlled by the ESP 8266 microcontroller. The ESP 8266 is a microcontroller with wireless capabilities that suits Internet of Things (IOT) applications. The microcontroller is connected to a network to facilitate the control of the system. The Network Service Set Identifier (SSID) is set within the microcontroller and this allows the microcontroller to automatically connect to the network when it is within range.

The camera position depends greatly on the Field of View (FOV) of the camera used. The FOV is the area of the world that can be seen via the camera at a specific angle and position in space. When a picture is taken, anything outside of the field of view is not captured. It is most frequently stated as an angle of view, or the

angular size of the view cone. For a normal lens, the diagonal (or horizontal or vertical) field of view can be calculated as:

$$\text{Angular field of view (AFOV) (radians)} = 2 * \arctan\left(\frac{\text{sensor size}}{2f}\right) \quad (1)$$

$$\text{AFOV (degree)} = 2 * \arctan\left(\frac{\text{sensor size}}{2f}\right) * \frac{180}{\pi} \quad (2)$$

where f is the focal length of the camera lens, the sensor size in the equation above is the size of the camera sensor used. Here the sensor size and focal length, f is in the same unit of length. Equation 1 is Angular field of view (AFOV) in radians while equation 2 is AFOV in degrees.

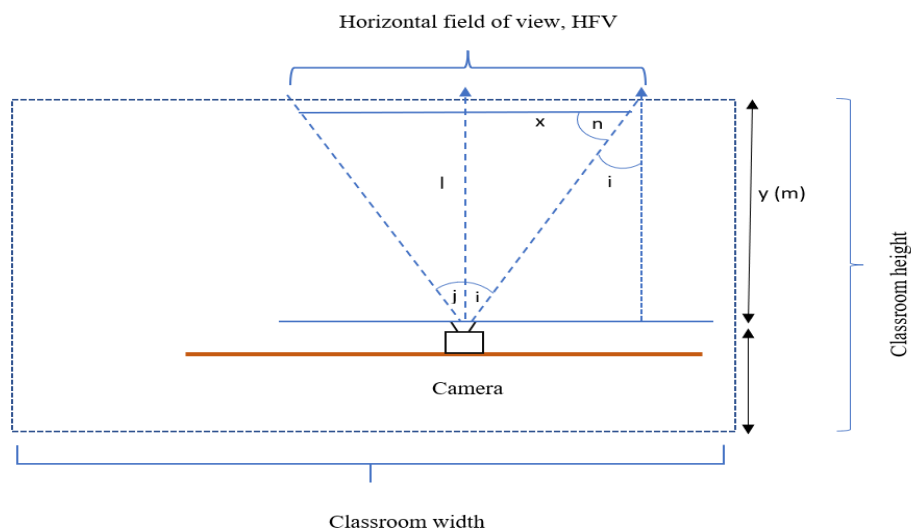


Fig. 6: Geometry of the camera position

The algorithm for the operation of the camera rail system is given below:

1. Start camera control system
2. Include necessary libraries
3. Connect the wireless network
4. Connect to MySQL database
5. Read the Stop positions as a list (data type) and the wireless camera IP.
6. Set Stepper speed
7. Connect to the wireless Camera
8. While the camera is connected and the frames are read:
 - For position in positions
 - Set position in stepper step method and move camera by the steps
 - If stepper is done stepping
 - Save the frame
 - Continue

9. Return camera to start position by adding a negative sign to the last position in the list.

Determination of the Camera Stop Positions

The stop positions of the camera are determined by the equations below:

x = is the field of view horizontal length

n = angle between the diagonal field of view and the horizontal plane

i = half of the field of view i.e. $\frac{AFOV}{2}$

l = maximum image distance (the distance between the camera position and the last object).

$t_{initial}$ = initial stop position of the camera

$$\text{Half of the field of view, } i = \frac{AFOV}{2} \quad (3)$$

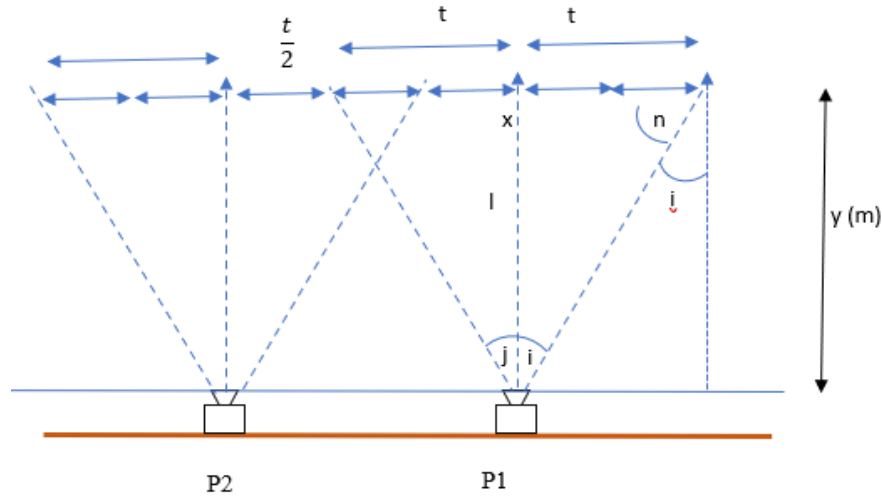


Fig. 7: Geometry for the determination of the next camera stop position

Consequently, for the initial movement position of the camera, $t_p = t_{initial}$

The next stop position, $NSP = t_p + 0.5 t_p$ (7)

where t_p = half of the horizontal view length x of the previous camera position.

Interaction of the Face Recognition System and the Camera System

Fig. 8 shows the interaction between the camera system and the face recognition system. The camera system transfers images captured from various sections of the class to the face

where $AFOV$ = Angular Field of view of the camera (in degrees).

By applying laws of trigonometry,

$$n^o = 90^o - i \text{ (law of alternate angles)} \quad (4)$$

$$t = \frac{l * (\sin i)}{\sin n} \quad (5)$$

$$x = 2 * t \quad (6)$$

To deduce the stop positions of the camera, the following consideration/assumptions are made:

- The horizontal-view-length of each stop position must overlap as shown in Fig. 7 using equation 6.
- The initial start position of the camera is not at point where $t = 0$ but at point $t_{initial}$ as determined by equation 5, hence, the start position of the camera is at point $t = t_{initial}$

recognition system for processing. The proposed system uses wireless camera on the rail and the Open CV library to capture the video feed from the camera through the camera's Internet Protocol (IP) address.

Here, P1- P4 corresponds to various camera positions, which are determined by the length of the rail with direct proportion to the size of the classroom. In Fig 8, Θ is the field of view angle of the camera being used. Frames from the camera at the various positions are saved in a directory within the face recognition application.

Python Open CV library can be used to capture the feed from the camera and to save the frames. These images are fed into the face detection phase of a face recognition system to extract the faces in each picture frame.

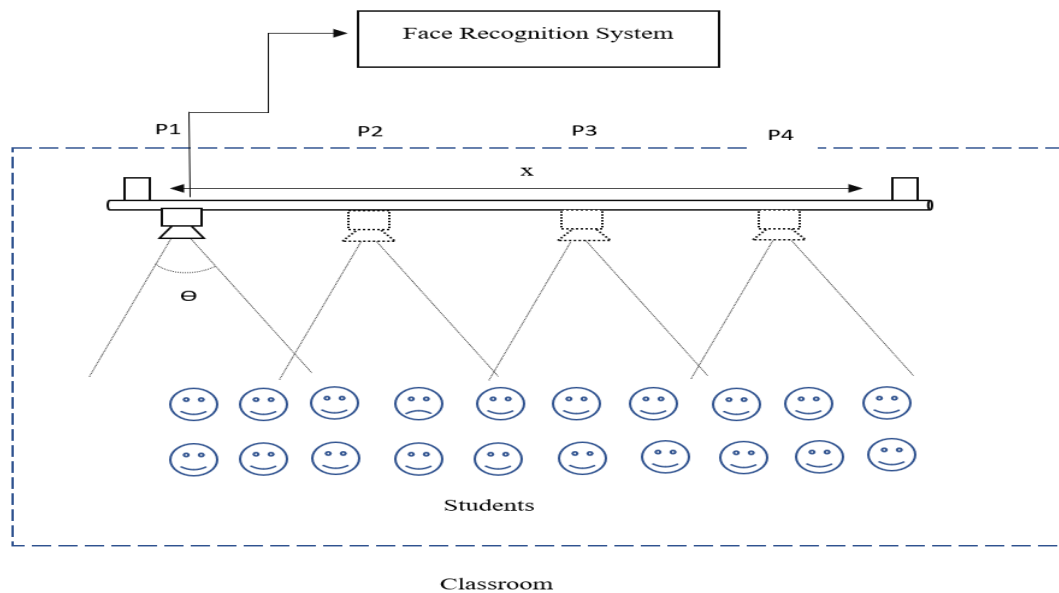


Fig. 8: The Interaction of the camera system with the face recognition system

System Implementation

The system is a microcontroller-based system with a stepper motor used to facilitate the movement of the wireless camera on the rail. The ESP 8266 microcontroller is used because of its wireless capability and it coordinates the

movements of the camera by reading the stop positions and passing these stop positions to the stepper motor as calculated steps. It also determines the storage locations of the captured image by the camera on the face recognition system to ensure correct images are processed for the corresponding class.

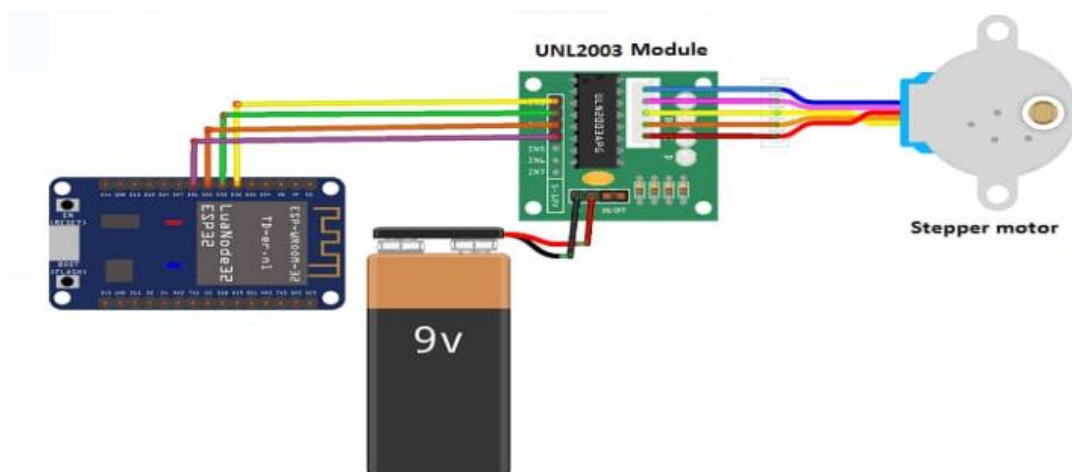


Fig. 9: Schematic diagram of the camera system

The components on the schematic diagram are described below:

- i. **ESP 8266WeMos module:** This is a microcontroller and a Wi-Fi module embedded in one chip.
- ii. **LM7805 voltage regulator:** Linear Monolithic, LM7805 is a positive

- voltage regulator that outputs +5 volts, which is used in powering up the circuit
- iii. **100 μ f Capacitor:** This is used in filtering the input and output current from the LM7805 IC
 - iv. **Stepper Motor:** This stepper motor 5V 4-Phase is an electromechanical device that converts electrical pulses into discrete mechanical movements. This is in charge of moving the camera to the various stop positions.

Other components used are:

- v. **Pulley:** This is attached at one end to facilitate the movement of the belt.
- vi. **A Belt:** This is attached to the camera base to help pull the camera base on the rail.

The camera is implemented as shown in Figure 10. It employs a belt and a pulley at one end to drag the camera on the rail. The stepper motor can move either in clockwise or anticlockwise direction to pull the camera to either end of the rail based on the number of steps programmed in the microcontroller.

The camera system circuit is implemented on a bread board and tested to ensure all the parts

worked well. Different current and voltage ratings of each component were taken into account to avoid damage to the components. The microcontroller was programmed using a Micro python. Micro Python is a complete Python compiler and runtime that operates on bare metal. It has an interactive prompt known as the Read Evaluate Print Loop (REPL).

The prototype consists of two steel rods 1.5 meters used to implement the rail for which the camera will slide on. The prototype is shown in Figure 11. Figure 12 shows a code snippet for the computation of the stop positions for the camera in python.

A user interface is created to set the configuration settings for the camera system. The interface is a web based interface and is implemented using the python flask framework for the backend and bootstrap for the frontend. Flask is a micro web framework written in python and does not require particular tools or libraries. Figure13 is an interface developed to interact with the wireless camera of the proposed system.

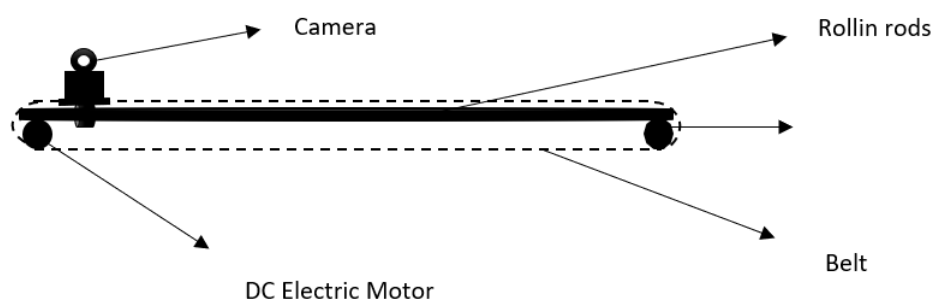


Fig. 10: Camera rail system



Fig. 11: A 1.5-meter camera rail system prototype

```

def compute_stop_positions(cam_fov,rail_lenght,max_obj_distance):
    import math
    from math import sin
    Stop_positions = []
    Nsp = 0
    Psp=0
    half_fov= int(cam_fov)/2
    n = 90 - int(half_fov)
    r = sin(math.radians(n))
    initial_pos= int(max_obj_distance) * sin(math.radians(int(half_fov))) / r
    r_initial_pos = round(initial_pos)
    if int(r_initial_pos) <int(rail_lenght):
        Psp = r_initial_pos
        Stop_positions.append(r_initial_pos)
        Nsp = Psp + 0.5*Psp
        IfNsp<= rail_lenght:
            Stop_positions.append(round(Nsp))
            whileNsp<= rail_lenght:
                Psp = Nspa
                Nsp = Psp + round(0.5*Psp)
                If Nsp<= rail_lenght:
                    Stop_positions.append(round(Nsp))
                Psp += Nsp
            else:
                #do nothing
                print("Warning: Next camera position is out of rail,current position is maintained!")
                returnStop_positions
        else:
            print("Warning: The Rail lenght is not sufficient for camera configuration, Minimum rail
            length required is " + str(r_initial_pos) + " meters.Median is used")
            Temp_pos = round(int(rail_lenght)/2)
            Stop_positions.append(round(Temp_pos))
            return Stop_positions
    return Stop_positions

```

Fig 12: Code snippet to compute the stop positions

The code in Figure 14 shows the implementation of the image capturing for the camera system. A folder is used to store the captured class image frames. A random string is generated and appended to the image frame for uniqueness.

Results and Discussion

Many of the researchers have implemented the face recognition system using either the single or multi camera architecture and the main aim of a face recognition system is to be able to take

the attendance of the attendees by processing images and video feeds from camera. These images must be captured in a way that all the faces of attendees are captured and the faces can be detected and processed by the face recognition system. The single architecture systems are implemented at stationary locations within the room and thus introducing blind spots which this design of a sliding camera on a rail eliminates.

Enter the Configuration Parameters

Field of view

Rail Length

Maximum Object Distance

Wireless Camera IP

Venue

submit

Fig 13: The configuration interface

```
def openvideo ():
    camip_addr = getcamip(str(lecture_venue))
    parent_dir = "C:\\Users\\UBONG\\face_recog_project\\class_assets\\"
    c_folder = getactive_class_folder(lecture_venue)
    cl_folder = os.path.join(parent_dir, c_folder)
    ran = ".join(random.choices(string.ascii_uppercase + string.digits, k = 4))
    msgv=""
    cap = cv2.VideoCapture('http://192.168.43.6:4747/video'
    camera, frame = cap.read()
    print(camera)
    if camera == False:
        print("Camera is not working")
    while(cap.isOpened()):
        status = cv2.imwrite(cl_folder + "/frame_" + str(ran)+'.jpg', frame)
        if(status == True):
            msgv = "<b style='color:green'>Image saved successfully!!</b>"
            continue
        else:
            msgv = "<b style='color:red'>Error Saving Image</b>"
            break
    cap.release()
    cv2.destroyAllWindows()
    return msgv
```

Fig. 14: Code snippet for image capturing

Results and Discussion

Many of the researchers have implemented the face recognition system using either the single or multi camera architecture and the main aim of a face recognition system is to be able to take the attendance of the attendees by processing images and video feeds from camera. These images must be captured in a way that all the faces of attendees are captured and the faces can be detected and processed by the face recognition system. The single architecture systems are implemented at stationary locations within the room and thus introducing blind spots which this design of a sliding camera on a rail eliminates.

With the design of this system, the cost of implementing face recognition-based attendance system will greatly reduce as multiple camera systems can be deployed at

different venues and they can all be connected wirelessly with the face recognition system that will process the images and record the attendance of the attendees.

For this research, a 1.5-meter rail was created as a prototype for this project and a 15 meters rail was used for testing. The results obtained shows that the system is able to calculate correctly the stop positions for the full coverage of the attendees and this is achieved irrespective of the camera's field of view.

The wireless camera attached was moved on the rails to the calculated positions and images captured by the wireless camera were successfully transferred to the face recognition system for further processing. Table 1 below shows the testing parameters used.

Table 1: Testing Parameters

Rail Length (meter)	Camera AFOV	Max object Distance(m)	Number of stops Calculated	Stop position in meter(s)
15	72	7	3	5.086, 7.629, 11.443
1.5	72	7	1	0.75



Fig. 15 (a): Class image at 5.1 meters position



Fig. 15(b): Class image captured at 7.6meters position



Fig. 15 (c): Class image captured at 11meters position

Conclusion

With the camera system developed, various sizes of the attendees can be comfortably captured. This system reduces the cost of procuring many cameras to capture different sections of the attendees as it requires just one camera for its operation. The limitation of camera blind spots and poor capture angle are eliminated, which improves the performance of the class attendance system.

References

- An, L., Kafai, M., & Bhanu, B. (2013). Dynamic Bayesian network for unconstrained face recognition in surveillance camera networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(2), 155-164. <https://doi.org/10.1109/jetcas.-2013.2256752>
- Baig, S., Geetadhari, K., Noor, M. A., & Sonkar, A. (2022). Face recognition-based attendance management system by using machine learning. *International Journal of Multi-disciplinary Research and Growth Evaluation*, 3(3), pp. 1-4. <https://doi.org/10.54660/anfo.2022.3.3.1>
- Chaoyang Li, Rosen, A., & Bourgeois, A. G. (2016). On K-full-view-coverage-algorithms in camera sensor net-works. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2218 - 2222.
- Credé, M., Roch, S. G., & Kieszczyńska, U. M. (2010). Class attendance in college: A meta-analytic review of the relationship of class attendance with grades and student characteristics. *Review of Educational Research*, 80 (2), 272-295.
- Fuzail, M., Nouman, H. M. F., Mushtaq, M. O., Raza, B., Tayyab, A., & Talib, M. W. (2014). Face detection system for attendance of class' students. *International journal of multidisciplinary sciences and engineering*, 5(4).
- Gomes, C., Chanchal, S., Desai, T. & Jadhav, D. (2020). Class Attendance Management System using Facial Recognition. In *ITM Web of Conferences* (Vol. 32, p. 02001). EDP Sciences.
- Harguess, J., Hu, C., & Aggarwal, J. K. (2009). *Fusing face recognition from multiple cameras*. *2009 Workshop on Applications of Computer Vision (WACV)*. <https://doi.org/10.1109/wacv.2009.5403055>
- Kalel, K., Shinde, A., & Shinde, A. S. (2022). AI Based Attendance System Using Faces Recognition. *Journal homepage: www.ijrpr.com ISSN, 2582, 7421*.
- Lim, J. H., Teh, E. Y., Geh, M. H., & Lim, C. H. (2017). Automated classroom monitoring with connected visioning system. *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 386-393. <https://doi.org/10.1109/apsipa.2017.8282063>
- Mothwa, L., Tapamo, J., & Mapati, T. (2018). Conceptual model of the smart attendance monitoring system using computer vision. *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 229-234. <https://doi.org/10.1109/sitis.2018.00042>
- Park, U., Choi, H. C., Jain, A. K., & Lee, S. W. (2013). Face tracking and recognition at a distance: A coaxial and

- concentric PTZ camera system. *IEEE transactions on information forensics and security*, 8(10), 1665-1677.
- Satpute, R., Sontakke, S., Gondaliya, B., Sonawane, T. & Suryawanshi, K. (2020). Attendance Management System Using Face Recognition. *biometrics*, 7(05).
- Trivedi, A., Tripathi, C. M., Perwej, Y., Srivastava, A. K., & Kulshrestha, N. (2022). Face Recognition Based Automated Attendance Management System. *Int. J. Sci. Res. Sci. Technol*, 9, 261-268.
- Wang, Y., Lu, K., & Zhai, R. (2014). Challenge of multi-camera tracking. In *2014 7th International Congress on Image and Signal Processing* (pp. 32-37).