Automatic Office Environment System for Employees Using IoT and Computer Vision

Devang Sharma
Department of Electronics and Communication Engineering, Institute of Technology, Nirma University
Ahmedabad, India
17bec104@nirmauni.ac.in

Himanshu Sharma
Department of Electronics and Communication Engineering, Institute of Technology, Nirma University
Ahmedabad, India
17bec105@nirmauni.ac.in

Dipesh Panchal
Department of Electronics and Communication Engineering, Institute of Technology, Nirma University
Ahmedabad, India
dipesh_panchal@nirmauni.ac.in

2020 IEEE 17th India Council International Conference (INDICON) | 978-1-7281-6916-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/INDICON49873.2020.9342455

***Abstract*— The present need of the world demands automation and the ease of work. Getting inspired from various home automation projects, this project article deals at inculcating the automation in the offices. The article aims at accomplishing the task of an automatic attendance system for the employees using face recognition and Amazon Web Services like S3 and QuickSight, and also adjusting the opening-closing process of the doors using the face recognition results. It also aims at triggering the electrical appliances like lights and fans of the office by detecting the movement of the people in the office. The article also describes the webserver wherein all the appliances of the office as well as the AWS and login based attendance of the employees can be accessed. All the sensors, cloud services and the face recognition algorithm are interfaced with the Raspberry Pi, that is used as a central unit in this project. This system, proposed to make an office smart, can be a good model for implementation at various government, and private offices and can be implemented at schools, colleges, and similar organisations as well.**

Keywords*— *Raspberry Pi, Face Recognition, OpenCV, AWS Cloud, Flask, PIR, Relay

I. Introduction

The present world scenario as seen in [1] is based more on the automation of all the routine tasks surrounding the people like speech based home automation, industrial automation in processing the machinery tasks, and the list continues. Getting inspired by [2], the present project is aimed to automate the various tasks in an office comprising many employees. This automatically processed office, also termed as SMART OFFICE would automate the lights, doors, and other electrical appliances of the complete office as well as the appliances of the cabin of a particular employee. Also, as per the reports, the COVID-19 pandemic situation has shifted people to work with their own set of instruments rather than using them on a sharing basis. So considering the same, the proposed project focuses on triggering the automation in the office appliances by detecting the motion throughout the office and the cabins using the PIR motion sensor. All the switching applications can be accessed from an exclusive webserver that is made using Flask. This webpage can be accessed by all the employees from their mobile phones using a common network inside the office. So rather than using a common switch to be used by all, this project aims at providing individual switches (on the webserver) for each employee and thus reduces the indirect contact between all. The office also facilitates an employee to get his attendance recorded without any physical contact with any device. The attendance is recorded using the Face Recognition algorithm that is developed using OpenCV, Haar Cascade classifiers, and LBPH algorithm. The personal attendance record is updated frequently on the AWS cloud storage services and this can be seen using the personal login on the AWS cloud. The link to the login based attendance check is also embedded on the web server of the office. Raspberry Pi has been used as the core unit for triggering all the processes in the office. All other devices like camera module, PIR sensor, servo motor, and the cloud services have been interfaced and made compatible to work with the Raspberry Pi.

II. Hardware Automation

A. Raspberry Pi

Raspberry Pi 4B model is used in this project as the central unit for the commencement and interconnection of the various processes occurring simultaneously. GPIO (General Purpose Input Output) ports on the Raspberry Pi are used to interface the various modules like PIR motion sensor, servo motor, relay with the Raspberry Pi. These 40 GPIO pins works on a 3.3V and gives up to 16mA of current, almost enough to drive the modules interfaced [3]. The main objective of using this model is the 64-bit ARM v8 Quad-core Cortex-A72 1.5 GHz processor used for the smooth governing of various sensors, face recognition algorithm, and cloud servicessimultaneously, and the 4 GB memory on-board memory used to store larger data for the attendance record of each employee. The pink box mark in Fig. 1(a) and Fig. 2(a) shows the power supply to the Raspberry Pi and the connected Raspberry Pi respectively.

B. Raspberry Pi

The camera module used to capture the images for the person on the door to start the face recognition process is the OV5647 5MP 1080P IR-Cut Camera for Raspberry Pi 4. This camera supports night vision is interfaced with the Raspberry Pi using the on-board CSI port. The camera works on the 5-megapixel OV5647 sensor [4] with 1080p resolution, and an embedded IR Cut filter that eliminates the colour distortion due to IR light in the daylight and thus helps capture the better quality picture at a resolution of up to 1080p during both day and night [5]. The dark blue box mark in Fig. 1(a) and Fig. 2(b) shows the camera attached to the door and the camera interfaced with the Raspberry Pi respectively.

C. Servo Motor

A servo motor is an electrical device capable of rotating an object at some specific angles or distance. It is used here for

978-1-7281-6916-3/20/$31.00 ©2020 IEEE

governing the opening-closing of the door [6]. On successful recognition of the face of an employee, the shaft undergoes a rotation of 45 degrees and opens the door. A delay of 5 seconds is given to generate a small pulse for which the shaft remains at the 45 degrees position and after 5 seconds the shaft comes to its original position, thus closing the door. This process repeats on every successful recognition of an employee. The green box mark in Fig. 1(a) and Fig. 2(b) shows the door and the servo motor attached to it respectively.

D. PIR Motion Sensor

The Passive Infra-Red (PIR) motion sensor detects infrared radiation, given off in the form of body heat, of a body moving across its view, and that can be used to detect the presence of a person or animal [6]. One PIR motion sensor is used at the entry side while the other is at the exit side. When someone enters the cabin and is detected by the sensor then the count is incremented by 1 in the program that is under execution on the Raspberry Pi. As soon as the count rises from 0 to 1 all the appliances of the cabin are turned ON as this leads to the detection of a person inside. The appliances will be ON until the cabin is empty. Whenever a person leaves the cabin, the exit side PIR motion sensor decreases the count by 1 and as soon as the count again reaches 0, the appliances will be turned OFF. The blue box marks in Fig. 1(b) show the PIR sensors connected to the relay module.

E. Relay

The output from the GPIO pins, i.e. 5V, is not sufficient to run the electrical appliances. These appliances run on 230V and this interfacing from 5V to 230V is accomplished using a Relay module. When it gets the HIGH signal, i.e 5V, the electrical contact is set to CLOSE, thus the appliance is turned ON and when it gets the LOW signal, i.e 0V, the electrical contact is set to OPEN, thus the appliance is turned OPEN. Whenever a motion on the entry side PIR sensor is detected, the count is incremented by one and the relay control pin is set to HIGH signal, i.e 5V, till the count is greater than one. So the relay switch is set to CLOSE that switches the appliances ON. The count is decremented by one on motion detection by the exit side PIR sensor and when the count is 0, i.e. nobody is present in the office, the relay control pin is set to LOW signal. So now the relay switch is set to OPEN that switches the appliances OFF. The yellow box mark in Fig. 1(b) shows the relay module interfaces with the Raspberry Pi and connected to the bulbs, represented as appliances, shown by the orange box mark in Fig. 1(a).

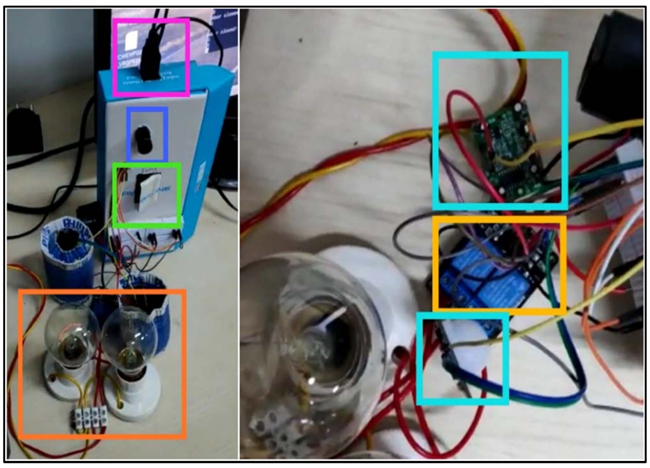

Fig. 1. (a) Hardware Setup (b) PIR and Relay interfacing

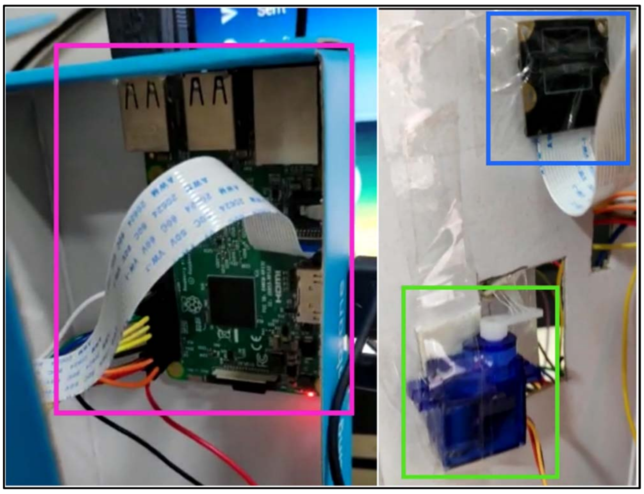

Fig. 2. (a) Raspberry Pi Setup (b) Camera and Servo Motor interfacing

III. Software Architecture

A. Facial Recognition

The base of the automatic attendance record and the automation in operating the door is formed by the use of the face recognition algorithm. This project includes the use of OpenCV, an open source computer vision library [7], and different algorithms like Haar Cascade Classifier and Local Binary Pattern Histogram (LBPH) algorithm to implement the face recognition algorithm successfully. The algorithm used comprises of four stages.

1) Dataset Creation
2) Dataset Training
3) Face Detection
4) Face Identification

The first stage is building a dataset of 100 front face images for each employee for training the face recognition algorithm. The process starts with capturing an image. This image is then converted to a gray scale image and using the “haar cascade frontal face” XML file, the features for the face, if present, in the image are extracted. These features are line features, edge features, box features. The original image is cropped by creating a box around the boundary features of the face extracted [8]. So the training data is a set of 100 frontal face gray scale images of each employee as shown in Fig. 3. The second stage is the training of the dataset created and assigning a label to each trained dataset. For training, Local Binary Pattern Histogram (LBPH) algorithm is used [9]. In this algorithm, a Local Binary Pattern of each image in the dataset is created. The image in the dataset is first transformed to a MxM matrix and each pixel is assigned a value based on its and its neighbouring pixels’ intensities using the local binary operator. Eq. (1) is defined as the local binary operator

$$LBP_{pixel} = \sum_{p=0}^{p-1} 2^{p}(i_{p} - i_{c}) \qquad (1)$$

In (1) ’i_c’ is the central pixel’s intensity, ’i_p’ is the neighbouring pixel’s intensity, and ’$s(i_p-i_c)$’ is a signum function that returns 0 if $i_p < i_c$ else returns 1. The median LBP_{pixel} value of the matrix is used as the threshold. If the LBP_{pixel} is greater than or equal to the median LBP_{pixel} value then the forms a binary pattern of 2^M bits and this value is converted to decimal for convenience. This value is known as the LBP value of the data under test or data under training. Rather than having a single LBP value for a data in the dataset, the LBP value is found for various circular regions with varying radius and varying neighbours. After getting all the LBP values, a histogram of the LBP values of the data is created and this histogram is called Local Binary Pattern Histogram, abbreviated as LBPH [10]. So for each image in the dataset, a LBPH is created and this LBPH is given the name of the employee as the label

The third stage is detecting the face from the test image that will be captured by the Pi Camera when a person stands in front of it at the door. It is the same as the detection of facial features, cropping the image, and converting it to a gray scale image that was done while creating the data set using the Haar Cascade Classifiers. Here the images are transformed to gray scale images because the LBP operator is robust against gray scale transformations. Now this detected face image, also called the test data in this case, is operated by a LBP operator and its LBPH is created. This LBPH is compared to the LBPHs in the trained dataset and the image is returned with a box around the face and the label of the closest histogram match found from the dataset LBPHs. The match probability is also shown with the label and this probability is the LBPH match percentage that can be calculated using methods like Euclidean distance, chi-square, absolute value, etc.

The major objective of using Haar Cascade Classifier for face feature extraction is its simplicity and ease of work. It compares the features with the pre-defined feature file for the frontal face and extracts the feature and is also completely compatible with OpenCV. The use of the LBPH algorithm for face identification can be justified by the following advantages of the algorithm [10].

- LBP operator is robust against gray scale transforms
- Less storage and less processing time
- Each dataset image is analyzed independently
- Able to represent local features in the images
- Able to recognise both side and front faces
- Not affected by illumination variations

Fig. 3. Sample 3x3 image matrix of the dataset

B. AWS Cloud Services – S3, QuickSight

During the execution of the facial recognition algorithm, whenever a successful employee match is found, the door is opened and the record of the employee is stored in the". employee-specific CSV (comma separated value) file locally on the Raspberry Pi. This CSV file contains the data in a monthly manner, e.g. a particular CSV file of a particular employee will have the attendance data of all the days in that particular month for that employee. The file has a column that is to be marked as PRESENT or ABSENT for any particular date. There is one more column that keeps the track of the total entries made by the employee in any particular day, it keeps on increasing that count everytime the employee's face is recognised at the door. These CSV files are updated on AWS S3 everytime any update is made in them locally. S3 is an abbreviation for Simple Storage Service and it is a data storage cloud service provided by the Amazon Web Services [11]. For a particular office, there is a single bucket, the terminology used in AWS as the root location of the data stored, and each bucket has subfolders for each employee.The local CSV file for each employee is updated on S3 under the subfolder of the particular employee.

The S3 bucket can be accessed publicly using the access keys provided by AWS. For generating the access credentials, AWS IAM, i.e. Identity and Access Management is used. Using IAM, one can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. For a specified resource, and user or group, the following credentials like Access Key and Secret Access Key are generated using AWS STS, i.e Security Token Service. AWS STS enables users, especially IAM users to have limited privileges for the resources allocated [12]. Here we have used the 'S3 FULL ACCESS' resource to use S3 buckets locally on the Raspberry Pi system. For interfacing the Raspberry Pi with AWS cloud services,'boto3' and the above mentioned credentials are necessary. Boto3 is the AWS SDK for python that enables the developers to configure and manage the AWS cloud services [13].

These stored files in S3 are analysed by AWS QuickSight and a per employee attendance report is generated. AWS QuickSight is a cloud service used for data visualization, provided by the Amazon Web Services [14]. In this project, it is used to create and update the employee attendance record from the CSV files stored in S3. It uses the URL of the manifest JSON file of each file stored in the S3 bucket for using them in creating the reports. A manifest JSON of a particular file in S3 consists of the dataset URL and its necessary information needed by AWS QuickSight to link with it. One Quicksight admin account can provide access to many users under it. So all the attendance reports are generated in a single account and can be analysed by the admin user while the admin has the right to set the permissions in a way that a particular employee can look at his/her report only. In this way, QuickSight provides a login based personalized report generation platform. Fig. 4 represents the complete cloud architecture for AWS that includes the flow of the services being used.

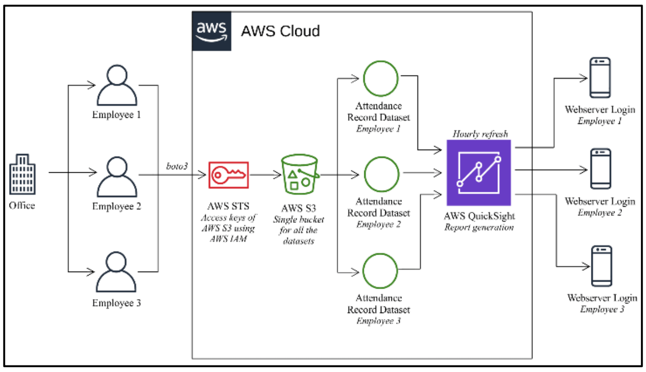

Fig. 4. AWS Cloud Architecture

C. Flask Webserver

Within the office, the appliances can also be triggered using the webserver that is developed using the Flask framework and can be accessed from any mobile or computer device connected to the same network as that of the Raspberry Pi. Flask is a micro web framework written in Python [15]. In this project, a flask python interfacing file is executed and the webpage would be active on the address 0.0.0.0:80. Here 80 is the port number that will be activated during the execution of the code. This webserver can be accessed through other web browsers by using the IP address of the used Raspberry Pi, connected to a common network [16]. The home webpage of the webserver consists of two elements.

The first one is the control of the appliances that is responsible for the simultaneous triggering of the electrical appliances. If the appliances are off then the button will show the TURN ON option and on clicking it the respective appliance will be switched on and now the button status will change to TURN OFF and on clicking it the respective appliance will be switched off. The second one is a link that redirects to the AWS QuickSight login page. On successful login, with proper employee-specific credentials, the respective employee becomes able to access his/her personal attendance report based on the data generated on the Raspberry Pi using the face recognition algorithm.

IV. Working Flow

Fig. 5 connects all individually discussed processes in a systematic flow for the successful implementation of the proposed approach. The requirement for the smooth governing of all the processes is a complete dataset of all the employees needed to train the face recognition algorithm. So the process starts by creating a dataset of frontal face gray scale images, 100 images, for each employee. This dataset is stored locally on the Raspberry Pi. Now looking at the front door of the office, a Pi camera is attached to it and interfaced with the Raspberry Pi. This camera captures the image of the person standing in front of it and with the use of the face recognition algorithm it identifies if the respective person is an employee of the office. If the person matches any of the employees in the dataset, the door gets opened up by giving a pulse signal to the servo motor for 5 seconds. After 5 seconds the door is closed and the face recognition process is reset. From this stage, the flow is divided into two parts, one of them is the automation in the office appliances using the components discussed in the earlier sections and the webserver, whereas the other one is the automatic attendance report generation using AWS cloud services.

Continuing with the automation part, after successful recognition of the employee at the door, immediately at the entry after the door, there is a PIR motion sensor that will keep a count of the number of people entering the office and will switch the appliances ON when there is even a single person inside the office. Now the appliances will be ON unless the cabin is empty and this will work with the help of PIR motion sensors. On the exit side also there is a PIR motion sensor that keeps a count of the people exiting the office and keeps on reducing the count generated on the entry side. If the count of the number of people in the office is 0 the appliances will be switched OFF. This process and these functionalities can also be provided to each cabin of the office. The appliances of the complete office can also be accessed using the webserver. The webserver has separate buttons for all the appliances and each such button displays TURN ON if the respective appliance is OFF or it shows TURN OFF if the appliance is ON.

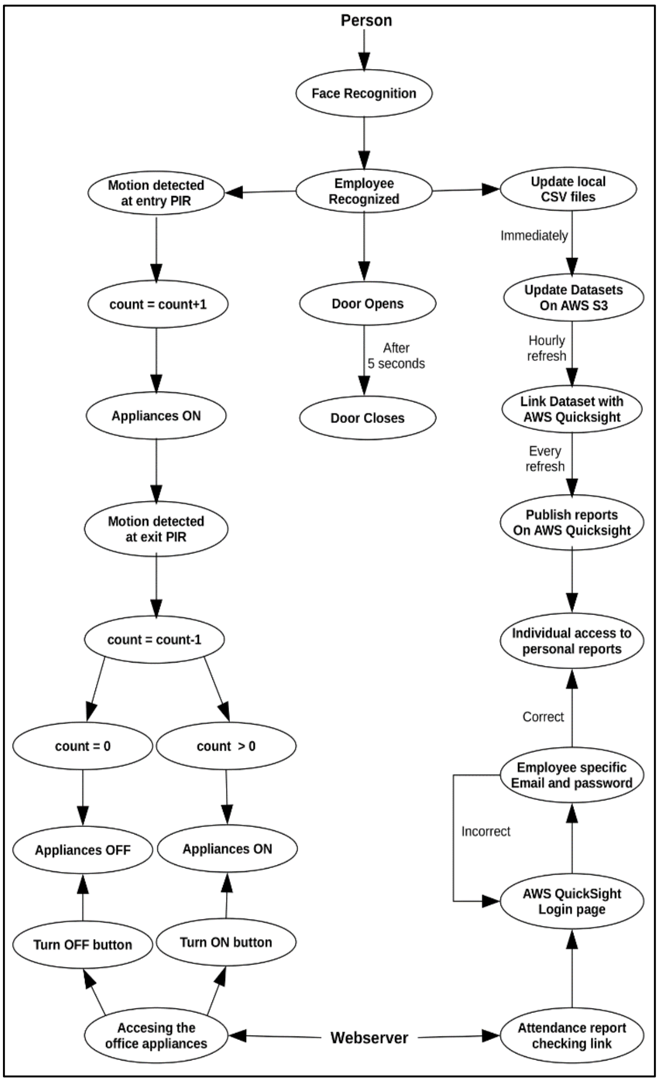

Fig. 5. Flowchart of the proposed approach

The attendance report generation process works parallel to this complete automation process. On the successful recognition of the employee, the CSV file of the respective employee is marked with P, i.e. PRESENT, or else at the end of the day it is marked as A, i.e. ABSENT. The next column, with respect to each row of each date of a particular month keeps a count of the number of successful recognition of the employee during that day. This CSV files on every update get updated on the AWS S3 bucket of the office. These CSV files are connected to AWS QuickSight that continuously updates the attendance report of the employees. All the reports are generated under a single office account but each employee has the right to access only his/her report. The link of this QuickSight login page is also provided on the webserver and on providing proper login credentials by the employee, exclusive for each, he/she can access the reports.

V. Experimental Results

Fig. 6 shows the developed webserver. Fig. 6(a) shows the webserver accessed on the computer's web browser where the orange box mark represents the local network address used to access the webserver, the green box marks are the buttons to control the office appliances, and the red box mark is the link that redirects to AWS QuickSight login page. Fig. 6(b) shows the webserver accessed using a mobile phone where the pink box mark is the IP address of the Raspberry Pi using which the webserver can be accessed remotely on the same network, and the red and green box marks are the same as that in Fig. 6(a).

The status of both the appliances as per the webserver shown in Fig. 6 is OFF and so the option to TURN ON is shown. In Fig. 7(b) the status of appliance-1, marked in the yellow box, is ON and so the option to TURN OFF is shown while the status of the other appliance is OFF. The hardware result for the same is also shown in Fig. 7(a). Similarly, when the status of both the appliances is ON, the option of TURN OFF will appear for both of them.

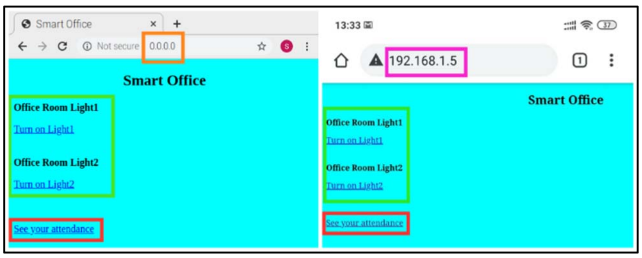

Fig. 6. (a) Webserver on computer (b) Webserver on mobile

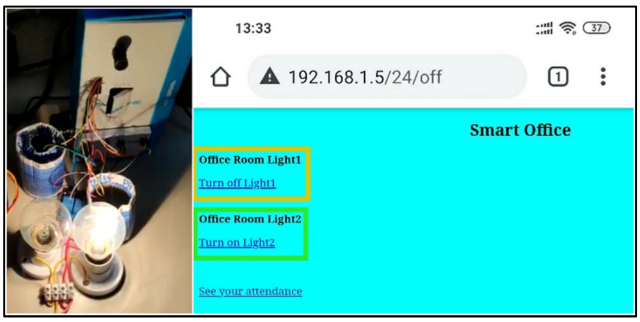

Fig. 7. One appliance ON (a) Hardware result (b) Webserver result

The results of the Face Recognition algorithm are shown in Fig. 8. Fig. 8(a) shows that the person detected is not a match with any of the employees in the dataset and so the image is labelled as UNKNOWN while Fig. 8(b) shows the successful recognition of the person as one of the employees. The image is labelled with the name of the employee and the probability of the recognition concerning the specific employee's dataset. The probability shown in Fig. 8(b) is 67.09%.

The idea of using only 100 images per employee in the dataset is due to the presence of a large number of employees. For the dataset of the complete organization, the total images would be 100 times the total number of employees. For a large number of employees, the dataset images will increase and that would cost in the memory storage of Raspberry Pi. Here, with the use of 100 images, the average probability of recognition was around 67%. According to the test observations, it can be concluded that the probability was raised by just 3%

VI. Conclusion

Apart from surveillance, the proposed system uses the camera module for face recognition that works as the main triggering part for the automatic attendance report generation. The attendance is thus marked without involving any physical contact and its association with the AWS brings security to the data and can also retrieve the data for a long time. The face recognition algorithm used, with the use of 100 dataset images per employee, is optimum as it saves cost, memory and also processing time as more images in the dataset would lead the test image to look for more trained images, or more lead the test image to look for more trained images, or more LBPHs, to find an appropriate match. More dataset images would improve the recognition results but that would cost a trade-off between cost, memory, and processing time. The automation in the office appliances

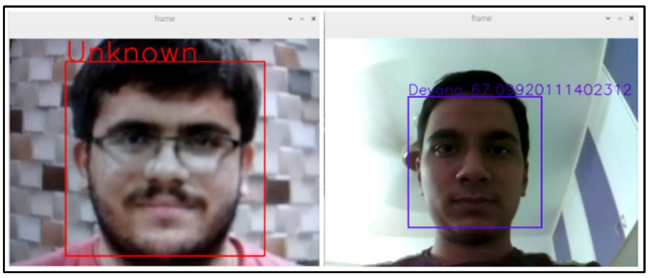

Fig. 8. Face Recognition (a) No match (b) Employee matched

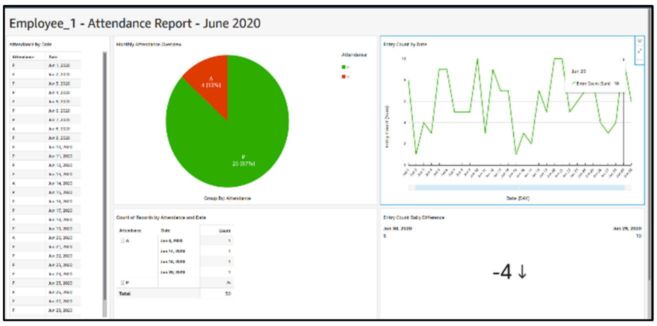

Fig. 9. Attendance report of an employee on AWS QuickSight

gives an employee the convenience of operating them automatically, by their motion in the office, as well as manually, using the webserver, without any common physical contact with the switches. So looking at the present world condition, this proposed approach of turning an office to a SMART OFFICE becomes ready to fit model for the government as well as the private offices that helps in maintaining comfort, regularity, discipline, and uniformity in the employees of the office with a reduction in the physical work.

REFERENCES

[1] Horch, Andrea,et al. "Why should only your home be smart?-a vision for the office of tomorrow." Smart Cloud (SmartCloud), 2017 IEEE International Conference on. IEEE, 2017.

[2] R. Sunchu, S. Palli, V. V. S. R. Datta and M. Shanmugasundaram, "Intelligent System for Office Environment Using Internet of Things," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 717-721.

[3] Richardson, Matt, and Shawn Wallace. Getting started with raspberry PI. " O'Reilly Media, Inc.", 2012.

[4] OV5647 Datasheet. 2020.

[5] Pagnutti, Mary A., Robert E. Ryan, Maxwell J. Gold, Ryan Harlan, Edward Leggett, and James F. Pagnutti. "Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes." Journal of Electronic Imaging 26, no. 1, 2017.

[6] Watkiss, Stewart. Learn Electronics with Raspberry Pi: Physical Computing with Circuits, Sensors, Outputs, and Projects. Apress, 2016.

[7] Howse, Joseph. OpenCV computer vision with python. Packt Publishing Ltd, 2013.

[8] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Vol. 1. IEEE, 2001.

[9] Wazwaz, Ayman A., et al. "Raspberry Pi and computers-based face detection and recognition system." 2018 4th International Conference on Computer and Technology Applications (ICCTA). IEEE, 2018.

[10] Deeba, Farah, et al. "LBPH-based enhanced real-time face recognition." Int J Adv Comput Sci Appl 10.5 (2019): 2019.

[11] Murty, James. Programming amazon web services: S3, EC2, SQS, FPS, and SimpleDB. " O'Reilly Media, Inc.", 2008.

[12] Anthony, Albert. Mastering AWS Security: Create and maintain a secure cloud ecosystem. Packt Publishing Ltd, 2017.

[13] Garnaat, Mitch. "boto Documentation." (2016).

[14] Huckman, Robert S., Gary P. Pisano, and Liz Kind. "Amazon web services." (2008).

[15] Oprea, Marin. "The Integration of IoT Projects in Undergraduate Education." Conference proceedings of» eLearning and Software for Education «(eLSE). Vol. 1. No. 15. " Carol I" National Defence University Publishing House, 2019.

[16] Ghosh, Atonu. "Intelligent appliances controller using Raspberry Pi." 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2016.