International Journal of Educational Research and Information Science
2018; 5(4): 31-37
http://www.openscienceonline.com/journal/eris

Open Science

Automated Student Attendance Management System Using Face Recognition

Ofualagba Godswill[1, *], Omijie Osas[1], Orobor Anderson[2], Ibhadode Oseikhuemen[1], Odiete Etse[1]

[1]Electrical and Electronics Engineering Department, Federal University of Petroleum Resources, Effurun, Nigeria

[2]Information and Communication Technology Department, Federal University of Petroleum Resources, Effurun, Nigeria

Email address

ofualagba.godswill@fupre.edu.ng (O. Godswill)

[*]Corresponding author

To cite this article

Ofualagba Godswill, Omijie Osas, Orobor Anderson, Ibhadode Oseikhuemen, Odiete Etse. Automated Student Attendance Management System Using Face Recognition. *International Journal of Educational Research and Information Science.* Vol. 5, No. 4, 2018, pp. 31-37.

Received: July 17, 2018; **Accepted**: August 13, 2018; **Published**: September 13, 2018

Abstract

Attendance management system is a necessary tool for taking attendance in any environment where attendance is critical. However, most of the existing approach are time consuming, intrusive and it require manual work from the users. This research is aimed at developing a less intrusive, cost effective and more efficient automated student attendance management system using face recognition that leverages on cloud computing (CC) infrastructure called FACECUBE. FACECUBE takes attendance by using IP camera mounted in front of a classroom, to acquire images of the entire class. It detect the faces in the image and compares it with the enrolled faces in the database. On identification of a registered face on the acquired image collections, the attendance register is marked as present otherwise absent. The system is developed on Open Source image processing library hence, it is not vendor hardware nor software dependent.

Keywords

Student Attendance, Face Recognition, Face Detection, FACECUBE, Cloud Computing

1. Introduction

Maintaining attendance is very important in all educational institutions. Every institution has its own method of taking student attendance. Some institutions uses paper based approach and others have adopted automated methods such as fingerprint biometric techniques. However, these methods subjects students to wait in a queue which consumes time and it is intrusive [1].

Humans often use faces to recognise individuals but advancement in computing capability over the past few decades now enable similar recognitions automatically [2].

Face recognition technology is one of the least intrusive and fastest growing biometric technology. It works by identification of humans using the most unique characteristics of their faces.

Face recognition has characteristics that other biometrics do not have. Facial images can be captured from a distance and any special action is not required for authentication. Due to such characteristics, the face recognition technique is applied widely, not only to security applications but also to image indexing, image retrievals and natural user interfaces [3].

Faces are highly challenging and dynamic objects that are employed as biometric evidence, in identity verification. Biometrics systems have proven to be an essential security tool, in which bulk matching of enrolled people and watch lists is performed every day [4].

The importance of developing a new solution to improve the performance of face identification methods has been highlightedand CC is one of the most promising technology paradigms that can be used to achieve it [5].

Face recognition technology has also been used for both verification and identification of students in a classroom. This research is aimed at developing a less intrusive, cost effective and more efficient automated student attendance management system using face recognition, leveraging CC infrastructure.

2. Related Works

Several work exist in automated attendance system [6-8]. However, very few have been implemented using a less intrusive approach such as face recognition. In this research, our focus is on face recognition and a cost effective architecture for its implementation such as CC.

Face recognition based attendance management system with raspberry pi 2 using Eigen faces algorithm has been proposed [9]. In the work, a camera is placed at the entrance of the class that interface with a raspberry pi 2 module for capturing students entering the class. The images are stored in the raspberry pi 2. The raspberry pi 2 module is used to achieve high speed of operation.

An attendance management system that uses CCTV camera fixed at the entry point of a classroom has also been proposed. The camera automatically captures the image of a person entry the classroom and compares the image with the face database using android enhanced smart phone. The Android mobile was used because of its rich face recognition capability [10].

Due to computationally intensive task involved in face detection and identification, a system that leverages cloud resources to improve facial identification performance in large video databases was proposed [5].

CC architecture which utilises PCA-characterised face instances has also been used to overcome the need to provide high computing resource for bulk image enrolment and matching [4].

3. Cloud Framework

Every day, sophisticated technologies emerge which becomes difficult for institutions to keep up with the trend in time of limited budget [11]. With the advancement of CC, many organisations are rapidly adopting IT enabled services that are hosted by cloud service providers [4].

Cloud is a large pool of easily usable and accessible virtualized resources (such as hardware, development tools and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale) and can be accessed by several users simultaneously (multi-tenant), thus allowing optimum resource utilization [12].

CC is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [13].

CC allow for scalable and flexible systems to be built with an easy on-demand access to virtual resources in a pay-as-you-go utility model [5].

In face detection and recognition, rapid image processing is essential in large-scale real time system. Computational cost of image processing is quite expensive. To minimise the burden of maintaining costly facilities for enrollment and recognition, it is suggested that organisations transfers this responsibility to third-party vendors who can maintain CC infrastructures for recognition [4]. CC presents an attractive option for delivering educational services in a more reliable and economic way without compromising quality [11].

CC offers a more convenient platform for image processing through large data management, distributed parallel processing, horizontal scaling etc, hence considering it ideal for FACECUBE implementation.

4. Proposed System

4.1. Research Method

To achieve the set objectives, FACECUBE components were developed as window service and web application. The system is designed to shift compute intensive task like face detection and recognition from client on-premise local server to cloud server, since image processing could be compute intensive especially when large data is involved. This enable client local server to be any commodity hardware when implementing FACECUBE.

FACECUBE window service component is a lightweight application that runs on the background of the on-premise local server. It is responsible for listening to attendance request, starting IP cameras irrespective of their location on the Campus Area Network (CAN) and continuous acquisition of images from the IP cameras mounted in front of the classrooms. The service sends all acquired images to the cloud web server for storage through web service mechanism where they are processed.

The web aspect of FACECUBE is developed to provide an intuitive and simple self-service interface for users to interact with the system. It performs all the task of image storage, image processing (face enrolment, face detection, face recognition) and attendance management. It also enables users perform all administrative task.

FACECUBE image processing engine is built on an Open Source image processing library. Hence, the system is not vendor hardware or software dependent.

4.2. System Description

FACECUBE is an automated student attendance management system that uses face recognition technology in its attendance process.

FACECUBE provides online access for 3 user roles: Student, Instructor, and Administrator role.

Student role: Student can create personalised account, enroll their faces on the system in 5 orientations using its self-service capability and could also register their courses as well as view lecture time-table. Every student that must participate in FACECUBE attendance process must be registered.

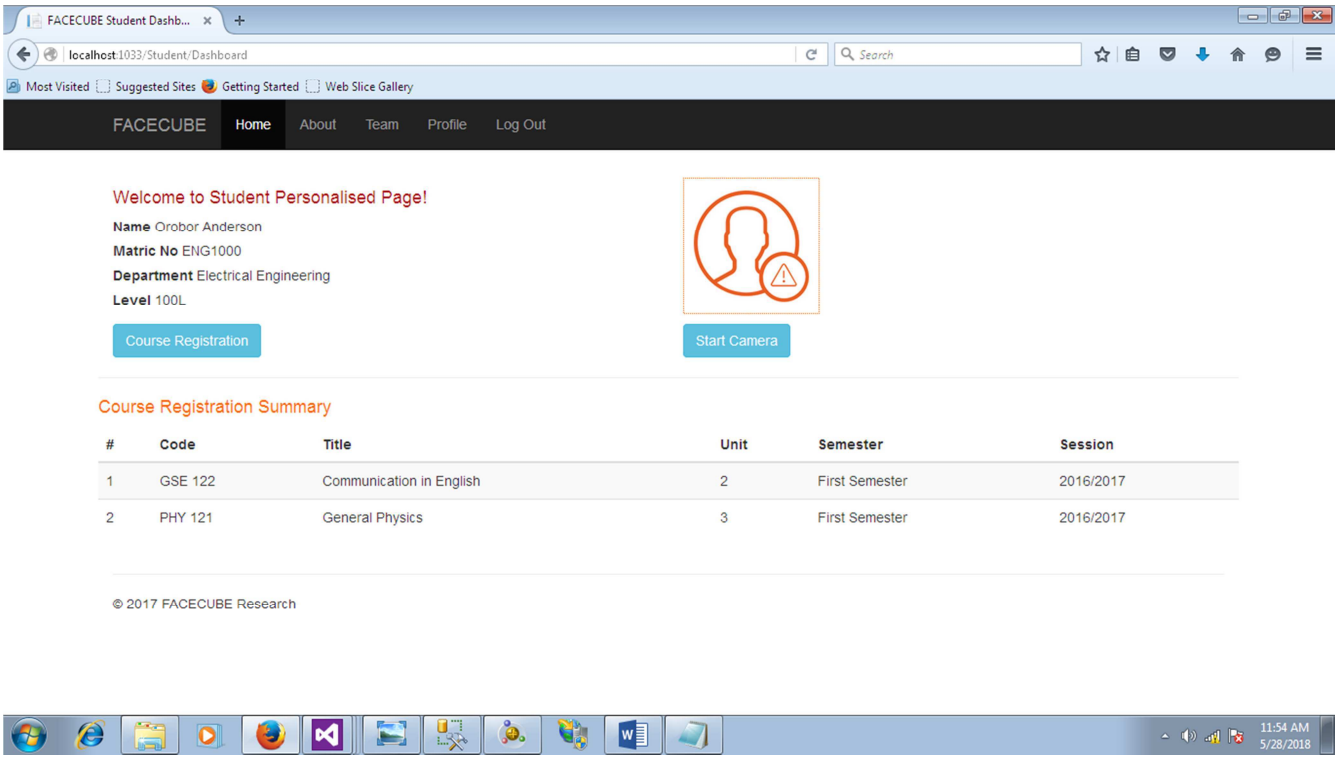


Figure 1. *Student profile.*

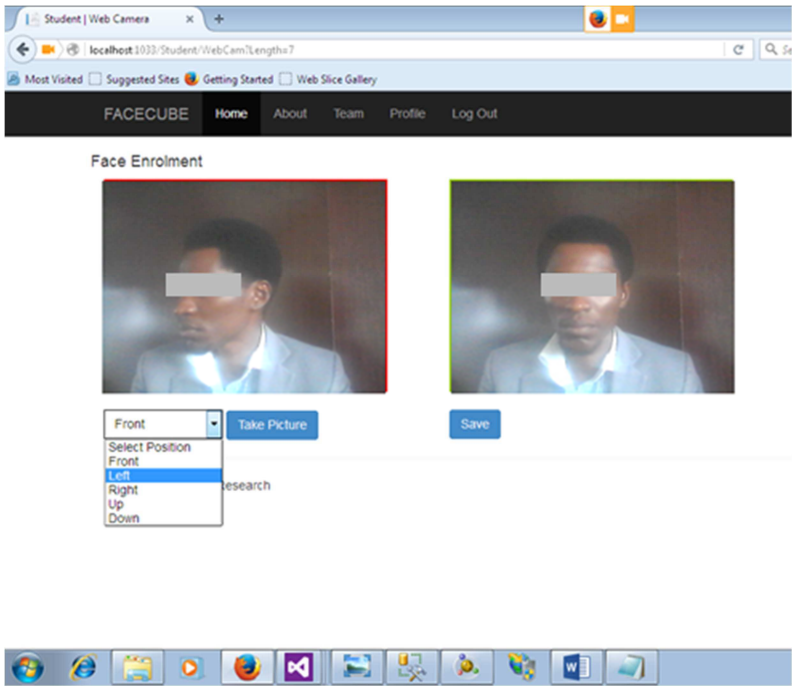


Figure 2. *Student face enrolment.*

Instructor role: The instructor can also create a personalised account, setup personal time-table, initiate request for attendance using his/her handheld device when in class and can generate attendance list after class anytime and anywhere.

Administrator role: The administrator is responsible for setting up IP cameras in different classrooms on the system as well as other administrative activities including user account management, course management etc.

FACECUBE sends notification to both student and instructor ahead of a class.

When a request for attendance is initiated by an instructor from the web platform, FACECUBE windows service takes the request details, get the classroom the request was made from, by using it IP address to locate and start the camera in that class to begin image acquisition.

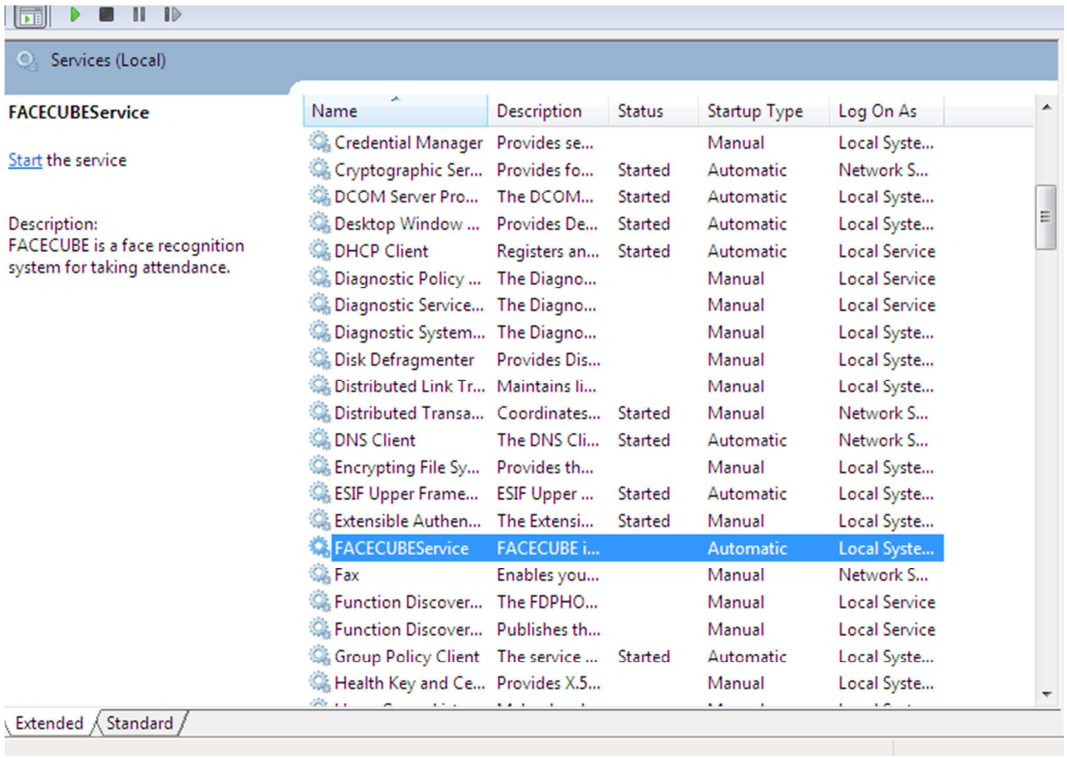


Figure 3. *FACECUBE windows service.*

The system uses IP Cameras mounted in front of a classroom irrespective of their location to continuously capture image of the entire class at set interval, throughout the period of a lecture and sends images over the internet to the cloud server for processing. The server processes the images by identifying/detecting the human faces contained, extract the faces and matches them with the enrolled faces of the students stored on the database.

During the process of face matching, only the students that registered for the course being taught when the images where captured is processed. On correct identification of a student face, the attendance register for the course is flagged as present for the student, otherwise absent.

4.3. System Design

FACECUBE is designed to run part of its components in on-premise local server as well as on a cloud server.

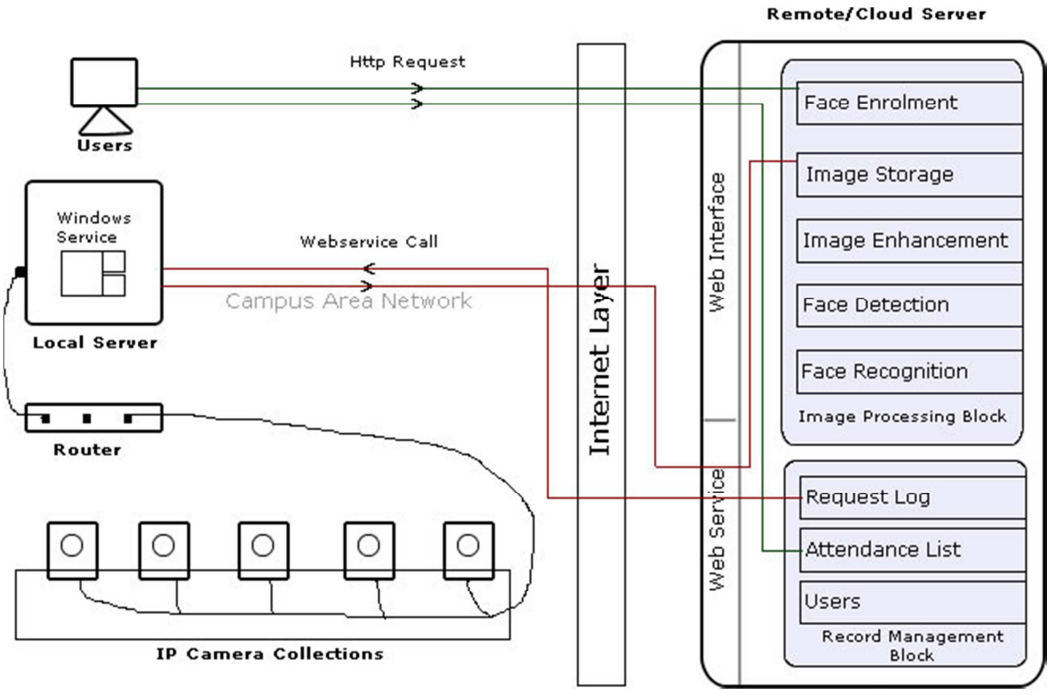


Figure 4. *FACECUBE architecture.*

The users interact with FACECUBE through its web application interface.

The local server connects all the IP cameras in different classroom locations using a router. It also host the windows service that runs on the background when the server is powered. The windows service gets or post request to the cloud server using a web service call.

The cloud server host the web application the windows service and users interact with. The web application components are:

Face Enrolment: The enrolment process involves creating face templates from set of images taken with a webcam. The face template are stored in the database for face recognition.

Image Storage: All images acquired from the IP camera video live feed as well as student enrolled face templates are converted to binary and stored in the database.

Image Enhancement: Most times, acquired images have darkness probably as a result of poor lightning in the environment. Before these images can be used for enrolment, detection or recognition purpose, it has to be normalized. Normalization enhances the accuracy of face detection and recognition. This process begins by converting the RGB image into a gray scale. Histogram normalization is then applied for contrast enhancement.

Face Detection: This is the ability to recognise human face in an image. FACECUBE uses Haar Classifier for its face detection. Haar classifier is a face recognition algorithm trained with thousands of human faces with different face positions and lighting condition. FACECUBE uses this algorithm to detect multiple faces in an image and draw a rectangle on each detected face. The face images are extracted and resized to 100x100.

Figure 5. *Detected faces.*

Face Recognition: FACECUBE uses Eigenface algorithm for face recognition. The algorithm is use to determine if enrolled face template stored in the database is found in the acquired classroom image.

4.4. Face Recognition and Detection

In face recognition, problem arise when performing recognition in a high-dimensional space. However, significant improvements can be achieved by first mapping the data into a lower dimensionality space [11].

Suppose G is an *N*2x1 vector, corresponding to an *N*x*N* face image *I*.

The idea is to represent G (F=G - mean face) into a low-dimensional space:

$$F\hat{} - mean = w1u1 + w2u2 + ...wKuK(K << N2)$$

Computation of the eigenfaces [14]

Step 1: obtain face images *I*1, *I*2,..., *IM* (training faces)

Step 2: represent every image *Ii* as a vector G*i*

Step 3: compute the average face vector Y:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M} \Gamma i$$

Step 4: subtract the mean face:

$$\phi i = \Gamma i - \Psi$$

Step 5: compute the covariance matrix *C*:

$$C = \frac{1}{M}\sum_{n=1}^{M} \phi_n \phi_n^T = \mathrm{AA}^{\mathrm{T}}$$

Where *A* = [F1 F2... F*M*]

Step 6: compute the eigenvectors *ui* of AA^T

Step 6.1: consider the matrix $A^T A$

Step 6.2: compute the eigenvectors *vi* of $A^T A$

$$\mathrm{A}^{\mathrm{T}}\mathrm{Avi} = \mu\mathrm{ivi}$$

Step 6.3: compute the *M* best eigenvectors of AA^T: *ui* = *Avi*

Step 7: keep only *K* eigenvectors (corresponding to the *K* largest eigenvalues)

Each face (minus the mean) F*i* in the training set can be represented as a linearcombination of the best *K* eigenvectors:

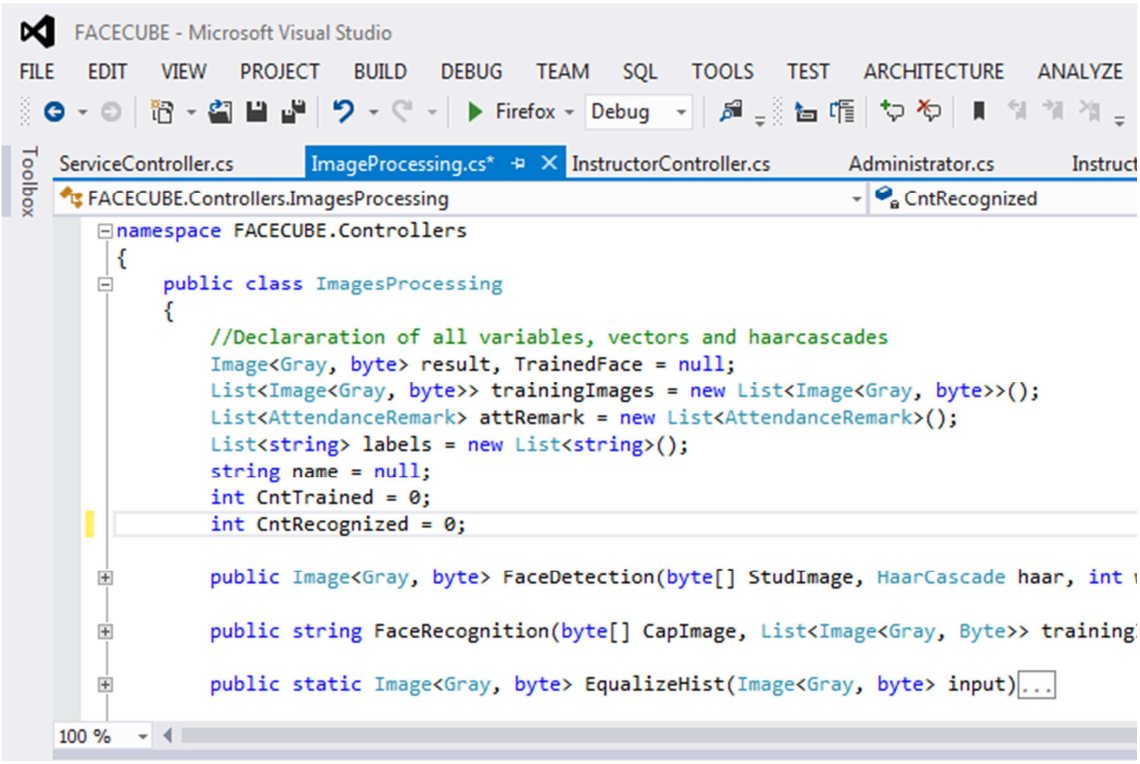
$\phi_i - mean = \sum_{j=1}^{K} \mathrm{Wjuj}$, ($wj=u^T_j Fi$)$u$ j 's is called *eigenfaces*)

Each normalized training face F*i* is represented in this basis by a vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \dots \\ w_k^i \end{bmatrix}, i = 1,2, \dots, M$$

4.5. System Implementation

FACECUBE was implemented using.NET technology on Visual Studio and Azure SDK. Visual Studio and the Azure SDK provide the capabilities of creating and deploying projects to Azure cloud infrastructure directly from the IDE. The source code was developed using C# ASP.NET and the face detection and recognition engine was built on EmguCV, a.NET wrapper to OpenCV image processing library. FACECUBE was tested in Intel(R) Celeron(R) CPU N3050 @1.60Hz computers with a 4GB RAM and windows 7 operating system.



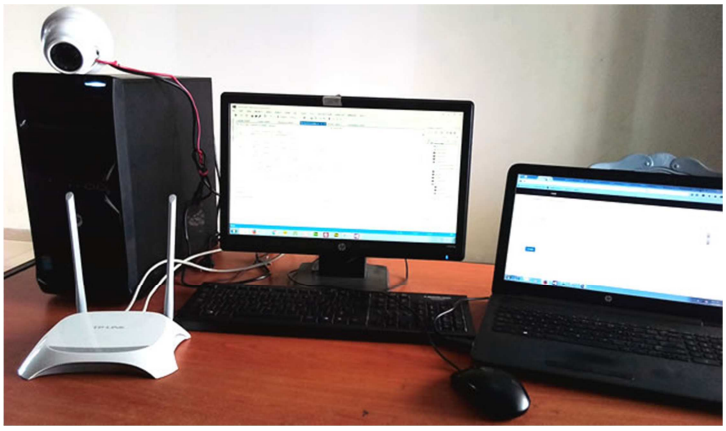


Figure 6. *FACECUBE experimental setup.*

Further test was carried out by deploying FACECUBE on Microsoft Azure cloud environment. Using cloud platform will make compute intensive task faster. This will also enable scalability from a single machine to clusters of servers on a public or private cloud infrastructure and improved performance. Cloud platform could also help in commercialising FACECUBE as a pay per use software service.

5. Conclusion

Attendance management system is a necessary tool for taking attendance in any environment where attendance is critical. Most of the existing approaches are time consuming, intrusive and required manual work from the users. In this research, we eliminated the above challenges by demonstrating the use of face recognition in student attendance management system called FACECUBE. FACECUBE takes attendance by using IP camera mounted in front of a classroom to acquire images of the entire class, detect the faces in the image and compare with the enrolled faces in the database. On identification of a registered student face on the acquired image collections, the attendance register is marked as present otherwise absent.

The system is designed to be cost effective with no specific vendor hardware and software required for deployment.

References

[1] Balcoh N. K Yousaf M. H, Ahmad W. and Baig M. I. (2012) Algorithm for Efficient Attendance Management: Face Recognition based approach. IJCSI International Journal of Computer Science Issues, 9 (4): 146-150.

[2] FBI (2017) Face Recognition. Available at: https://.fbi.gov%2Ffile-repository%2Fabout-us-cjis-fingerprints_biometrics-biometric-center-of-excellences-face-recognition.pdf. Accessed 14 January 2017.

[3] Sato A., Imaoka H., Suzuki T. and Hosoi T. (2005) Advances in Face Detection and Recognition Technologies. NEC Journal of Advance Technology, 2 (1): 28-34.

[4] Kisku R. and Rana S. (2016) Multithread Face Recognition in Cloud. Available at: https://www.hindawi.com/journals/js/2016/2575904/.Accessed 14 January 2017.

[5] Caldeira R. (2013) FaceID-Cloud: Face Recognition Leveraging Utility and Cloud Computing. Available at: https://fenix.tecnico.ulisboa.pt/downloadFile/395145529147/dissertacao.pdf. Accessed 28 May 2017.

[6] Dhanalakshmi N., Kumar S. and Sai Y. (2017) Aadhaar Based Biometric Attendance System Using Wireless Fingerprint Terminals. Available at: https://ieeexplore.ieee.org/iel7/7974917/7976734/07976871.pdf. Accessed 6 August 2018.

[7] Mohamed M. (2017) Class Attendance Management System Using NFC Mobile Devices. Journal Intelligent Automation & Soft Computing, 23 (2): 251-259.

[8] Rahim A., Ismail N., Razak F., Zulkifli Z., Jamian H., Razi F., and Mohammad N. (2017) Automated Attendance Management and Alert System. Journal of Fundamental and Applied Sciences, 9 (65): 59-80.

[9] Gaddam S. C. 251, Ramesh N. V. and Dhanekula H. (2016) Face Recognition Based Attendance Management System with Raspberry Pi 2 Using Eigen Faces Algorithm. ARPN Journal of Engineering and Applied Sciences, 11 (13): 8107-8112.

[10] Muthu K. and VeeraA. (2013) Smart Application for AMS Using Face Recognition. Computer Science & Engineering: An International Journal (CSEIJ), 3 (5): 13-20.

[11] Orobor A. and Orobor H. (2018) Cost-Saving and Internally Generated Revenue in Tertiary Educational Institutions: The Role of Cloud Computing. International Journal of Education and Information Technology, 3 (2): 7-16.

[12] Vaquero L., Rodero-Merino M., Caceres J. andLindner M. (2008) A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Computer Communication Review, 39 (1): 50-55.

[13] Mell P. and Grance T. (2009) The NIST Definition of Cloud Computing. Available at: http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf. Accessed: 24 February 2017.

[14] Turk M. and Pentland A. (1991) Eigenfaces for Face Detection/Recognition. Journal of Cognitive Neuroscience, 3 (1): 71-86.