

# Real-Time Facial Recognition Smart Door

Sumit Kumar<sup>1</sup>, Kartik Kumar<sup>2</sup>, Sarita Yadav<sup>3</sup>

<sup>1</sup>Student, Dept. of Information Technology, BVCOE, New Delhi, India

<sup>2</sup>Student, Dept. of Information Technology, BVCOE, New Delhi, India

<sup>3</sup>Professor, Dept. of Information Technology, BVCOE, New Delhi, India

\*\*\*

**Abstract** - In this paper we discuss the real time face recognition with the idea of secure locking automation utilizing iot for door unlocking system to provide necessary security to our homes, office, hospitals, banks etc. and related control operations. We know that in daily life security is major concern about every people so we work on this project to provide security by using smart door. Security caution through the azure database. It uses the image capturing technique in an embedded system based on raspberry pi and azure server system. R-pi (Raspberry pi) controls the video camera for catching it for turning on a relay for door unlocking. The system module contains a secured face recognition for automatic door unlocking. The camera catches the facial picture and compares with the given image stored in our local machine. If the image is found in the given database then the door unlock automatically otherwise it does not open the door and it will produce a message of an unknown person is trying to gain access and door remain lock.

**Key Words:** Facial Recognition Door, Home Security, Raspberry Pi, Iot.

## I. INTRODUCTION

Home security systems are trending in modern technology or we can say that it is the technology to provide security. A self-built system is not more expensive but difficult to installation and its efficient to use [1]. With the introduction of Microsoft's services, facial recognition applications are now more accessible to makers than ever before [1]. In this project it's needs a basic Webcam, Raspberry Pi and an internet connection to create a door that unlocks itself via facial recognition. If the visitor at the door is recognized, the door will unlock [2]. So, the project that was assigned to us was something regarding security and smart home automation or even public interaction. So, the final details of the project were decided and from there we came across the "Smart Door – Facial Recognition" [2].

In this project, we were told to use Microsoft Windows IOT core dashboard for the whole interaction with the Raspberry pi Board. Yes, it's true, we were using raspberry pi board for this whole project [2]. As Raspberry pi board is easily available and the processing was quite enough.

For the project like this, also to deploy the code and run it on the Raspberry pi board is convenient and was easily correctable, if any error persists. There we many a service we could have possibly used, but there were least (I would not say so), but yes, we faced lesser amount of challenges and efforts that the other services and platforms like Linux based processing on the Board [2]. Here we simply design a code or a program which detects human faces and then use that data to process in the raspberry pi board and then pass it to perform the desired output. This looks so simple but it is not actually, we have to create facial data processing platforms, where the real comparison takes place.

## 2. HOW DOES THIS WORKS

We were given the task of understanding cloud computing and the facial API'S calling, and the azure cognitive services in order to understand how things will be working in this project and what will be the flow of the whole data processing that will take place in the raspberry pi board [2]. Hereby , we were asked to create Azure account and buy a free pay-as-you-go services called , azure keys and API's , also we were told to first test all the raspberry pi GPIO(general purpose input output registers ) pins and also were told to blink an simple led from azure account and call it from the direct method[3]. At first it took a lot of time for me to understand what all was going in this project, then later I was able to understand the whole working of the project in parts. Our mentor, helped us lot in the whole process, at every stage he stood by us and made us understand the whole scenario of INTERNET OF THINGS and it partial working in this project. Azure services is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data Center [3]. This provides Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS) and supports many different types of programming languages, frameworks and tools, including both Microsoft specific and third-party software and application programmable interface (API) [3].



Fig -1: Adding user images into whitelist.



Fig -2: Capturing images into whitelist

## 2.1 SAAS – Software as a Service

• It is a software to delivery and licencing model in which software is licensed on a subscription basis and pro features facility. It is sometimes mentioned as "on-demand software", and was formerly mentioned as "software plus services" by Microsoft. SaaS is usually accessed by users employing a thin client, e.g. via a web browser [3]. SaaS has become a model for many business applications, including messaging software, office software, payroll processing software, DBMS software, management software, CAD software, development software, gamification, collaboration, accounting, visualization, enterprise resource planning (ERP), Management Information Systems (MIS), customer relationship management (CRM), invoicing, human resource management (HRM), talent acquisition, learning management systems, content management (CM), service desk management (SDM), Geographic Information Systems (GIS). SaaS has been incorporated into the many strategical of all leading enterprise software companies.

## 2.2 PAAS- Platform as a Service

• Platform as a service or platform-based service is a part of cloud computing services this platform allowing to customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically related to developing and launching an app [3]. PaaS provide a product delivered in three ways: As a public cloud service from a provider, where the consumer controls software deployment with minimal configuration options, and the provider provide

the servers, storage, network, operating system, middle-ware as Java Run-time, .NET , integration, etc.), database and other services to host the consumer's application.

• As a personal service (software or appliance) behind a firewall. As software deployed on a public infrastructure as a service [3] So these services were used to create an platform for the interaction between the facial API's and raspberry pi. No, we were about to start the project, so we first decided in which language will we be coding for this project, since everyone was okay with the C# language and also, this project works on IOT CORE DASHBOARD so we will have to choose between c++ and c# [3]. So, for us it was convenient to code as well as function the program in C# (also known as c sharp). So, we first made all the libraries and the necessary data to be coded in the language. This was provided to us by the Company and it was given in Raw form, which we then worked upon and made it working for our project. Later we installed the main dashboard on which we were about to start testing the processing and the whole integration of the project, that is Visual Studio 2015. It was the main compiler used for this project and we used it as interact or mediator later wards once we started deploying the code onto the raspberry, but that's long-long away. First, we made all the libraries, the strings. CS file, the each and every functioning page in GUI for the camera feeds and live camera feed in the gateway. It consists of all the GUI of the stages. It mainly had Welcome interface, add user interface, under which it had 3 more interfaces regarding the deletion and the doorbell ringing option [4]

## 3. RELATED LITERATURE AND COMPARISON WITH EXISTING MODEL

We first installed Windows IOT Core on the Raspberry Pi to run our UWP Application on it. We executed the project on a Local Machine using Visual Studio 2017 for initial testing. In the project, we have used the Face API of Microsoft Azure for face verification [4].

- If anyone wants to enter through the door, the App will capture his/her picture on pressing the doorbell and send it on the cloud to the Face API(to determines whether this is a whitelisted face or not).
- If the person is recognized then it sends the information to the Raspberry Pi board which is connected to the relay through GPIO pins to unlock the gate. If there is an unrecognised visitor, the door remains locked. Whenever the door is unlocked, the corresponding information about the visitor is sent to IOT hub [4].
- If there is a guest, we have introduced a feature to unlock the door using an android application which makes a direct method call to our device client to unlock

the door (using IOT Hub). Cognitive Services are APIs to help build intelligent application without having direct AI or data science skills.

- AZURE Cognitive Services are APIs (application programmable interface), SDKs (software development kit), and these services are available to help developers build intelligent applications without having direct AI or data science. It also enables the service for developers to easily add cognitive features into their applications. The goal of Azure Cognitive Services is help to developers create applications, hear, speak, understand, and see. Azure Cognitive Services can be categorized into five main parts - Vision, Speech, Language, Web Search, and Decision [5].

- H. Lwin.et al. [4] has proposed a door lock access system which is consist of 3-subsystems: to be specific automated door access control, face recognition and face detection. Face recognition is actualized by using the PCA (Principal Component Analysis). The door of office, room and other security access system itself open for the known person when command forward to microcontroller and caution will ring for the unknown person. The Demerit of this system is that it taken input images through a web camera continuously until the we can't push 'stop camera' button. So, we required somebody at that location to check unauthorized person's images and check status of the system and take further appropriate action. Personal computer system; PCs; is associated with the microcontroller if these occur some error or problem then malfunctioning entire system, not work properly and system will be crashed or Nonfunction.

- M. Chowdhury.et al. [5] had implemented security system where if any person came at the door it was notified to the home owner via e-mail or twitter then the user could see those persons which standing at the door and using camera to capture images of those people from remote location. The image of the person got captured and sent to twitter and e-mail. They stated that user couldn't control the door remotely. It is concluded that this system was useful for prevent unauthorized access user. The limitation of this system was that these generate alert message sent to the mail or twitter account but if the user didn't have internet connection on phone, they couldn't check the mail and couldn't recognize that any unauthorized person was trying to access the door without using internet connection.

- S. Jogdand.et.al [6] proposed a work on Implementation of automated accessing door system with the features of Face Design and Recognition in which they used two algorithms first is Viola Jones method for face detection and second is PCA (Principal Component Analysis) for the comparison of images. The

limitation of this work was that it is not robust and efficiency is less

- U. Sowmiya.et al [7]. Developed to connect any door with internet connectivity, In this system user also implemented PIR sensor and camera, PIR sensor used for detecting person capture image and camera also used for capturing the video of the person who comes at the door. The video was sent using 3g dongle (internet service) to authorized person. It also discussed some advantages of this system. It's concluded that use of this system in banks, hospitals and other security concern system etc. But these proposed models didn't provide the features of sending messages to the authorized people through twitter and mail.

- G. senthilkumar.et.al. [8] proposed a work on Embedded Image Capturing System Using Raspberry Pi. In this work, it captured the image of person and compared it with the given database if the image found in database then door will unlock otherwise door will remain lock. But limitation of this system is that it couldn't work properly in the ambient light condition that's why it's not working in dark condition.

- So, these references gave us idea about how to work in a flow with all the knowledge and after the whole lot of research work, we came across that the Image capturing system using raspberry pi will work for us. In this work, they captured the image and compared it with the database but the limitations were about to be discovered. We will later discuss the limitations and all the improvements in later part in this project report.

- Now that we have started deploying out code into Visual Studio 2015, we created a project which obviously was specific for the C# language and later started making the code and the logics work.

- Face recognition technology emulates the capabilities of human eyes to detect faces. This is done by smart computing that creates "face bunch" that consists of 70 nodal points. This Features are extract from the face and saved as templates. These templates are compared to the face detected. For this research, we interfaced an LCD, Camera and a Motor to the Raspberry Pi board [8].

#### 4. PROPOSED METHODOLOGY

##### Hardware

Microsoft HD Live Webcam

Raspberry Pi 3 Model B+

Display Monitor

Solenoid

Generic Mouse

Audio Speaker

Breadboard, Push Buttons and Connecting Wires

Generic Keyboard

Relay Module 1-way 5V

**Software**

Microsoft FACE API

Microsoft Azure subscription

Microsoft Visual Studio 2015

WINDOWS 10 IOT Dashboard Manager

UI deployed in XAML and backend in C#

WINDOWS 10 IOT

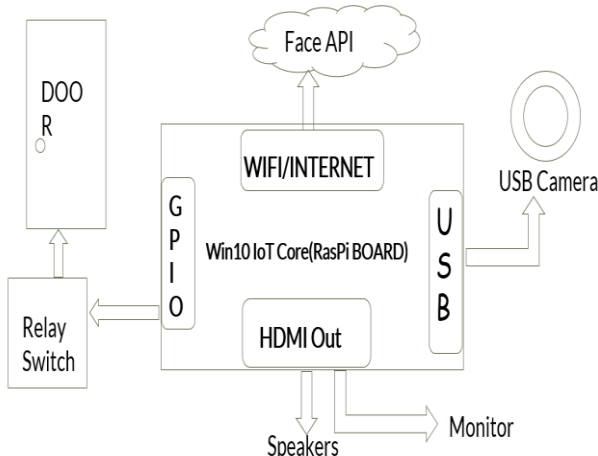
This system will work in two different categories. The first is for capturing and creating a database by storing the images. And the second one is to compare the images with the stored image in the database. For that feature extraction we will use Eigen faces methodology [5] and Euclidian distances will be used for recognition of that face. Camera module is work as module in which raspberry pi interfacing to the camera module. It is used to capture images and then send the capture images to the raspberry pi module and communicate to other devices. Camera contains LEDs flash light to handle that light condition by using infrared sensor is available inside the camera its automatically adjusted light according to its surrounding light, these light conditions are known as ambient light conditions. Raspberry pi module: raspberry pi 3 module is a very small computer board [7]. When an image is taken by raspberry pi it is compared with database images at the Azure face API. For the first time when we capture an image to Create a database raspberry pi module [7] captures many images to create a database in the system and this database is compared with the live captured images [8].

**5. CONCLUSION**

The AZURE based face recognition system for door lock control using RASPBERRY PI has been developed in this study. All related work which lead to the completion of the study, including the development of AZURE application programmable interface face recognition system and auto-switching magnetic lock mechanism with servomotor using relay. The performance and functionality of the designed system showed that the transmission of data from Azure to microcontroller is successfully done after that image is recognized. The system can be improved to become fully automatic face recognition system by adding another feature that automatically capture image with a sensor when there is the human present at the entrance area.

**REFERENCES**

- [1] Januzaj, Y., Luna, A., Ramaj, V. 2015 Real time access control based on Facial Recognition.
- [2] Lwin, H., Khaing, A., Tun, H. 2015. Automatic door access system using face recognition.
- [3] Chowdhury, M., Nooman, S. 2013. Access Control of Door and Home Security by Raspberry Pi through Internet.
- [4] Senthikumar, G., Gopalkrishnan, K., Sathish Kumar, V. 2014 Embedded Image Capturing System Using Raspberry Pi System.
- [5] Çarıkçı, M., Özen, F. 2012 A Face Recognition System Based on Eigen faces Method.
- [6] Jogdand, S., Karanjkar, M. 2015 Implementation of Automated Door Accessing System with Face Design and Recognition.
- [7] Sowmiya, U, Shafiq mansoor, J. 2015 Raspberry pi-based home door security.
- [8] Sahani, M., Nanda, C., Sahu, A., Pattnaik, B. 2015 Web Based Online Embedded Door Access Control and Home Security System Based on Face Recognition.
- [9] Gubbi, Jayavardhan. (2013) Internet of Things (IOT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29.7: 1645-1660.
- [10] Amazon 2018b. AWS IoT Core. URL: <https://aws.amazon.com/iot-core/pricing>
- [11] Microsoft Azure 2018. IoT Hub Pricing. URL: <https://azure.microsoft.com/enus/pricing/details/iot-hub/>. Accessed: 27 April 2018.
- [12] Satyanarayanan, M. OpenFace: URL: <https://www.cs.cmu.edu/~satya/docdir/CMU-CS-16-118.pdf>.
- [13] What is the Internet of Things? explains. URL: <http://www.wired.co.uk/article/internet-of-things-what-is-explained-iot>.
- [14] Condliffe, J. 2017: URL: <http://www.businessinsider.com/facial-recognition->



**Fig 3.** Architecture of facial recognition smart door.

controversyimprovement-2017-  
9?r=US&IR=T&IR=T.

[15] Delbiaggio, N. 2017. URL:  
<https://www.theseus.fi/handle/10024/132808>.

[16] Berdy, N. Betts, D. 2018. What is  
Azure.URL:<https://docs.microsoft.com/enus/azur>

[e/iot-hub/about-iot-hub](https://iot-hub/about-iot-hub).

[17] Fielding, R. 2000. Architectural Styles and the  
Design of Network-based Software  
Architectures.URL:[https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf).