

Automated Hostel Visitor's Management System Using Cognitive Services

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

SUSHMA EPHRIN E (Reg. No. 37110758)

SNEHA JOHNSON (Reg. No. 37110723)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119

MARCH – 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with —All grade by
NAAC Jeppiaar Nagar, Rajiv
Gandhi Salai,
Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **SUSHMA EPHRIN E (Reg. No. 37110758)** and **SNEHA JOHNSON (Reg. No. 37110723)** who carried out the project entitled —**Automated Hostel Visitor's Management System Using Cognitive Services** under my supervision from **November 2020** to **March 2021**.

**Internal Guide
MRS.DEEPA D, M.E.,(Ph.d.,)**

Head of the Department

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I **SUSHMA EPHRIN E** and **SNEHA JOHNSON** hereby declare that the Project Report entitled —**Automated Hostel Visitor's Management System Using Cognitive Services**” is done by us under the guidance of **MRS.DEEPA D, M.E.,(Ph.d.,)** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide , **MRS.DEEPA D, M.E.,(Ph.d.,) Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

As the technology developed at its peak, people, in general, were extremely dependable on its applications. Face Detection is one of all such technological inventions of humans that assures the safety of organizations to an excellent extent. Recognition software has been a major problem in the field of image analysis systems; it plays an important role in overcoming the style of applications from life sciences, police work, protection, identification to authentication. Many algorithms are being used to identify and distinguish human faces effectively. In this paper, we have used Image process techniques wherever the system acknowledges the faces of the guests. The system features a dataset of illustrious faces such that once an unknown face is detected within the organization, he/she is recognized as an associate unwelcome person. This may shield the scholars from any quiet invasion or attack. During this paper, we have mentioned various techniques which might be accustomed to implement the image process and assure the security of the scholars.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
1	INTRODUCTION	1
	1.1 Face Recognition	1
	1.2 Background Of Study	2
	1.3 Statement Of Research Problem	3
2	Literature Review	4
	2.1 Review Of Relevant Literatures	4
	2.1.1 Design and Implementation	4
	2.1.2 Biometrics and Security System	5
	2.2 Advantage And Disadvantage Of Biometric System	8
	2.3 Summary Of Literature	9
3	SYSTEM DESIGN AND METHODOLOGY	10
	3.1 Steps Involved in Face Detection	10
	3.2 Applications of Face Recognition System	10
	3.3 Limitation Of Facial Recognition Technology	11
	3.4 Solution Approach	13
	3.4.1 Steps to Implement Microsoft Azure Face API	13
	3.4.2 Image	14
	3.4.3 Face Detection	14
	3.4.4 Pre-Processing	15
	3.4.5 Feature Extraction	15
	3.4.6 Face Recognition	16
4	Experimental or materials and methods; algorithm used	18

	4.1 Software Requirements	18
	4.2 Hardware requirements	18
	4.3 .Net Framework	18
	4.4 Visual Studio	21
	4.5 Azure Cognitive services	27
	4.6 Azure SDK for .NET	32
	4.7 Proposed work	33
5	RESULTS AND CONCLUSION	37
	REFERENCES	42
	APPENDIX	
	A. PLAGARISM REPORT	
	B. SOURCE CODE	

LIST OF FIGURES

Figure no	Figure name	Page no
3.1	Azure cognitive services home page	13
3.2	Modules in face detection and recognition	14
3.3	Face recognition using azure cognitive services	17
4.1	Visual studio installer	26
4.2	Packages needed to be installed	26
4.3	Visual studio home page	27
4.4	Cognitive services home page	30
4.5	Creating a resource group	30
4.6	Resource group deployment completion page	31
4.7	Keys and endpoint section	31
4.8	Azure services from .NET applications	33
4.9	Architecture diagram	35
5.1	Creating a person group	36
5.2	Output screenshot of physical security's application	37
5.3	Output screenshot of warden / administrative application	38

LIST OF TABLES

Table no	Table name	Page no
4.1	Tools in designer	24
4.2	Other tools in visual studio	25

CHAPTER 1

INTRODUCTI

ON

Face recognition is the task of identifying an already detected object as a known or unknown face. Often the problem of face recognition is confused with the problem of face detection. Face Recognition on the other hand is to decide if the "face" is someone known, or unknown, using for this purpose a database of faces in order to validate this input face.

1.1 FACE RECOGNIZATION:

DIFFERENT APPROACHES OF FACE RECOGNITION:

There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

Recognition algorithms can be divided into two main approaches:

1. Geometric: Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features.
2. Photometric stereo: Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals.

1.2 Background of Study

Security is the major concern of today. The need for security is required everywhere from home to commercial, official and from defense point of view. Over the days the security provided or accessing system is developed based on password access, mechanical system based access and biometrical features based access. In biometrical based systems, fingerprint, iris, and face recognition and detection are widely used. Security systems have been developed over the years with different discrete access codes being employed, but also in the recent past, with the ability of face recognition principles or ideas being applied to a wide range of problems like criminal identification and crowd surveillance. Online Biometric security is one of the tools that Information Technology (IT) is trying to maximize to full potential. Today, security is one of the most important challenges we face in the day-to-day activities of life; there are several security systems, most of which are either electronic or manual. However, the most significant security check systems are those that use biometrics – which is unique to each individual. Iris, fingerprints, palm check, face recognition are all examples of biometric tools. Face recognition is a popular type of biometric system. Such systems are essentially pattern recognition systems that operate by fetching an individual's biometric data, extracting a feature set from the acquired data (in this case the face), and comparing this feature against the template set in the database. Depending on the application context, a biometric system may operate either in verification mode or identification mode. Face detection has gained significant importance in recent times. Law Enforcement Agencies, security organizations, personal identification systems and mass surveillance applications all use this technology. Human face detection has traditionally been a challenging task due to a number of factors like face size, image size, type and other conditions involved. The pictures containing faces may have different poses and angles; also there is a lot of difference in pictures and images with respect to resolution and contrast.

1.3 Statement of Research Problem

Security is one of the main objectives of every individual, everyone wants to feel safe within their home and work environment. This security requirement increases as the importance of the premises increases. For this reason, security personnel in places such as banks and other establishment of high importance such as military departments and many others may need assistance when it comes to security control. An efficient face recognition system can be of great help in forensic sciences, identification for law enforcement, surveillance, and authentication for banking and security system. The use of face recognition system can help to limit or inhibit the disadvantages and weaknesses of human based security systems thereby solving many of the problems and providing a much better solution for security control.

CHAPTER 2

LITERATURE

REVIEW

2.1 Review of Relevant Literatures

2.1.1 Design and Implementation of Facial Recognition Application System

Face recognition includes face verification and face identification. Face verification is one-to-one matching, which has been implemented in mobile phone and personal computer login systems. Face identification is one-to-many matching, where a huge database needs to be compared, and thus more challenging. It is envisioned in the future that face identification can be also carried out with a mobile terminal interface, which captures face images and accesses face recognition processor or database through wireless network. For instance, after a policeman captures a suspect's facial image, he can access the face recognition system remotely through a hand-held unit and identify the suspect on the spot. It will be more interesting if an un-manned surveillance system works independently and sends out alerts whenever somebody suspicious is in vision. The challenges in realizing this vision include: A wireless face recognition system requires considerable energy, computation, and bandwidth for image acquisition, processing, and transmission. Face recognition techniques need to be robust and accurate, and it should scale with the size of user group. Face detection plays an important role in applications such as human computer interface, face recognition video surveillance and face image database management. With the pervasive presence of low-cost, high quality cameras on mobile devices, we are witnessing the beginning of an explosive growth of embedded computer vision applications, such as mobile augmented reality, mobile image search and real-time image content analysis. There is some relevant work on wireless face recognition: Zaeri, Mokhtarian and Cherri discuss face recognition for wireless surveillance systems.

Rajani and Yan propose a prioritized transmission scheme for a wireless system especially designed for face recognition. Ikdong, Jaechang, Jason and Wayne

implement a wireless face recognition system based on ZigBee transmission protocol and Eigenface method with low power consumption. Though not designing for a face recognition system, Robert and Leonard propose a simulation framework and performance criteria for a distributed sensor processing over an adhoc wireless network, where a model of three conceptual layers in a wireless sensor network is proposed. The motivation of this research is that there are many benefits from constructing a wireless face recognition system. A wireless network is very suitable for constructing a flexible face recognition system, because a wireless setup incurs minimal modification to existing infrastructures, and a wireless network provides scalable and flexible monitoring region. If a wireless face recognition system has been set up, once the security level changes and the network needs to be moved, the wireless network can be moved and reused easily. Face recognition enhances the functionality and security of the wireless network, whilst a face recognition system is integrated with existing systems. Face recognition provides identity and expression information for event analysis to constitute a smart environment, and face recognition can replace or concert with other security measures to enhance the security.

2.1.2. Biometrics and Biometric Security System

A brief background of biometric and biometric security systems will provide a greater understanding of the concept of security. Biometrics is defined as the unique (personal) physical/logical characteristics or traits of human body (Jain, 2004). These characteristics and traits are used to identify each human. Any details of the human body which differs from one human to other will be used as unique biometric data to serve as that person's unique identification (ID), such as: retinal, iris, fingerprint, palm print and DNA. Biometric systems will collect and store this data in order to use it for verifying personal identity. The combination of biometric data systems and biometrics recognition/identification technologies creates the biometric security systems. Biometric security system is a lock and capture mechanism to control access to specific data. In order to access the biometric security system, an individual will need to provide their unique

characteristics or traits which will be matched to a database in

the system. If there is a match, the locking system will provide access to the data for the user. The locking and capturing system will activate and record information of users who accessed the data. The relationship between the biometric and biometric security system is also known as the lock and key system. The biometrics security system is the lock and biometrics is the key to open that Lock (Jain, 2006). There are seven basic criteria for biometric security system: uniqueness, universality, permanence, collectability, performance, acceptability and circumvention (Schuckers, 2001). As mentioned above, uniqueness is considered as the priority one requirement for biometric data. It will indicate how differently and uniquely the biometric system will be able to recognize each user among groups of users. For instance, the DNA of each person is unique and it is impossible to replicate. Universality is the secondary criteria for the biometric security. This parameter indicates requirements for unique characteristics of each person in the world, which cannot be replicated. For example, retinal and iris are characteristics which satisfy this requirement. Thirdly, a permanence parameter is required for every single characteristic or trait which is recorded in the database of the system and needs to be constant for a certain period of time. Following the permanence parameter is the collectability. The collectability parameter requires the collection of each characteristic and trait by the system in order to verify their identification. Then, performance is the next parameter for the system which outlines how well the security system works. The accuracy and robustness are main factors for the biometric security system. These factors will decide the performance of the biometric security system. The acceptability parameter will choose fields in which biometric technologies are acceptable. Finally, circumvention will decide how easily each characteristic and trait provided by the user can lead to failure during the verification process. DNA is believed to be the most difficult characteristic leading to the failure of the verification process.

Application fields for Biometrics Technology Physical access control refers to the process that requires the physical characteristics. On the other hand, logical access control is the schemes, procedures and techniques which are used in the system. The difference between logical and physical access control is really small and it can be confused easily because physical access control is controlled by

logical access control.

Physical access control:

Physical access control covers identity authentication processes which require users to provide physical characteristics. It is used in high security locations such as: hospitals, police stations, and the military. The most common use for the physical access control application is the access devices which are applied at doors or computers. This application is confidential and important and is entrusted with a high level of security. The physical access control reduces the risk of human problems. It also covers the aspect of data loss in the system. The system helps to eliminate the process of identifying long and complex pass codes with different processes. Physical access control is not only effective and efficient but also safe, secure and profitable in the workplace (O'Neill,2019)

Logical access control:

Logical access control refers to a process of a scheme control over data files or computer programs. These contain personal or privacy information of many different users. Logical access control is used by militaries and governments to protect their important data with high security systems using biometric technology. The only difference between logical access control and physical access control is that the logical access control is used for computer networks and system access control. It helps to reduce the burden of long and complex password requirements for users. Moreover, it is more secure and effective in the way of protecting and maintaining privacy over data in the system

2.1.3. Advantages and Disadvantages of Biometrics security system

Advantages:

The first advantage of using this new technology is the uniqueness and it is also the main characteristic which allows biometrics technology to become more and more

important in our lives. With uniqueness of biometrics technology, each individual's identification will be single most effective identification for that user. A chance of two users having the same identification in the biometrics security technology system is nearly zero (Tistarelli,2019). Secondly, the highly secure way of identifying users makes this technology less prone for users to share access to highly sensitive data. For example, users can share their fingerprints, iris and so forth allowing other users access to secure information. Each trait used during identification is a single property of that user. In other words, it is extremely hard or impossible to make duplicate or share biometrics accessing data with other users. This makes it ever more secure allowing user information and data to be kept highly secure from unauthorized users (Tistarelli, 2009). Lastly, this identification of users though biometrics cannot be lost, stolen or forgotten. This aspect of biometrics technology allows it to become more popular in its use. This method of identifying and giving access to user makes user identification a lot easier. Finally, most biometrics security systems are easy to install and it requires small amount of funding for equipment (except modern biometrics technology such as: DNA/retinal/irisrecognition) (Tistarelli, 2019).

Disadvantages:

Even though, there are many advantages of biometrics security system, it still has many flaws in its system. Each biometrics application method has weaknesses which can cause problems for its users. For example, if the biometrics security system uses fingerprints to identify its users and an accident causes a user to lose his/her finger then it can be a problem during the verification process. For voice recognition methods, illnesses such as strep throat can make it hard for authorized users to get access to their information. Another factor that can influence voice recognition systems is the continuous aging of its users. Noise in an environment where voice recognition is used to identify its users can also make it hard for users to be identified (PBworks, 2019). For iris or retinal scanning applications, users may find it very intrusive. Users may also have the concern for the safety of their eyes during the iris or retinal scan. Furthermore, databases used to store user

identification data will be

very large which might form a potential threat. For scanning retinal/iris characteristics and storing large amount of database, biometrics system requires new and modern technology. Therefore, the cost for equipment is also expensive (PBworks, 2020). Finally, lots of people are still concerned about biometrics technology in different aspects such as: security, adaptability to rate of change in life, scalability, accuracy, privacy and others.

2.3 Summary of Literature

The proposed system of this project is an enhancement of the face recognition systems reviewed above. With the aid of the literature review, decisions were made for the actualization of the face recognition system. The idea is to use still facial images captured to enhance the functionality of the system and to widen the scope of application of the system.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 Steps involved in face detection

While humans can recognize faces without much effort, facial recognition is a challenging pattern recognition problem in computing. Facial recognition systems attempt to identify a human face, which is three-dimensional and changes in appearance with lighting and facial expression, based on its two-dimensional image. To accomplish this computational task, facial recognition systems perform four steps. First face detection is used to segment the face from the image background. In the second step the segmented face image is aligned to account for face pose, image size and photographic properties, such as illumination and grayscale. The purpose of the alignment process is to enable the accurate localization of facial features in the third step, the facial feature extraction. Features such as eyes, nose and mouth are pinpointed and measured in the image to represent the face. The so established feature vector of the face is then, in the fourth step, matched against a database of faces.

3.2 Applications of face recognition system

- i. Law enforcement and justice solutions: Links victims to offenders, perform rapid and accurate searches.
- ii. Identification solutions: with regards to primary identification documents, the use of face recognition for identification programs has several advantages.
- iii. Homeland defense: This includes everything from preventing terrorists from boarding aircraft to protecting critical infrastructure.
- iv. Airport security: The use of biometric identification, can enhance security efforts already underway at most airports and other major transportation hubs.

- v. Immigration: Biometric technology, particularly face recognition software, can enhance the effectiveness of immigration and customs personnel.
- vi. Scene analysis and surveillance solution.

3.3 Limitations of facial recognition technology

Several factors limit the effectiveness of facial recognition technology:

- i. Image quality: Image quality affects how well facial-recognition algorithms work. The image quality of scanning video is quite low compared with that of a digital camera. Even high-definition video is, at best, 1080p (progressive scan); usually, it is 720p. These values are equivalent to about 2MP and 0.9MP, respectively, while an inexpensive digital camera attains 15MP. The difference is quite noticeable.
- ii. Image size: When a face-detection algorithm finds a face in an image or in a still from a video capture, the relative size of that face compared with the enrolled image size affects how well the face will be recognized. An already small image size, coupled with a target distant from the camera, means that the detected face is only 100 to 200 pixels on a side. Further, having to scan an image for varying face sizes is a processor-intensive activity. Most algorithms allow specification of a face-size range to help eliminate false positives on detection and speed up image processing.
- iii. Face angle: The relative angle of the target's face influences the recognition score profoundly. When a face is enrolled in the recognition software, usually multiple angles are used (profile, frontal and 45-degree are common). Anything less than a frontal view affects the algorithm's capability to generate a template for the face.

The more direct the image (both enrolled and probe image) and the higher its resolution, the higher the score of any resulting matches.
- iv. Processing and storage: Even though high-definition video is quite low in resolution when compared with digital camera images, it still occupies significant

amounts of disk space. Processing every frame of video is an enormous undertaking, so usually only a fraction (10 percent to 25 percent) is actually run through a recognition system.

To minimize total processing time, agencies can use clusters of computers. However, adding computers involves considerable data transfer over a network, which can be bound by input-output restrictions, further limiting processing speed. Ironically, humans are vastly superior to technology when it comes to facial recognition. But humans can only look for a few individuals at a time when watching a source video. A computer can compare many individuals against a database of thousands. As technology improves, higher-definition cameras will become available. Computer networks will be able to move more data, and processors will work faster. Facial-recognition algorithms will be better able to pick out faces from an image and recognize them in a database of enrolled individuals. The simple mechanisms that defeat today's algorithms, such as obscuring parts of the face with sunglasses and masks or changing one's hairstyle, will be easily overcome. An immediate way to overcome many of these limitations is to change how images are captured. Using checkpoints, for example, requires subjects to line up and funnel through a single point. Cameras can then focus on each person closely, yielding far more useful frontal, higher-resolution probe images. However, wide-scale implementation increases the number of cameras required. Evolving biometrics applications are promising. They include not only facial recognition but also gestures, expressions, gait and vascular patterns, as well as iris, retina, palm print, ear print, voice recognition and scent signatures. A combination of modalities is superior because it improves a system's capacity to produce results with a higher degree of confidence. Associated efforts focus on improving capabilities to collect information from a distance where the target is passive and often unknowing. Clearly, privacy concerns surround this technology and its use. Finding a balance between national security and individuals' privacy rights will be the subject of increasing discussion, especially as technology progresses.

3.4 SOLUTION APPROACH:

3.4.1 steps to implement Microsoft Azure Face API

Create Face API service in MS Azure, search for cognitive services in the search bar.

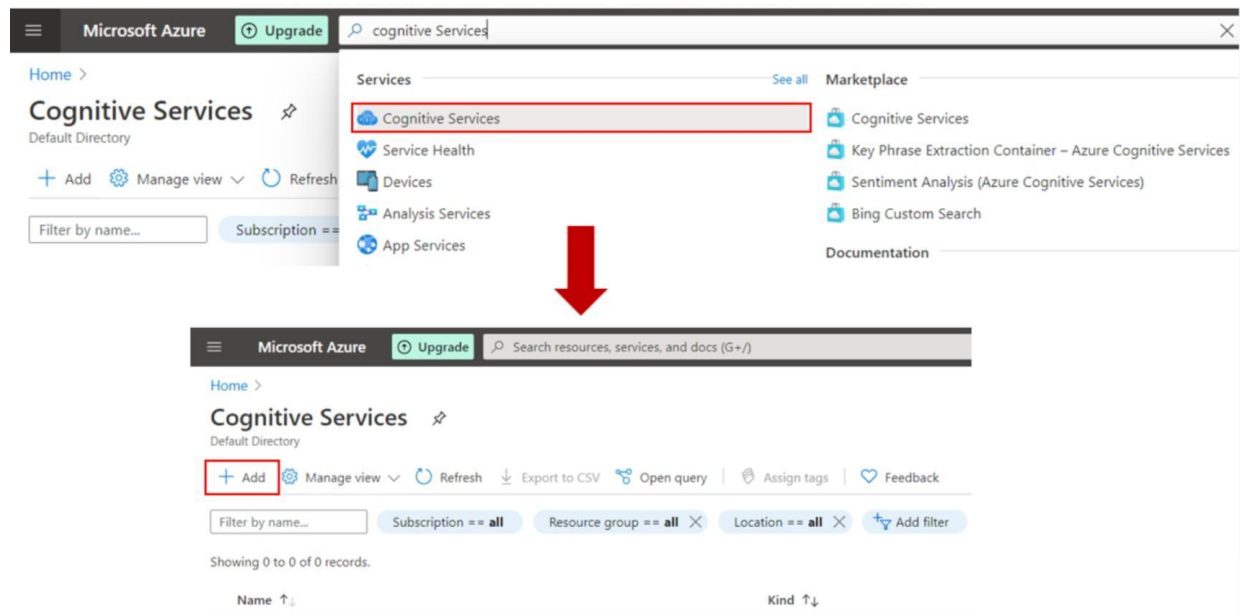


Fig 3.1 Azure cognitive services home page

Go to —All Services menu, where you can see your services listed. Note the subscription id, end point, and access keys which we need later in our program.

Select the appropriate price tier for creating the service. In this example, we are using the —Free Price Tier for demo purpose.

Authorize API call: While using face API client services, you should authorize yourself for a valid subscription key.

Create person group: Person group acts as a container of all persons. In the following code snippet, replace **personGroupId** with your group name as per your requirement (example: mygroup, group1).

Add person to group: Here we are adding persons to previously created group.

Add person's faces (Take Training images)

Train person group: The person group must be trained prior to performing an identification using it. Moreover, it has to be re-trained after adding or removing any person, or if any person has their registered face edited.

Identify Person: If the person's image is already trained then, it will identify their faces.

3.4.2 Image:

First, we need to store the images in the azure cloud with the people we want to provide access to and train our model. The service support until 1 million different persons and 248 pictures from each person. Once the training process has finished, you can use the face.identify method to check if the incoming face image corresponds to any of the people registered in the cloud.

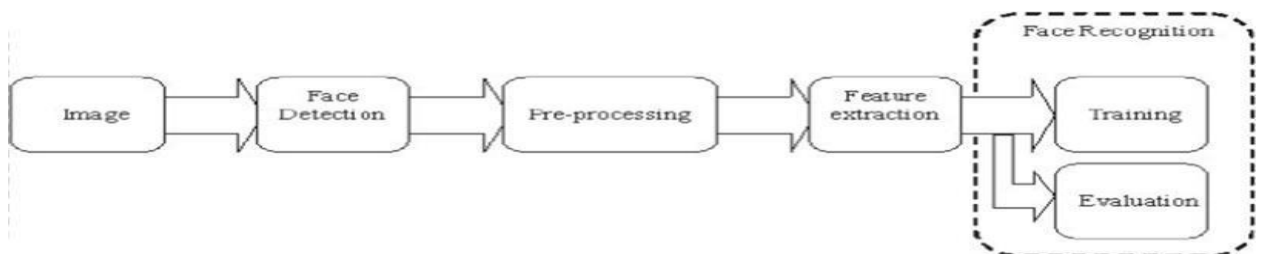


Fig 3.2 Modules in face detection and recognition

3.4.3 Face Detection

The Face service detects human faces in an image and returns the rectangle coordinates of their locations. Optionally, face detection can extract a series of face- related attributes, such as head pose, gender, age, emotion, facial hair, and glasses. It gives us all these operations for managing person groups. Once our person group is created, we can add n number of persons to that person group. After we add a person

to a person group, then we add n number of faces to each person. Once we've added as many persons and faces to our person group that we want, we then invoke the Face API method to train our person group. Once our person group is trained, we're ready to do face identification. Face identification always starts with face detection. The key result is that we get a unique faceId for each face in face detection. Once we invoke detection, we can finally invoke face identification using the results of our face detection calls along with the person group we want to use for face identification.

3.4.4 Pre-Processing

Quality of image plays a vital role in increasing face recognition rate. A good quality image gives better recognition rate than noisy images. Recent research on face recognition reveals that image capturing devices such as video cameras, also impacts the recognition rate as sometimes they capture low quality and/or noisy images due to their poor quality. In such case the images captured will have low resolution, noisy, contrast variation and/or varying brightness throughout the image. It is more difficult to extract features from such noisy images which in-turn reduces face recognition rate. To overcome problems occurred due to low quality image, pre-processing is done before extracting features from the image

3.4.5 Feature Extraction

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set. The technique of extracting the features is useful when you have a large data set and need to reduce the number of resources without losing any important or relevant information. Feature extraction helps to reduce the amount of

redundant data from the

data set. In the end, the reduction of the data helps to build the model with less machine's efforts and also increase the speed of learning and generalization steps in the machine learning process.

3.4.6 Face recognition

Face recognition is an important research problem spanning numerous fields and disciplines. This is because face recognition, in addition to having numerous practical applications such as bankcard identification, access control, Mug shots searching, security monitoring, and surveillance system, is a fundamental human behavior that is essential for effective communications and interactions among people. A formal method of classifying faces was first proposed. The author proposed collecting facial profiles as curves, finding their norm, and then classifying other profiles by their deviations from the norm. This classification is multi-modal, i.e. resulting in a vector of independent measures that could be compared with other vectors in a database. Progress has advanced to the point that face recognition systems are being demonstrated in real world settings. The rapid development of face recognition is due to a combination of factors: active development of algorithms, the availability of a large database of facial images and a method for evaluating the performance of face recognition algorithms. In the literatures, face recognition problem can be formulated as: given static (still) or video images of a scene, identify or verify one or more persons in the scene by comparing with faces stored in a database. When comparing person verification to face recognition, there are several aspects which differ. First, a client – an authorized user of a personal identification system – is assumed to be co-operative and makes an identity claim. Computationally this means that it is not necessary to consult the complete set of database images in order to verify a claim. An incoming image (referred to as a probe image) is thus compared to a small number of model images of the person whose identity is claimed and not, as in the recognition scenario, with every image (or some descriptor of an image) in a potentially large database. Second, an automatic authentication system must operate in near-real time to be acceptable to users. Finally, in recognition

experiments, only images of people from the training database are presented to the system, whereas the case of an imposter (most likely a previously unseen person) is of utmost importance for authentication. The most famous early example of a face recognition system is due to Kohonen, who demonstrated that a simple neural net could perform face recognition for aligned and normalized face images.

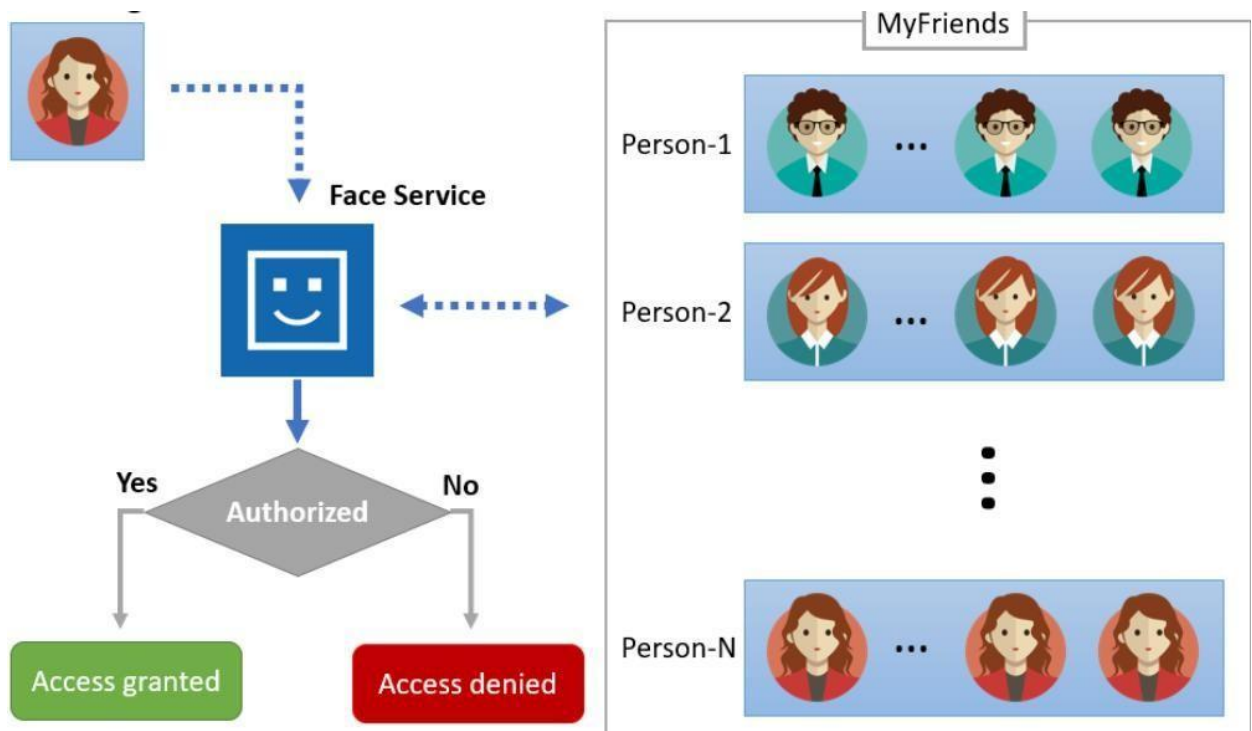


Fig 3.3 Face recognition using azure cognitive services

The type of network he employed computed a face description by approximating the eigenvectors of the face image's autocorrelation matrix; these eigenvectors are now known as eigenfaces. While several were successful on small databases of aligned images, none successfully addressed the more realistic problem of large databases where the location and scale of the face is unknown. Later introduced an algebraic manipulation which made it easy to directly calculate the eigenfaces, and showed that fewer than 100 were required to accurately code carefully aligned and normalized face images. Then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery and to determine the precise location and scale of faces in an image.

CHAPTER 4

Experimental or materials and methods; algorithm used

4.1 Software Requirements:

- .Net Framework
- Visual Studio
- Azure cognitive services
- Azure .Net SDK

4.2 Hardware requirements :

- laptop or desktop with web-camera.
- 4 GB RAM

4.3 .Net Framework

.Net Framework is a software development platform developed by Microsoft for building and running Windows applications. The .Net framework consists of developer tools, programming languages, and libraries to build desktop and web applications. It is also used to build websites, web services, and games.

The .Net framework was meant to create applications, which would run on the Windows Platform. The first version of the .Net framework was released in the year 2002. The version was called .Net framework 1.0. The Microsoft .Net framework has come a long way since then, and the current version is .Net Framework 4.7.2.

The Microsoft .Net framework can be used to create both - Form-based and Web-based applications. Web services can also be developed using the .Net framework. The framework also supports various programming languages such as Visual Basic and C#. So developers can choose and select the language to develop the required application. In this chapter, you will learn some basics of the .Net framework.

Visual Studio is the development tool that is used to design and develop the .NET applications. For using Visual Studio, the user has to first install the .NET framework on the system.

In the older version of Windows OS like XP SP1, SP2 or SP3, .NET framework was integrated with the installation media. Windows 8, 8.1 or 10 do not provide a pre- installed version 3.5 or later of .NET Framework. Therefore, a version higher than 3.5 must be installed either from a Windows installation media or from the Internet on demand. Windows update will give recommendations to install the .NET framework .It is a complete development and execution environment. That allows the developer to develop, run, and deploy the following application.

- Console application.
- Windows application
- Web application
- Service-oriented application (WCF)
- Web service
- Windows service
- WPF, WWF, etc.

.NET Framework supports the object-oriented programming model for multiple languages such as C#, VB, vC++, J#, etc. It supports language interoperability. Interoperability implies that each language can use the code written in some other language. It provides a common set of tools and enables easy integration of modules developed with one another. It also enables the developers to create a shareable component to be used in distributed computing architecture.

Need for .NET Framework:

- .NET Framework was designed to fulfil the following goals.
- Interoperability
- Cross-platform Support
- Language Independence

Base class library.
Easy development
Security

Interoperability:

.NET Framework supports interoperability between new applications and existing applications. It implies that the code written in one language can be used in some other language.

Cross-Platform support:

Each .NET compatible language, such as C#, VB, VC++ provides its own compiler to compile code to MSIL. After that, the common runtime engine compiles MSIL code to native code with the help of the Just In Time (JIT) compiler and then run the application.

Cross Platform:

Language Independence

.NET framework provides multiple language support by using a feature known as a common type system. That is built into common language runtime.

Base Class Library:

The base class library provides the fundamental building block for any application you develop. These applications could be windows, web, console, or service-oriented. The BCL mostly serves as the main point of interaction with the CLR.

Easy Deployment:

.NET Framework makes the task of employment easier. In most cases, installing an application along with a component on the target computer is easier.

Security:

.NET Framework provides a security model that allows you to authorize and authenticate the user.

4.4 Visual Studio:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

As of March 2021 the current production-ready Visual Studio version was 2019, with older versions such as 2012 and 2013 on Extended Support, and 2015 and 2017 on Mainstream Support.

Visual Studio Editions:

There are 3 editions of Microsoft Visual Studio as follows:

1. Community: It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than \$1 Million(US Dollars) then you are not permitted to use this edition. In a non- enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem(Access to thousands of extensions) and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.
2. Professional: It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility(Professional developer tools for building any application type), Productivity(Powerful features such as CodeLens improve your team's productivity), Collaboration(Agile project planning tools, charts,

etc.) and Subscriber benefits like Microsoft software, plus Azure, Pluralsight, etc.

3. Enterprise: It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high- quality software.

To improve functionality at many levels, plug-ins may be added to provide support for source-control systems, while domain-specific language editors and visual designers are examples of new tool sets that may be added via these plug-ins. Built-in language services offer support for different well-known languages (like C#, C++, VB, and F#), while language services may be installed separately to support languages, including M, Python, and Ruby, among other supported languages. There are Visual Studio versions that support individual languages, which are a cost-effective solution for beginners who only need Visual Basic, or advanced users who work on fast processing applications and continually need visual C# or visual C++, but are not interested in low-speed Visual Basic.

Features of Visual Studio:

Visual Studio provides the most advanced features that the latest developers are looking for. Few of the most important features are described as follows.

Code Editor:

The Visual Studio has the best in class Code editor which supports multiple functionalities. It enables bookmarks in code to incorporate Quick Navigation. In addition to the normal text, VS also provides Incremental Search, Regex Search, Multi-item Clipboard, and Task-list. The Editor is also capable to support Code refactoring, Parameter Reordering, Variable and Method Renaming, Interface Extraction and many more user-friendly functionalities.

Debugger

The remarkable Debugger of the MS Visual Studio can work both as a Machine Level Debugger as well as a Source Level Debugger. It is capable to debug Mapped Code as well as Native Code. It is capable to compile any programming language that is supports VS.It is capable to attach itself to the currently running process to conduct monitoring and debugging procedures on the code, provided if the source code is available. VS can incorporate Memory Dumps and Multi-threading as well.

Designer

Visual Studio includes a number of visual designers to help within the development of applications. Some of the tools are listed below.

Table 4.1 Tools in designer

Designer	Description
WPF Designer	WPS authorizes user interfaces for Windows Presentation Foundation.
Windows Forms Designer	Windows Forms designer creates GUI applications using Windows Forms.
Class Designer	The Class Designer authorises and edits the classes including its members and their access using UML modelling.
Web Designer	Visual Studio additionally includes a web-site editor and designer that permits websites to be authored by dragging and dropping widgets.
Mapping Designer	Mapping designer is employed by LINQ to SQL to design the mapping between information schemas and therefore the classes that encapsulate the data.
Data Designer	The data designer is used to graphically edit information schemas, as well as written tables, primary and foreign keys and constraints.

Table 4.2 Other tools in visual studio

Tool	Description
OpenTabs Browser	Properties editor edits the properties in the GUI in VS.
Properties Editor	It enlists all the properties available for classes, objects, web pages.
Server Explorer	Server Explorer manages Database connections on a computer.
Object Browser	Object browser is used to browse the namespaces in managed assemblies.
Team Explorer	Team Explorer Integrates the Azure DevOps Into the VS IDE.
Data Explorer	Data Explorer manages MS SQL databases and instances.
Solution Explorer	Solution Explorer manages and browses the files in a solution.

Extensibility

Visual Studio permits developers to write down extensions for Visual Studio to extend its capabilities. These extensions —plug into Visual Studio and extend its functionality. Extensions are available in the form of macros, add-ins, and packages.

Why should you choose Visual Studio?

The main reason why Visual Studio became so popular is that it has everything which any programmer expects from any code editor with some additional and useful features. It's lightweight, fast, open-source and cross-platform nature along with other cool features gives it an extra edge over any other editor

Install Visual Studio:

Step 1: Download Microsoft Visual Studio

Google for the latest version of Visual Studio and download the installer file into your local system and then run the installer file as an administrator.

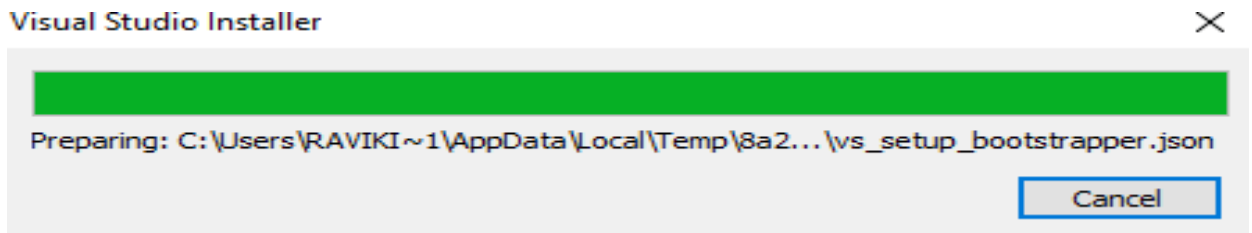


Fig 4.1 Visual studio installer

Step 2: Select the .NET Desktop Development Package

Once you run the Installer, the Visual Studio Editor will be successfully downloaded into your local system, Later a dialogue box will be displayed on your desktop screen asking for a particular package you need in your system. Here, you need to select the .NET Desktop Development package.

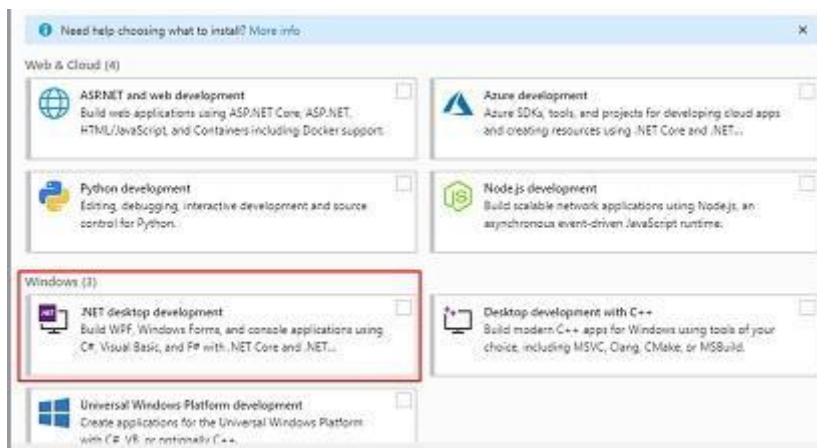


Fig 4.2 Packages needed to be installed

Step 3: Set C# Environment

Once your packages for .NET Development are downloaded, then another dialogue box will be displayed on your screen asking for the development environment you are looking for. Here, you might select the environment for C# or any other language, depending upon your requirement.

Step 4: Create your First Project

Once the environment is set, you are all good to go. Start your Visual Studio and select create new project option in the displayed dialogue box.

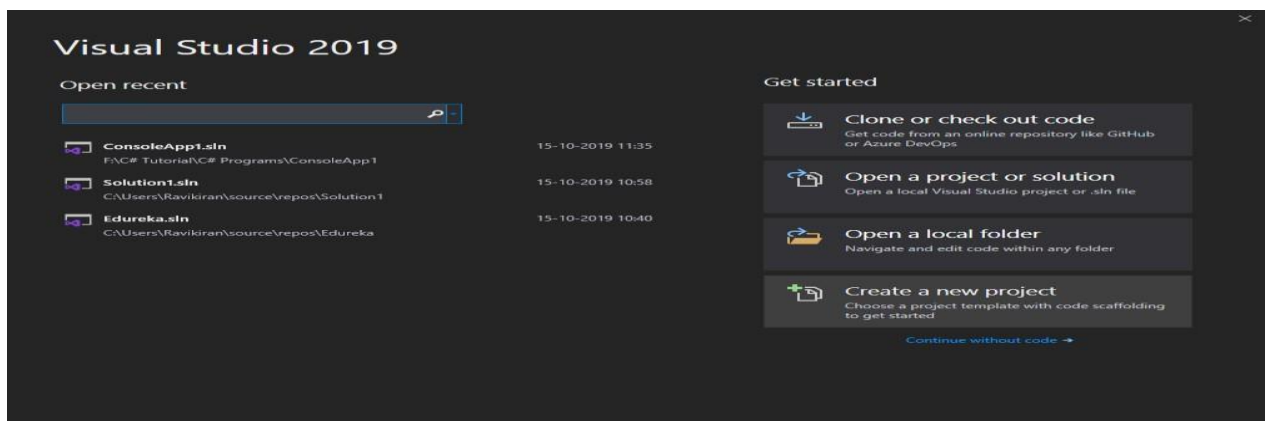


Fig 4.3 Visual studio home page

You will be redirected to the next dialogue box and there you need to select Class library as .NET Standard

In the next dialogue box, you will be asked to Configure your project. Configure it and you are now in the Editor.

4.5 Azure Cognitive services

Azure Cognitive Services is a comprehensive family of AI services that helps you build intelligent software applications. No machine machine-learning expertise is

required to use them; an API call alone can embed AI capabilities into your existing applications. Azure Cognitive Services includes 30 APIs that belong to the following categories:

Decision APIs—Help you to make decisions faster, detect anomalies, or moderate content

Vision APIs—Assist you in identifying and analyzing content in images, videos, and digital ink

Speech APIs—Integrate speech processing into apps (e.g., speech to text)

Search APIs—Give you the power of BING in a box

Language APIs—Allow you to easily infuse natural language processing, text analytics, and technology into chatbot development

Azure Cognitive Services can be deployed anywhere in the cloud, and, if data privacy is a concern, containers can be used to give you total flexibility and control over API consumption. The consumption of Azure Cognitive Services has been standardized. It typically involves the following steps:

1. Provision the Cognitive Service in the Microsoft Azure Portal.
2. Take note of the API key.
3. Build a REST request with the required parameters.
4. Invoke the request and parse the API response.

Initially released in 2016, the Face API belongs to the Microsoft Cognitive Services set of AI cloud services and gives you access to some of the most advanced face recognition algorithms that are currently on the market. The main features of the Face API are centred around face detection with attributes, face recognition and emotional recognition which allow you to:

- Detect human faces
- Find similar faces

- Group faces based on similarity
- Identify previously tagged people using their faces

The properties of the Azure Cognitive Services, like their service level agreement, their pricing, and how you can scale them. It is important to know these things if you are going to use the Cognitive Services. We start by looking at the service level agreement. The Azure Cognitive Services are hosted in Azure. From the Azure data centers, endpoints are exposed for the different Cognitive Services APIs that your applications can use. Once you start using the services, your application will rely on them, and you need them to work. Currently, Microsoft guarantees that the Cognitive Services are available 99.9% of the time. This is the current number, and it is likely to go up. 99.9% is pretty good, but this does mean that you should assume that the services are down for about 43 minutes every month. Onto pricing the first thing that is that pricing is part of Azure. You create Cognitive Services in your Azure subscription and their costs go into that, so they show up on your Azure bill. Alright, there are different pricing models for the Azure Cognitive Services. The first model is the free pricing tier.

The Basic pricing tier is designated for production sites, supporting smaller sites, as well as development and testing scenarios. The Basic pricing tier includes core Azure Web Sites abilities, allowing you to run production sites at a lower cost. Basic pricing tier offers up to three instances of dedicated Virtual Machine running your web sites, SSL support, Scheduled Jobs, Endpoint Monitoring and more. The free tier is limited to 5000 calls at the moment, and is a lot slower than the paid tier.

Vision Services – these are used to analyze static or moving images, face recognition has been around for a while (it's probably built into the digital camera you are using), but these services can recognize age, gender, emotion (happy, angry, surprised etc.) and be trained to recognize specific people. In addition, it can recognize and describe objects. We all take far too many pictures with digital cameras, because it's so easy right? As a result, we have hundreds (or thousands if you're me) of unorganized

pictures. Imagine indexing all those pictures and categorizing them by the people in them, or the pictures where people are happy or the ones with my red car in. You will start to get an idea of the power of Vision Services.

- Login to portal.azure .
- To create a resource, click in create a resource.
- Search for cognitive services , and give —Createll.

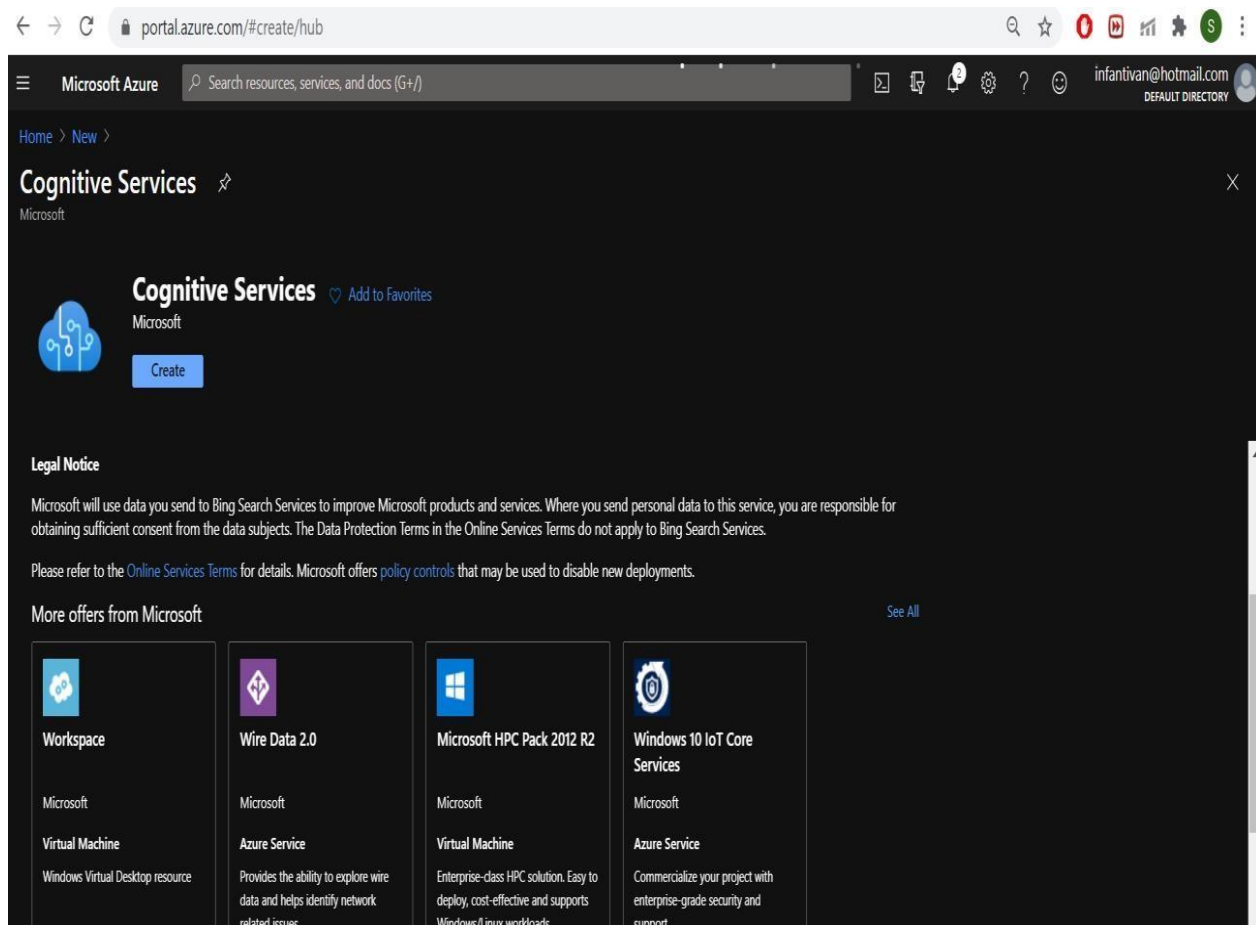


Fig 4.4 Cognitive services home page

- Give a resource group name , instance name and select the pricing tier and press —Review+Createll.

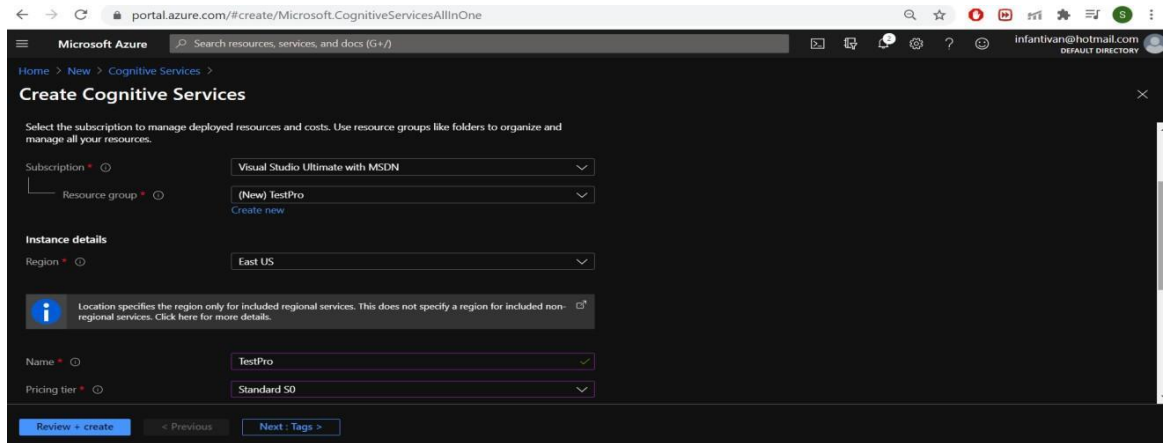


Fig 4.5 Creating a resource group

- When it is created , you will get as in the image below

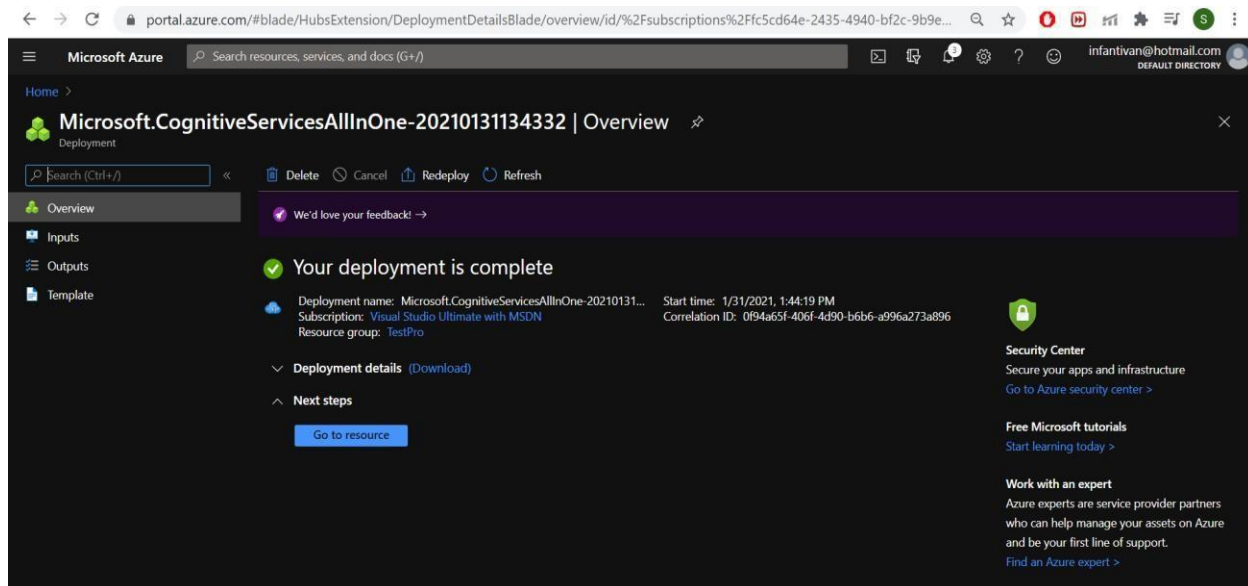


Fig 4.6 Resource group deployment completion page

- You need to click on —Go To resourcell
- In that you need to click on —Keys and Endpointll. You need to note it down or store it somewhere. By using those key and endpoint we will be able to access to azure portal.

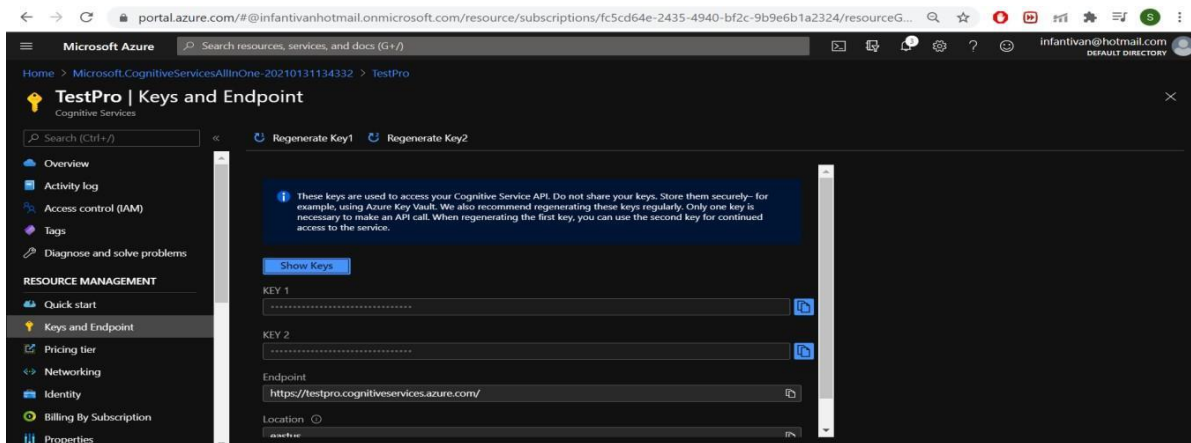


Fig 4.7 Keys and endpoint section

4.6 Azure SDK for .NET

The Azure SDK for .NET is designed to make it easy to use Azure services from your .NET applications. Whether it is uploading and downloading files to Blob Storage, retrieving application secrets from Azure Key Vault, or processing notifications from Azure Event Hubs, the Azure SDK for .NET provides a consistent and familiar interface to access Azure services.

The Azure SDK for .NET is available as series of NuGet packages that can be used in both .NET Core (2.1 and higher) and .NET Framework (4.6.1 and higher) applications. Use the Azure SDK for .NET in your applications

To use an Azure SDK package in one of your .NET applications, you want to follow these steps.

Locate the appropriate SDK package - Use the package list to find the appropriate package for the Azure service you are working with. Be advised that most services have a client package for working with the service and a management package for creating and managing instances of the service. In most cases, you will want the client package. Install this package in your application using NuGet.

Set up authentication for your application - To access Azure resources, your application will need to have the appropriate credentials and access rights assigned in Azure. Learn how to configure authentication in Authenticating .NET applications to Azure.

Write code using the SDK in your application - When working with Azure services, your code will first create a client object to work with the service and then call methods on that client object to interact with the service. Both synchronous and asynchronous methods are provided. Examples of using each individual SDK package are provided throughout the Azure documentation.

Configure logging for the SDK (optional) - If you need to diagnose issues between your application and Azure, you can enable logging in the Azure SDK for .NET.

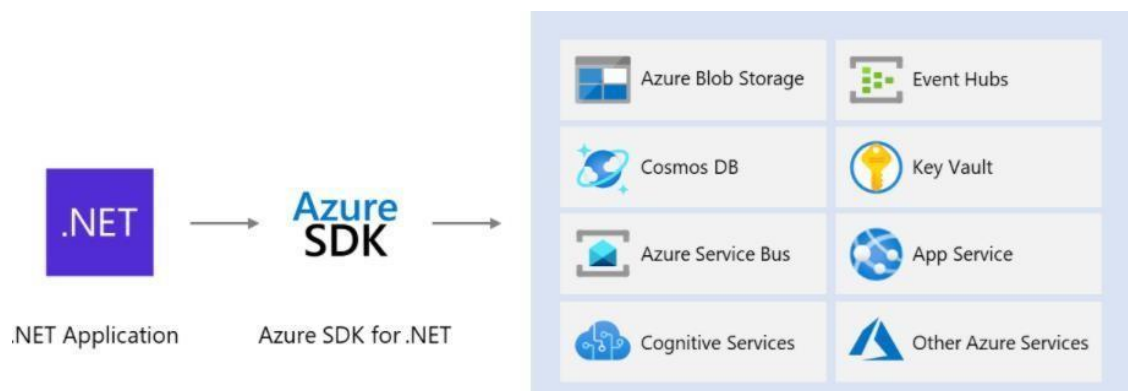


Fig 4.8 Azure services from .NET applications

4.7 Proposed work :

This system can detect faces in real time via a camera module. In Person's Face Training module each student can register up to a specific no of visitors. To register their faces in the database, they need to provide the photographs of each of them in all different angles in order to get better results. Then their photographs will be stored

in Azure Cloud Services and those particular faces will be trained. After that the identification module whenever someone enters the camera module, Our system will detect the face of the person if the system is unable to detect the face then it will show error message so at that time the physical securities can override the system, If the system detects the person's face without any issue then it will check whether the person detected is a registered visitor or a new visitor . In case of registered visitor he/she will be allowed inside if it is a permitted visiting hour, if the person is a new visitor who doesn't come under list of registered visitor they can speak to higher authorities for permission as per our system it will not allow them further , but if they have proper permission then the physical securities can allow them inside. As an enhancement to our system, it will also detect whether the visitor is wearing a mask inorder ensure whether they follow the covid norms.

There are two modules in this project , they are :

1. Person's Face Training
2. Identification and Mapping

Person's Face Training :

Each student residing in the hostel will be allowed to register a specific no of visitors. To register their faces each one should provide photographs of their visitors in many different angles (eg : top, bottom, left, right, with mask and without mask). These images of the visitors will be uploaded in Azure Cloud Portal by using our API .We are allowed to add more photographs of the visitor so that the accuracy will be increased. When we train the system with more number of photographs of visitors then the probability of correctly identifying a person's face will be high. Each photograph of the visitors will be mapped with their name. In order to store the faces in Azure cloud portal we need to create an account, then create cognitive service resource, you can choose your pricing scale according to your need. Once the resource is created it will generate a subscription key and endpoint. Using the subscription key the API

consumers will get access to the API's published through an API Management instance and the Endpoint is a network interface which connects securely to the service powered by azure.

Person's Face Identification and Mapping :

In Identification mapping module, Whenever a visitor goes through the camera module, their faces will be detected, by using the detected image the system will check if the face detected is a registered visitor or an unauthorized person accordingly it will give the output as whether the person is allowed to enter or not. If you want to add a visitor then you need to add the person's face in the person's face training module. Our system will also show a warning message if the visitor doesn't wear mask. If there is any problem in detecting a face at that time the physical securities can override our system and do the needful action.

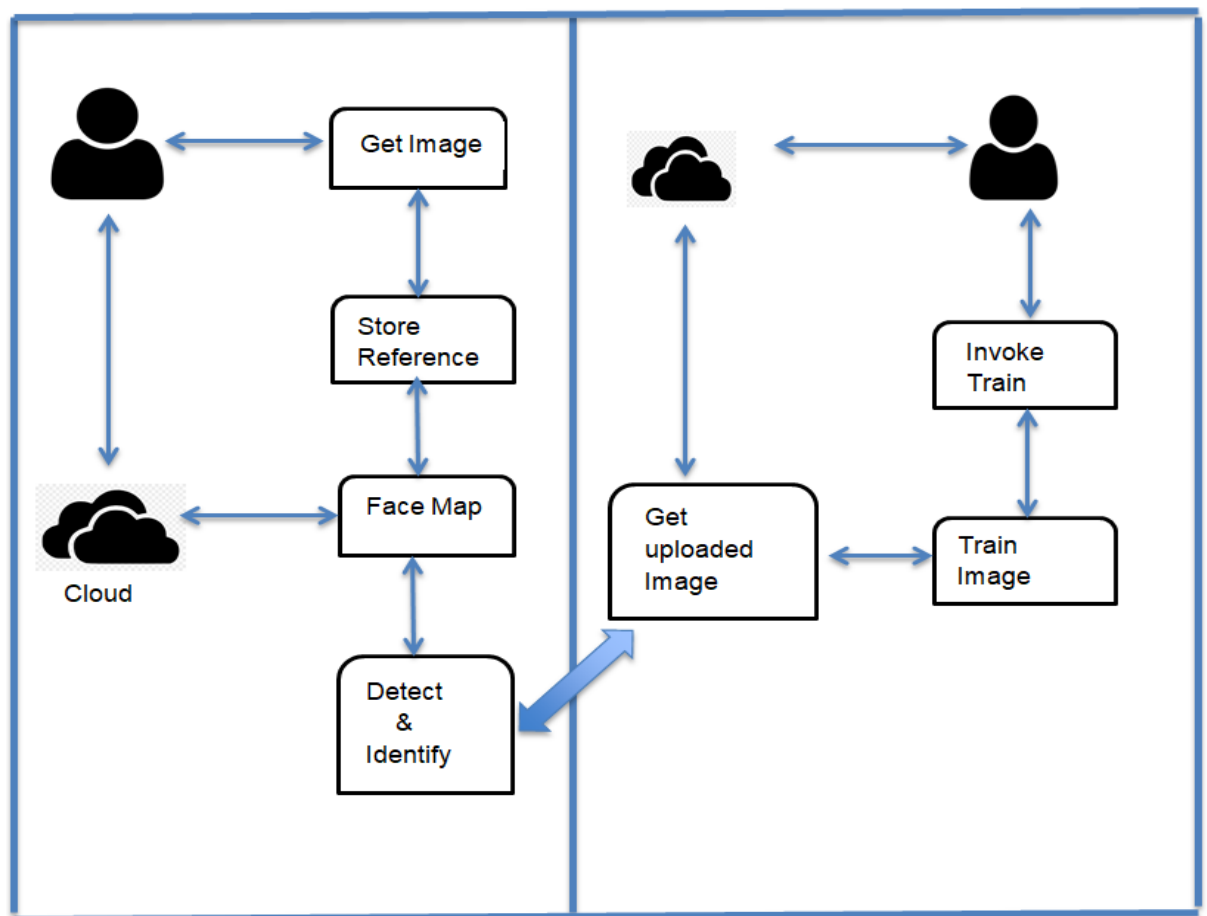


Fig 4.9 Architecture diagram

CHAPTER 5

RESULTS AND

CONCLUSION

This system can be used in many other areas to keep track of their visitors like schools, colleges etc. This system reduces the work pressure of physical securities during a busy day. It is the easiest and safest method of security, as it will not grant access to any unregistered visitor to enter inside the premises. If we give RFID card for the visitors there is a possibility of misusing the access. It will give an alarm if any banned visitors try to enter. It reduces the manual work of securities, also improves the hostel visitors experience and overall security of the campus. It has faster processing speed, hence saves the time. It is very easy to manage, it detects the faces in real time We can also ensure covid- specific norms like, wearing marks is followed

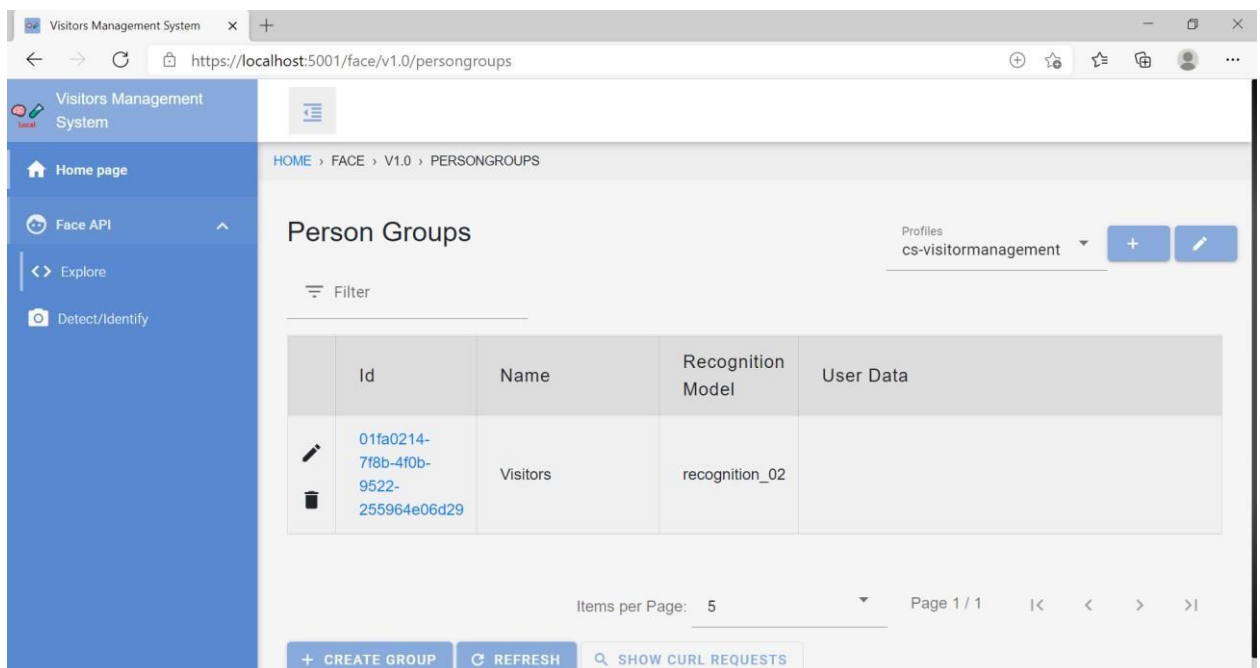


Fig 5.1 Creating a person group

This is the Visitor management system, at first we need to create a person group by adding Id, name and recognition model .After creating a person group we need to create person. We can create as many no of person according to our requirement. After adding persons to the group we need to train the group. There are two modules one module is exclusively for the warden which contains all the details about the visitor. The other module is to be kept near the security gate, so that the physical security can allow the registered visitor inside.

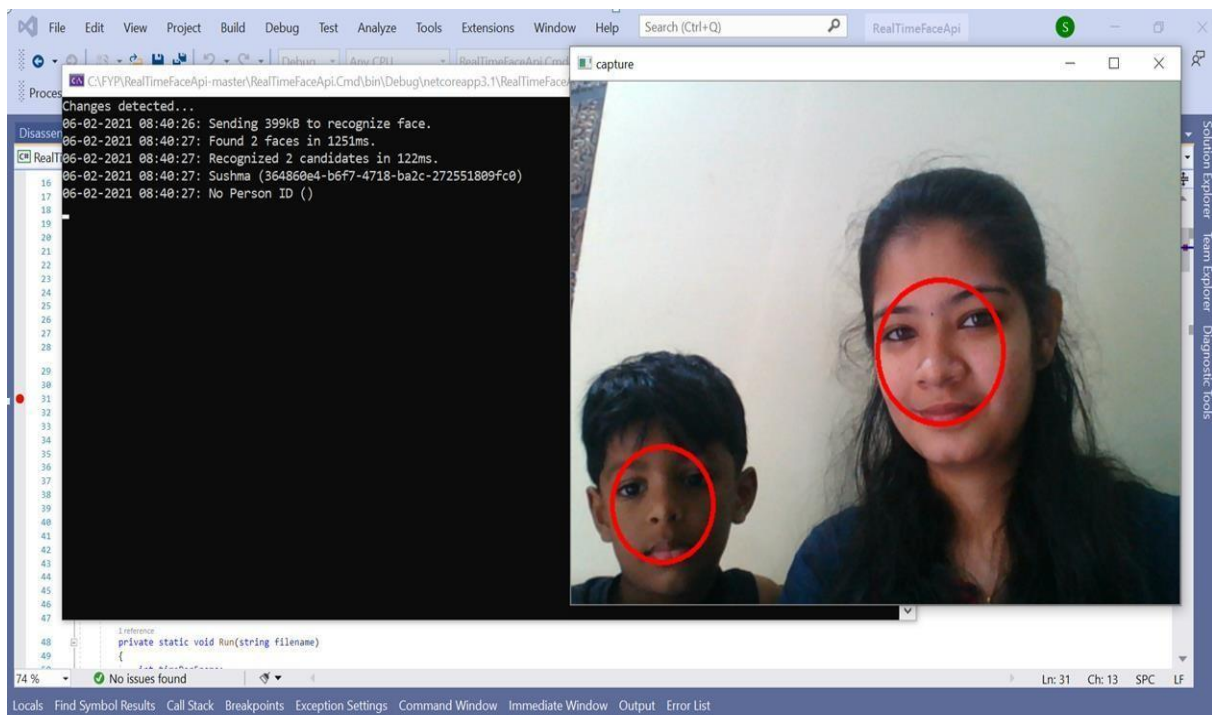


Fig 5.2 : Output screenshot of physical security's application

Now our system is able to detect faces and recognize whether he/she is a registered visitor, it will print their name along with their personId if he/she is a registered visitor or else it will print as there is no specific personId for this face.

When a face is detected in the screen, then it will print the no of faces detected and their name and personId, if there it detects a person who is a unregistered visitor then it will print as no personID found. It can detect as many no of faces according to the pricing tier which we choose in Azure cloud portal.It shows the confidence when it

identifies the person. You can click on the personID to view more details about the person.

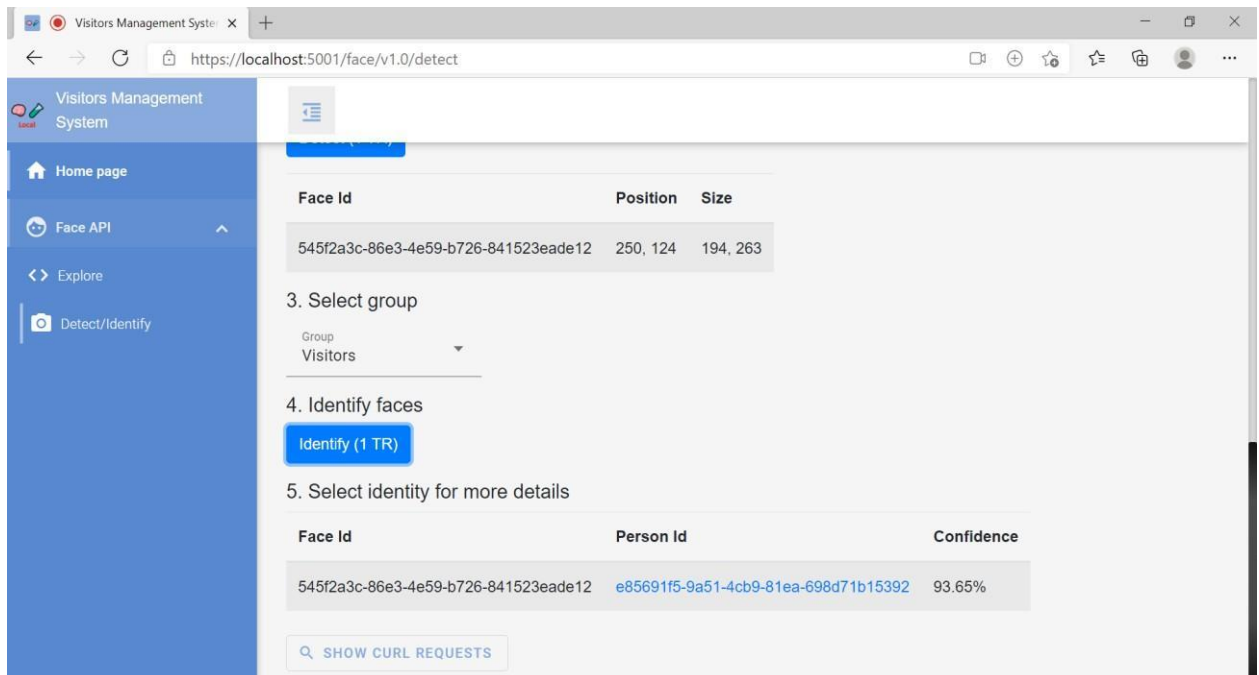


Fig 5.3: Output screenshot of warden/administrative application

This system reduces the work pressure of physical securities during a busy day. It is the easiest and safest method of security, as it will not grant access to any unregistered visitor to enter inside the premises. If we give RFID card for the visitors there is a possibility of misusing the access. It will give an alarm if any banned visitors try to enter. It reduces the manual work of securities, also improves the hostel visitors experience and overall security of the campus. It has faster processing speed, hence saves the time. It is very easy to manage, it detects the faces in real time We can also ensure covid- specific norms like, wearing marks is followed This is the Visitor management system , at first we need to create a person group by adding Id , name and recognition model .After creating a person group we need to create person . We can create as many no of person according to our requirement. After adding persons to the group we need to train the group. There are two modules one module is exclusively for the warden which contains all the details about the visitor. The other module is to be kept near the security gate,

so that the physical security can allow the

registered visitor inside. Now our system is able to detect faces and recognize whether he/she is a registered visitor, it will print their name along with their personID if he/she is a registered visitor or else it will print as there is no specific personID for this face. When a face is detected in the screen, then it will print the no of faces detected and their name and personID, if there it detects a person who is a unregistered visitor then it will print as no personID found. It can detect as many no of faces according to the pricing tier which we choose in Azure cloud portal. It shows the confidence when it identifies the person. You can click on the personID to view more details about the person.

Security is an all-time global concern, everyday a large number of human lives and properties are lost because of insecurity. The face recognition system has become very imperative in our daily lives, organizations of high importance use this system in their daily routines, it aids them in carrying out proper security measures in their environments. We propose an efficient system that aids security personnel's in security control. The system accepts 2D images of faces and compares them in order to attain if both are of the same or not. The system is sub-divided into three modules the verification, enrolment and identification module, the comparison of images are done on the identification module while the enrolment module is where new profiles are set and stored in the database. The aim of this project is to design and implement a face recognition system for security control which is affordable.

Researchers are looking for a solution to strengthen security in our environments, the proposed system is an efficient system that will aid security personnel secure and control their environment. From our experience, Security is the major concern of every individual the need for security is required everywhere from home to commercial, official and from defense point of view.

Hence a face recognition system for security control has had a major effect in transforming the way security personnel's control their environment. There are many applications and solutions of face recognition technology used in security systems. It

has many advantages which can improve our lives such as: improved security and effectiveness, reduced fraud and password administrator costs, ease of use and makes live more comfortable. Even though the face recognition security system still has many concerns such as information privacy and physical privacy, users cannot deny the fact that this new technology will change our lives for the better.

REFERENCES

- [1] Mengzi San, Megan Zhao, Zoning Ru, energy Zhang — VarGFaceNet: academic degree economical Variable cluster Convolution Neural Network for light-weight Face Recognition, Horizon Computing, 2019.
- [2] Ali, Abdelmgeid A., Tarek ABD El-Hafeez, and Yosra Khalid Mohany. —A System for Face Detection and Recognition, Journal of Advances in arithmetic and laptop Science (2019): 1-19.
- [3] Kumar, inheritable la Ashish, et al. —Smart human activity Marking System mistreatment face recognition, analysis Journal of Science and Technology eleven. (2019): 101-108.
- [4] Abdullah, Ahmed SS, Majid Ali Abed, and Israa Al Barazanchi. —Improving face recognition by Elma neural network victimization curve let remodel, and HSI color space, Periodicals of Engineering and Natural Sciences seven.2 (2019): 430-437.
- [5] S. F. Kay and F. M. Mustafa, ‘Smart Home System supported Face Recognition in Real-Time’, in 2019 International Conference on Science and Engineering, ICO ASE 2019, 2019, pp. 40–45, DOI:10.1109/ICOASE.2019.8723673.
- [6] S. Arya, H. Mesariya, and V. Parekh, —Smart group action system exploitation CNN, arXiv preprint arXiv:2004.14289, 2020.
- [7] Jacob, I. J. (2019), Capsule Network-Based Biometric Recognition System, Journal of AI, 1(02), 83-94.
- [8] Face Detection and Recognition exploitation The Support Vector Machine, International Journal of Engineering and Advanced Technology (IJ EAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019.
- [9] A. Priadana and M. Habibi, —Face Detection exploitation Haar Cascades to Filter Selfie Face Image on Instagram, 2019 International Conference of AI and data Technology (ICAIT), Yogyakarta, Indonesia, 2019, pp. 6-9.
- [10] I. Masi, A. T. Tran, T. Hassner, G. Sahin, and G. Medioni, —Face-specific knowledge augmentation for free face recognition, International Journal of laptop Vision, vol. 127, no. 6-7, pp. 642–667, 2019.

A. PLAGARISM REPORT:

Submission date: 01-Mar-2021 08:42AM (UTC-0500)

Submission ID: 1521253429

File name: March1.docx (2.34M)

Word count: 2865

Character count: 14970

March1

PRIMARY SOURCES

ORIGINALITY REPORT

19%

SIMILARITY INDEX

0%

INTERNET SOURCES

19%

PUBLICATIONS

0%

STUDENT PAPERS

B.SOURCE CODE:

using

CognitiveServices.Explorer.Application.FaceApi;

using CognitiveServices.Explorer.Domain.Face;

using

MediatR;

using

System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CognitiveServices.Explorer.Application.ViewModels.FaceApi

```

{
    public class DetectViewModel : BaseFaceApiViewModel
    {
        private readonly HttpRequest _personGroupListRequest;
        private HttpRequest _detectBinaryRequest;
        private HttpRequest
            _detectUrlRequest; private
            HttpRequest _identifyRequest;

        public DetectViewModel(IMediator mediator)
            : base(mediator)
        {
            _personGroupListRequest = PersonGroupRequestGenerator.List();
            _detectBinaryRequest = FaceRequestGenerator.Detect(Array.Empty<byte>());
            _detectUrlRequest = FaceRequestGenerator.Detect(string.Empty);
            _identifyRequest = FaceRequestGenerator.Identify(string.Empty,
                Array.Empty<string>());

            UpdateRequestList();
        }

        public string? SelectedPersonGroupId { get;
            set; } public string SelectedDetectionModel {
            get; set; } =
            FaceApiConstants.DetectModelV02;

            public string SelectedRecognitionModel { get; set;
            } = FaceApiConstants.RecognitionModelV01;
    }
}

```

```

public List<PersonGroupDto>? PersonGroups { get; private
set; } public List<DetectedFaceDto>? Faces { get; private set;
}

public List<IdentityCandidate>? Candidates { get; private set; }

public List<HttpRequest> Requests { get; } = new List<HttpRequest>();


public async override Task OnInitializedAsync()
{
    await base.OnInitializedAsync().ConfigureAwait(false);

    PersonGroups = await
MakeRequest<List<PersonGroupDto>>(_personGroupListRequest);

    SelectedPersonGroupId = PersonGroups?.Select(g =>
g.PersonGroupId).FirstOrDefault();
}


public async Task ProfileChanged()
{
    // This information is no longer valid as the Face IDs are not shared
between accounts.

    Faces = null;

    Candidates = null;

    await LoadLatestConfig();
    PersonGroups = await
MakeRequest<List<PersonGroupDto>>(_personGroupListRequest);

```

```
}
```

```
public async Task Detect(byte[] data)
```

```
{
```

```
    Faces = null;
```

```
    Candidates = null;
```

```
    _detectBinaryRequest = FaceRequestGenerator.Detect(
```

```
        data,
```

```
        detectionModel: SelectedDetectionModel,
```

```
        recognitionModel: SelectedRecognitionModel);
```

```
    UpdateRequestList();
```

```
    Faces = await
```

```
    MakeRequest<List<DetectedFaceDto>>(_detectBinaryRequest).ConfigureAwait(fa  
lse
```

```
)
```

```
    ?? new List<DetectedFaceDto>(0);
```

```
}
```

```
public async Task Detect(string imageUrl)
```

```
{
```

```
    Faces = null;
```

```
    Candidates = null;
```

```
_detectUrlRequest =  
    FaceRequestGenerator.Detect( imageUrl,  
    detectionModel: SelectedDetectionModel,  
    recognitionModel: SelectedRecognitionModel);
```

```
UpdateRequestList();
```

```
Faces = await  
MakeRequest<List<DetectedFaceDto>>(_detectUrlRequest).ConfigureAwait(false)  
    ?? new List<DetectedFaceDto>(0);  
}
```

```
public async Task Identify()  
{  
    if (string.IsNullOrEmpty(SelectedPersonGroupId))  
    {  
        Error = "Person Group Id is not  
        set!"; return;  
    }  
}
```

```
Candidates = null;
```

```
        _identifyRequest =  
FaceRequestGenerator.Identify(SelectedPersonGroupId!, Faces.Select(f =>  
f.faceId));
```

```
        UpdateRequestList();
```

```
        Candidates = await  
MakeRequest<List<IdentityCandidate>>(_identifyRequest).ConfigureAwait(false)
```

```
        ?? new List<IdentityCandidate>(0);
```

```
    }
```

```
    private void UpdateRequestList()
```

```
    {
```

```
        // Both requests are changing every time we do a request and because the  
reference is changing, we need to rebuild the list.
```

```
        Requests.Clear();
```

```
        Requests.Add(_detectBinaryRequest);
```

```
        Requests.Add(_detectUrlRequest);
```

```
        Requests.Add(_identifyRequest);
```

```
        Requests.Add(_personGroupListRequest);
```

```
    }
```

```
}
```

```
}
```