INTRODUCTION

Modernization is leading to a remarkable increase number of crimes, especially robbery. In the report, the law enforcement agencies throughout the US showed an overall increase of 1.7 percent in the number of violent crimes, which are brought to their attention for the first 6 months of 2015; and, robbery has been increased by 1 percent from 311,936 cases in 2014. Therefore, Security systems have a crucial role to safeguard people. It is necessary to have a robust system which can distinguish between people and respond differently based on their privileges.

A number of methods are available for detecting and recognizing faces with various levels of complexities. Face recognition facilitates automation and security. It has already been used in many applications including ID issuance, law enforcement, border control, and many other commercial products. The state-of-art recognizers using convolutional neural networks (CNN) outperform the human's recognition rate; however, these systems are not automatically improving. Another issue with these systems is that it requires adequate data to be trained before it is actually being deployed. It is essential that the system is robust to recognize people and that the training should be accomplished without much difficulty.

And with Google Brain's second-generation system, TensorFlow is a deep learning framework released by Google. TensorFlow is flexible, portable, feasible, and completely open source. It also extensively interacts with different hardware such as smartphones, and embedded computers. With the advancement in Web Application we develop the Security based on Face Recognition System (SoF).

LITERATURE SURVEY

Home security has been an essential feature in smart home and received a growing interest in recent years. Various home security systems have been used in the market for many pre-potent companies such as ADT, Vivint, and Protect America.

However, none of them have the face-recognition feature in their systems because of moderate confidence and exhausted computational requirements. Along with the rapid development of smart devices, Netatmo presented a device using the deep neural network to realize the face, but their system is still far from expectation. This smart camera does not keep up with the competition as a webcam or a security cam. It is slow at everything; the live feed is lagged; the notifications are delayed, and it takes a while to learn faces.

In research problems, there are several security systems using face recognition technology. Facial recognition system for security access and identification presented by Jeffrey S. Coffin uses custom VLSI Hardware and Eigenspaces method, and security systems presented by Shankar Kartik also uses Eigenfaces method for face identification which gives unfavorable results with moderate accuracy. We use a deep learning algorithm for face recognition problems which is closing the gap to human-level performance in face verification.

The robustness of face recognition systems depend on the changes in conditions of light or expression or even in the partial blocked of the face can be considered. Several papers have proposed various techniques for face recognition under those conditions. Eigenfaces are variant extracted feature to above factors. Facenet using deep convolutional neural network with the architecture of Inception model from Google and uses a novel online triplet mining method to train instead of an intermediate bottleneck layer. On the widely used Labeled Faces in the Wild (LFW) dataset, Facenet system achieved a new record accuracy of 99.63%. However, unfortunately, not only the size of the database increases but also its computational cost increases and recognition accuracy declines accordingly. That is why incremental learning is a learning algorithm which approaches to handle large-scale training data to efficiency and accuracy. A brief definition of incremental learning is that learning is a gradual process with new data. The idea is the existed classifiers that are identified with the new classes to be learned. Its key idea is to begin learning on low-resolution images and then gradually increase to high-resolution image. We use 96x96 pixels cropped as the input data which is also mentioned in Facenet for training data size.

Face detection algorithms have been addressed in many papers such as Haar Cascades, Kalman Filter and applied in various fields. Besides, OpenCV is a robust open source which supports many methods to detect faces. However, in this paper, we use the Dlib C++ library which supports machine learning algorithms and uses histogram-of-oriented-gradient algorithm. Face detection is not only necessary for the camera node detecting the face but also helpful in pre-processing the input data. In this paper, we also describe a novel method to collect the data from social media by using Facebook API and ask from human interaction to label the unidentified peoples, that directly incremental learn for the neural network model with new data. The interface is also designed with easy uses in many types of devices.

2.1 Use Cases of Face Detection System

The system can be used in the following places to identify people known and unknown :

- Offices – Manufacturers, retail, other SMEs and corporate giants
- Colleges / Schools
- Hospitals/healthcare organizations
- Airports and railway stations
- Sports venues
- Entertainment and hospitality industry
- Densely populated areas

TOOLS/ENVIRONMENT USED

3.1 HARDWARE REQUIREMENTS:

Processor	:	Pentium IV 2.4 GHz.
Hard Disk	:	250 GB.
Monitor	:	15 VGA Color.
RAM	:	1 GB
Mouse	:	Optical
Keyboard	:	Multimedia

(These are Minimum Configuration)

3.2 SOFTWARE REQUIREMENTS:

Operating system	:	Windows XP Professional / Windows7 or More
Coding Language	:	Python
IDE	:	Pycharm, Python 3.7.4/3.9.4

3.3 Pycharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

Choose the best PyCharm for you

PyCharm is available in three editions:

- *Community* (free and open-sourced): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.
- *Professional* (paid) : for professional Python, web, and data science development, including code assistance, refactorings, visual debugging, version control integration, remote configurations, deployment, support for popular web frameworks, such as Django and Flask, database support, scientific tools (including Jupyter notebook support), big data tools.
- *Edu* (free and open-sourced): for learning programming languages and related technologies with integrated educational tools.

Supported languages

To start developing in Python with PyCharm you need to download and install Python from python.org depending on your platform.

PyCharm supports the following versions of Python:

- **Python 2:** version 2.7
- **Python 3:** from the version 3.6 up to the version 3.10

PyCharm is a cross-platform IDE that works on Windows, macOS, and Linux. Check the system requirements:

Requirement	**Minimum**	**Recommended**
RAM	4 GB of free RAM	8 GB of total system RAM
CPU	Any modern CPU	Multi-core CPU. PyCharm supports multithreading for different operations and processes making it faster the more CPU cores it can use.
Disk space	2.5 GB and another 1 GB for caches	SSD drive with at least 5 GB of free space
Monitor resolution	1024x768	1920×1080
Operating system	Officially released 64-bit versions of the following: Microsoft Windows 8 or later macOS 10.13 or later. Any Linux distribution that supports Gnome, KDE, or Unity DE. PyCharm is not available for some Linux distributions, such as RHEL6 or CentOS6, that do not include GLIBC 2.14 or later. Pre-release versions are not supported.	Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL)

You can install PyCharm using Toolbox or standalone installations. If you need assistance installing PyCharm, see the installation instructions: Install PyCharm

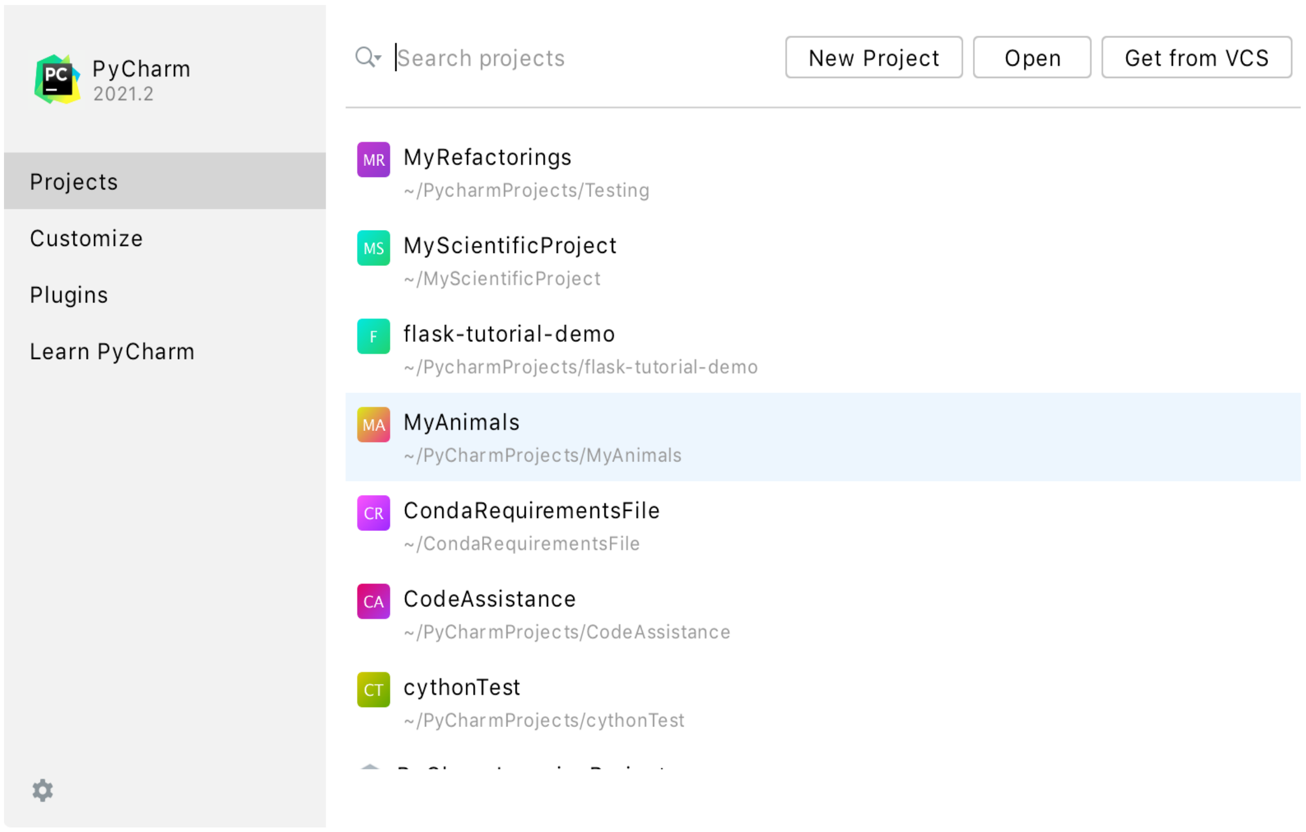
Start with a project in PyCharm

Everything you do in PyCharm, you do within the context of a project. It serves as a basis for coding assistance, bulk refactoring, coding style consistency, and so on. You have three options to start working on a project inside the IDE:

- Open an existing project
- Check out a project from version control
- Create a new project

Open an existing project

Begin by opening one of your existing projects stored on your computer. You can select one in the list of the recent projects on the Welcome screen or click **Open**:

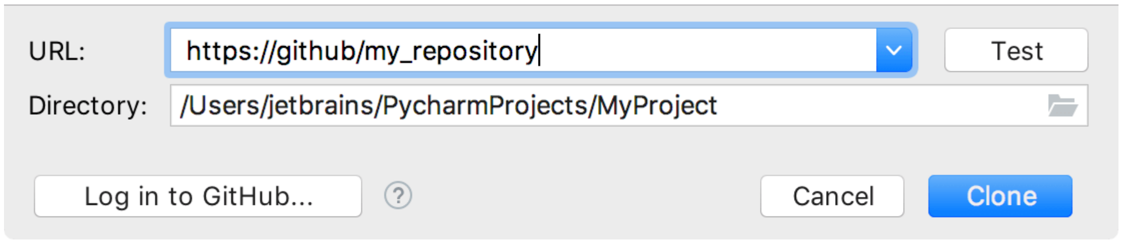


Otherwise, you can create a project for your existing source files. Select the command **Open** on the **File** menu, and specify the directory where the sources exist. PyCharm will then create a project from your sources for you. Refer to the section Importing Project from Existing Source Code for details

Check out an existing project from Version Control

You can also download sources from a VCS storage or repository. Choose Git (GitHub), Mercurial, Subversion, Preforce (supported in Professional edition only), and then enter your credentials to access the storage.

Then, enter a path to the sources and clone the repository to the local host:


URL: https://github/my_repository

Directory: /Users/jetbrains/PycharmProjects/MyProject

Create a new project

To create a project, do one of the following:

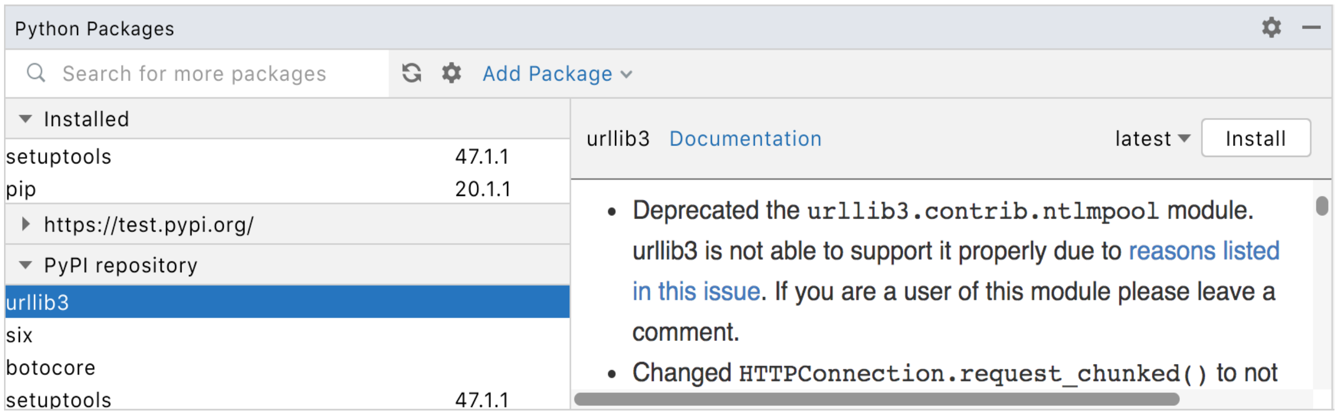
- From the main menu, choose **File | New Project**
- On the Welcome screen, click **New Project**

In PyCharm Community, you can create only Python projects, whereas, with PyCharm Professional, you have a variety of options to create a web framework project.
Community and Edu Professional

See more details in Create a Python project.

When creating a new project, you need to specify a Python interpreter to execute Python code in your

project. You need at least one Python installation to be available on your machine.


Python Packages

Search for more packages | Add Package

Installed	
setuptools	47.1.1
pip	20.1.1
https://test.pypi.org/	
PyPI repository	
urllib3	
six	
botocore	
setuptools	47.1.1

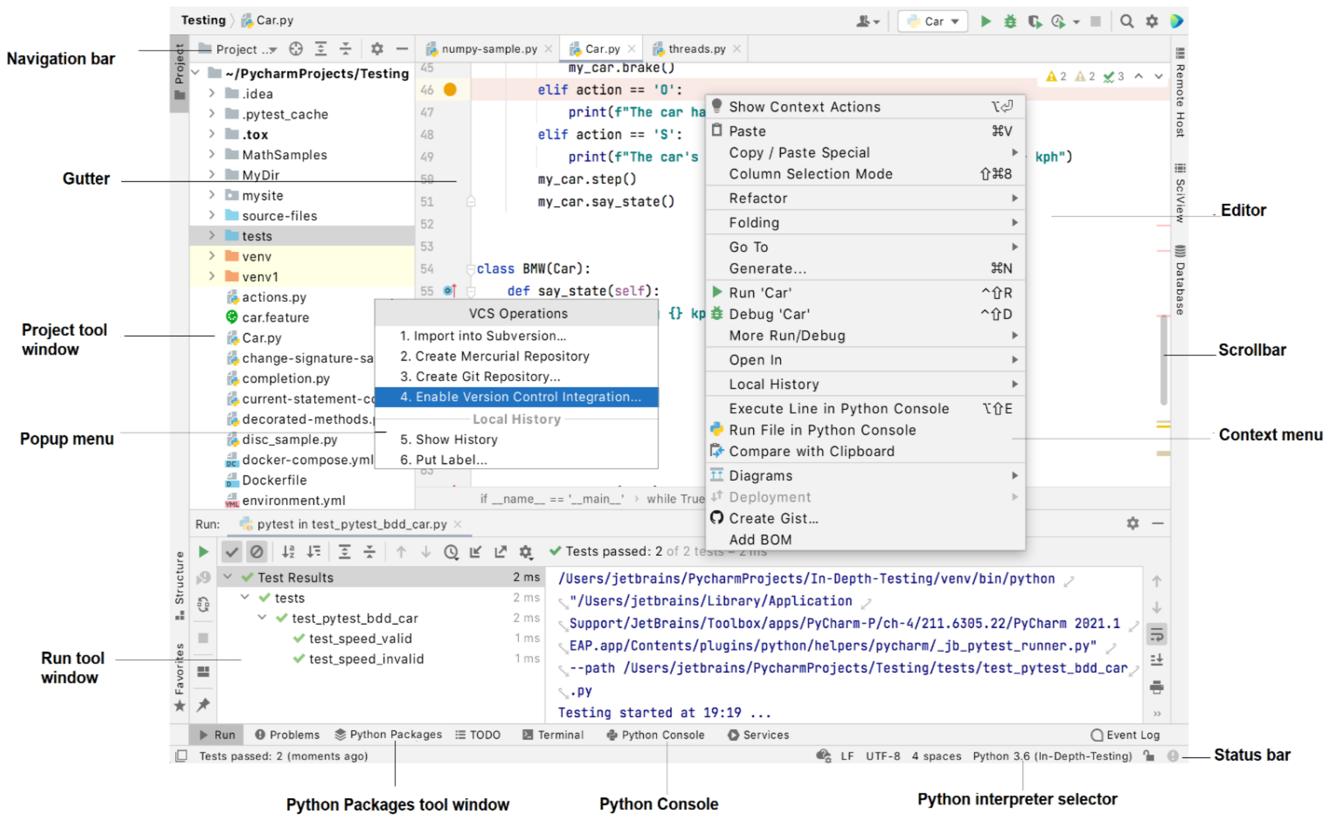
urllib3 Documentation latest Install

- Deprecated the `urllib3.contrib.ntlmpool` module. urllib3 is not able to support it properly due to reasons listed in this issue. If you are a user of this module please leave a comment.
- Changed `HTTPConnection.request_chunked()` to not

For more details see Configure a Python interpreter.

When you launch PyCharm for the very first time, or when there are no open projects, you see the Welcome screen. It gives you the main entry points into the IDE: creating or opening a project, checking out a project from version control, viewing documentation, and configuring the IDE.

When a project is opened, you see the main window divided into several logical areas. Let's take a moment to see the key UI elements here:



```
        my_car.brake()
    elif action == 'O':
        print(f"The car ha
    elif action == 'S':
        print(f"The car's
    my_car.step()
    my_car.say_state()

class BMW(Car):
    def say_state(self):
```

```
/Users/jetbrains/PycharmProjects/In-Depth-Testing/venv/bin/python
"/Users/jetbrains/Library/Application
Support/JetBrains/Toolbox/apps/PyCharm-P/ch-4/211.6305.22/PyCharm 2021.1
EAP.app/Contents/plugins/python/helpers/pycharm/_jb_pytest_runner.py"
--path /Users/jetbrains/PycharmProjects/Testing/tests/test_pytest_bdd_car
.py
Testing started at 19:19 ...
```

Tests passed: 2 of 2 tests

Test Results 2 ms
- tests 2 ms
  - test_pytest_bdd_car 2 ms
    - test_speed_valid 1 ms
    - test_speed_invalid 1 ms

Tests passed: 2 (moments ago) — LF UTF-8 4 spaces Python 3.6 (In-Depth-Testing)

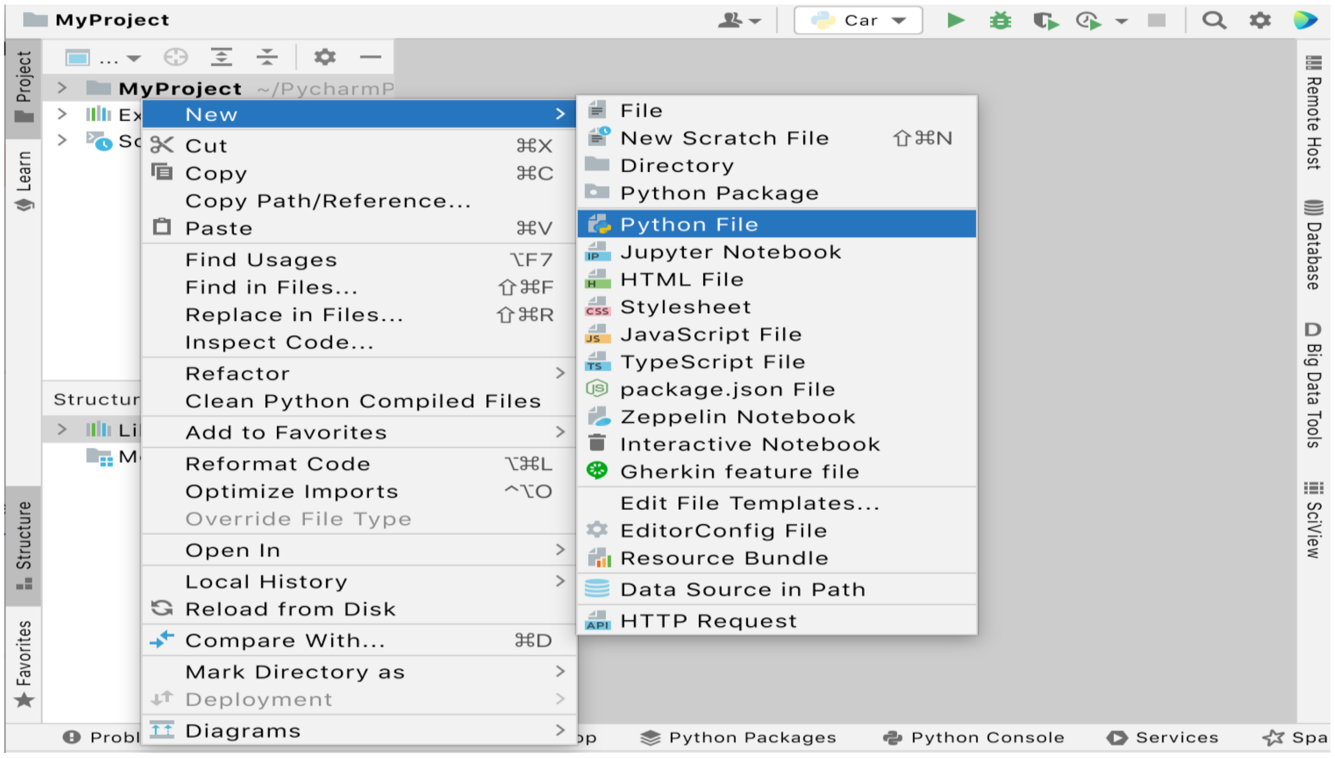
1. Project tool window on the left side displays your project files.
2. Editor on the right side, where you actually write your code. It has tabs for easy navigation between open files.
3. Navigation bar above the editor additionally allows you to quickly run and debug your application as well as do the basic VCS actions.
4. **Gutter**, the vertical stripe next to the editor, shows the breakpoints you have, and provides a convenient way to navigate through the code hierarchy like going to definition/declaration. It also shows line numbers and per-line VCS history.
5. **Scrollbar**, on the right side of the editor. PyCharm constantly monitors the quality of your code and always shows the results of its code inspections in the gutter: errors, warnings, and so on. The indicator in the top right-hand corner shows the overall status of code inspections for the entire file.
6. Tool windows are specialized windows attached to the bottom and sides of the workspace and provide access to typical tasks such as project management, source code search and navigation, integration with version control systems, and so on.
7. The status bar indicates the status of your project and the entire IDE, and shows various warnings and information messages like file encoding, line separator, inspection profile, and so on. It also provides quick access to the Python interpreter settings.

Also, in the bottom-left corner of the PyCharm window, in the Status bar, you see the button or This button toggles the showing of the tool window bars. If you hover your mouse pointer over this button, the list of the currently available tool windows show up.

When you have created a new project or opened an existing one, it is time to start coding.

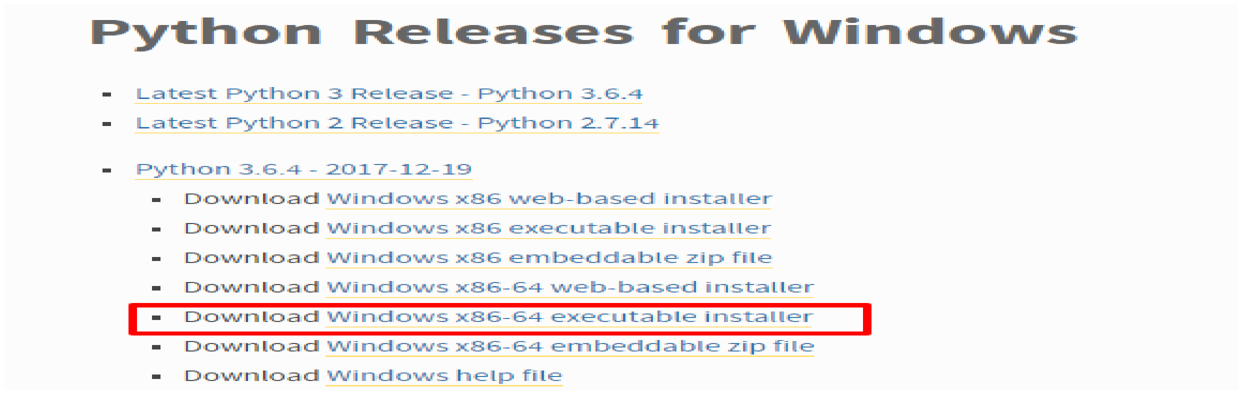
Create a Python file

In the **Project** tool window, select the *project root* (typically, it is the root node in the project tree), right-click it, and select **File | New**



3.4 Python installation

Steps 1 – Download the latest Python 3.x version. At the time of writing this article latest version was Python 3.7.4. Download Windows x86 – 64 executable file only as installer will automatically install 3or 64 bit of Python according to the system configuration.

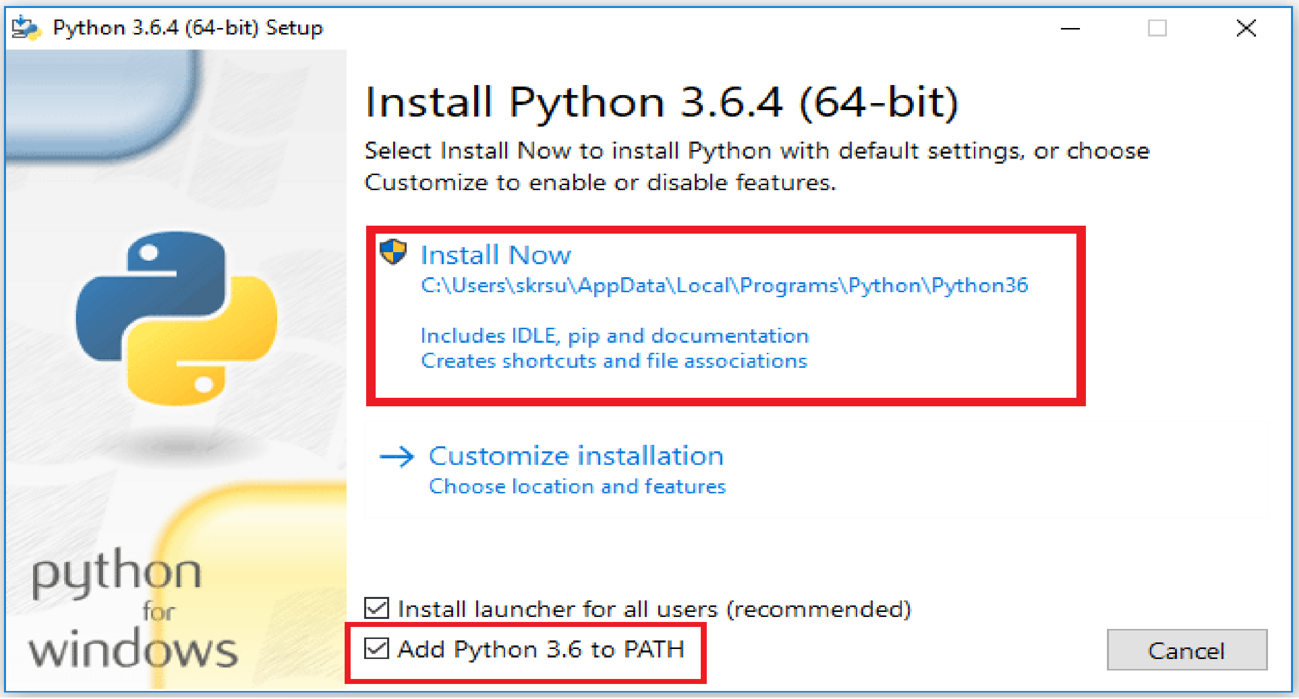


Python Releases for Windows

- Latest Python 3 Release - Python 3.6.4
- Latest Python 2 Release - Python 2.7.14
- Python 3.6.4 - 2017-12-19
  - Download Windows x86 web-based installer
  - Download Windows x86 executable installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86-64 web-based installer
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 embeddable zip file
  - Download Windows help file

Python Releases for Windows

Step 2 – Open the executable file and Check the Add Python 3.6 to PATH. Then click the Install Now button. It will show the installation progress.



Python 3.6.4 (64-bit) Setup

Install Python 3.6.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

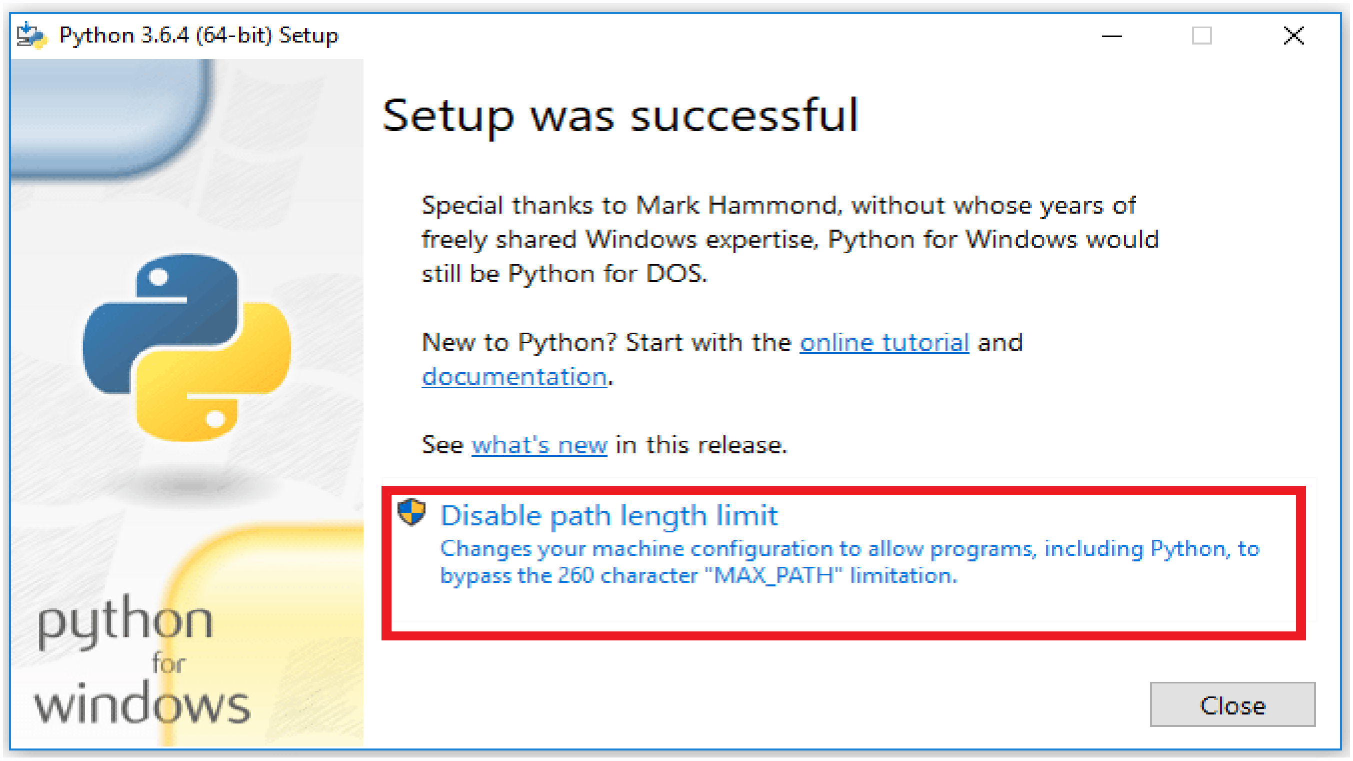
Install Now
C:\Users\skrsu\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

Step 3 – When the installation progress is completed, you will see the Disable path length limit. Now you must be thinking what is it and what will happen if I will disable it. The answer is clear; it will remove the limitations on MAX_PATH variable. **It will allow using long path names for the Python.** We recommend you to not disable this option as it will remove any path related issues while working in Windows. Therefore click on the close button to finish the installation.



Setup was successful

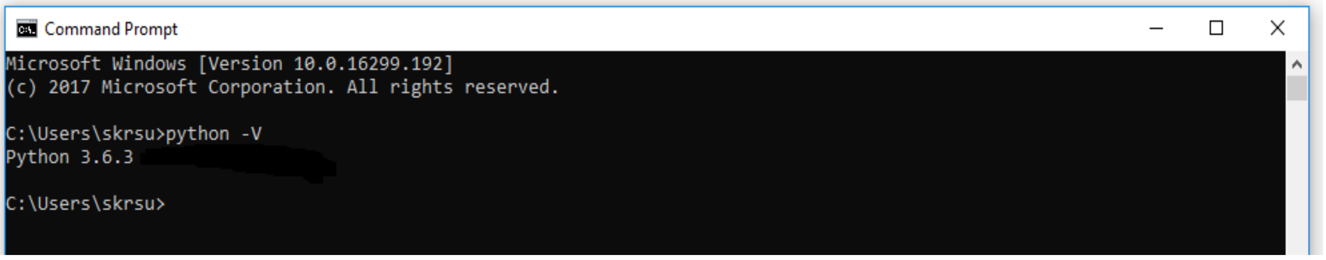
Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Step 4 – Now, the Python 3.6 is installed. You can check it either it is properly installed or not. You can do it through Command Prompt. Open the command prompt and type the following command. It will output the version of the Python.



```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\skrsu>python -V
Python 3.6.3 [redacted]

C:\Users\skrsu>
```

Application of theories to real life situation

Python is the most versatile language in the programming world and it has applications in various domains. It helps us in taking care of our current programming task as well as lets us focus on the core functionality of Python programming languages.

Some of the best application that is developed in the python are instagram, spotify, google Netflix and so on.

We can Choose Python Development for real time instance for the following-

- GUI based desktop applications
- Image processing and graphic design applications
- Scientific and computational applications
- Games
- Web applications
- Enterprise
- Business applications
- Operating system
- Language development
- Prototyping

Machine learning is an application of python which involves computer to get trained using a given data set, and uses this training to predict the properties of a given new data. For example, we can train computer by feeding it 1000 images of cats and 1000 more images which are not of a cat, and tell each time to computer whether a picture is cat or not. Then if we show the computer a new image, then from the above training, computer should be able to tell whether this new image is cat or not.

There are various machine learning algorithms like Decision trees, Naive Bayes, Random forest, Support vector machine, K-nearest neighbor, K-means clustering, etc.

In this post, we will provide an overview of the common functionalities of NumPy and Pandas. We will realize the similarity of these libraries with existing toolboxes in R. This similarity and added flexibility have resulted in wide acceptance of python in the scientific community lately.

ANALYSIS DOCUMENT

4.1 AN OVERVIEW OF OUR SYSTEM

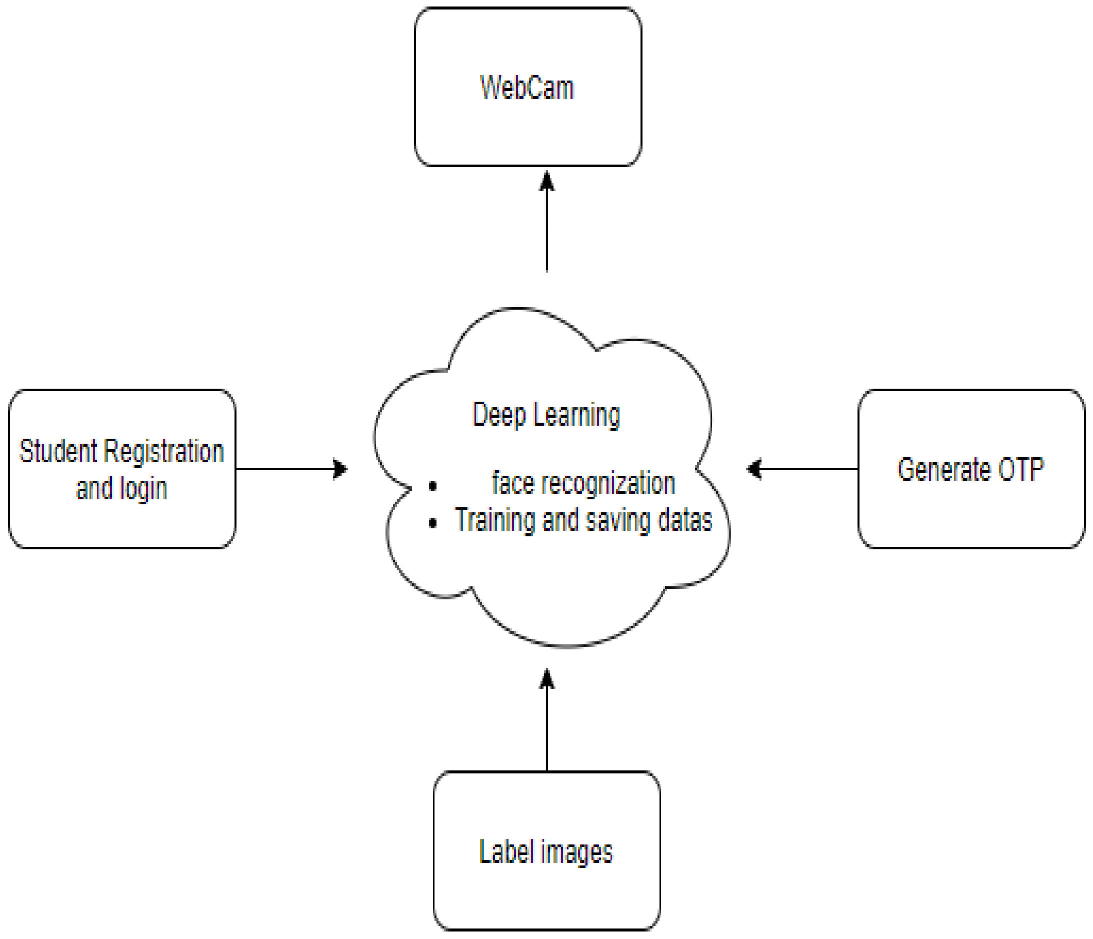
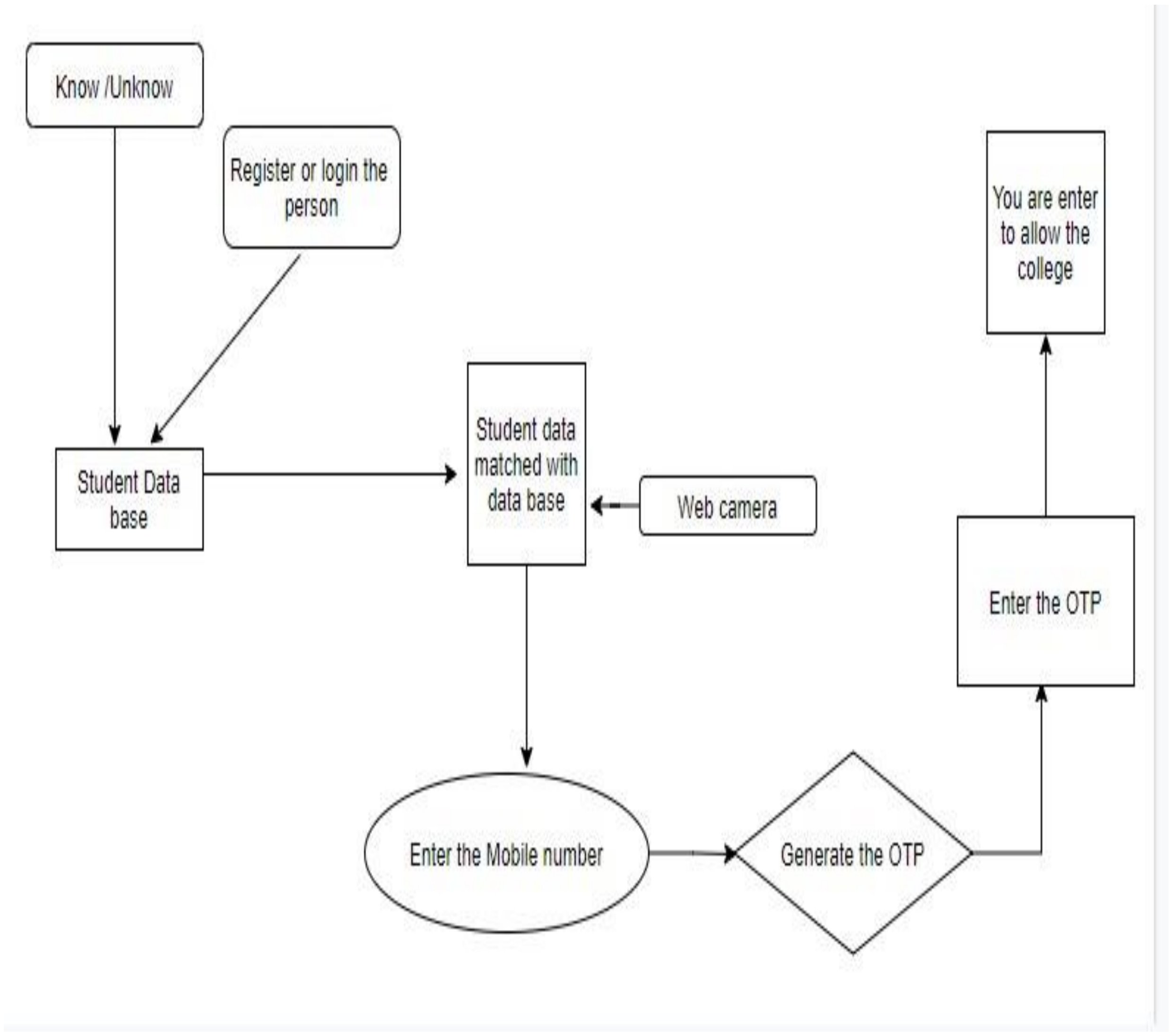
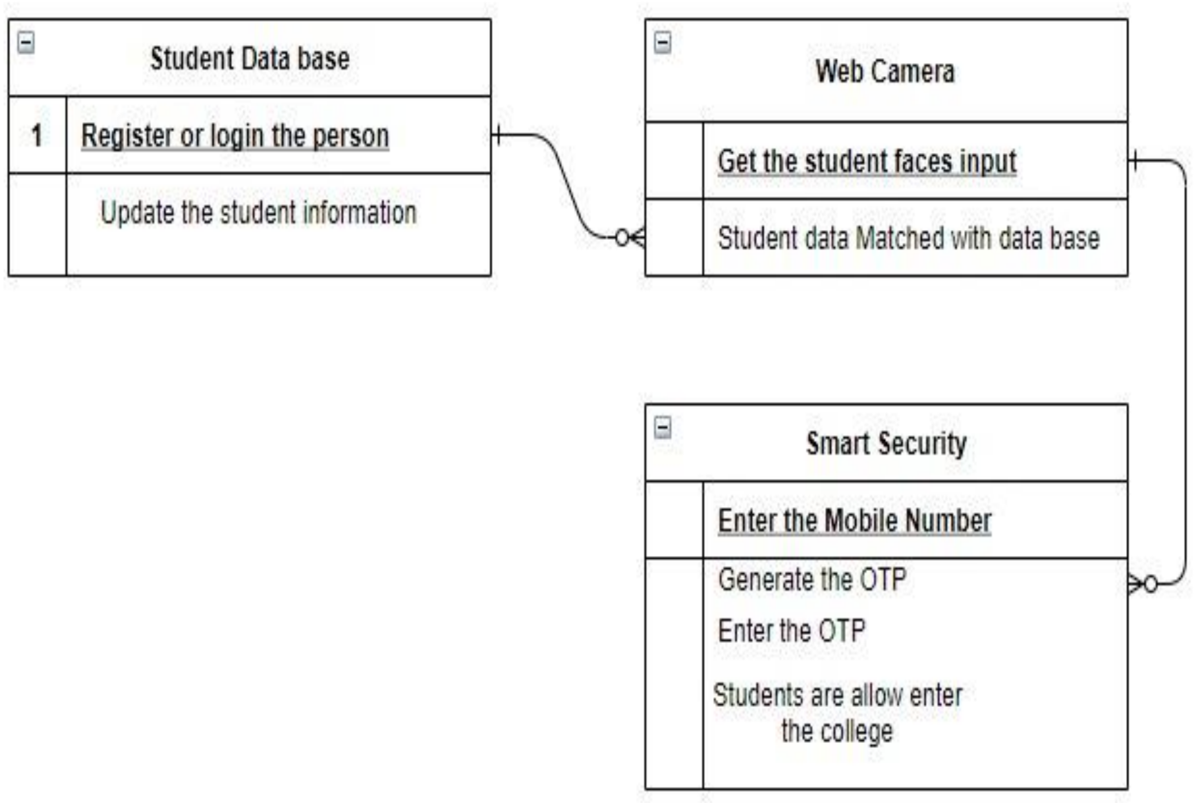


Figure: Smart security System Architecture.

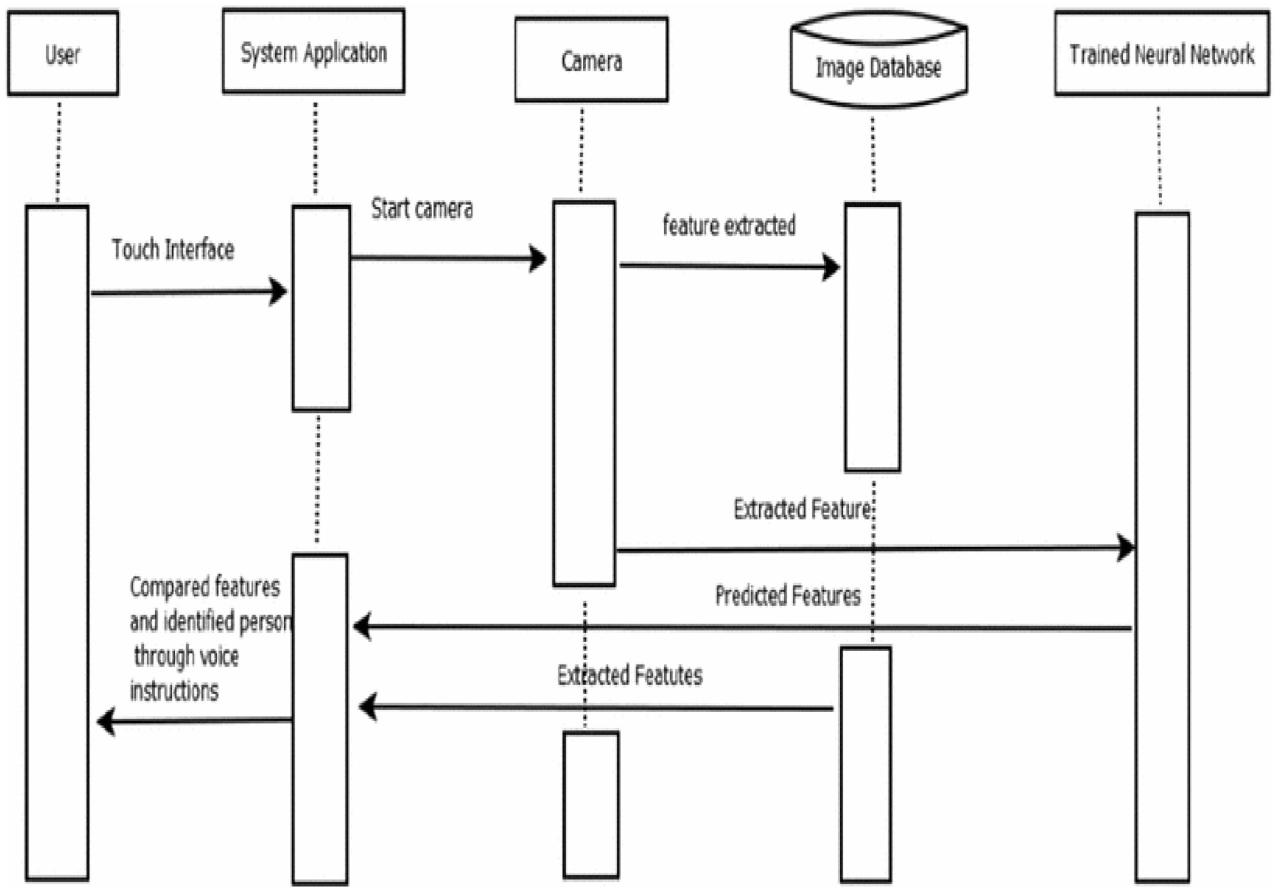
4.2 ER DIAGRAM


Know /Unknow
Register or login the person
Student Data base
Student data matched with data base
Web camera
You are enter to allow the college
Enter the OTP
Enter the Mobile number
Generate the OTP

4.3 DATA FLOW DIAGRAM

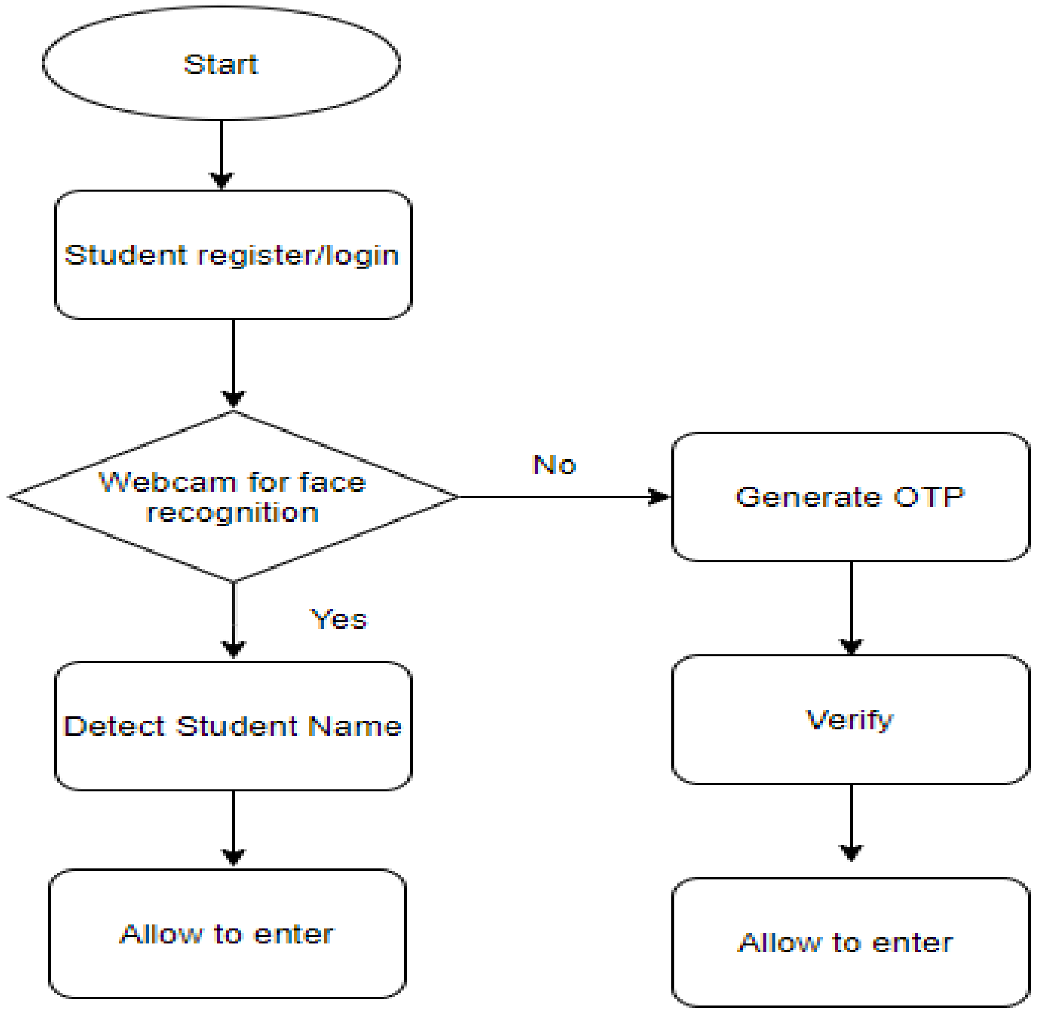

Student Data base
1
Register or login the person
Update the student information
Web Camera
Get the student faces input
Student data Matched with data base
Smart Security
Enter the Mobile Number
Generate the OTP
Enter the OTP
Students are allow enter the college

4.4 SEQUENCE DIAGRAM

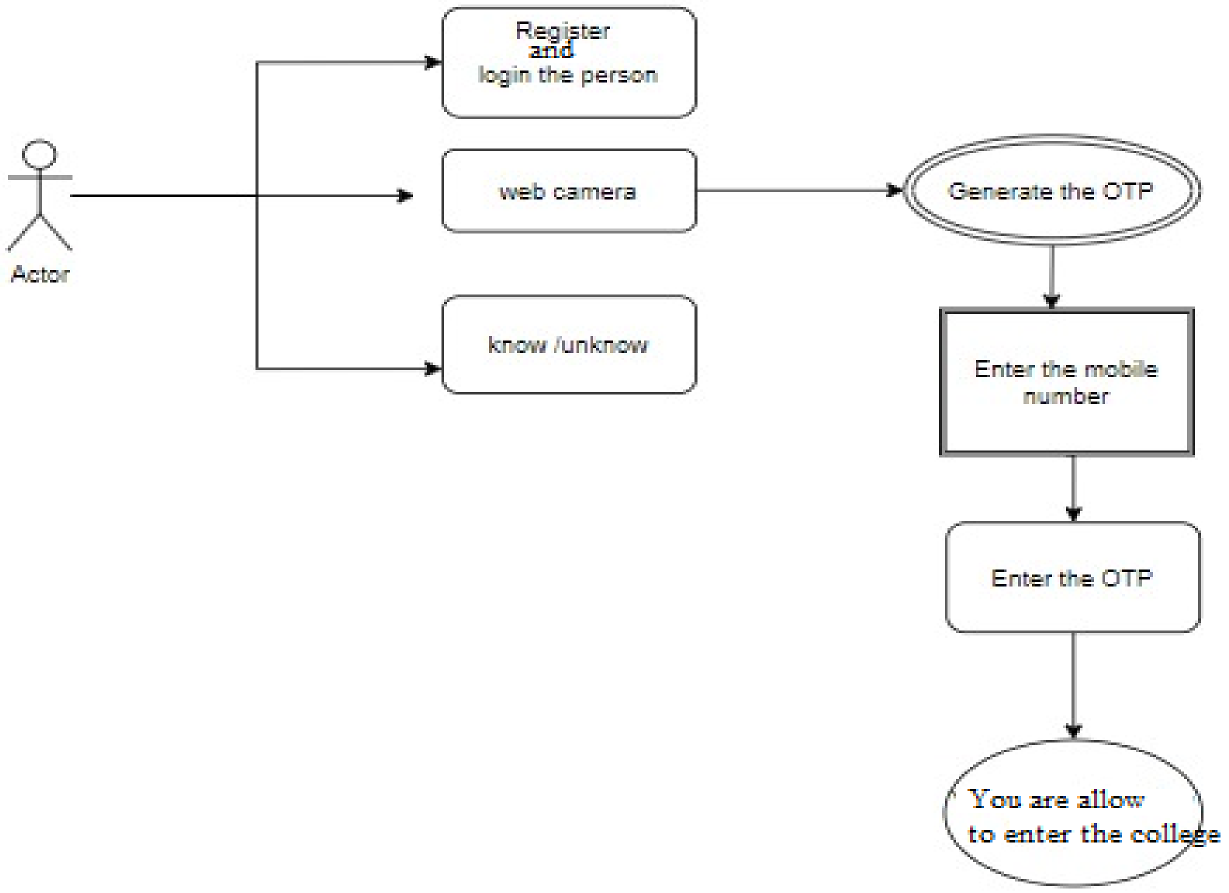


METHODOLOGY

We propose this paper with twin objective of creating a Binary face classifier which can detect faces in any orientation irrespective of alignment and train it in an appropriate neural network to get accurate results. The model requires inputting an RGB image of any arbitary size to the model. The model's basic function is feature extraction and class prediction. The output of the model is a feature vector which is optimized using Gradient descent and the loss function used is Binomial Cross Entropy. represents the end to end pipeline of our method along with sample demonstration of obtained output at each step.



5.1 USECASE DIAGRAM



5.2 FACE RECOGNITION ARCHITECTIRE

Based on the architecture and published model from the Openface and Inception-v3 model citeabadi2016tensorflow. We trained a new model with a new database set. The face recognition architecture is shown in Figure. The input data is aligned by using a face detection method and then goes through the deep convolutional neural network to extract an embedding feature. We can use the feature for similarity detection and classification.

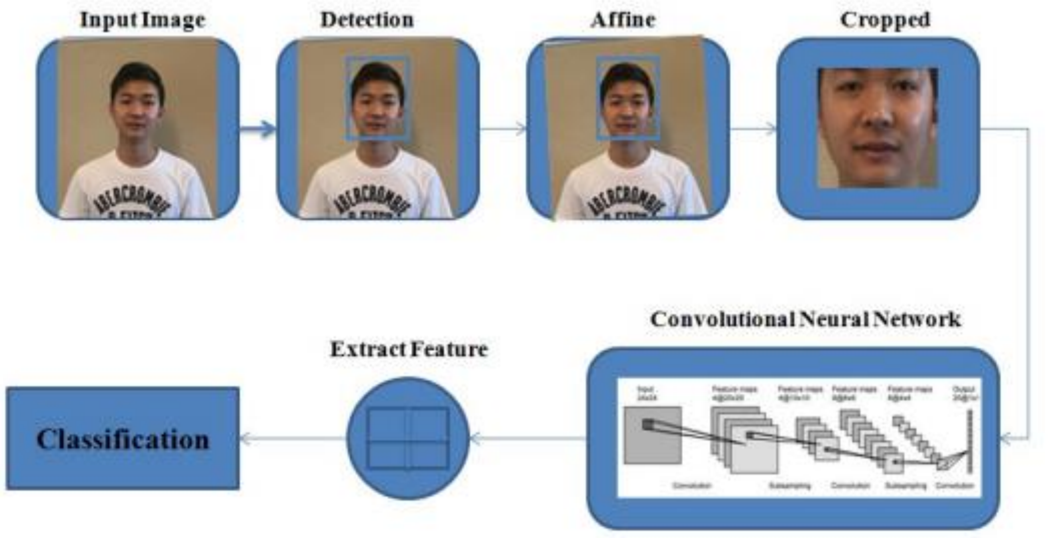


Figure: Face Recognization Architecture.

When processing an image, face detection (Dlib library) first find a square around faces. Each face is then passed separately into the neural network, which expects a fixed sized input, currently 96x96 pixels as mentioned in Facenet, which is the best size giving the highest accuracy and low training time. We reshape the face in the square to 96x96 pixels. A potential issue with this is that faces could be looking in different directions and we have to rotate the images. We use align faces method described in OpenFace by first finding the locations of the eyes and nose with Dlib's landmark detector, and if the face is undetected or unaligned which will be eliminated before going to the neural network. Finally, an affine transformation is performed on the cropped image to make the eyes and nose appear at about the same place.

5.3 DEEP LEARNING

Deep Learning is a class of machine learning which performs much better on unstructured data. Deep learning techniques are outperforming current machine learning techniques. It enables computational models to learn features progressively from data at multiple levels. The popularity of deep learning amplified as the amount of data available increased as well as the advancement of hardware that provides powerful computers. This article comprises of the evolution of deep learning, various approaches to deep learning, architectures of deep learning, methods, and applications.

Deep learning techniques which implement deep neural networks became popular due to the increase of high-performance computing facility. Deep learning achieves higher power and flexibility due to its ability to process a large number of features when it deals with unstructured data. Deep learning algorithm passes the data through several layers; each layer is capable of extracting features progressively and passes it to the next layer. Initial layers extract low-level features, and succeeding layers combines features to form a complete representation. Section 2 gives an overview of the evolution of deep learning models. Section 3 provides a brief idea about the different learning approaches, such as supervised learning, unsupervised learning, and hybrid learning. Supervised learning uses labeled data to train the neural network. In supervised learning, the network uses unlabeled data and learns the recurring patterns. Hybrid learning combines supervised and unsupervised methods to get a better result. Deep learning can be implemented using different architectures such as architectures like Unsupervised Pre-trained Networks, Convolutional Neural Networks, Recurrent Neural Networks, and Recursive Neural Networks, which are described in section 4. Section 5 introduces various training methods and optimization techniques that help in achieving better results. Section 6 describes the frameworks which allow us to develop tools that offer a better programming environment. Despite the various challenges in deep learning applications, many exciting applications that may rule the world.

Deep Learning Approaches Deep neural networks are successful in Supervised learning, Unsupervised learning, Reinforcement learning, as well as hybrid learning.

5.3.1 Supervised Learning

In supervised learning, the input variables represented as X are mapped to output variables represented as Y by using an algorithm to learn the mapping function f.

$$Y = f(X)$$

The aim of the learning algorithm is to approximate the mapping function to predict the output (Y) for a new input (X). The error from the predictions made during training can be used to correct the output. Learning can be stopped when all the inputs are trained to get the targeted output . Regression for solving regression problems, Support Vector machines used for classification , Random forest for classification as well as regression problems.

5.3.2 Unsupervised Learning

In unsupervised learning, we have the input data only and no corresponding output to map. This learning aims to learn about data by modeling the distribution in data. Algorithms can be able to discover the exciting structure present in the data. Clustering problems and association problems use Unsupervised learning. The unsupervised learning algorithms such as K-means algorithm is used in clustering problems, Apriori algorithm is used in association problems

5.3.3 Reinforcement Learning

Reinforcement learning uses a system of reward and punishment to train the algorithm. In this, the algorithm or an agent learns from its environment. The agent gets rewards for correct performance and penalty for incorrect performance. For example, consider the case of a self-driving car, the agent gets a reward for driving safely to destination and penalty for going off-road. Similarly, in the case of a program for playing chess, the reward state may be winning the game and the penalty for being checkmated. The agent tries to maximize the reward and minimize the penalty. In reinforcement learning, the algorithm is not told how to perform the learning; however, it works through the problem on its own.

5.3.4 Hybrid Learning

Hybrid learning refers to architectures that make use of generative (unsupervised) as well as discriminative (supervised) components. The combination of different architectures can be used to design a hybrid deep neural network. They are used for action recognition of humans using action bank features and are expected to produce much better results.

5.4 Applications of Deep Learning

Deep learning networks can be used in a variety of applications such as self driving cars, Natural Language Processing, Google's Virtual Assistant, Visual Recognition, Fraud detection, healthcare, detecting developmental delay in children, adding sound to silent movies, automatic machine translation, text to image translation, image to image synthesis, automatic image recognition, Image colorization, earthquake prediction, market-rate forecasting, news aggregation and fraud news detection.

ALGORITHM

6.1 Convolutional Neural Networks:

Convolutional Neural Networks have a distinct design than regular Neural Networks. Regular Neural Networks usually transform an input by sending it through a series of hidden neural network layers. Each layer is created of a collection of neurons, where every layer is fully connected to all neurons within the layer before. Finally, there is a last fully-connected layer — the output layer — that represents the predictions. Convolutional Neural Networks are a bit different. First of all, the layers are organized into 3 dimensions: width, height, and depth. Further, the neurons in one layer do not connect to all the neurons within the next layer however solely to a tiny region of it. Lastly, the final output is going to be reduced to one vector of likelihood scores, organized along the depth dimension. CNN is composed of two major parts:

• Feature Extraction: In this part, the Neural Network performs a series of convolution operations and pooling operations throughout which the features are detected. If you would have a picture of a zebra, this is the part where the network would acknowledge its stripes, two ears, and 4 legs.

• Classification: Here, the fully connected layers function as a classifier on top of these extracted features. They're going to assign likelihood for the object on the image being what the algorithm predicts it's.

• There are squares and lines inside the red dotted region which we will break down later. The green round circles inside the blue dotted region on the right side of the image are named classification. This neural network or multi-layer perceptron area acts as a classifier. The inputs to this layer (feature extraction) came from the preceding part named feature extraction.

• Feature extraction is the part of CNN design from wherever this network derives its name. Convolution is the mathematical operation that is central to the effectiveness of this algorithmic program. Let's perceive on a high level what happens within the red self enclosed region. The input to the red region is that the image that we would like to classify and therefore the output may be a set of features. Think about features as an example of the image, as an example, a picture of a cat may need features like whiskers, two ears, four legs and etc. A handwritten digit image may need features like horizontal and vertical lines or loops and curves. Later I will show you how to extract such features from the image.

CNN Algorithm steps

• Step 1: Convolution Operation The name “Convolutional neural network” indicates that the network employs a mathematical operation called Convolution. Convolution is a specialized kind of linear operation. Convnets are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers

• Step 1(b): ReLU Layer The second part of this step will involve the Rectified Linear Unit or ReLU. In this step we apply the rectifier function to increase non-linearity in the CNN. Images are made of different objects that are not linear to each other. Without applying this function the image classification will be treated as a linear problem while it is actually a non-linear one.

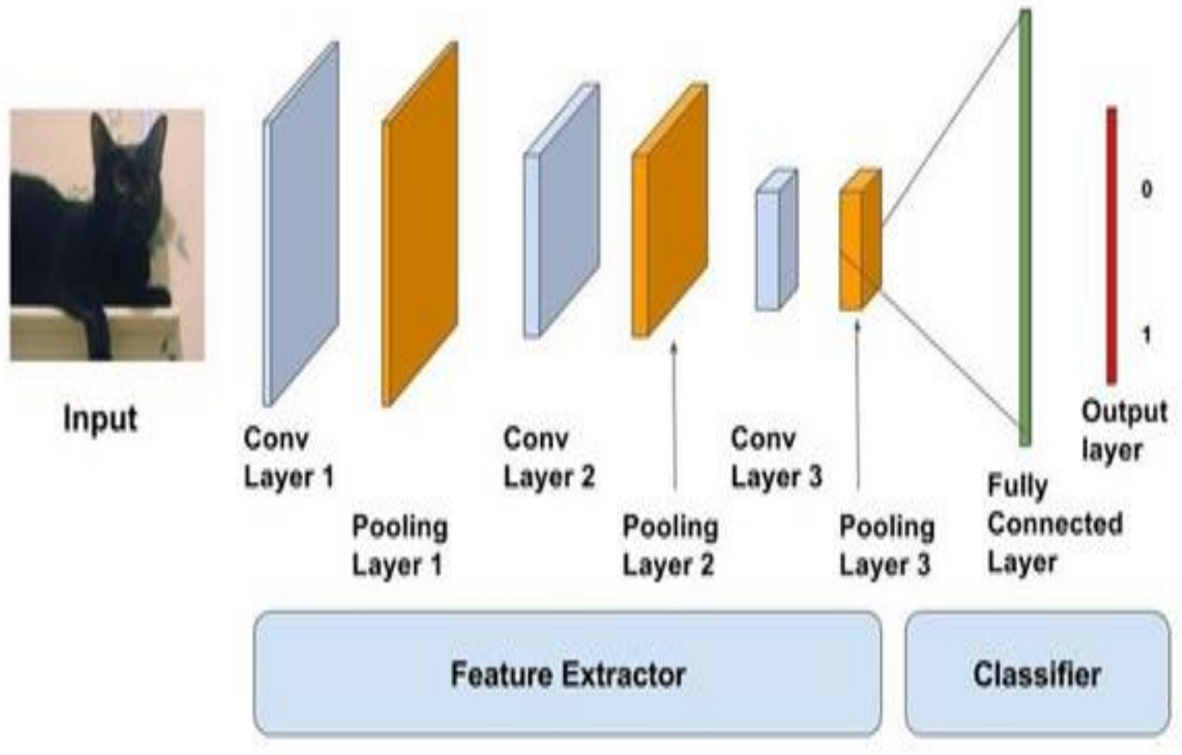
• Step 2: Pooling

The pooling layer is also called the down sampling layer as this is responsible for reducing the size of activation maps. A filter and stride of the same length are applied to the input volume. Less significant data is ignored by this layer hence image recognition is done in a smaller representation. This layer reduces over fitting. Since the amount of parameters is reduced using the pooling layer, the cost is also reduced. The input is divided into rectangular pooling regions and either maximum or average is calculated, which returns maximum or average consequently. Max Pooling is a popular one.

• Step 3: Flattening Once the pooled featured map is obtained, the next step is to flatten it. Flattening involves transforming the entire pooled feature map matrix into a single column which is then fed to the neural network for processing.

• Step 4: Full Connection After flattening, the flattened feature map is passed through a neural network. This step is made up of the input layer, the fully connected layer, and the output layer. The fully connected layer is similar to the hidden layer in ANNs but in this case it’s fully connected. The output layer is where we get the predicted classes. The information is passed through the network and the error of prediction is calculated. The error is then back propagated through the system to improve the prediction. The final figures produced by the neural network don’t usually add up to one.

However, it is important that these figures are brought down to numbers between zero and one, which represent the probability of each class. This is the role of the Soft max function.



FEASIBILITY SYSTEM

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- OPERATIONAL FEASIBILITY

7.1 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

7.2 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

7.3 OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

PROPOSED SYSTEM

Proposed Methodology:

We propose a two-stage architecture for detecting Known and Unknown faces and localizing them. represents our proposed system architecture(input image taken from the dataset by. It consists of two major stages. The first stage of our architecture includes a Face Detector, which localizes multiple faces in images of varying sizes and detects faces even in overlapping scenarios. The detected faces (regions of interest) extracted from this stage are then batched together and passed to the second stage of our architecture, which is a CNN based Known Face Classifier. The results from the second stage are decoded and the final output is the image with all the faces in the image correctly detected and classified as either Known or Unknown faces.

We are using a camera for capturing live video. The video is then processed frame-by-frame. By using image processing people in the video are identified if there is no minimum distance between the people then the device gives an alert by speaking.

Advantages of Proposed System:

- Memory.
- User Friendly: The proposed system is user friendly because the retrieval and storing of data is fast and data is maintained efficiently. More over the graphical user interface is provided in the proposed system, which provides user to deal with the system very easily.

- Very Less Paperwork: The proposed system requires very less paper work. All the records of each field is fetched into the computer immediately and reports can be generated through computers. More over hardware copy will reduced.

- Computer operator control: Computer operator control will be there so there's no chance of errors. More over storing and retrieving of information is easy. So work can be done speedily and in time.

EXISTING SYSTEM:

Traditional object detection uses a multi-step process. A well-known detector is the Viola-Joins detector, which is able to achieve real-time detection. The algorithm extracts feature by Haar feature descriptor with an integral image method, selects useful features, and detects objects through a cascaded detector. Although it utilizes integral image to facilitate the algorithm, it is still very computationally expensive. In for human detection, an effective feature extractor called HOG is proposed, which computes the directions and magnitudes of oriented gradients over image cells. Later on, deformable part-based model (DPM) detects objects parts and then connects them to judge classes that objects belong to. Rather than using handcrafted features, deep learning based detector demonstrated excellent performance recently, due to its robustness and high feature extraction capability. There are two popular categories, one-stage object detectors and two-stage object detectors. Two-stage detector generates region proposals in the first stage and then fine-tune these proposals in the second stage. The two-stage detector can provide high detection performance but with low speed. The seminal work R-CNN is proposed by R. Girshick et al. R-CNN uses selective search to propose some candidate regions which may contain objects. After that, the proposals are fed into a CNN model to extract features, and a support vector machine (SVM) is used to recognize classes of objects. However, the second-stage of R-CNN is computationally expensive, since the network has to detect proposals on a one-by-one manner and uses a separate SVM for final classification.

DRAW BACKS OF EXISTING SYSTEM

Due to the limited size of the known face dataset, it is difficult for learning algorithms to learn better features. As deep learning based methods often require larger dataset, transfer learning is proposed to transfer learned knowledge from a source task to a related target task. According to, transfer learning has helped with the learning in a significant way as long as it has a close relationship. In our work, we use parts of the network pretrained on a large scale face detection dataset - Wider Face, which consists of 32,203 images and 393,703 annotated faces. In Retina Face, only the parameters of the backbone and neck are transferred from Wider Face, the heads are initialized by Kaiming’s method. In addition, pretrained Imagenet weights are considered as a standard initialization of our backbones in basic cases.

Post processing on the predicted known person obtained is performed so that the irregularities in the region can be filled and to remove the unwanted errors (which may have crept during the processing). This we perform by first passing the mask through Median filter and then performing the Closing Operation.

PROGRAM CODE

Check_face.py

```
# import the necessary packages
import face_recognition as fr
import os
import cv2
import face_recognition
import numpy as np
from time import sleep
from imutils.video import VideoStream
import random
import requests
import imutils

def get_encoded_faces():
    """
    looks through the faces folder and encodes all
    the faces
    :return: dict of (name, image encoded)
    """
    encoded = {}
    for dirpath, dnames, fnames in os.walk("./faces"):
        for f in fnames:
            if f.endswith(".jpg") or f.endswith(".png") or f.endswith(".jpeg"):
                face = fr.load_image_file("faces/" + f)
                encoding = fr.face_encodings(face)[0]
                encoded[f.split(".")[0]] = encoding
    return encoded
def unknown_image_encoded(img):
    """
    encode a face given the file name
    """
    face = fr.load_image_file("faces/" + img)
    encoding = fr.face_encodings(face)[0]
    return encoding

def classify_face(img):
    """
    will find all of the faces in a given image and label
    them if it knows what they are

    :param im: str of file path
    :return: list of face names
    """
    faces = get_encoded_faces()
    faces_encoded = list(faces.values())
    known_face_names = list(faces.keys())

    # img = cv2.imread(im, 1)
    #img = cv2.resize(img, (0, 0), fx=0.5, fy=0.5)
    #img = img[:,:,::-1]

    face_locations = face_recognition.face_locations(img)
    unknown_face_encodings = face_recognition.face_encodings(img, face_locations)

    face_names = []
    for face_encoding in unknown_face_encodings:
        # See if the face is a match for the known face(s)
        matches = face_recognition.compare_faces(faces_encoded, face_encoding)
        name = "Unknown"
        # use the known face with the smallest distance to the new face
        face_distances = face_recognition.face_distance(faces_encoded, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = known_face_names[best_match_index]

        face_names.append(name)

        for (top, right, bottom, left), name in zip(face_locations, face_names):
            # Draw a box around the face
            cv2.rectangle(img, (left-20, top-20), (right+20, bottom+20), (255, 0, 0), 2)

            # Draw a label with a name below the face
            cv2.rectangle(img, (left-20, bottom -15), (right+20, bottom+20), (0, 255, 0), cv2.FILLED)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(img, name, (left -20, bottom + 15), font, 1.0, (255, 255, 255), 2)



    # Display the resulting image

    return face_names
# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to have a maximum
    # width of 400 pixels
    frame = vs.read()
    frame = imutils.resize(frame, width=800)
    faces = classify_face(frame)
    # loop over the detected face locations and their corresponding
    # locations
    # show the output frame
    print("hi")
    print(faces)
    cv2.imshow("Frame", frame)
    for names in faces:
        if names == 'Unknown':
            cv2.destroyAllWindows()
            vs.stop()
            os.system('python main.py')
            #break
        #else:
            #continue

    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
       break

# do a bit of cleanup
#cv2.destroyAllWindows()
#vs.stop()
```

main.py

```
import sqlite3
import os

conn = sqlite3.connect('college_database')
cur = conn.cursor()
try:
  cur.execute('''CREATE TABLE Student (
  id integer Primary key  AUTOINCREMENT,
  Student_name varchar(20),
  email varchar(50),
  password varchar(20),
  Mobile_No int(10),
  Register_No varchar(50),
  Branch varchar(50),
  address varchar(100),
  year varchar(60))''')
except:
  pass

import requests
import math, random
from flask import Flask, render_template, url_for,request, flash, redirect, session
saved_otp =0
app = Flask(__name__)

@app.route('/user_login',methods = ['POST', 'GET'])
def user_login():
  conn = sqlite3.connect('college_database')
  cur = conn.cursor()
  if request.method == 'POST':
    Register_No = request.form['Register_No']
    password = request.form['psw']
    count = cur.execute('SELECT * FROM Student WHERE Register_No = "%s" AND password =
"%s"' % (Register_No, password))
    conn.commit()
    #cur.close()
    print(count)
    if len(cur.fetchall()) == 1:
      print(count)
      flash('{} You have been logged in!'.format(Register_No), 'success')
      session['logged_in_d'] = True
      # cur.execute("update Student set avaliability='true' where Roll_No='%s'" %Roll_No)
      # print("""update Student set avaliability='true' where Roll_No='%s'" %Roll_No""")
      conn.commit()
      cur.execute('select * from Student where Register_No="%s"' % Register_No)
      s = cur.fetchall()
      print(s)
      return render_template('user_account.html', a=Register_No,b=s)
    else:
      flash('invalid register Number and password')
      return redirect(url_for('user_login'))
  return render_template('user_login.html')

@app.route('/enter_mobile')
def enter_mobile():
  return render_template('enter_mobile.html')

@app.route('/user_register',methods = ['POST', 'GET'])
def user_register():
  conn = sqlite3.connect('college_database')
  cur = conn.cursor()
  if request.method == 'POST':
    Student_name = request.form['Student_name']
    email = request.form['email']
    password = request.form['psw']
    Register_No  = request.form['Register_No']
    Branch = request.form['Branch']
    address = request.form['address']
    year = request.form['year']
    Mobile_No = request.form['Mobile_No']
    cur.execute("insert into
Student(Student_name,email,password,Register_No,Branch,address,year,Mobile_No)values('%s','%s'
,'%s','%s','%s','%s',%s,%s)" % (Student_name, email, password,
Register_No,Branch,address,year,Mobile_No))
    conn.commit()
    # cur.close()

    return redirect(url_for('user_login'))

  return render_template('user_register.html')

@app.route('/server')
def server():
  import pythonchat.pyserve

@app.route('/user_account',methods = ['POST', 'GET'])
def user_account():
  if request.method == 'POST':
    Register_No = request.form['Register_No']
    return render_template('user_account_edit.html', em = Register_No)
```

```
@app.route('/')
@app.route('/home')
def home():
  if not session.get('logged_in'):
    return render_template('home.html')
  else:
    return redirect(url_for('user_account'))

@app.route('/about')
def about():
  return render_template('about.html')



@app.route('/enter_mobile' ,methods = ['POST', 'GET'])
def otp():
  global saved_otp
  conn = sqlite3.connect('college_database')
  cur = conn.cursor()
  if request.method == 'POST':
    number = request.form['number']
    saved_otp = random.randint(1111,9999)
    print("IN otp functin",saved_otp,number)
    url = "https://www.fast2sms.com/dev/bulkV2"
    payload = "sender_id=TXTIND&message=ur otp is {}
&route=v3&numbers={}".format(saved_otp, number)
    headers = {
        'authorization':
"hYTK1cSdW6R8iJMLVvyOu9Pn37tG0QgaIFqDreUzfABkpXslZHJeUnIigMSyYGX7Kc4Nr25u9j
Bo1fTZ",
        'Content-Type': "application/x-www-form-urlencoded",
            'Cache-Control': "no-cache"
            }

response = requests.request("POST", url, data=payload, headers=headers)
      print(response.text)
      return render_template('verify.html')
   return render_template('enter_mobile.html')
@app.route('/validateOTP',methods = ['POST','GET'])
def validateOTP():
    global saved_otp
    entered_otp = int(request.form['otp'])
    print("in validate OTP fun",saved_otp,entered_otp,type(saved_otp),type(entered_otp))
    if(entered_otp == saved_otp):
      print('otp matched')
      #flash('otp matched')
      return render_template('allow.html')
    else:
      print('otp wrong')
      flash('you have entered wrong otp')
      return render_template('user_login.html')
@app.route("/logoutd", methods = ['POST','GET'])
def logoutd():
      session['logged_in_d'] = False
      return home()
@app.route("/logout")
def logout():
   session['logged_in'] = False
   return home()

if __name__ == '__main__':
   app.secret_key = os.urandom(12)
   app.run(debug=True)
```

pyserve.py

```
#!/usr/bin/env python3
"""Server for multithreaded (asynchronous) chat application."""
from socket import AF_INET, socket, SOCK_STREAM
from threading import Thread


def accept_incoming_connections():
    """Sets up handling for incoming clients."""
    while True:
        client, client_address = SERVER.accept()
        print("%s:%s has connected." % client_address)
        client.send(bytes("Greetings from the cave! Now type your name and press enter!", "utf8"))
        addresses[client] = client_address
        Thread(target=handle_client, args=(client,)).start()


def handle_client(client):  # Takes client socket as argument.
    """Handles a single client connection."""

    name = client.recv(BUFSIZ).decode("utf8")
    welcome = 'Welcome %s! If you ever want to quit, type {quit} to exit.' % name
    client.send(bytes(welcome, "utf8"))
    msg = "%s has joined the chat!" % name
    broadcast(bytes(msg, "utf8"))
    clients[client] = name


    while True:
        msg = client.recv(BUFSIZ)
        print (msg)
        print("clients",clients[client])
        if msg != bytes("{quit}", "utf8"):
            broadcast(msg, name+": ")
        else:
            client.send(bytes("{quit}", "utf8"))
            client.close()
            del clients[client]
            broadcast(bytes("%s has left the chat." % name, "utf8"))
            break

def broadcast(msg, prefix=""):  # prefix is for name identification.
    """Broadcasts a message to all the clients."""

    for sock in clients:
        sock.send(bytes(prefix, "utf8")+msg)
clients = {}
addresses = {}

HOST = ''
PORT = 33000
BUFSIZ = 1024
ADDR = (HOST, PORT)
SERVER = socket(AF_INET, SOCK_STREAM)
SERVER.bind(ADDR)
#if __name__ == "__main__":
SERVER.listen(5)
print("Waiting for connection...")
ACCEPT_THREAD = Thread(target=accept_incoming_connections)
ACCEPT_THREAD.start()
ACCEPT_THREAD.join()
SERVER.close()
```

about.html

```
{% extends "layout.html" %}
{% block content %}


       <article class="media content-section">
       </article>
        <div class="form-group">
        </div>
{% endblock content %}
```

```
{% extends "layout.html" %}
{% block content %}


       <article class="media content-section">
       </article>
        <div class="form-group">
        </div>
{% endblock content %}
```

allow.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>enter page</title>
</head>
<body background="static/1.jfif">
<center><h1>Your are allow to enter</h1></center>
</body>

</html>
```

entermobile.html

```
{% extends "layout.html" %}
{% block content %}

<form action="/enter_mobile" method="post">
   <div>
      <label>Enter Mobile Number</label></br></br>
      <input type="text" name="number" placeholder="91 ***"/>
      <button type="submit">Get OTP</button>
   </div>
</form>

{% endblock content %}
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='login.css') }}">

   <title>{{ title }}</title>
</head>
<body background="static\1.jpg">
<div class="topnav">
   <a href="about"><button>About</button></a>
<!--    <a href="user_login"><button
onclick="document.getElementById('id01').style.display='block'">User</button></a>-->
   <a href="user_login"><button
onclick="document.getElementById('id02').style.display='block'">Student Details</button></a>
```

```
   <a  href="/home"><button class="active">Home</button></a>
</div>
<div class="slideshow-container">
<!--<div class="mySlides fade">-->
<!--  <img src="{{url_for('static', filename='e.png')}}" align="middle" />-->
<!--</div>-->
<!--<div class="mySlides fade">-->
<!--  <img src="{{url_for('static', filename='y.jpg')}}" align="middle" />-->
<!--</div>-->
<!--<div class="mySlides fade">-->
<!--  <img src="{{url_for('static', filename='c1.jpg')}}" align="middle" />-->
<!--</div>-->
<!--<div class="mySlides fade">-->
<!--  <img src="{{url_for('static', filename='z1.jpg')}}" align="middle" />-->
<!--</div>-->
</div>
<h1><center>Smart Security System For College</center></h1>
<br>
<div style="text-align:center">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div>
<script src="{{url_for('static', filename='login.js')}}"></script>
<script src="{{url_for('static', filename='main.js')}}"></script>

</div>
</body>
<style>
body {
  background-image: url("/static/1.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
</style>
</html>
```

layout.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='login.css') }}">


    <title>Title</title>
</head>
<body background="static/1.jfif">
<div class="topnav">
    <a href="about"><button>About</button></a>
    <a href="user_login"><button
onclick="document.getElementById('id02').style.display='block'">Student Details</button></a>
    <a  href="/home"><button class="active">Home</button></a>
</div>
          {% block content %}{% endblock %}
</body>
<style>
body {
  background-image: url("/static/1.jfif");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 1400px;
    height: 100px;
    max-width: 100%;
    overflow: hidden;
}
</style>
</html>
```

useraccount.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
    <meta charset="UTF-8">
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='login.css') }}">
    <title>Title</title>
</head>
<body background="static/1.jfif">
    <div class="topnav">
         <form action="/logoutd" method="POST">
             <a><button type="submit" class="btn">Sign out</button></a>
<!-- <a class="active" href="/Verification"><button type="submit"
class="btn">Verification</button>></a>-->
         </form>
            <a  href="/enter_mobile"><button class="active">verification</button></a>
<!-- <input type="hidden" name="email" value="{{a}}">-->
    </div>
    <center><div>{% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
              {% for category, message in messages %}
                <div class="alert alert-{{ category }}">
                  {{ message }}
            </div>
          {% endfor %}
        {% endif %}
    {% endwith %}</div></center>
<h>Student_name : {{ b[0][1] }}</h><br>
<h>Email ID : {{ b[0][2] }}</h><br>
<!-- <h>Password : {{ b[0][3] }}</h><br> -->
<h>Mobile No : {{ b[0][4] }}</h><br>
<h>Register_No : {{ b[0][5] }}</h><br>
<h>Address : {{ b[0][7] }}</h><br>
    <h>Branch : {{ b[0][6] }}</h><br>
    <h>year : {{ b[0][8] }}</h><br>
<form action="/user_account" method="POST">
<!-- <a><button type="submit" class="btn">Edit Details</button></a>-->
            <input type="hidden" name="email" value="{{a}}">
    </form>
</body>
<style>
body {
  background-image: url("/static/1.jfif");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 1400px;
    height: 100px;
    max-width: 100%;
    overflow: hidden;
}</style>
</html>
```

userlogin.html

```
{% extends "layout.html" %}
{% block content %}
<center><div  class="col-md-8">{% with messages = get_flashed_messages(with_categories=true)
%}
            {% if messages %}
              {% for category, message in messages %}
                <div class="alert alert-{{ category }}">
                  {{ message }}
                </div>
              {% endfor %}
            {% endif %}
    {% endwith %}
    </div></center>
<form action="/user_login" method="post">
    <div class="container">
      <label for="Register_No"><b>Register_No</b></label>
      <input type="text" placeholder="Enter your Register_No" name="Register_No" ><br />
      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" >
      <button type="submit">Login</button>
      <label><b>New user</b></label>
            <a href="/user_register" class ='btn2' type="submit">register</a>
    </div>
</form>
{% endblock content %}
```

userregister.html

```
p{% extends "layout.html" %}
{% block content %}
  <form action="/user_register" method="post">


    <div class="container">
      <label for="Student_name"><b>Student_name</b></label>
      <input type="text" placeholder="Enter Student Name"  name= "Student_name"
onkeypress="return /[a-z]/i.test(event.key)" ><br />
      <label for="email"><b>Email</b></label>
      <input type="email" placeholder="Enter Email" name="email" ><br />
      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" >
<!--        <label for="gender"><b>gender</b></label>-->
<!--        <input type="text" placeholder="" name="gender" />-->
        <label for="Register_No"><b>Register_No</b></label>
        <input type="text" placeholder="Enter your Register_No" name="Register_No"
class="Register_No"   />
        <label for="address"><b> address</b></label>
        <input type="text" placeholder="address" name="address" class="address" />
        <label for="Mobile_No"><b> Mobile_No</b></label>
        <input type="text" placeholder="Mobile_No" name="Mobile_No" class="Mobile_No"
maxlength="10" />
        <label for="Branch"><b> Barnch</b></label>
        <input type="text" placeholder="Enter Your Branch" name="Branch" class="Branch" />
      <label for="year"><b> year</b></label>
      <input type="text" placeholder="Enter your year" name="year" class="year" />
        <button type="submit">Submit</button>
    </div>
  </form>
{% endblock content %}
```

TESTING

11.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

11.2 TYPES OF TESTING

11.2.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

11.2.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

11.2.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: Identified classes of invalid input must be rejected.

Functions: Identified functions must be exercised.

Output: Identified classes of application outputs must be exercised.

Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

11.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

11.2.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

11.2.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

11.2.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

11.3 TEST CASES:

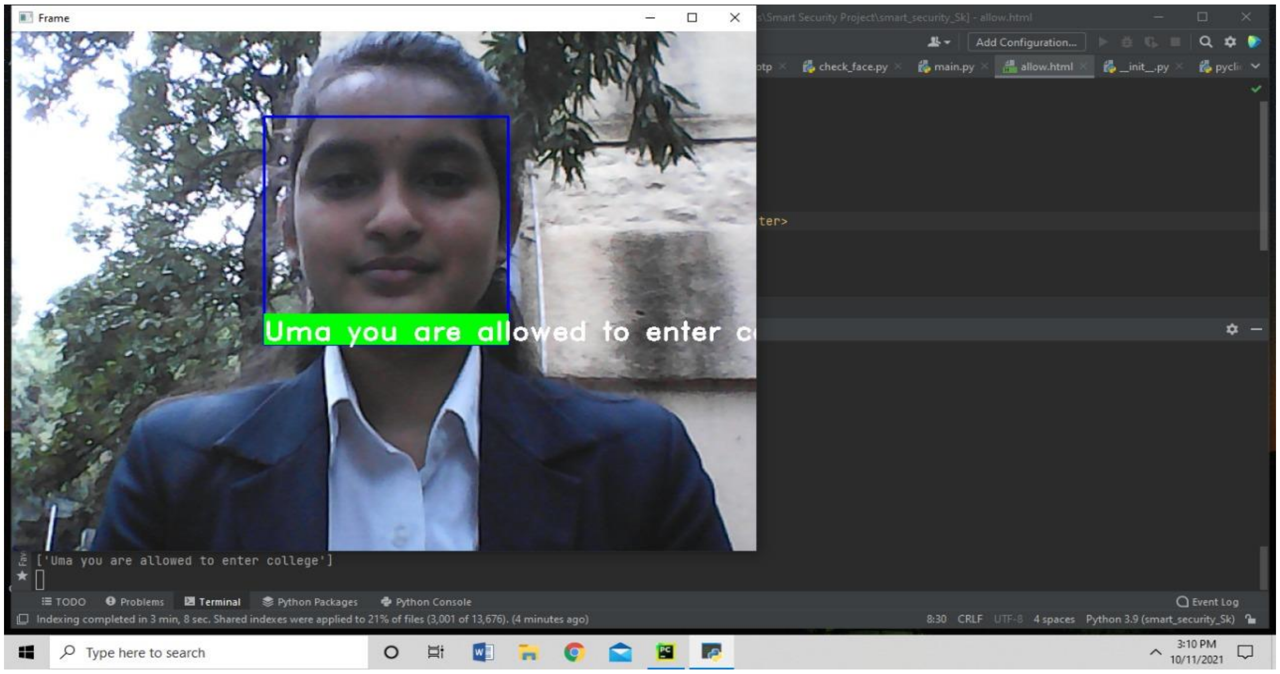
Test Case ID	SSSC-01
Test Case Title	Face Recognition
Test Case Description	User’s face matches with the stored faces and user is allowed to enter the college.
Test Input	A live Stream of a Person’s face.
Test Output	Face is recognized and name of the detected person and a message will be displayed.
Entry Criteria	Camera will be opened.
Exit Criteria	User’s face is recognized.

Test Case ID	SSSC-02
Test Case Title	Login Page
Test Case Description	Correct user-Id and password will allow access else denies access to login.
Test Input	Correct username and strong password.
Test Output	Logged-in to security system.
Entry Criteria	Displays the application home page.
Exit Criteria	Successful log-in of user.

Test Case ID	SSSC-03
Test Case Title	Registration
Test Case Description	Enter the user’s valid details in registration form.
Test Input	User Details.
Test Output	Successful Registration.
Entry Criteria	User enters the Valid details in registration form.
Exit Criteria	Successful in Registration.

Test Case ID	SSSC-04
Test Case Title	OTP Verification
Test Case Description	Correct Phone number and OTP will allow the user to enter the college.
Test Input	Correct Phone number and OTP
Test Output	Displays a page with the message that user is allowed to enter the college.
Entry Criteria	Displays a Verification Page.
Exit Criteria	Successful in OTP Verification.

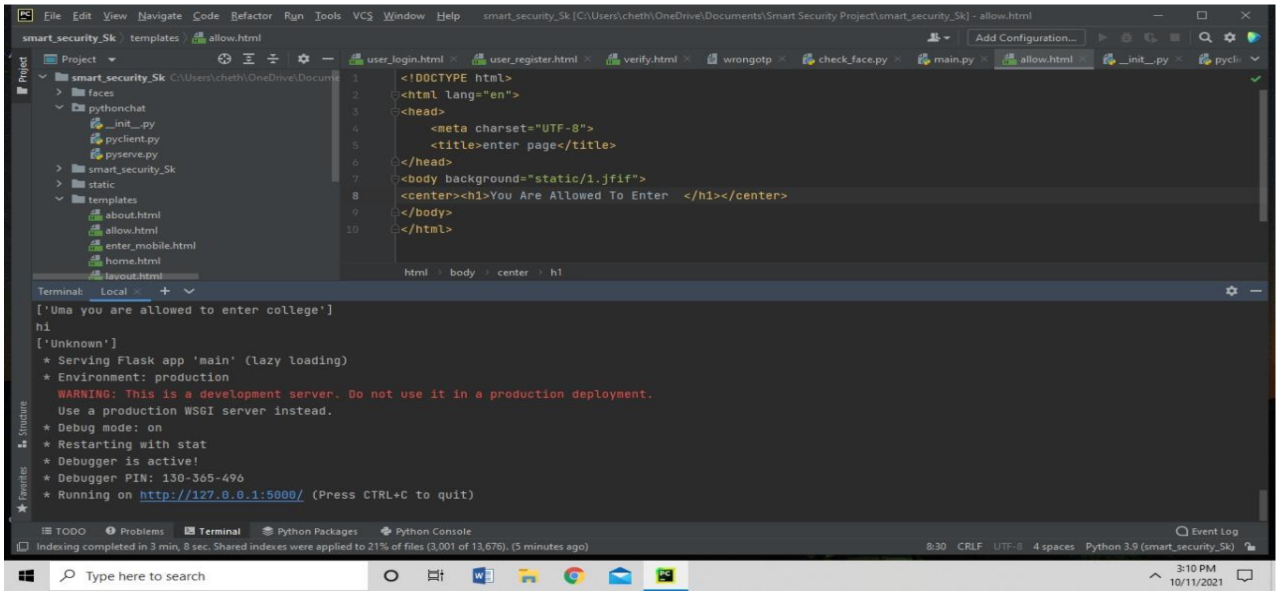
INPUT AND OUTPUT SCREENS



```
['Uma you are allowed to enter college']
```

```
Indexing completed in 3 min, 8 sec. Shared indexes were applied to 21% of files (3,001 of 13,676). (4 minutes ago)
```

- First, the camera will be opened and it captures the person in front of the camera. If the face matches with the stored faces then a message will be displayed and the person is allowed to enter the college .

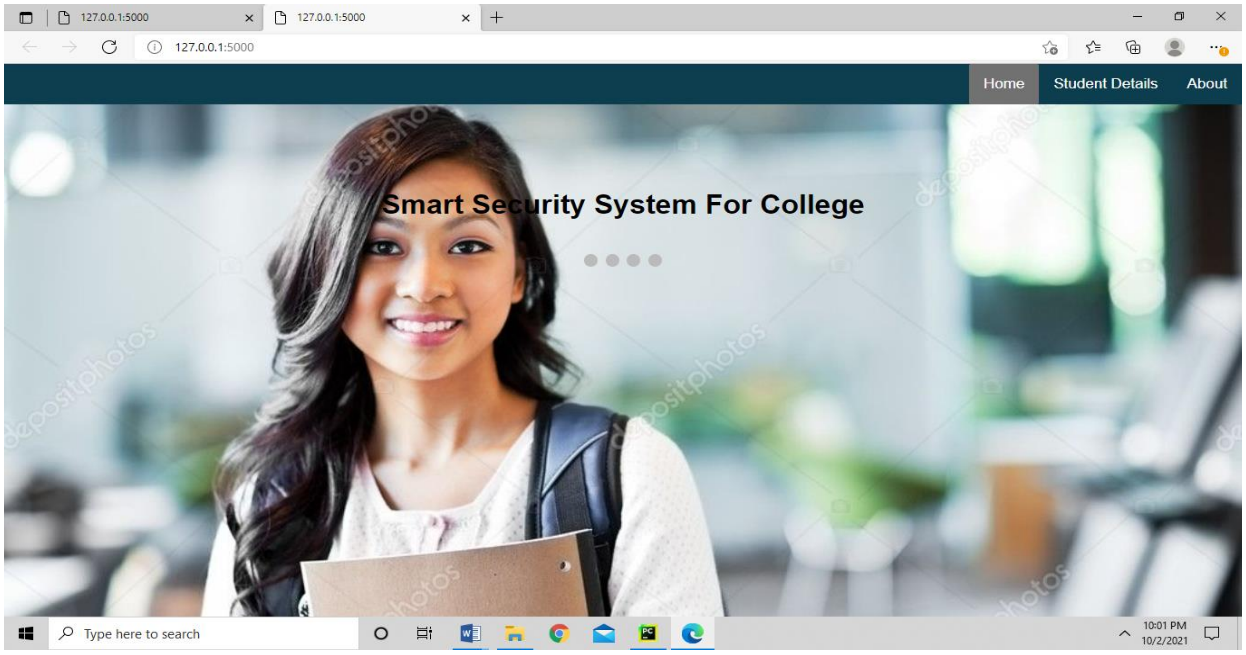


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>enter page</title>
</head>
<body background="static/1.jfif">
<center><h1>You Are Allowed To Enter  </h1></center>
</body>
</html>
```

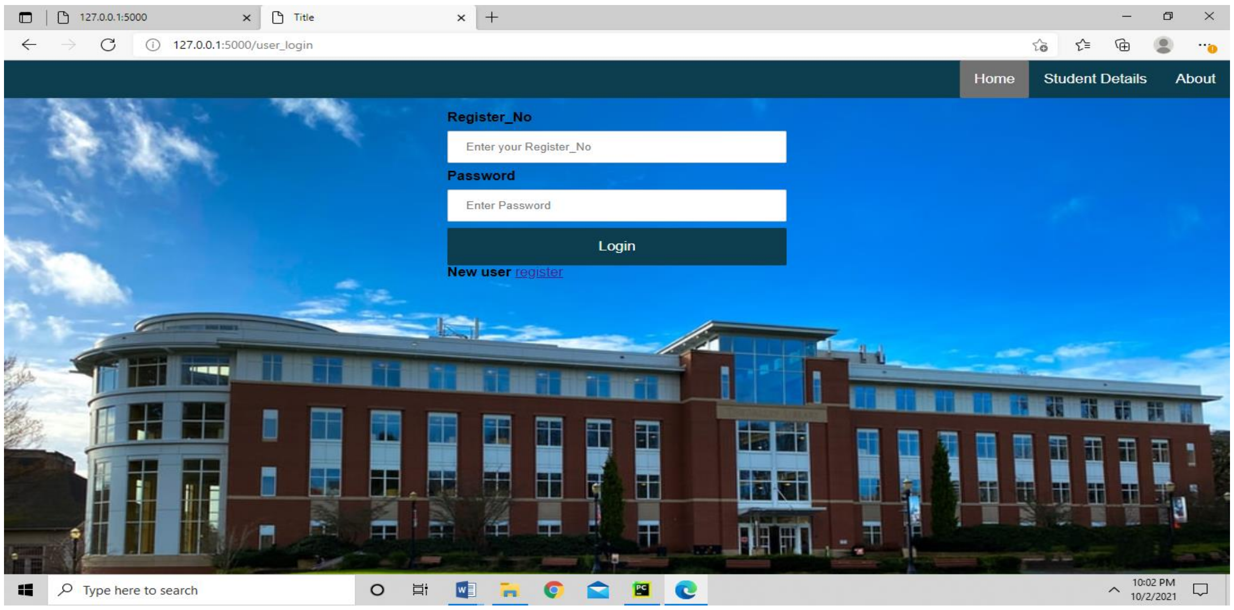
```
['Uma you are allowed to enter college']
hi
['Unknown']
 * Serving Flask app 'main' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 130-365-496
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
Indexing completed in 3 min, 8 sec. Shared indexes were applied to 21% of files (3,001 of 13,676). (5 minutes ago)
```

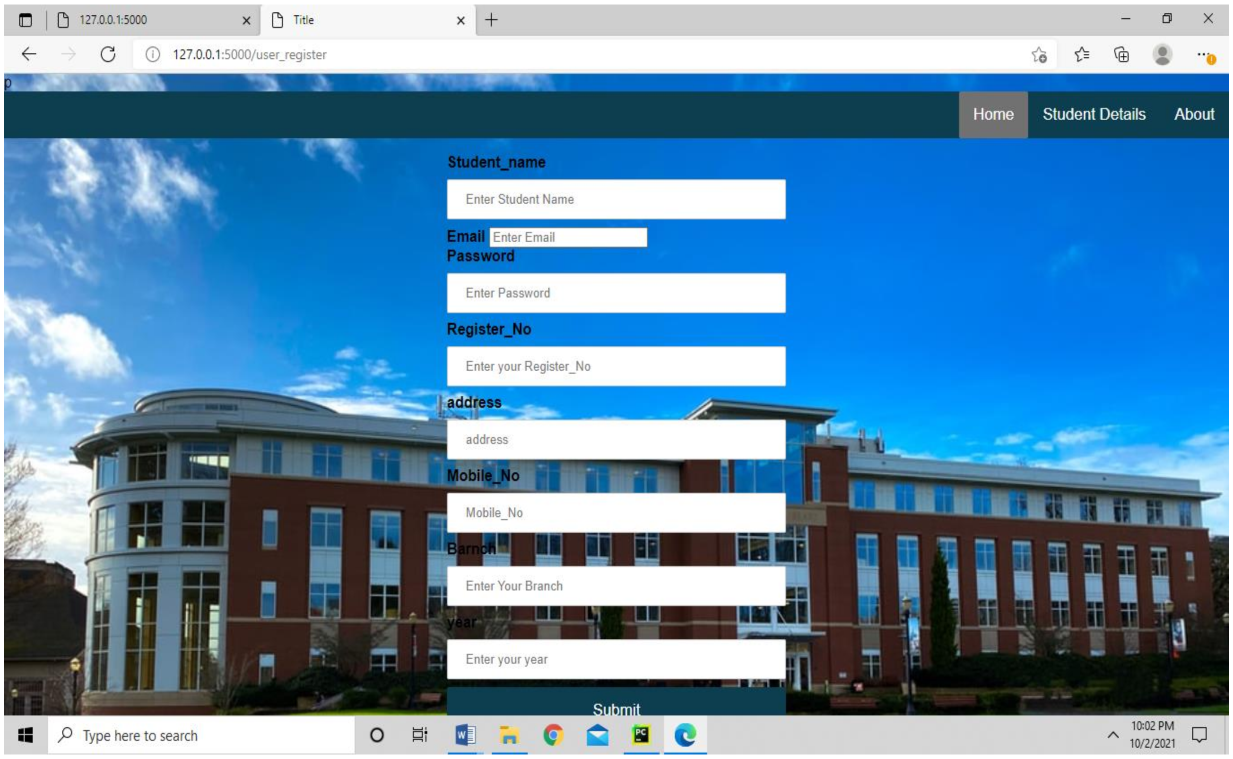
- If the person is not recognized (or if he is detected as **Unknown**) the person has to click on the link as shown above.



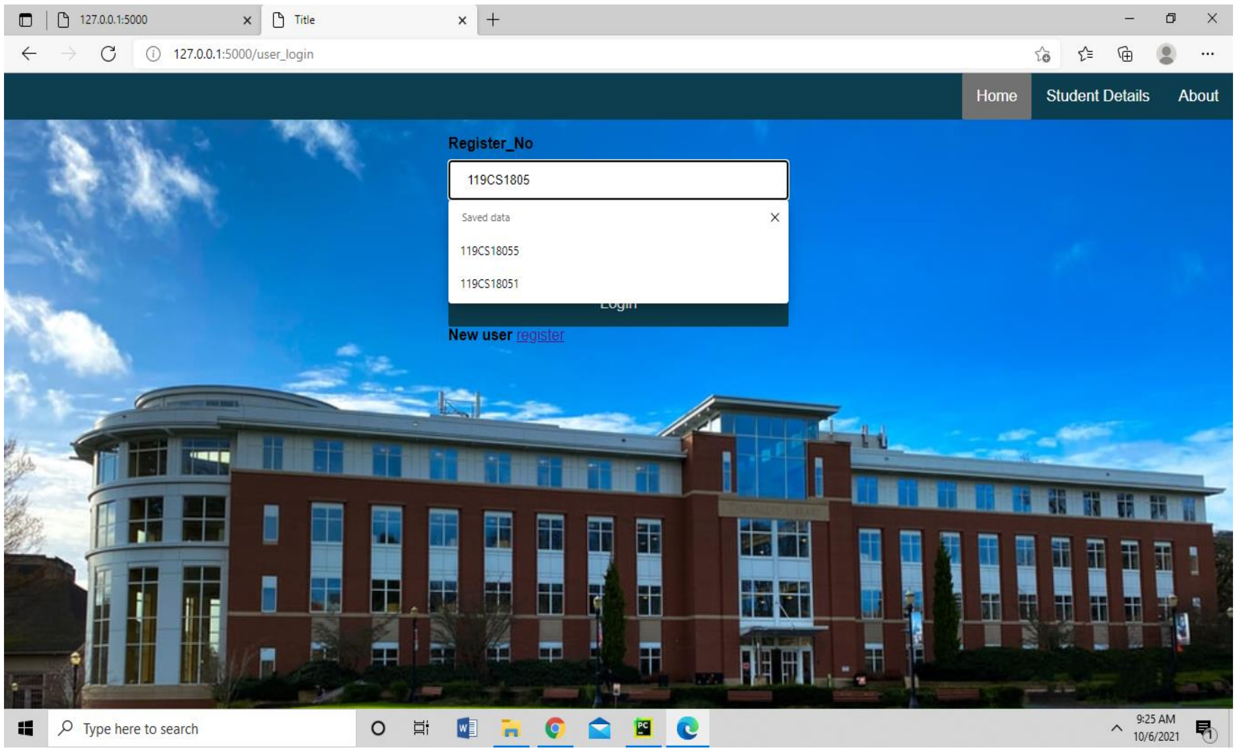
- And the above Website starting page will be appeared.



- Then click on “Student Details”. The Login screen will be displayed as shown above.
- The Student should first register herself and create Username and Password.

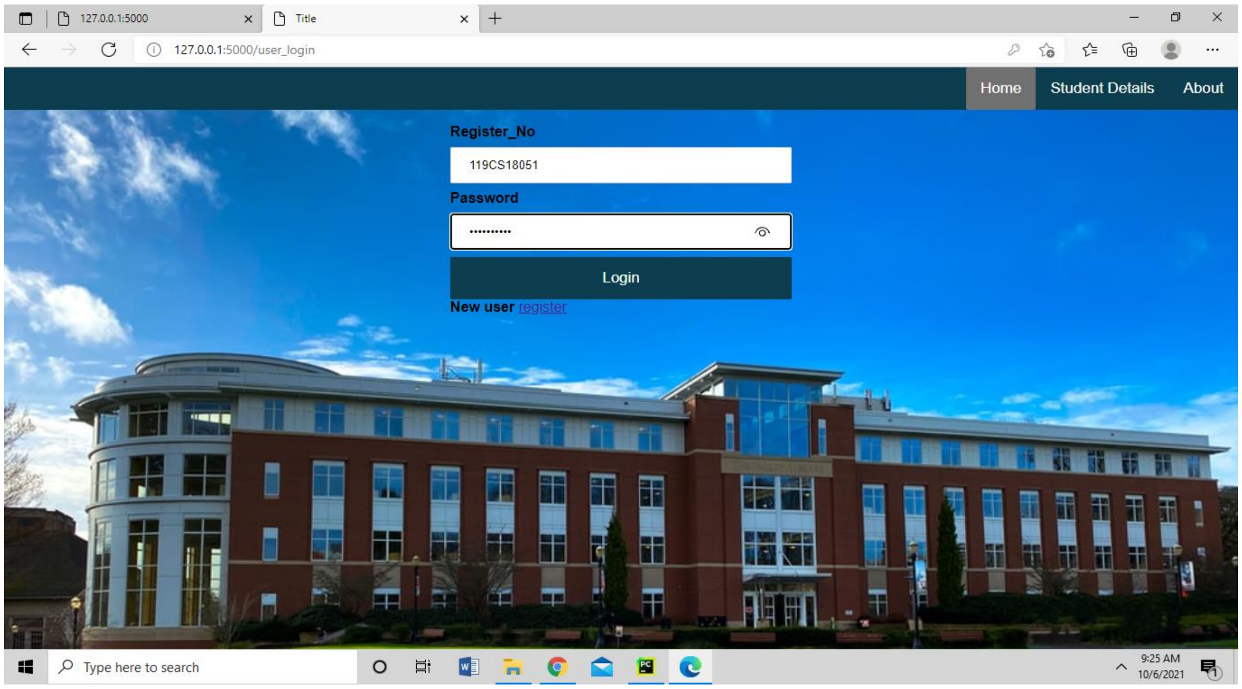


- After Successfully Registering. The User should enter the Username and Password.

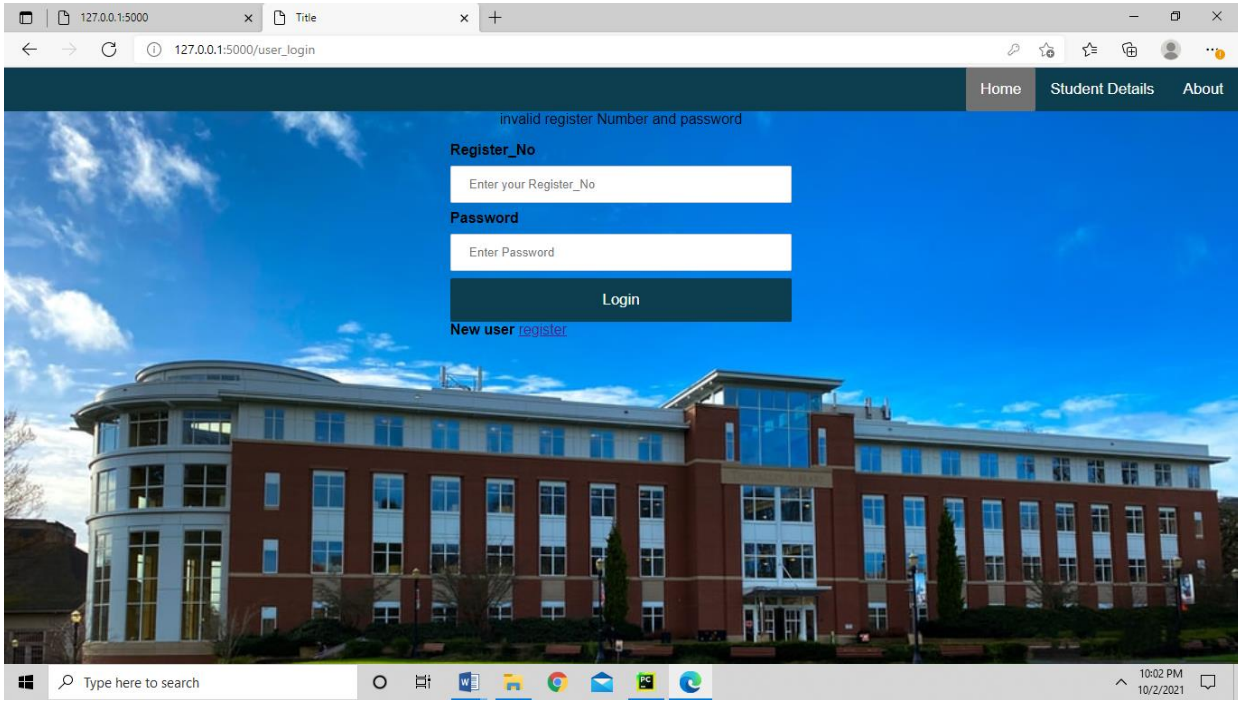


- The Username and Password is entered by the User.

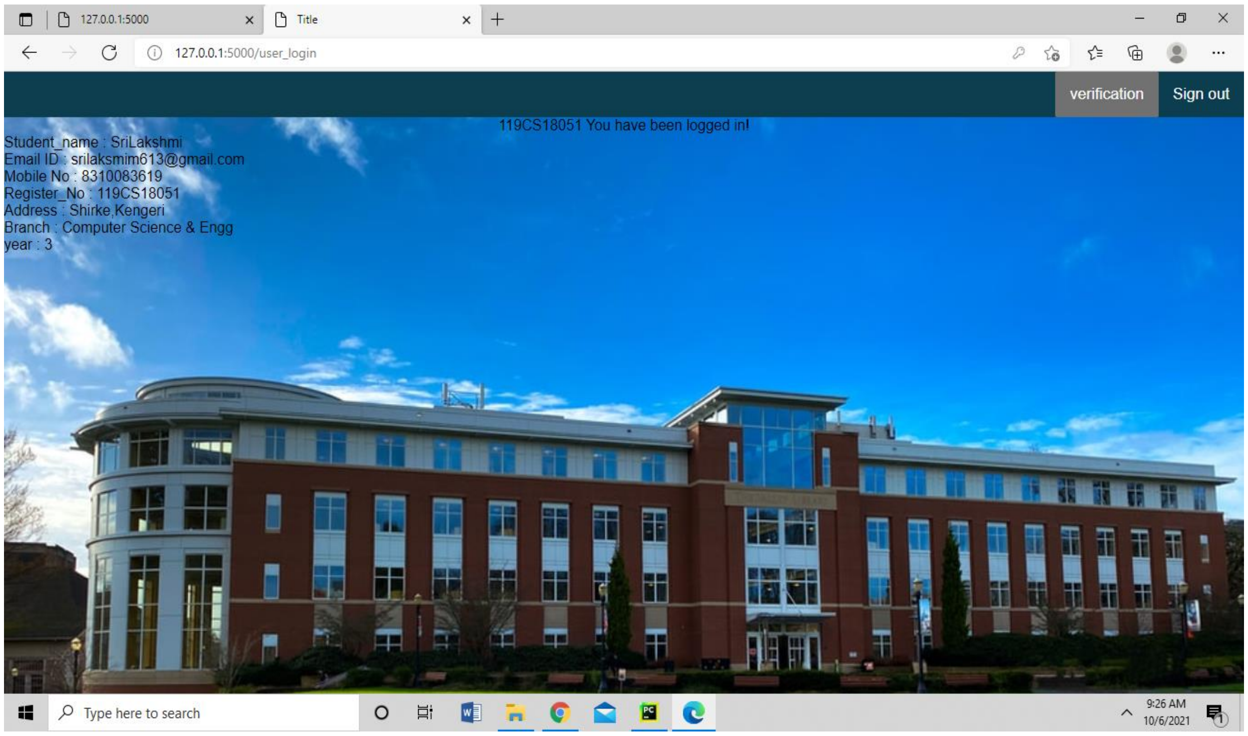
- Then click on the login Option.
- If Username and Password are correct then user can successfully login in.



- If the Username and Password are incorrect then the user cannot login.



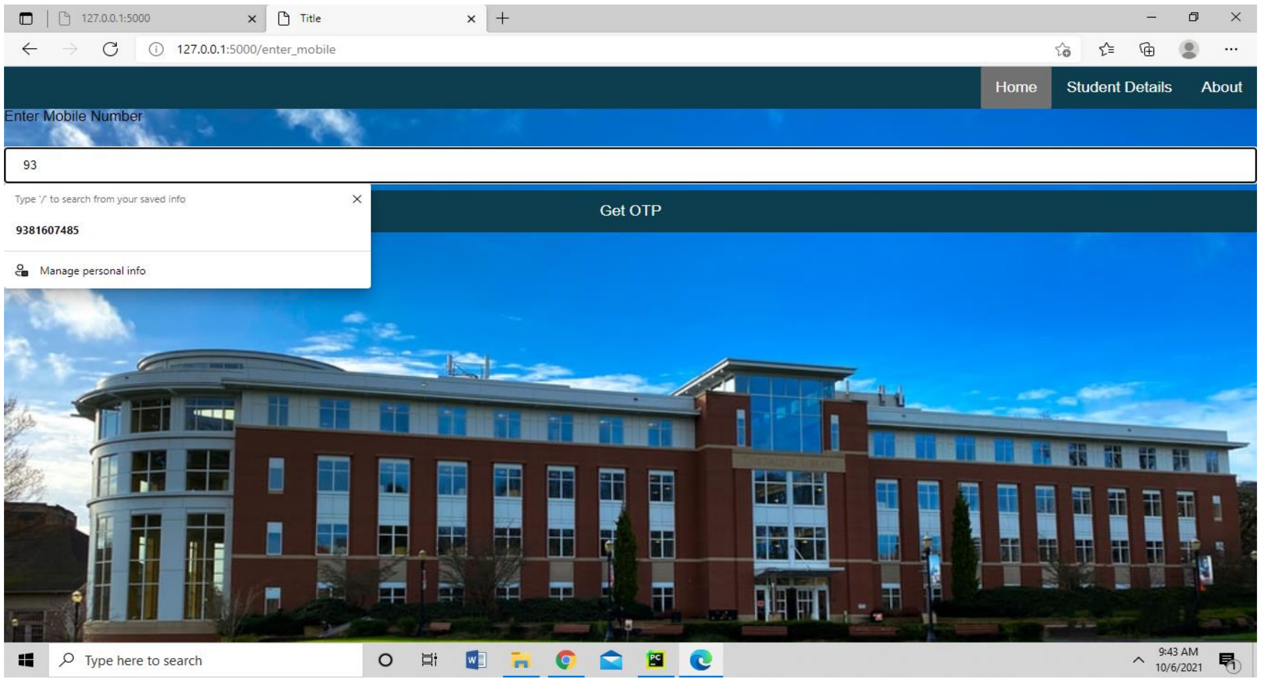
- After Logging in. The Student Details will be displayed on the screen.



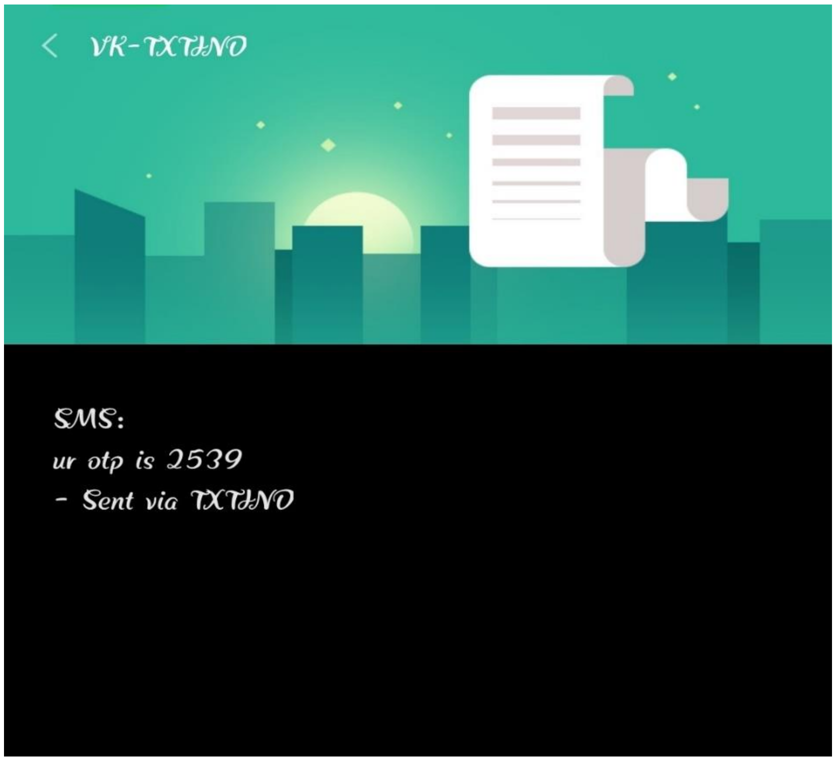
119CS18051 You have been logged in!

Student_name : SriLakshmi
Email ID : srilaksmim613@gmail.com
Mobile No : 8310083619
Register_No : 119CS18051
Address : Shirke,Kengeri
Branch : Computer Science & Engg
year : 3

- For Verification the User should click on verification to generate OTP.

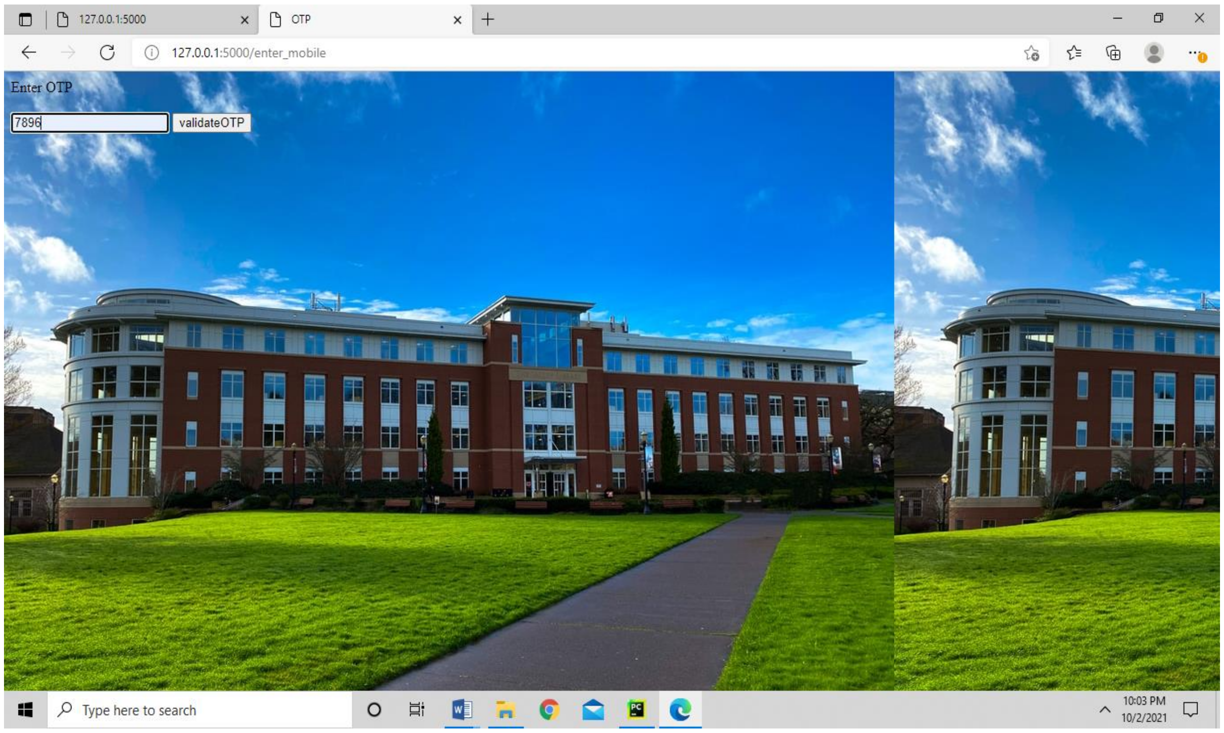


- The User should enter Mobile number and click on **Get OTP.** The OTP will be sent to the user's Phone number.



SMS:
ur otp is 2539
- Sent via TXTINO

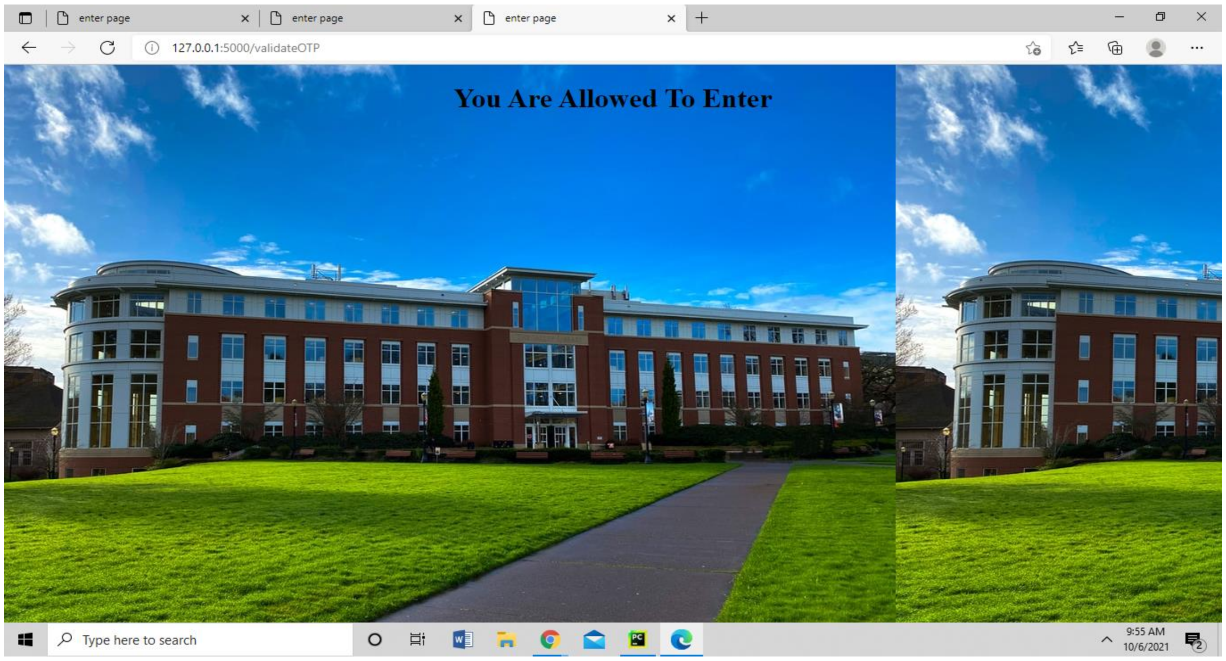
- Enter the OTP.



Enter OTP

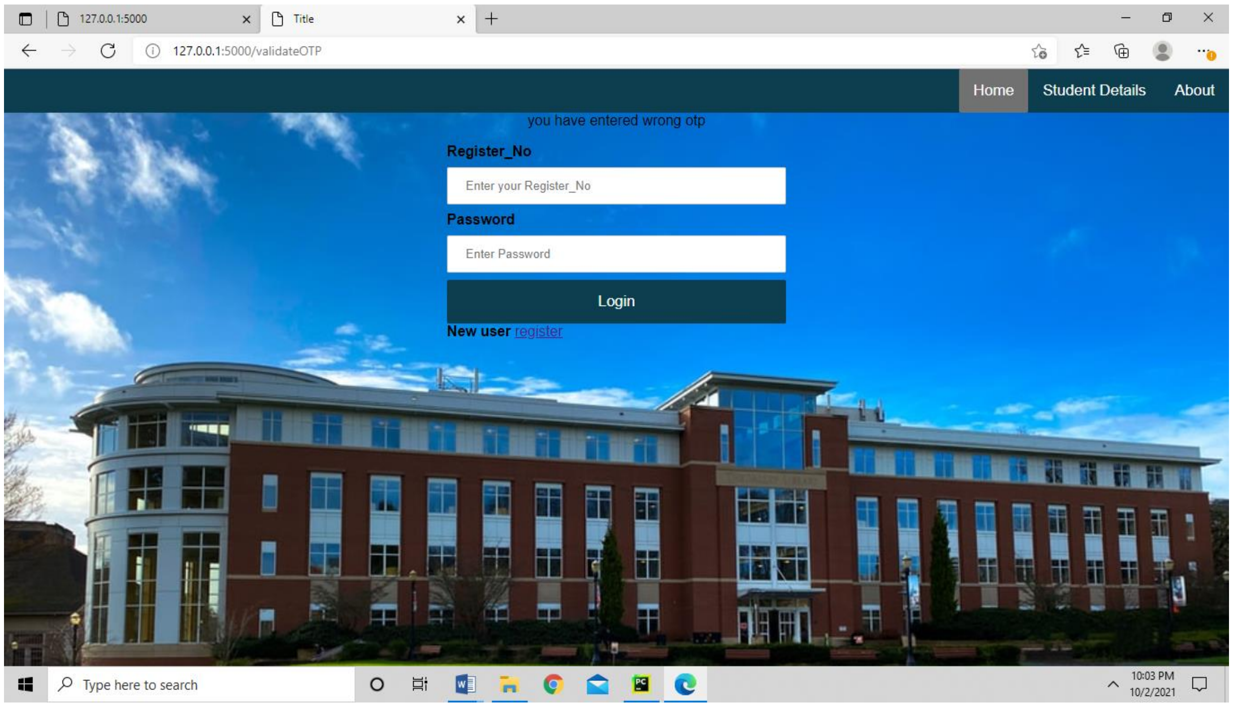
7896

- After entering the OTP. If the entered OTP is valid the below screen will be displayed and the user is allowed to enter the college.



You Are Allowed To Enter

- If the entered OTP is Invalid the below screen will be displayed and the user is not allowed to enter the college.



you have entered wrong otp

Register_No

Enter your Register_No

Password

Enter Password

New user register

Limitations of existing System:

- Time consuming more.
- Accuracy is high.
- Not user friendly: The existing system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently.
- Time consuming: Every work is done manually so we cannot generate report in the middle of the session or as per the requirement because it is very time consuming.
- Managing a very large

CONCLUSION

In this paper, we introduced a new method of obtaining data for training a security system from social media and human interaction. There are several advantages of our system which can be described. First, we should mention that using TensorFlow is adaptive, powerful, and visualizing. Also, training time is acceptable to compare with other frameworks and faster if using distributed TensorFlow

There are abundant interesting projects which are leading in Artificial Intelligent and Deep Learning developed in TensorFlow with huge support from Google. In addition, computation in parallel mode will dramatically drop the training time.

By using the method mentioned in the Face net paper, we succeeded in reaching a robust accuracy comparison with other algorithms as in Table 2. More importantly, the accuracy is improving as long as the system is used with new data from college student interaction.

REFERENCES

[1] Federal Bureau of Investigation United States Department of Justice. Crime in the united states, 2015, 2015.

[2] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

[3] Duy Tran, Ha Manh Do, Weihua Sheng, He Bai, and Girish Chowdhary. Real-time detection of distracted driving based on deep learning. IET Intelligent Transport Systems, 2018.

[4] Ha Manh Do, Minh Pham, Weihua Sheng, Dan Yang, and Meiqin Liu. Rish: A robot-integrated smart home for elderly care. Robotics and Autonomous Systems, 101:74–92, 2018.

[5] Ha Manh Do. Developing a home service robot platform for smart homes. Master's thesis, Oklahoma State University, 2015.

[6] ADT. Security systems, home automation, alarms and surveillance, 2016. [Accessed: 2016-09-09].

[7] Vivint Smart Home Security Systems. Smart home security solutions., 2016. [Accessed: 2016-09-09].

[8] Protect America. Affordable home securitysystems for everyone., 2016. [Accessed: 2016-09-09].

[9] K Denmead. Netatmo is the weather station for the rest of us. wired magazine.[online]. http. archive. wired. com/geekdad/2013/02/netatmo/.[Accessed 2014-04-02], 2013.

[10] Jeffrey S Coffin and Darryl Ingram. Facial recognition system for security access and identification, November 23 1999. US Patent 5,991,429.

[11] J Shankar Kartik, K Ram Kumar, and VS Srimadhavan. Security system with face recognition, sms alert and embedded network video monitoring terminal. International Journal of Security, Privacy and Trust Management (IJSPTM) Vol, 2, 2013.

[12] Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1):71–86, 1991.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 815–823, 2015.

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[15] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.