SMART Campus

by

ASC Muhammad Siddiqi

PC Hamza bin Atique

ASC Guftar Raza

ASC Muhammad Kashif Saeed

Supervised by

Supervisor. Prof. Dr. Hammad Afzal

Co-supervisor. Lt Col. Khawar Mahmood

In Partial Fulfillment

Of the Requirements for the degree

Bachelors of Engineering in Software Engineering (BESE)

Military College of Signals

National University of Sciences and Technology

Rawalpindi, Pakistan

June 2023

In the name of Allah, the Most Benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPORVAL

This is to officially state that the thesis work contained in this report

"SMART Campus"

is carried out by

ASC Muhammad Siddiqi

PC Hamza bin Atique

ASC Guftar Raza

ASC Muhammad Kashif Saeed

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Software Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor

Prof Dr. Hammad Afzal

Date: June 1, 2023

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else

ACKNOWLEDGEMENTS

Allah Subhan’Wa’Tala is the sole guidance in all domains.

Our Parents, colleagues, and most of all supervisor Prof Dr. Hammad Afzal your guidance it would never have been possible for us to complete our Final Year Project

Plagiarism Certificate (Turnitin Report)

We hereby declare that this project report entitled “SMART Campus” submitted to the “DEPARTMENT OF COMPUTER SOFTWARE ENGINEERING”, is a record of an original work done by us under the guidance of Supervisor “PROF DR HAMMAD AFZAL” and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Computer Science. This thesis has 8% similarity index. Turnitin report endorsed by Supervisor is attached.

Team Members	**Signature**
ASC Muhammad Siddiqi	______________________
PC Hamza Bin Atique	______________________
ASC Malik Muhammad Kashif Saeed	______________________
ASC Guftar Raza	______________________

Supervisor	**Signature**
Prof Dr. Hammad Afzal	______________________

Date:

June 1, 2023

Place:

Military College of Signals, NUST,

Rawalpindi Pakistan

ABSTRACT

Smart Campus is an innovative approach to creating a learning environment that leverages advanced technologies to improve the quality of education and enhance the student experience. It involves data analytics, and machine learning algorithms to enable real-time monitoring and control of campus operations, enhance campus security, streamline resource management, and optimize energy consumption.

One of the key features of a Smart Campus is the use of machine learning algorithms to provide personalized and context-aware services to students and staff. Security is also a critical aspect of a Smart Campus. Advanced security systems, such as facial recognition, Registration Plate reading, smoking detection, crowd activity analysis, no of people counting can be used to enhance campus security and ensure the safety of students and staff. In addition, data analytics can be used to identify potential security threats and alert campus security personnel in real-time. The implementation of a Smart Campus is not without its challenges. One of the main challenges is the need for significant investments in technology infrastructure and the training of staff to use the new systems. There are also concerns around data privacy and security, as the collection and analysis of large amounts of data can pose risks to personal privacy. Despite these challenges, the benefits of a Smart Campus are significant. By improving the learning environment, enhancing the student experience, and reducing energy waste, a Smart Campus can help institutions to attract and retain students, improve academic outcomes, and reduce operating costs. Some of the best practices and case studies from around the world illustrate the potential of Smart Campus to transform higher education and create a more sustainable future.

Table of Contents

List of Figures

Chapter 1. Software Requirements Specification

1.1 Introduction

1.1.1 Purpose.

This is a software requirement document for web-based system of SMART Campus. SMART Campus is suite of different tools and technologies, uses advanced network infrastructure and internet-connected devices to provide supportive and engaging experiences. It joins people, devices, and applications and allows universities to make insight-driven decisions to improve security and maximize resources. The system will provide Facial recognition-based attendance, facial expression recognition, number plate reading for smart parking system and crowd activity analysis.

This is the initial draft for the SRS and it will be used for the extensions. This document is prepared by following IEEE[1] conventions for software requirement specification.

1.1.2 Document Conventions

There are no specific document conventions.

1.1.3 Intended Audience and Reading Suggestions

Anyone with some basic knowledge of programming can understand this document. The document is intended for Developers, Software architects, Testers, Project managers and Documentation Writers. But anyone with programming background and some experience with UML can understand this document.

It is divided into Several phases with sections being intended for developers and software managers, but other sections can be understood by anyone having little knowledge about software.

This Software Requirement Specification also includes:

- Overall description of the product
- External interface requirements
- System Features
- Other non-functional requirements

To design meaningful test cases and provide useful feedback to developers, testers must first grasp the system's functionality. The developers must understand the specifications of the software product they are creating.

This paper is intended for broad conversations about Smart Campus System implementation decisions. The product's user should understand the fundamentals of Web development, SQL, interfaces, and classes.

1.1.4 Product Scope

Speech, fingerprint, facial photos, and other bioinformatics are heavily used in modern security systems. Furthermore, determining a user's emotional state through facial and voice analysis is critical in human-machine interaction (HMI) systems because it uses nonverbal indicators to assess the user's emotional state. This software system will be able to recognise emotions, mark attendance, analyse crowds, and manage vehicles in parking lots using camera footage. The system's simple user interface allows the user to either capture instant video/real time or upload an existing video to the system and do emotion identification, crowd analysis, and other functions. If some strange activity occurs, this system allows administrators to take appropriate action.

This project is built on the educational institute system, and this application provides the most services in a single software package utilized by teachers and system administrators. This project is built as a web-based application running on a local area network that shares data amongst departments at a university.

This project incorporates ReactJs and SQL. ReactJs is used to create the graphical user interface (GUI), which allows users to interact with software applications

The SQL Server is used to create the database in which various data will be stored. The major goal of this project is to deliver the finest user interface and various modules in a single product. Because the admin has complete access to the system, he or she can view and alter all the information saved in the database via the application.

1.1.5 References.

1. IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

1.2 Overall Description

1.2.1 Product Perspective

This is a web based project that provides a mechanism for facial recognition based attendance, crowd activity analysis and cars entry and parking system.

1.2.2 Product Functions

The SMART Campus System will be able to do following:

1. Detect and Recognize faces and expressions
    a. Mark Attendance
    b. Teacher Presence
    c. Person Tracking
    d. Sleep detection during class
    e. Smoking
2. Crowd Activity Analysis
    a. Tracking no of people in crowd
    b. Entries to restricted areas
    c. Abnormal crowd movement/activity
    d. Crowd movement tracks and flow
3. Cars
    a. Number Plate Reading
    b. In/out record

   c. Parking management
4. A web-based interface

1.2.3 User Classes and characters

There are three main user classes.

1. Highest Authority(Commandant/Rectors/HODs):

These users will be allowed to monitor the campus using this system at any time. They will have to login to access the system. They will have the authority to generate warnings, create rules and make announcements.

2. Control Room.

The control room users will monitor the campus continuously. They will create reports, make announcements, amend system rules, provide insights to higher authorities and respond in case of emergency.

3. Students.

Students will only be allowed to login and see their attendance, warning issued against them and announcements.

1.2.4 Operating Environment

Operating environment for the SMART Campus is as listed below.

- Distributed database
- Client/server system
- Operating system: Chrome OS/ Safari/ Opera
- database: sql+ database
- platform: React/Node/Python

1.2.5 Design and implementation constraints

- The information of all users, events must be stored in a database that is accessible by the website.
- MySQL Server will be used as SQL Engine and database.
- The system must be accessible 24 hours a day.
- Access should be limited to the local network of the campus
- Users must be registered with system before each semester.

1.2.6 Assumptions and Dependencies

The product needs following third party products

- Google Cloud Facial Recognition system
- Microsoft MySQL server to store database.
- React/Python/Node to develop the product.

1.3 External Interface Requirements

1.3.1 User Interfaces

User Interface will have following components:

1.3.1.1 Authentication Screens

The homepage will have a form for username and password to authenticate different users mentioned in User Classes and characters. On click of the 'LOGIN' button, user will be authenticated and taken to the dashboard.

1.3.1.2 Dashboard.

Dashboard will be same for all users. There will be a navigation tab on left side of the interface. It will contain following links

1. Cameras
2. Profiles
3. Warnings
4. Cars Record
5. Help

1.3.1.3 Cameras

In this tab, list of all active cameras will be listed, user will click on required camera to view its footage. The camera will encircle the detected faces and cars and show their name and Registration number respectively. User may click on it and will the navigated to the target profile.

1.3.1.4 Profiles

This tab will list all students in their respective classes and sections and faculty. User will click on required item and will be navigated to their profile page. Profile page will enlist following:

1. Complete Biodata
2. Enrollment Details (Students)
3. Class Schedules (Teachers/students respectively)
4. Attendance (Teachers/Students respectively)
5. Student Feedbacks (Teachers)
6. Results (Students)

1.3.1.5 Cars Record

This tab will enlist all the cars that entered the campus and left, along with all the details like Registration Number, Owner, Time of entry, Time of Exit, Gate.

1.3.1.6 Help
This tab will enlist the all the user manuals, How to do something and FAQs.

1.3.2 Hardware Interfaces
This system needs following hardware requirements.

1. Cameras to monitor the campus
2. PC/Mobile/Laptop to run the client side application
3. Server, Cameras and other client devices will connect to the system using local network to send/receive the information.

1.3.3 Software Interfaces
The software interface should follow the Model-View-Controller (MVC) model for rendering and modelling data objects. The interface must be able to connect to a database to attendance, retrieve biodata and store warnings. Source and destination formats for data must include JavaScript Object Notation (JSON), Comma Separated Value (CSV), and American Standard Code for Information Interchange (ASCII).

1.3.4 Communication Interfaces
The communication architecture must follow the client-server model. Communication between the client and server should utilize a REST-compliant web service and must be served over HTTP Secure (HTTPS). The client-server communication must be stateless. A uniform interface must separate the client roles from the server roles.

1.4 System Features
In this subsection, we will examine the features of the system in detail by categorizing them according to their functionality. For each of the feature, we will give an introduction, purpose, diagram and a stimulus/response sequence. Introduction part will give basic background information about the feature. After that, we will show a diagram for the feature representing flow of events. Alternative flow of events will be given in stimulus/response subsection

1.4.1 Facial Recognition
The Facial recognition system is further divided into five modules

1.4.1.1 Description
The system will be able to detect and recognize faces using cameras. The cameras will record and stream the live recordings to the server, the server will detect faces and recognize using the database records.

1.4.1.2 Stimulus/Response Sequence

The sequence of events is as follows:

1. The camera continuously records the video and sends to the server.
2. Server continuously monitors the video and detects the faces
3. If there is a face, server detects it.
4. The detected face is matched with the database.
5. If the records match, the output is sent to the user interface.
6. User interface encircles the face and labels the face with the output name.
7. If the records are not match, server sends “Not Matched”
8. The user interface encircles the face with red circle.
9. This indicates that the person is not registered in the database.

This is illustrated in the diagram below.

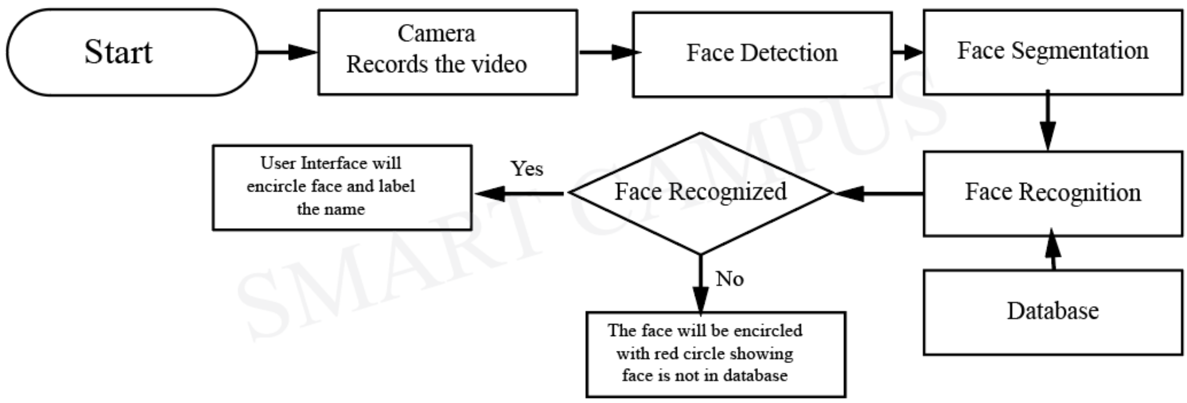

Figure 1. Facial Recognition Response

1.4.1.3 Functional Requirements.

Following are the functional Requirements for the facial recognition system.

1.4.1.3.1 Mark Attendance/Teacher Presence/Person Tracking

Mark Attendance: REQ-1

Teacher Presence: REQ-2

Person Tracking: REQ-3

After the face is recognized, the system will check if the person is staff, faculty or a student.

If the person is student and camera is the class of respective student, the system will mark that student present and will note the time. Now it will continuously detect the faces, the moment any face disappears from the camera, system will wait for 5 seconds, still if student’s face is not appearing the system will note the date/time and will mark absent for that specific time slot.

Here is the flow chart for the process:

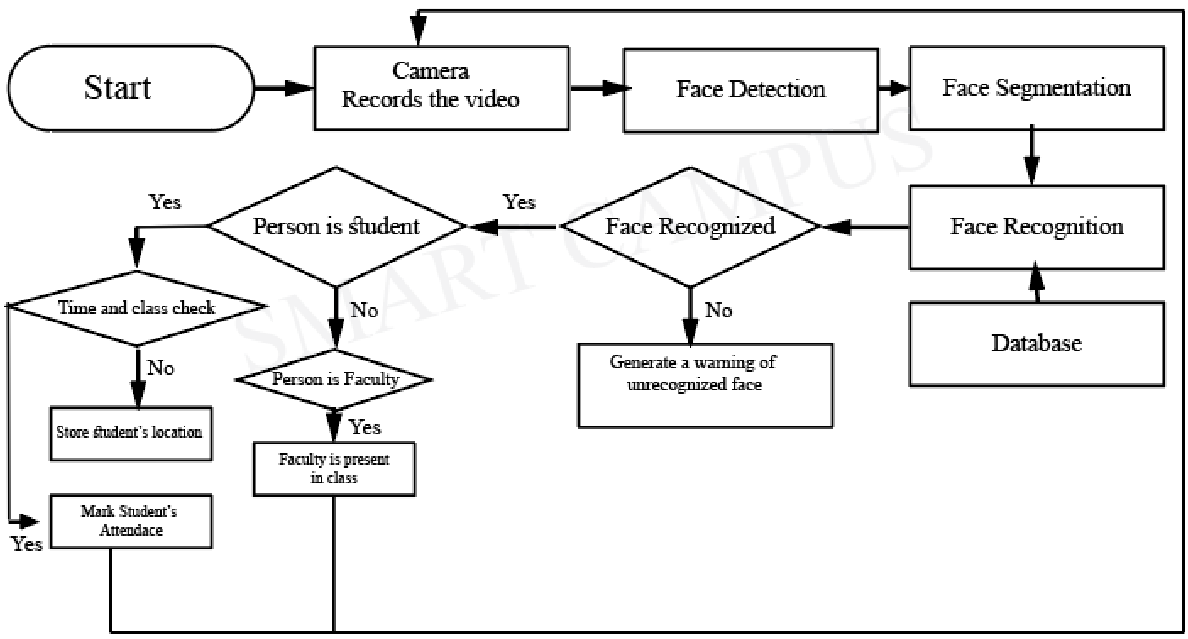

Figure 2. Attendance System

1.4.1.3.2 Sleep Detection

ID: REQ-4

The system will detect if student is sleeping or not, in case of sleeping the system will mark him/her absent during that specific time.

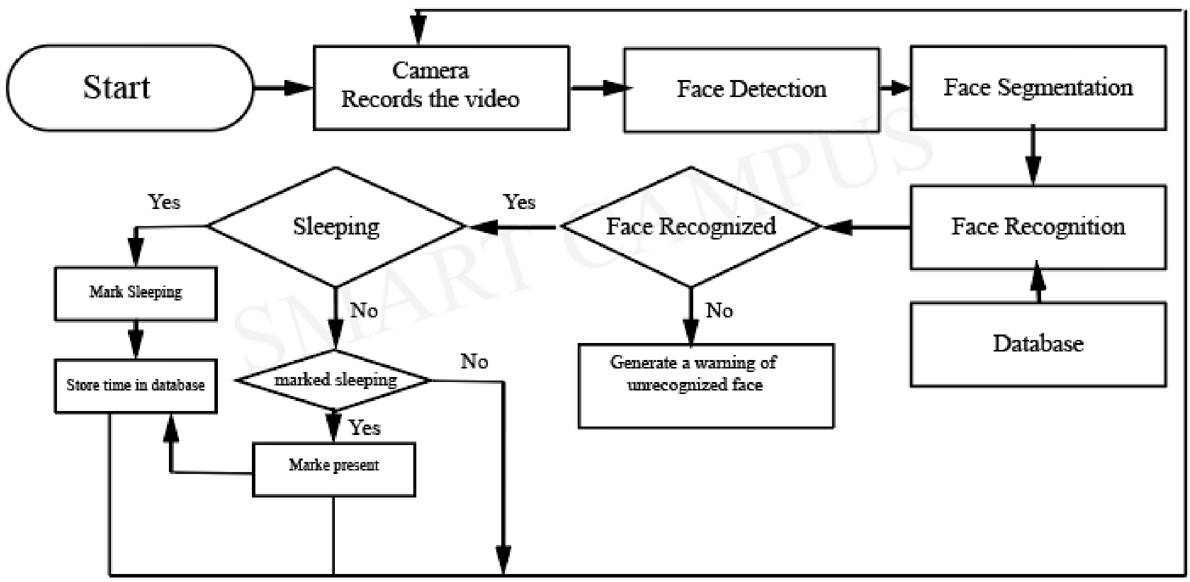

Figure 3. Sleep Detection

1.4.1.3.3 Smoking Detection

ID: REQ-5

The system will be able to detect whether a student in smoking or not. If smoking the system will generate a warning to notify the authorities and will impose a fine on the student.

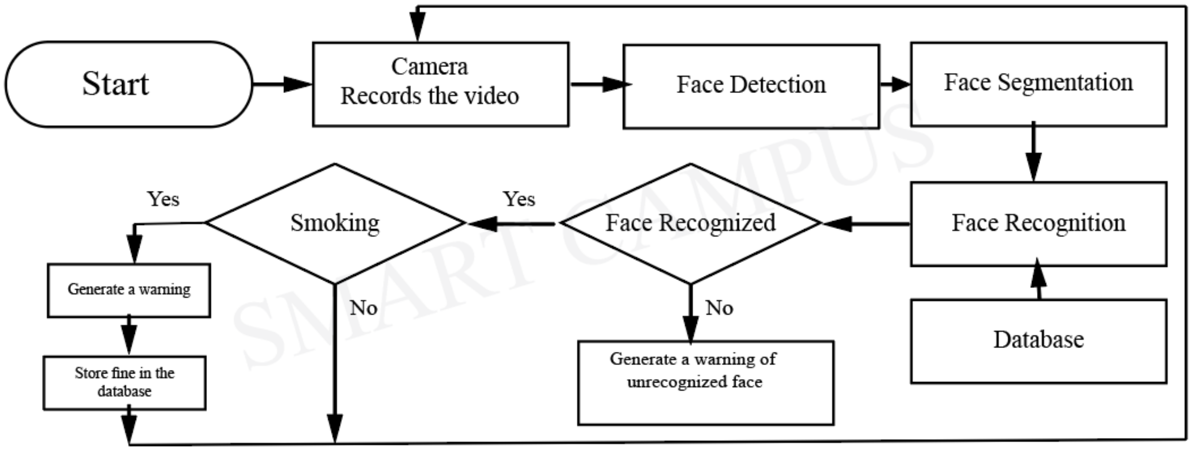

Figure 4. Smoke Detection

1.4.2 Crowd Activity Analysis

The system will be able to analyze some activities of crowd.

1.4.2.1 Description

Following are the activities that the system should detect.

1.4.2.1.1 Tracking No. of people

ID: REQ-6

The system will be able to count number of faces detected in a camera footage. The camera will live stream the video to server, server will detect face and count them.

The sequence diagram is shown below.

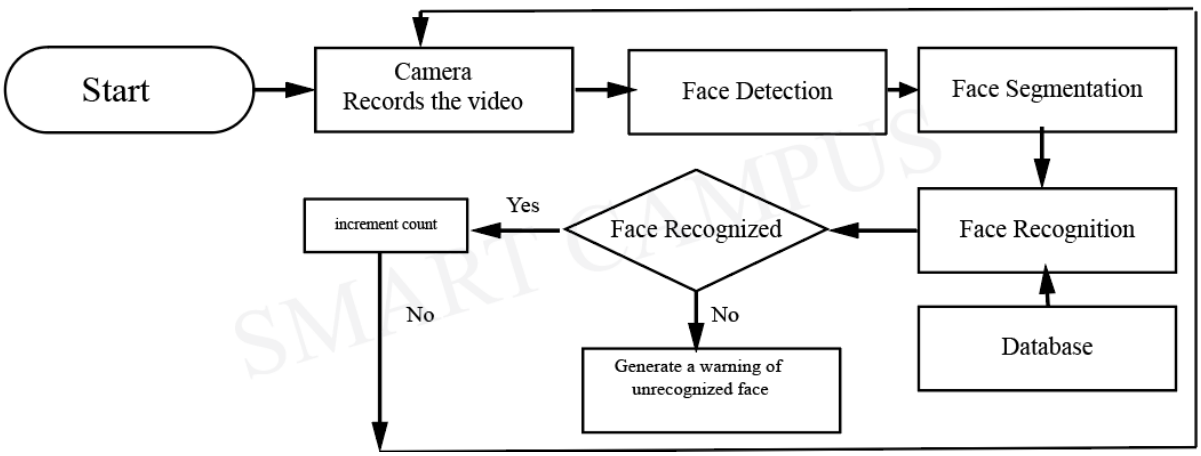

Figure 5Track No. of People

1.4.2.1.2 Other activities

The system is also capable of detecting following activities

a. Entries to restricted areas. ID: REQ-7
b. Abnormal crowd movement/activity. ID: REQ-8
c. Crowd movement tracks and flow. ID: REQ-9

Face detected in specific camera dedicated to restricted area will be marked as violation of rule. The system will track where the crowd is located by keeping the record of how many people are in specific area.

The flow chart showing the activities is shown on next page.

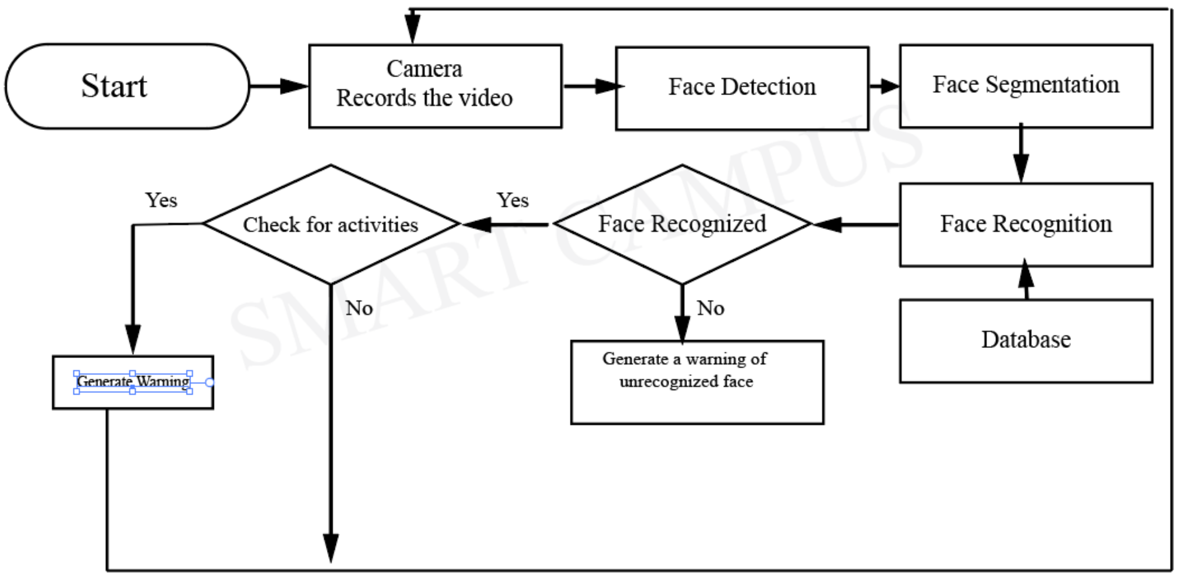

Figure 6. Crowd Activities

1.4.3 Cars

The system will be able to register the car using their registration numbers. The input required will be the registration number, owner name, owner's rank, license number. Following are some requirements.

1.4.3.1 IN/OUT

ID: REQ-10

Only registered vehicle will be allowed to enter the campus. The system will read the registration plate and match it with the database. If match is successful, the vehicle will be allowed to enter the campus else manual entry process will be executed. The process diagram is given below.

1.4.4 Requirements Prioritization

Requirement	*Priority (1-9)*
REQ-1	9

REQ-2	9
REQ-3	5
REQ-4	4
REQ-5	5
REQ-6	7
REQ-7	8
REQ-8	5
REQ-9	7
REQ-10	8

Figure 7. Requirment Priortization

1.5 Other Non-functional Requirements

1.5.1 Performance Requirements

Performance requirement of the system is one of the main NFRs of smart campus system as the project is going to work on facial recognition system and number plate readings. The response rate of facial or number plate recognition must be less than 2 seconds which is the maximum boundary line of the response time. In case of rush/crowd the system should recognize more than one faces simultaneously the minimum capacity will vary from module to module according to given area of coverage. For example, in case of classroom the facial recognition system should recognize more than 40 faces at a time as 40 students are the normal capacity of a class and in case of "CARS" module the recognition capacity of incoming or outgoing cars should be according to the road width and area available to scanning camera. The system should be responsive and error free in all the conditions for example: harsh weather, high load on system, etc.

1.5.2 Safety Requirements

Safety is sub-set of the requirements which are directly related to ensuring safe operations and sometimes requirements on protection systems which are designed to protect against accidents. For physical safety the project components should place on safe positions to avoid possible accidents/harms by human factors or natural factors. The main controlling/admin system should place on a safe position to entertain only authorized bodies to avoid any harm by unauthorized bodies. The system should perform an operation only in optimal situations and should trigger the alarm only when it is needed. For example, in case of crowd detection module the system should not trigger alarm if it detects crowd in book shop area or in cafe where crowd is a usual thing.

1.5.3 Security Requirements

Ensure the integrity of the system from accidental or malicious damage. Make sure all data inside the system or its part will be protected against malware attacks or unauthorized access. We have to define the rules of ACL to provide access to only authorized persons and to make our sensitive data and information safe from unauthorized persons. The system should be resilient and resistive to outside factors affecting it`s functionality and other operations. Use secure databases to store the information and data of the students, faculty, cars and others objects that we are going to use in our Machine Learning model of Smart Campus. This information should only be accessible to authorized members of admin panel according to ACL. The access permissions for system data may only be changed by the system's data administrator All system data must be backed up every 24 hours and the backup copies stored in a secure location which is not in the same building as the system.

1.5.4 Software Quality Attributes

1.5.4.1 Adaptability

Smart Campus system should adapt the new conditions but should not change it`s behaviour, system should work normally and according to given algorithms.

1.5.4.2 Maintainability

Maintainability is often measured at code level using Cyclomatic complexity so the Smart Campus project should not contain dead code(code which is not normally used by other functions or modules).

1.5.4.3 Reliability

The system should be reliable under all given conditions, and all the modules should work simultaneously without interfering each other`s performance. No component should disturb the input or output of the other component or module.

1.5.4.4 Portability

The Smart Campus system solution is for all educational and training institutes so it should work perfectly in different environments or institutes as it will execute the same results and will work the same in other institutes.

1.5.4.5 Usability

This can be measured in terms of ease of use. The admin side panel or application interface must be user friendly. There must be no complexity in the system usability.

1.5.4.6 Correctness

The Smart Campus Project should be correct in terms of its functionality, calculations used internally, and the navigation should be correct. This means that the project should adhere to functional requirements.

1.5.4.7 Testability

The system should be easy to test and find defects. If required, it should be easy to divide into different modules for testing.

1.5.4.8 Flexibility

Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.

1.5.4.9 Reusability

Software reuse is a good cost-efficient and time-saving development method. Different code library classes should be generic enough to be easily used in different application modules. Divide the application into different modules so that modules can be reused across the application.

1.5.4.10 Interoperability

Interoperability of one system to another should be easy for the product to exchange data or services with other systems. Different system modules should work on different operating system platforms, different databases, and protocol conditions.

1.5.5 Business Rules

Only the ACL/admins can access the personal information and recorded data of the Smart Campus System from the databases as the data contains the personal information of the students, faculty and cars. As this data is personal so this should not provide to other organizations and companies in order to protect the privacy of every individual.

Chapter 2. Design and Development

2.1 Introduction

The Software Design Specification (SDS) document's introduction provides an overview of the complete SDS, including its goal, scope, definitions, acronyms, abbreviations, references, and overview. This document's purpose is to discuss in depth the functional and non-functional features of the SMART Campus project, which employs image processing techniques and deep learning to recognize faces, number plates and crowd activity analysis. This document provides thorough descriptions and visualisations of the SMART Campus.

2.2 Purpose.

This is a software requirement document for web-based system of SMART Campus. SMART Campus is suite of different tools and technologies, uses advanced network infrastructure and internet-connected devices to provide supportive and engaging experiences. It joins people, devices, and applications and allows universities to make insight-driven decisions to improve security and maximize resources. The system will provide Facial recognition-based attendance, facial expression recognition, number plate reading for smart parking system and crowd activity analysis.

This is the initial draft for the SRS and it will be used for the extensions. This document is prepared by following IEEE[1] conventions for software requirement specification.

2.3 Product Scope

Speech, fingerprint, facial photos, and other bioinformatics are heavily used in modern security systems. Furthermore, determining a user's emotional state through facial and voice analysis is critical in human-machine interaction (HMI) systems because it uses nonverbal indicators to assess the user's emotional state. This software system will be able to recognise emotions, mark attendance, analyse crowds, and manage vehicles in parking lots using camera footage. The system's simple user interface allows the user to either capture instant video/real time or upload an existing video to the system and do emotion identification, crowd analysis, and other functions. If some strange activity occurs, this system allows administrators to take appropriate action.

This project is built on the educational institute system, and this application provides the most services in a single software package utilized by teachers and system administrators. This project is built as a web-based application running on a local area network that shares data amongst departments at a university.

This project incorporates ReactJs and SQL. ReactJs is used to create the graphical user interface (GUI), which allows users to interact with software applications.

The SQL Server is used to create the database in which various data will be stored. The major goal of this project is to deliver the finest user interface and various modules in a single product. Because the admin has complete access to the system, he or she can view and alter all the information saved in the database via the application.

2.4 Overview

This document is about the detailed architectural design of SMART Campus. For simplicity the document is divided into various sections. Section 1 introduces the document and provides overview for executive purposes. Section 2 includes detailed description of the system with various diagrams and charts. This section includes all the architectural details of system under development. Section 3 describes all the modules and components of the system in detail one by one. Section 4 compares this product to various other similar products available in market.

Section 5 throws light on the design decisions and tradeoffs. In the last section pseudo code of all the components in provided.

This document is intended for developers, testers, users, documentation writers, project clients, project supervisor and project evaluators. A copy of this document will be made available to all stakeholders

2.5 System Overview

This section provides detailed system architecture of Smart Campus System. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

2.6 Overview of Modules

Smart Campus System requires several modules to work. These modules can be divided into three parts, Facial Recognition, crowd detection and car parking. Following is the brief overview of all these modules. Detailed descriptions of these modules are presented in section 3.

2.6.1 Web Application.

The Web Application basically has two sub-modules in terms of usage which include.

1. Admin/Staff Side
2. Students side

2.6.2 Admin/Staff Side.

Admin/Staff side of Web Application module has further Three components which include.

1. face registration
2. live video acquisition from server
3. notification alerts / results

2.6.3 Face Registration.

Face Registration module will be used by admin/staff to register new students or faculty staffs to add their data in database for usage in other modules like attendance, sleep and smoke detection and for faculty the data will be used in teacher presence module. By fetching this data from database, the model will generate decisions, alerts and notification.

2.6.4 Live Video Acquisition from server.

This component of Web app module will fetch processed video from server and will display it to the monitor of admin/staff. This processed video will contain some alerts/notification and real time detections to take some actions.

2.6.5 Notification Alerts/Results.

This component will display notification on the admin/staff screen to take some actions.

2.6.6 Student side.

Student side of the web app module further has Two components which includes.

1. view attendance
2. view notification

2.6.6.1 View Attendance.

Students could see their attendance by this component of web application module through which they could keep track of their attendance as the attendance will be marked and generated on the basis of model at the backend.

2.6.6.2 View Notification.

Students will have access to view notifications generated based on model at the backend.

2.7 Server Module

Main functionality is at Server side so this module contains the main and detailed components which includes.

1. End Point Rest API
2. Video Acquisition from camera
3. Video pre-processing
4. Feature extraction
5. Facial recognition
6. Smoke/sleep detection
7. Teacher presence
8. Crowd activity
9. Car parking management
10. Number plate reading
11. Notification generation

2.7.1 End-Point rest API.

This component is the main contact point between the web application and the server-side. It receives the video from the cameras and send them to desktop after processing. This component of system interfaces with Video fetched and Display Results components to the web application. Endpoint – REST API component would receive videos from cameras and send results to the web application.

2.7.2 Video Acquisition from camera.

This feature enables the server-side to interpret the input received at the endpoint. video is processed at live time for detection. This component of system interfaces with the Endpoint – REST API to obtain the information received at the endpoint.

2.7.3 Video pre-processing.

This component is responsible for the dividing the video into required number of frames so that they undergo feature extraction. This component of system interfaces with the Video Acquisition module to obtain the video.

2.7.4 Feature extraction.

This component would basically perform feature extraction on frames. Feature Extraction component would process the incoming frames and store the feature data for each frame. This is then fed to the Smart Campus System model.

2.7.5 Facial Recognition.

This major component covers two requirements. One is that the model loads the feature data for each of the frames into the neural networks. And assess that data to classify the facial recognition. This component of system interfaces with the Feature Extraction component to obtain the feature data.

2.7.6 Smoke/sleep detection.

Live time cameras` video will be processed at server side and the feature extraction and model implementation will automatically detect someone who is sleeping and will generate alerts on web applications interface. This component is dependent on the Registration component which stores the data and images of students to tell who is smoking or sleeping.

2.7.7 Teacher presence.

Teacher presence will be detected through live camera after processing the video at server side. The detection will be performed through area of stage where teacher has to be present. So the detection will be made on that basis and the notification or alert will be generated accordingly.

2.7.8 Crowd activity.

Crowd will be detected at the places where crowd in normally not expected and on the basis of that crowd activity will be detected and alerts will be generated.

2.7.9 Car Parking Management.

The server will track the record for number of vehicles entered in specific parking. When the parking will be occupied, it will generate a alert for full space.

2.7.10Number Plate Reading.

The sever will extract number plates of cars from live video frame, and maintain in/out record for vehicles.

2.7.11Generating Alerts/Results.

This last component is responsible for producing the relevant alerts/results based on the classification done by the model. And then send that notification to the application through the Endpoint. Generating Alerts/Results properly formats the response which includes the classification note and the resultant video and sends it to the Endpoint for it to be transmitted to web application.

2.8 System Architecture

This section covers the overall architectural description of SMART Campus. It encompasses the high-level and low-level descriptions of the project including block diagrams of the application and the deep learning model. Moreover, a complete object-oriented description which includes class diagrams, sequence diagrams and others. Finally, the rationale for the design pattern is provided.

2.9 Architectural Design.

The architectural design has been divided into two portions, one covers the application architecture while the other depicts the architecture of the deep learning model used.

2.9.1 System Block Diagram.

This diagram shows the higher-level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

All the modules mentioned above are incorporated in this system block diagram.

First of all, we have the deep learning model which we have trained on datasets of videos for crowd Activity analysis and car parking system., we wrap that model using Django into a web service so that the user can use the model to run predictions. For facial recognition we will use Google Facial Recognition API. Then

In order to achieve mobility and scalability, we construct a Docker Image that would serve as a mini-VM to provide the model with environment it needs to execute including all the dependencies such as TensorFlow, Python, and Python Libraries. This Docker Image would communicate with the Web Application via the End Point based on REST API.

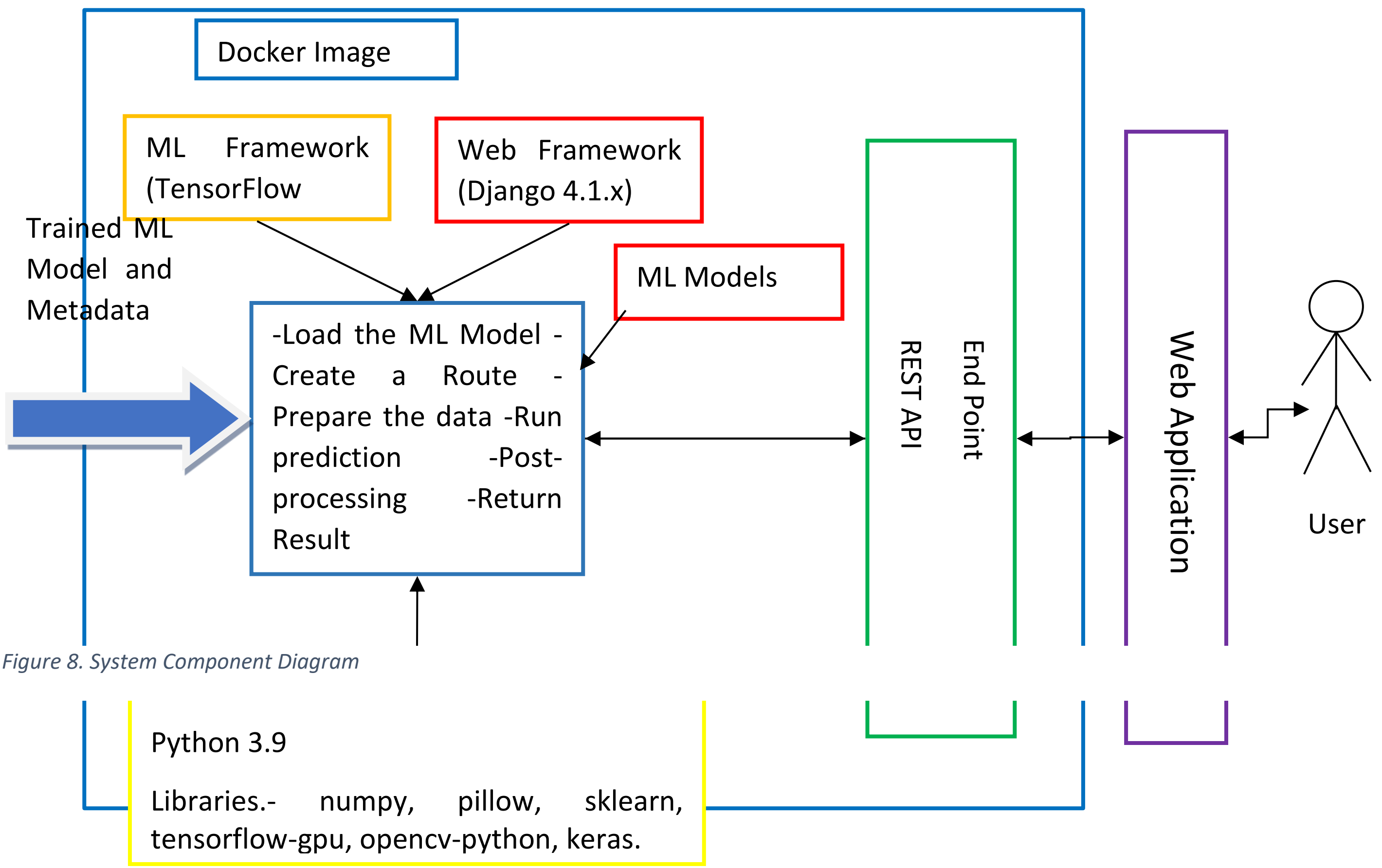

Figure 8. System Component Diagram

2.9.2 System Block Diagram (Crowd Activity/Density)

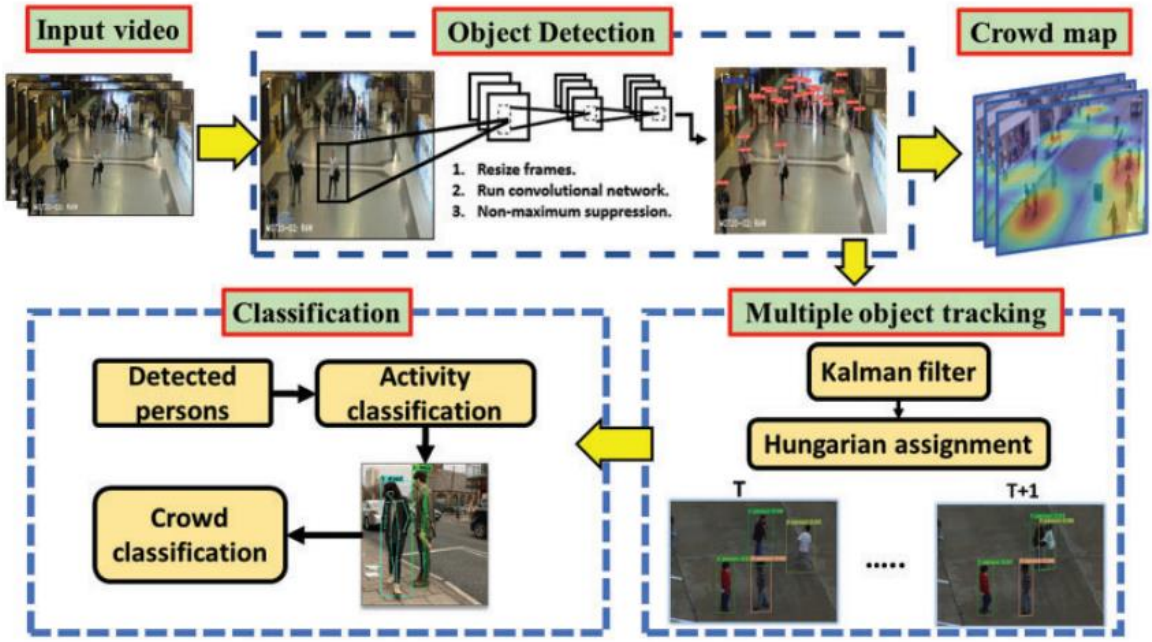

Figure 9. Crowd Activity Analysis Block Diagram

2.9.2.1 Object Recognition of people

2.9.2.1.1 Pre-processing of Human detection

Pre-processing the human detector provides real-time model input. Datasets are randomly partitioned into training, validation, and test sets by hand. The input video feed must be framed. Resize and smooth each frame separately. Normalizing the original frame speeds up detection without losing frame data. Identifying everyone, especially in a crowd, improves performance accuracy. It is done for computing parameters, implementation, and detection. After processing, the YOLO model outputs. After classifying the detection model's anchor box size, pre-processing would finish.

2.9.2.1.2 YOLOv5-S Model Architecture

YOLO uses neural networks to recognise and detect items (here individuals). It has outperformed the prior detection method due to its better performance and 7MB memory use. YOLO uses regression to detect humans and provide bounding boxes and class probabilities. YOLOv5-S, one of the YOLO algorithms, contains three primary components that distinguish it. The CNN backbone model generates key features from pictures using a Cross Stage Partial network (CSP) and a Focus interlaced sampling splicing structure. CSPNET addresses recurring gradient information in large-scale backbones. Feature maps included gradient changes. Lowering model parameters and FLOPS (floating-point operations per second) minimises computations and model size, improving inference speed and accuracy. The Neck model creates feature pyramids. Feature pyramids improve model scaling generalisation. It helps identify objects in varied sizes and scales.

PANET and SPP form YOLOv5's neck model. It increases lower-layer accurate location signals and information flow, boosting object location accuracy. Subsampling reduces the spatial scale of convolved features in this version. Reducing dimensionality reduces data processing computation. The head of YOLOv5, the YOLO layer, completes detection. A multi-scale prediction of three feature maps lets the model handle tiny, medium, and large objects. Generalised intersection over union (GIoU-loss) increases the overlap between the predicted bounding box of the detected object and the ground truth. SGD was used. ADAM is the adaptive learning rate. The final network layer uses SoftMax to describe class usage. As shown in Fig, the training stage gets transformed detection stage findings.

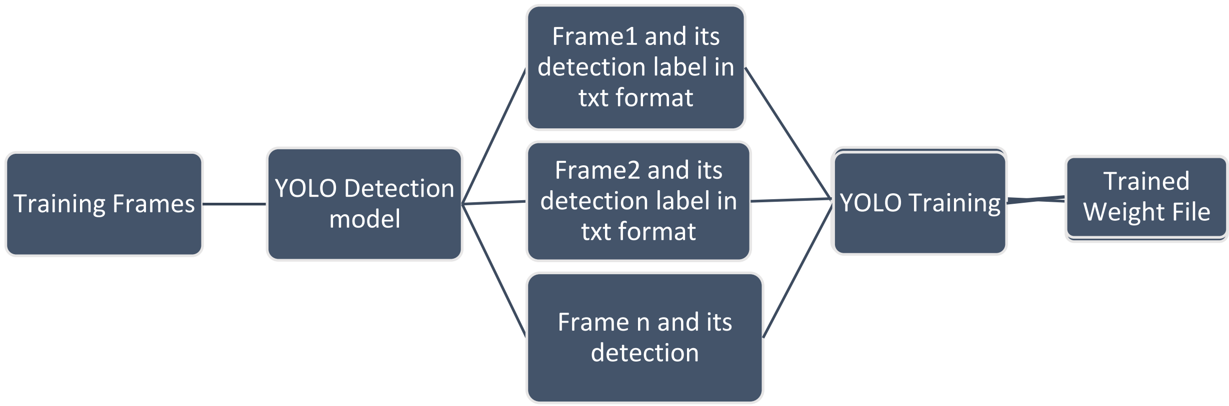

Figure 10. The flowchart for the training of YOLOv5's

Once the Yolov5 original model has been applied, we have provided a model with a special additional layer to increase efficiency.

To enhance object detection, particularly when dealing with occlusion issues, we will add a new scale that included specific convolutional layers with activation functions.

2.9.2.1.3 Crowd Counting

In order to determine each person's representation in the crowd, detection-based approaches are used to count the crowd. These techniques consider the crowd as a separate group of individuals who can identify specific individuals by creating bounding boxes and then counting those bounding boxes. Based on the total number of bounding boxes in each frame, crowd size is calculated.

2.9.2.2 Multiple Individual Tracking

Following detection, people are tracked using bounding boxes and given individual identifiers. tracking individuals listed in Algorithm 1 using SORT tracking as a framework for Kalman filtering and Hungarian optimization. The estimation for the current state at time t is calculated using a recursive estimator known as the Kalman filter at time t + 1. The estimated state from the previous time step and the current measurement are all that are needed. As a result, even if the human's vision is obscured, the human's location can still be determined. by determining whether a person in the current frame is the same person previously detected.

2.9.3 System Block Diagram (Number Plate Detection)

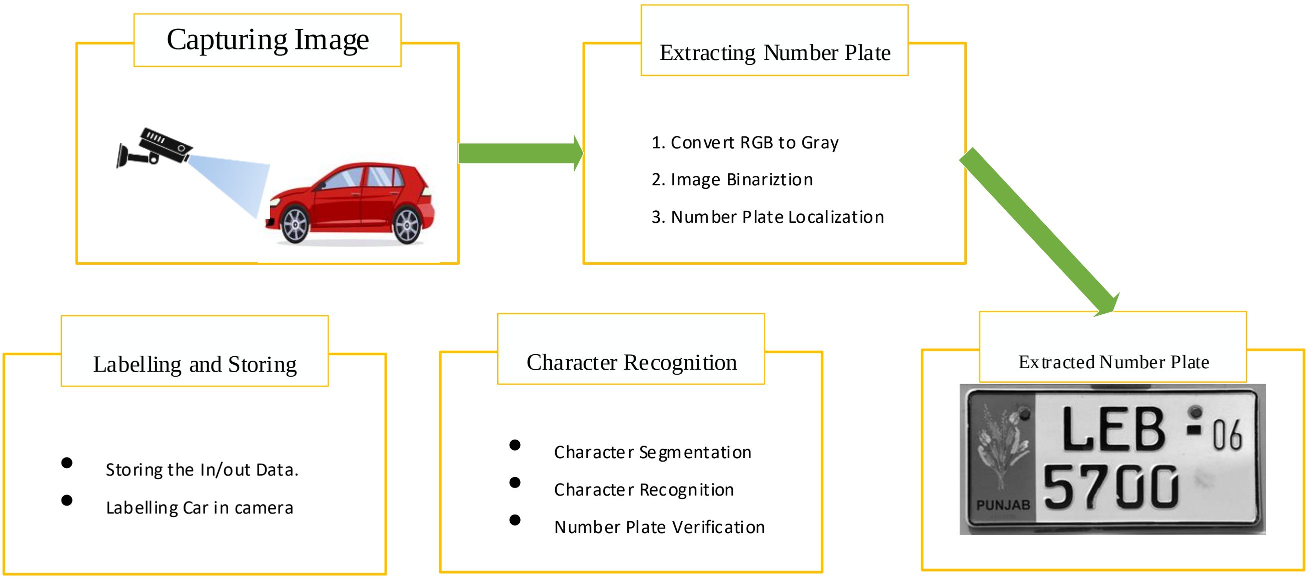

Figure 11. Number Plate Recognition Block Diagram

We'll create an effective method for removing the license plate that may be utilized to identify cars in traffic, find lost vehicles, and also for parking arrangement systems. The photo is taken from a fixed angle parallel to the horizon because the car should be immobile. Number plate characters are recognised using alphanumeric characters.

We presented a rapid method for automobile license detection (CLPD) and specifically highlight Character division, optical character recognition (OCR), and format coordination as three major advancements. To recognize the car license plate, use the ANPR (Automatic Number Plate Recognition) framework. The vehicle's image would first be captured by the framework when it approached the checking area. The division process is then used to extract the captured images. Utilizing optical character identification, the letters are recognized. This is a component of the security system that is frequently used.

2.9.3.1 Vehicle image captured by the camera

With the help of a high-tech 13-megapixel camera, the image of the car whose license plate has to be identified is captured.

2.9.3.2 Extraction of Number Plate

In this process, the color image of a vehicle's license plate is converted to a Grayscale.

Here, Sobel edge operations are employed to determine the edge boundary in addition to scientific morphology to discover the region. We then get a dilated image as a result. To obtain a suitable binary image, the infill function is then used to fill in the blanks.

2.9.3.3 Segmentation and Extraction of Plate Character.

The bounding box technique is used to segment the characters. It is used to evaluate the image's qualities in the targeted area. Recognizing the characteristics and dimensions of the number plate is a crucial first step in number plate recognition. Here, the retrieved image is multiplied by the grayscale image so that we only get the region of the vehicle's number plate.

2.9.4 System Block Diagram (Face Recognition)

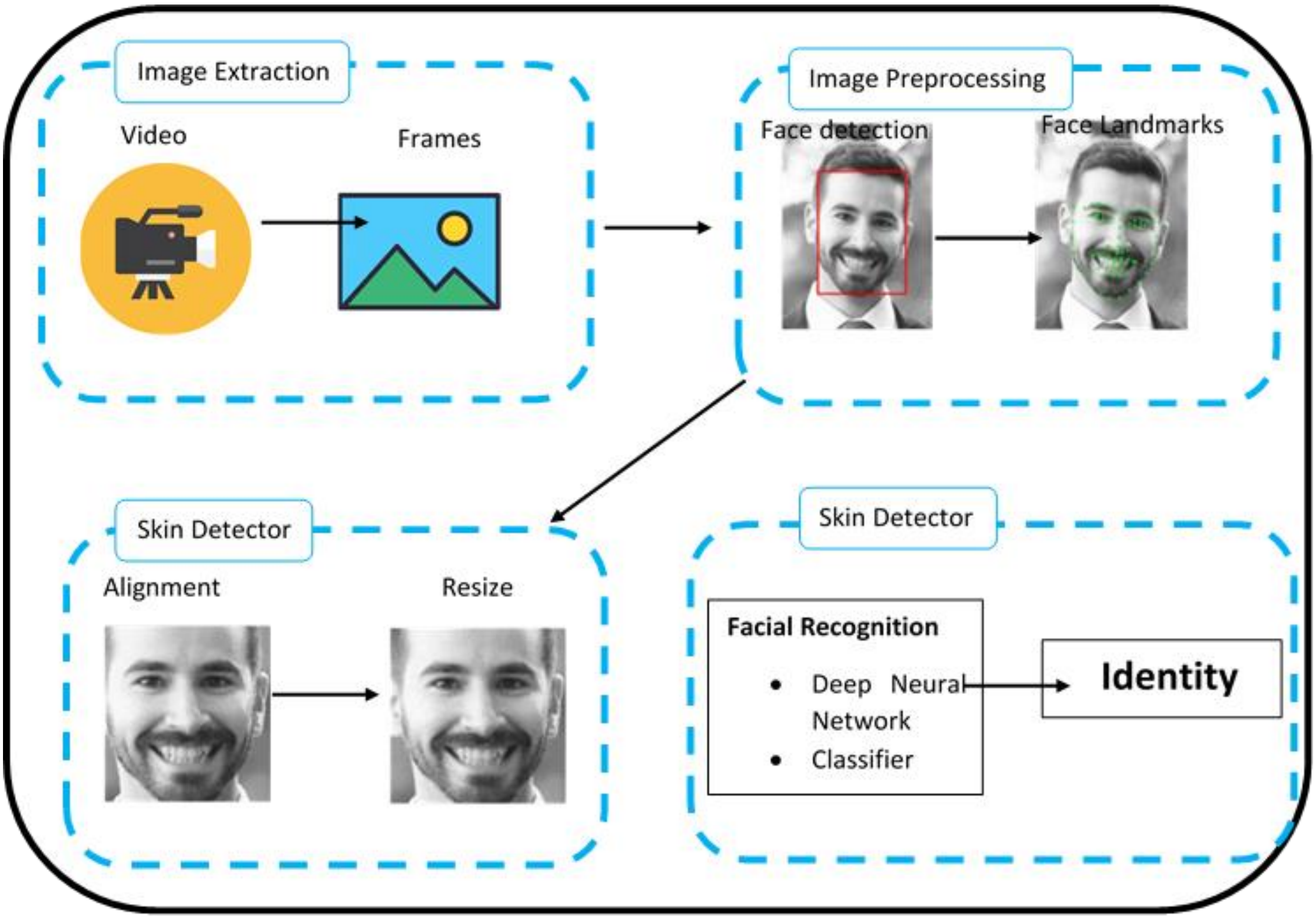

Figure 12. Face Recognition Block Diagram

2.9.4.1 Face Detection and Feature Extraction

The acquisition of multispectral images is the first step in the suggested procedure. These images can be obtained using a number of mono-spectral cameras, each of which captures an image of a specific spectral band, or by imaging equipment that can capture images at different spectral intervals. At this point, the only requirement is that the images be taken at the same time in order for them to represent the same person and have the same lighting and pose.

We will convert the image to grayscale and extract the facial landmarks in order to recognize human faces (i.e., eyes, nose, ear, etc.). The module that completes this task is shown as the image processing module in block diagram.

The presentation attack detector module is a component of the proposed facial recognition system that is responsible for spotting and alerting users to potential presentation attacks. In order to perform skin detection, this module makes use of all available multispectral images, shielding the facial recognition system from any potential presentation attacks.

Facial landmarks discovered in the image processing module are used to align the face in the following module, called facial processing. The eyes' horizontal landmarks on the face are horizontally aligned thanks to the image's rotation. The image is then scaled down to 144 144 pixels. Before presenting the image to the DCNN, the main goal of this module is to normalize it.

The facial recognition module then performs the person identification after receiving the normalized images.

2.9.4.2 Facial Recognition

Two main elements make up the facial recognition module. a classifier and a DCNN. The DCNN is used to extract embeddings that are representative of the person who needs to be identified. The identity of the person will then be ascertained by the classifier using these embeddings.

This architecture enables the use of multiple (N) channels by assigning a spectral band, or spectral range, to each channel (if several spectral ranges are being used in the same band). Each channel can also use a backbone DCNN that has been trained on a sizable dataset of visible-light facial images. This enables the use of transfer learning techniques, i.e., adapting the corresponding backbone DCNN to the spectral properties of that channel using a small dataset of labeled images for each channel.

The LightCNN serves as the foundational DCNN in this work. This DCNN differs from other comparable DCNNs in that it uses Max-Feature Map (MFM), an extension of the Maxout activation function, in place of the rectified linear unit (ReLU) activation function in its base architecture. The number of parameters obtained by LightCNN through this activation function is decreased. This network produces embeddings with 256 dimensions that are indicative of the identity of the person from 128 x 128-pixel greyscale images as its input.

To adapt the model to the spectral bands of the corresponding channel, different layers in the LightCNN are modified. The DCNN is not altered if the channel is assigned to the visible spectral band because the domain of use is the same as the domain in which the DCNN was pre-trained. The transfer learning methodology is used for the other channels. Given the limited number of training multispectral images in most existing datasets, reusing the weights of a pre-trained DCNN in a face recognition database with a high number of facial images is a way to avoid overfitting the training in a new domain.

At the network's end, a Fully Connected Layer (FCL) is added to combine all channels' classifications. The vector of size N 256 that is produced by concatenating the various channel embeddings is fed into this layer. The classifier block will use the final 256-d embedding produced by the FCL to identify the input face.

2.10Decomposition Description.

2.10.1Sequence Diagram.

2.10.1.1 Video Input.

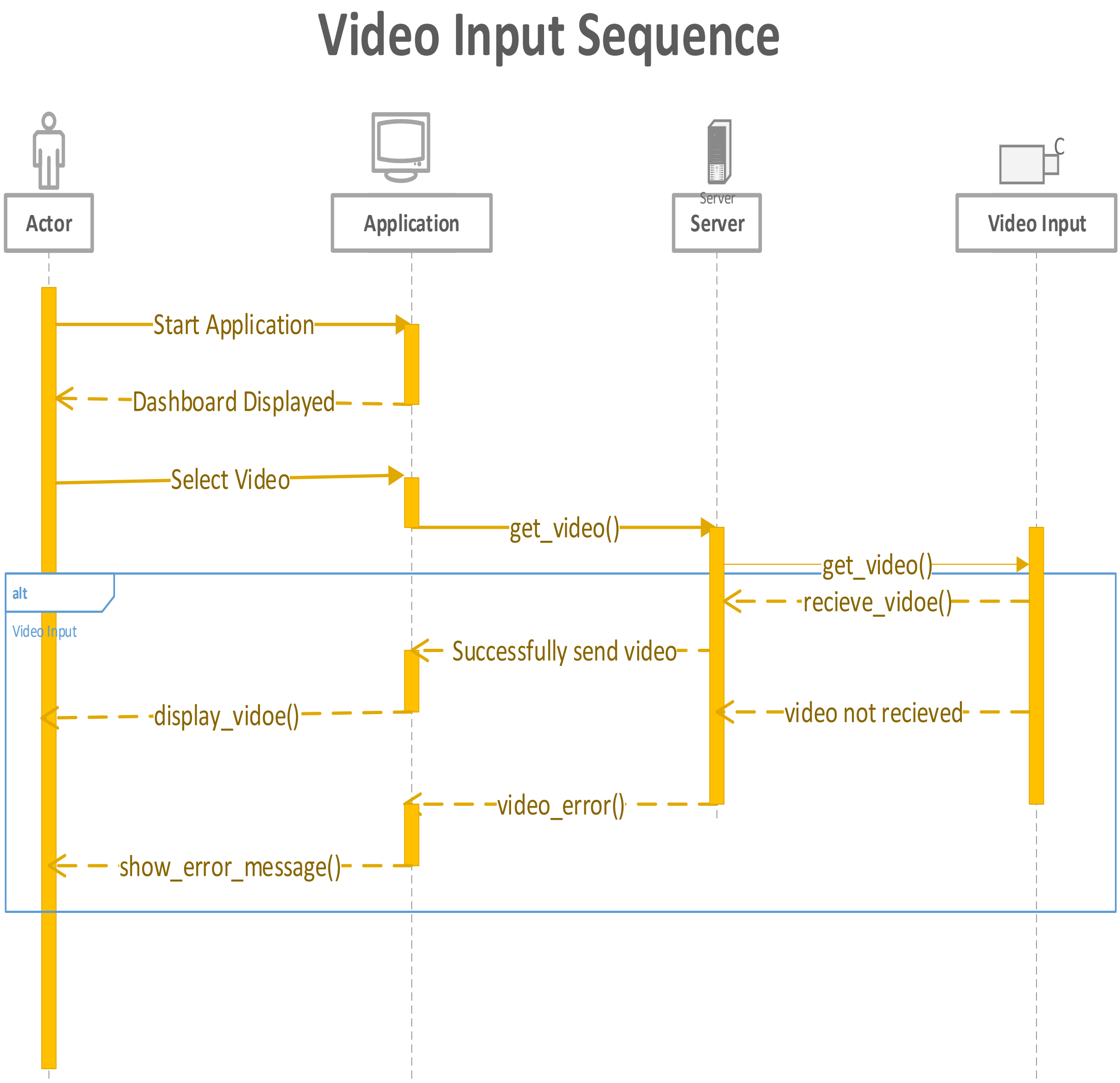

Figure 13. Video Input Sequence Diagram

2.10.1.2 Facial Recognition Attendance

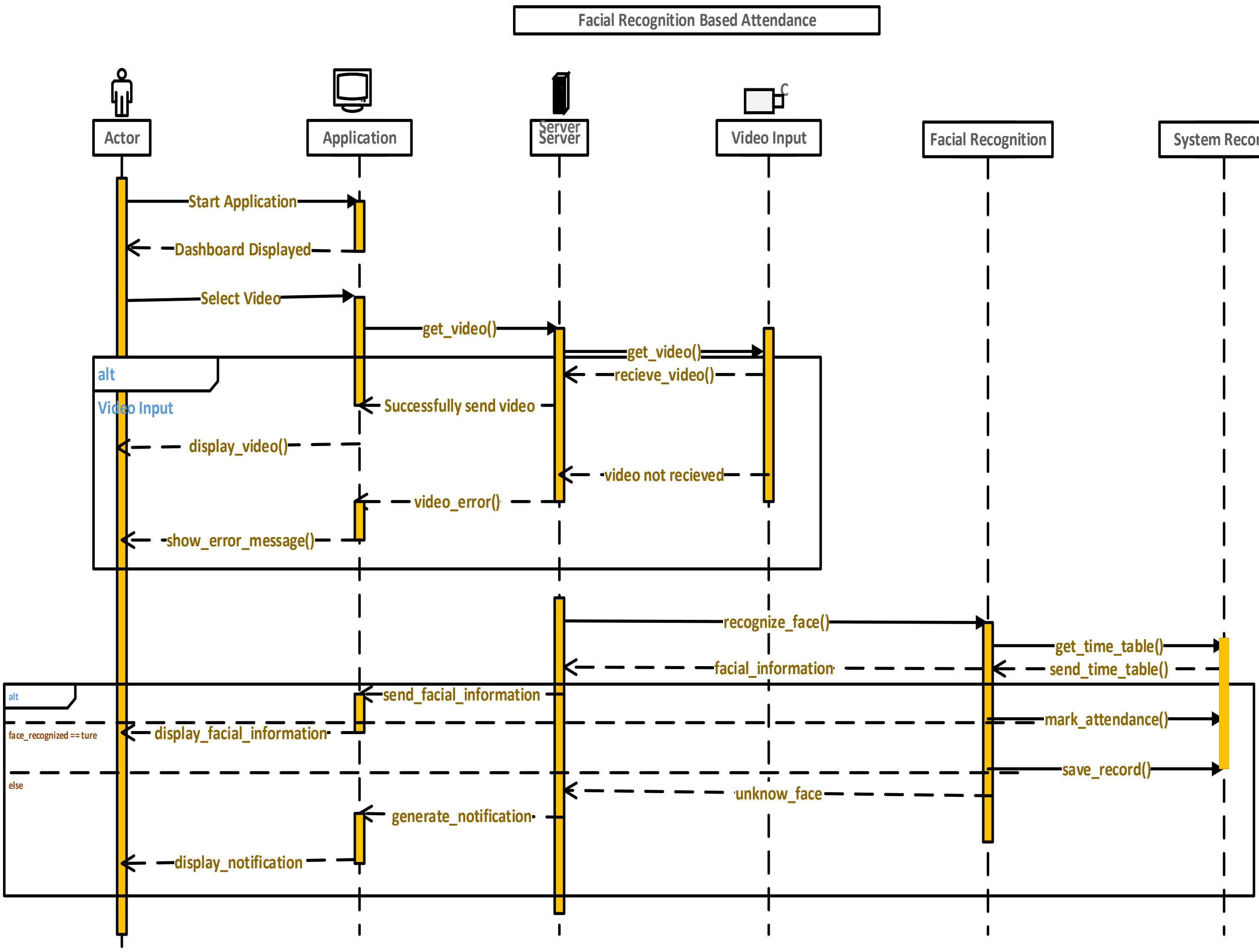

Figure 14. Facial Recognition Sequence Diagram

2.10.1.3 Car Parking System

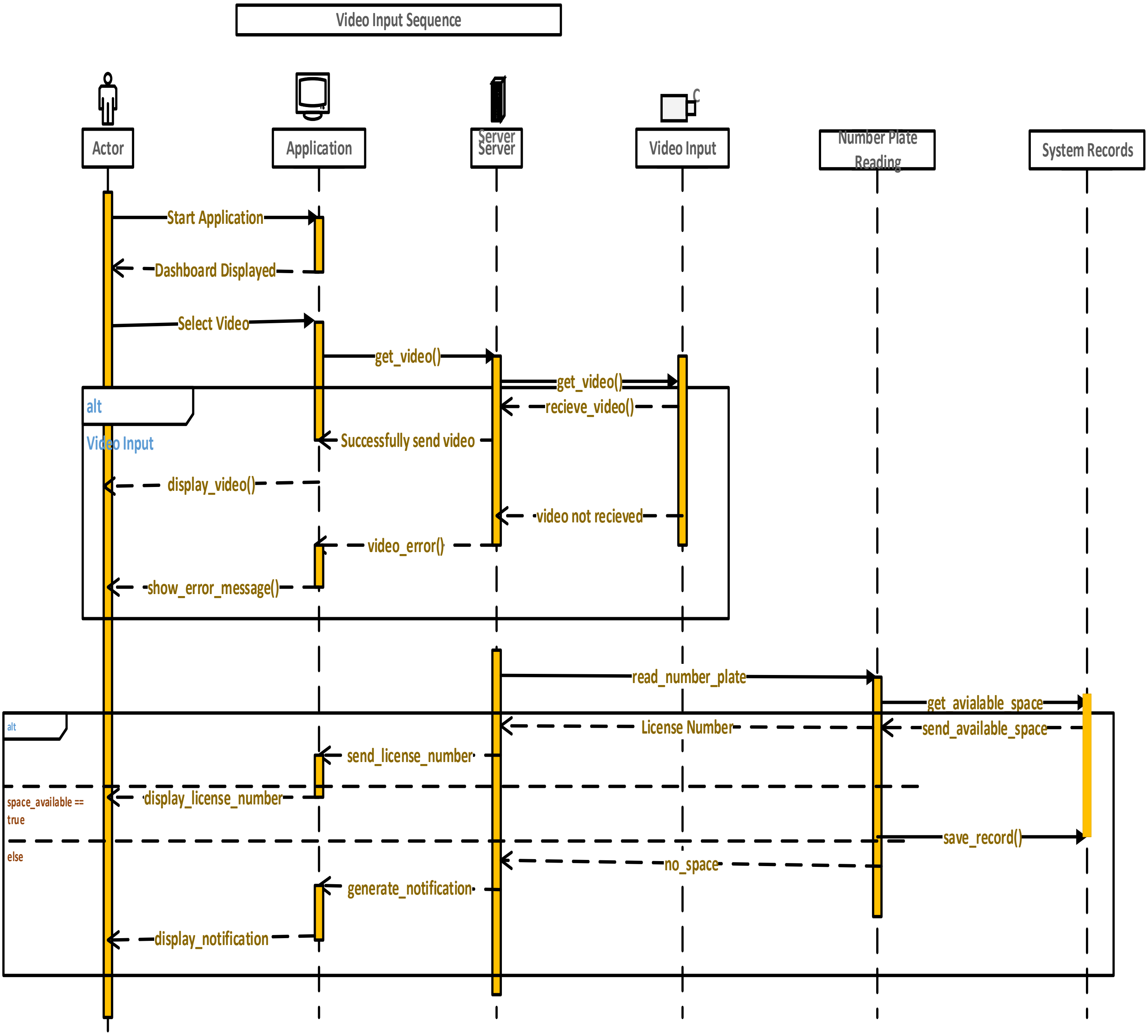

Figure 15. Car Parking System Sequence Diagram

2.10.1.4 Crowd Activity Analysis (Activity Diagram)

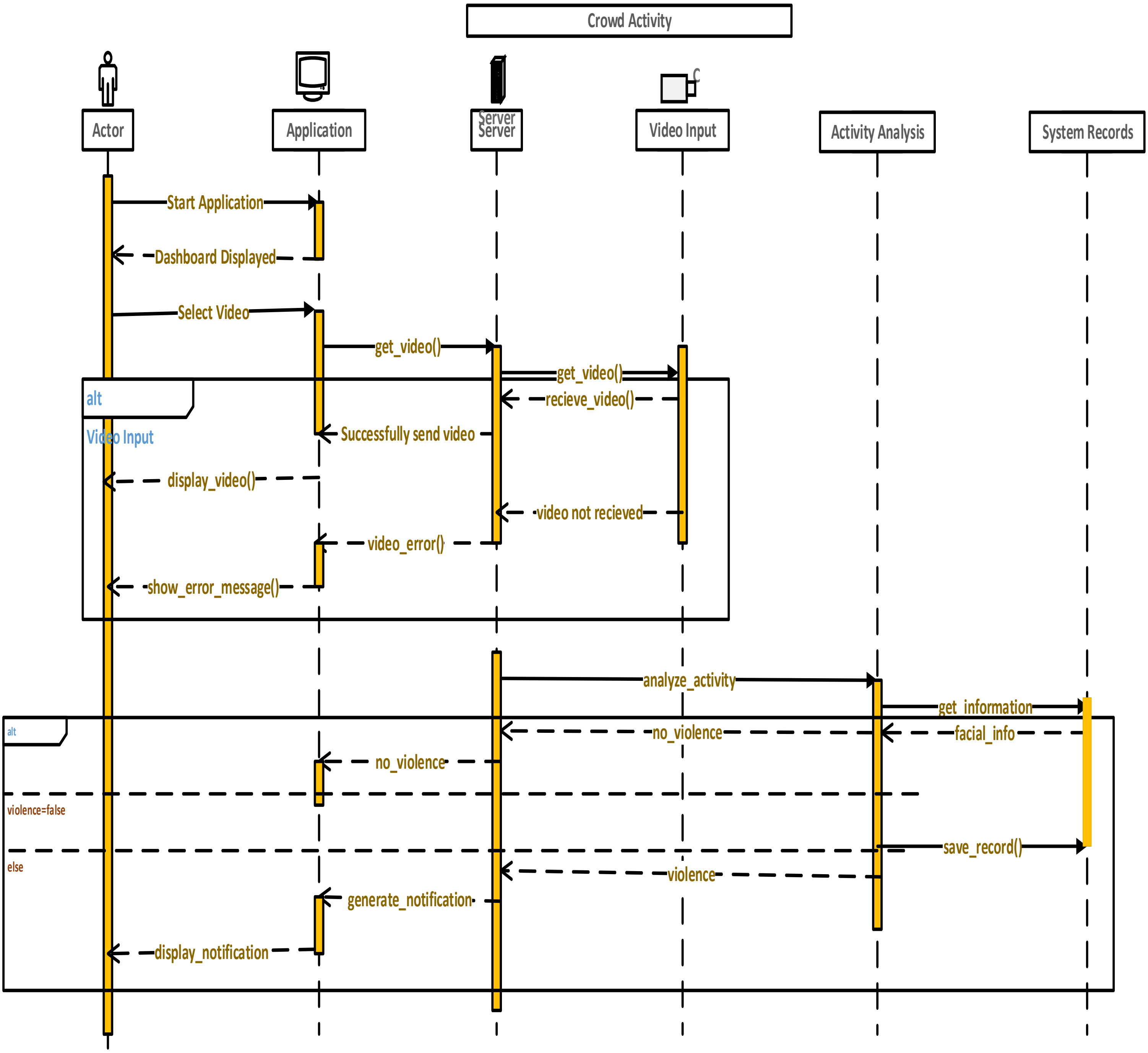

Figure 16. Crowd Activity Analysis Sequence Diagram

2.10.1.5 Sleep and Smoke Detection.

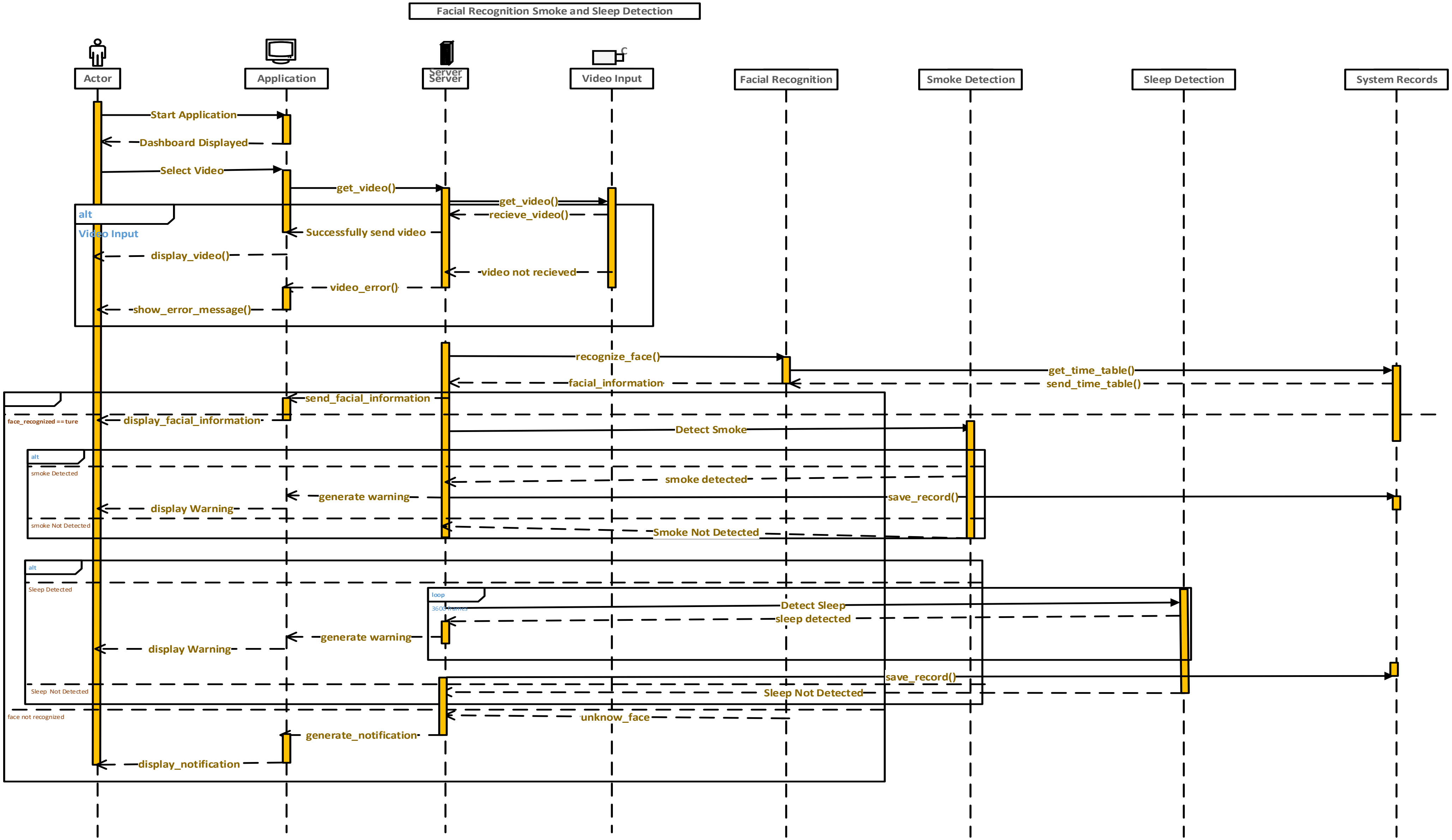

Figure 17. Smoke and Sleep Detection Sequence Diagram

2.10.1.6 Login Activity Diagram

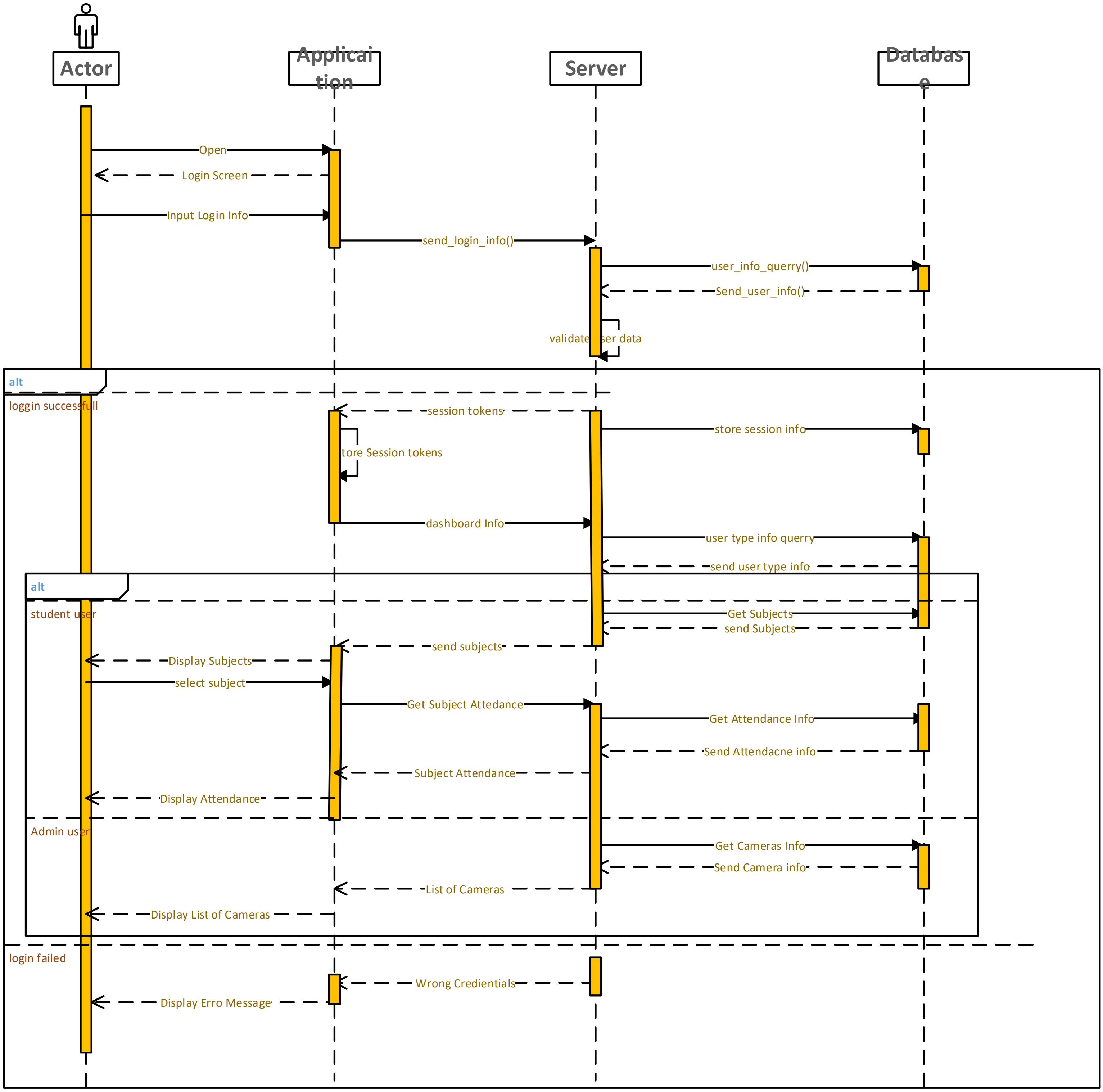

Figure 18. Login Authentication Sequence Diagram

2.10.1.7 Register New Faces.

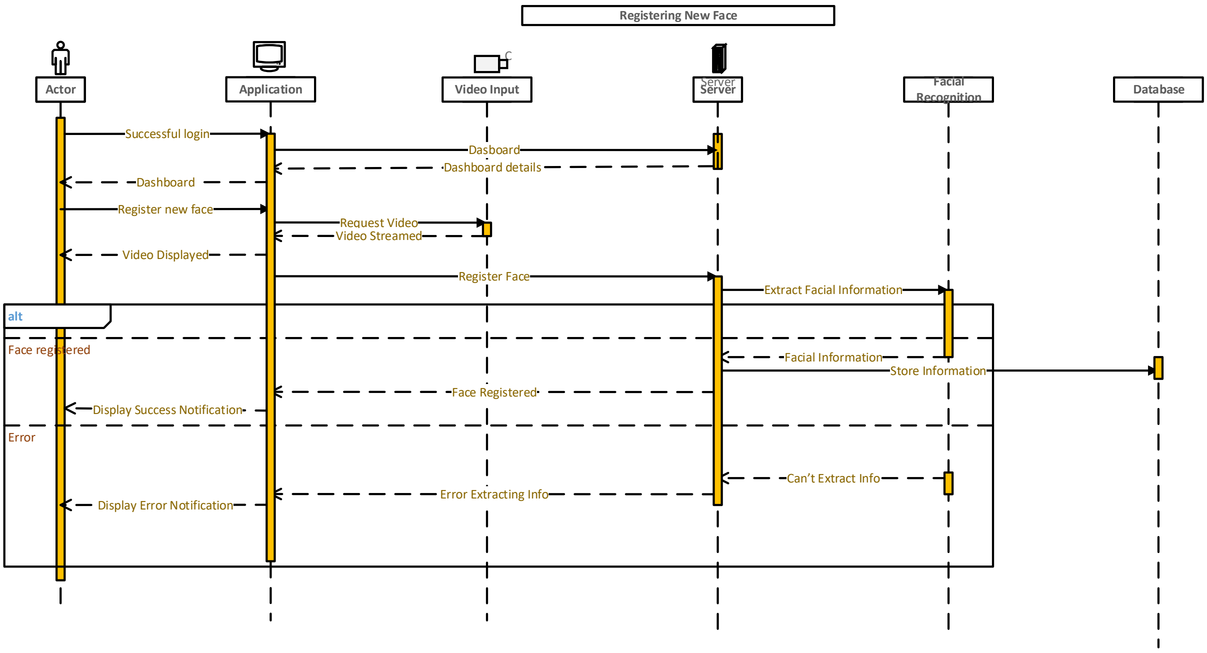

Figure 19. New Face Registration Sequence Diagram

2.10.2Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

2.10.2.1 System Activity Diagram.

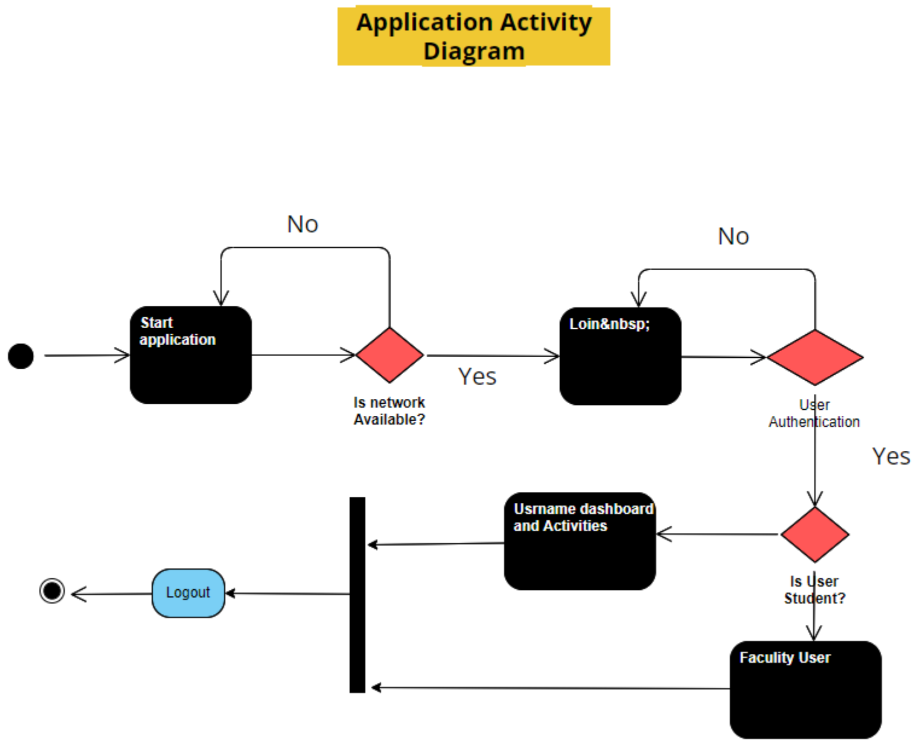

Figure 20. System Activity Diagram

2.10.2.2 *Facial Recognition based Attendance and Teacher Presence.*

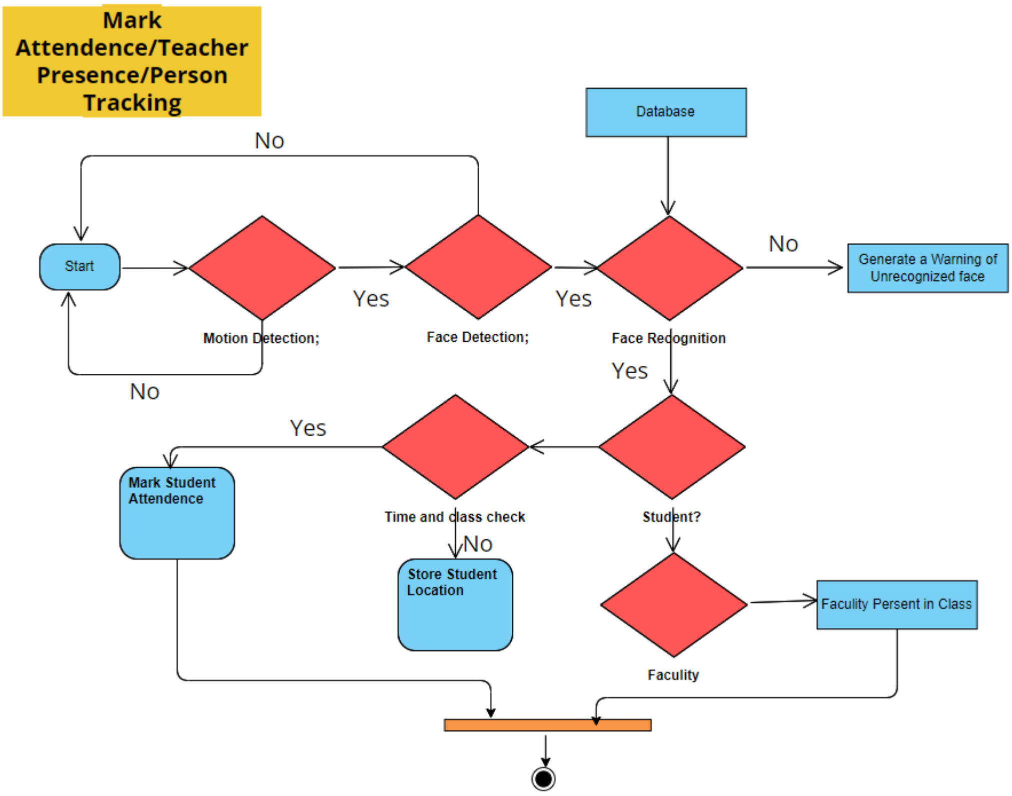

Figure 21. Facial Recognition Based Attendance Activity Diagram

2.10.2.3 Admin Activity Diagram

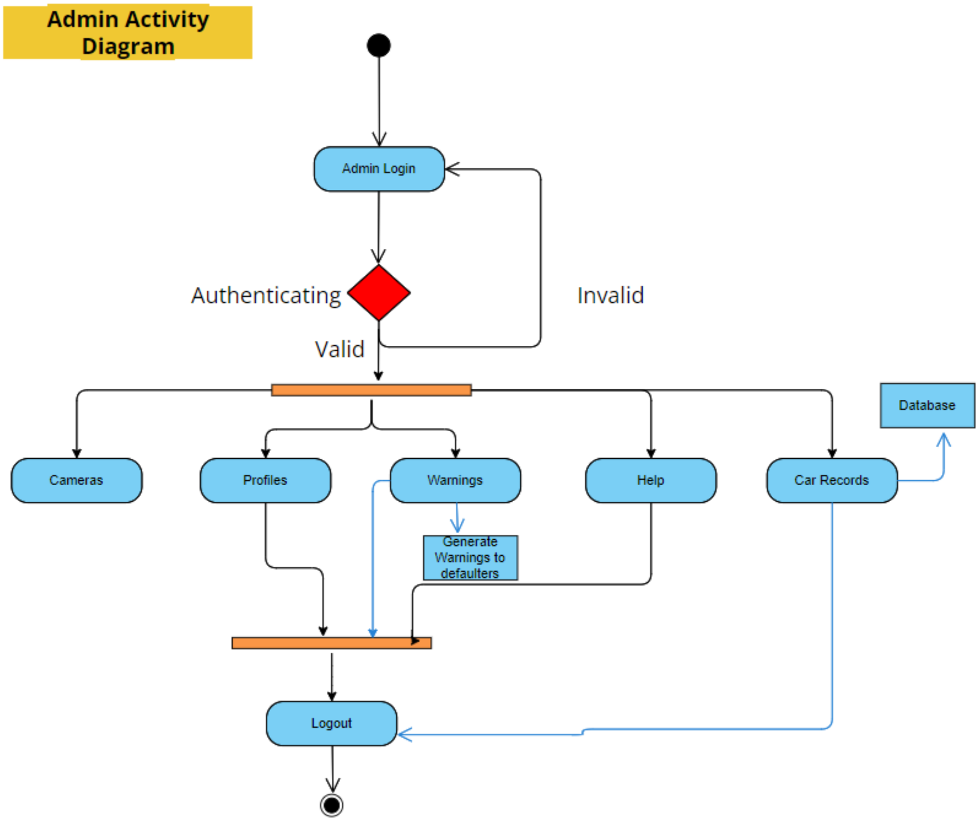

Figure 22. Admin Activity Diagram

2.10.2.4 Server-Side Activity Diagram.

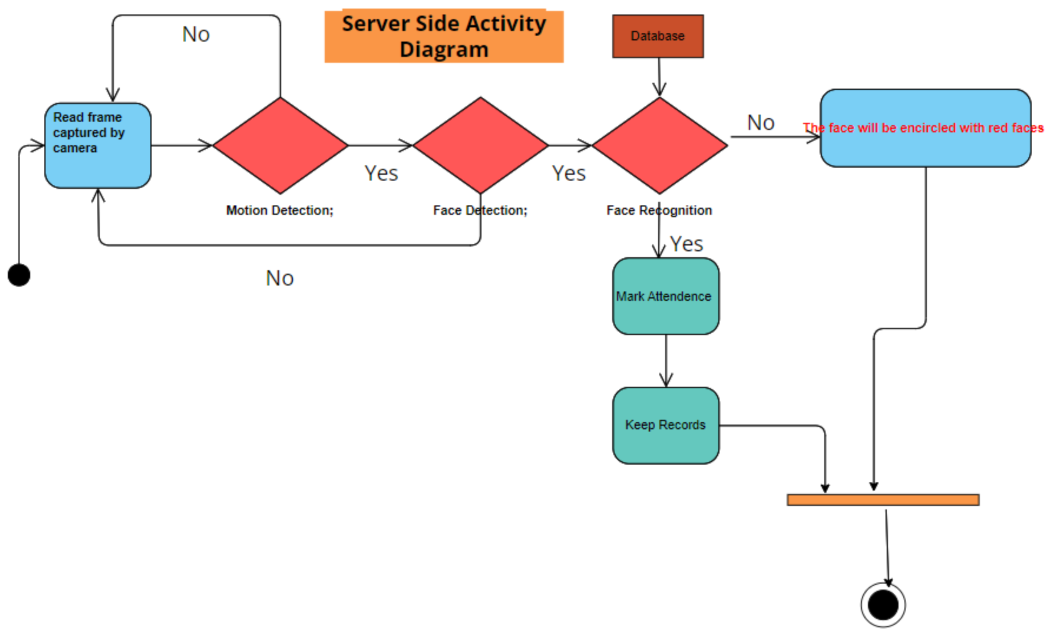

Figure 23.Server Side Activity Diagram

2.10.2.5 Sleep Detection Activity Diagram

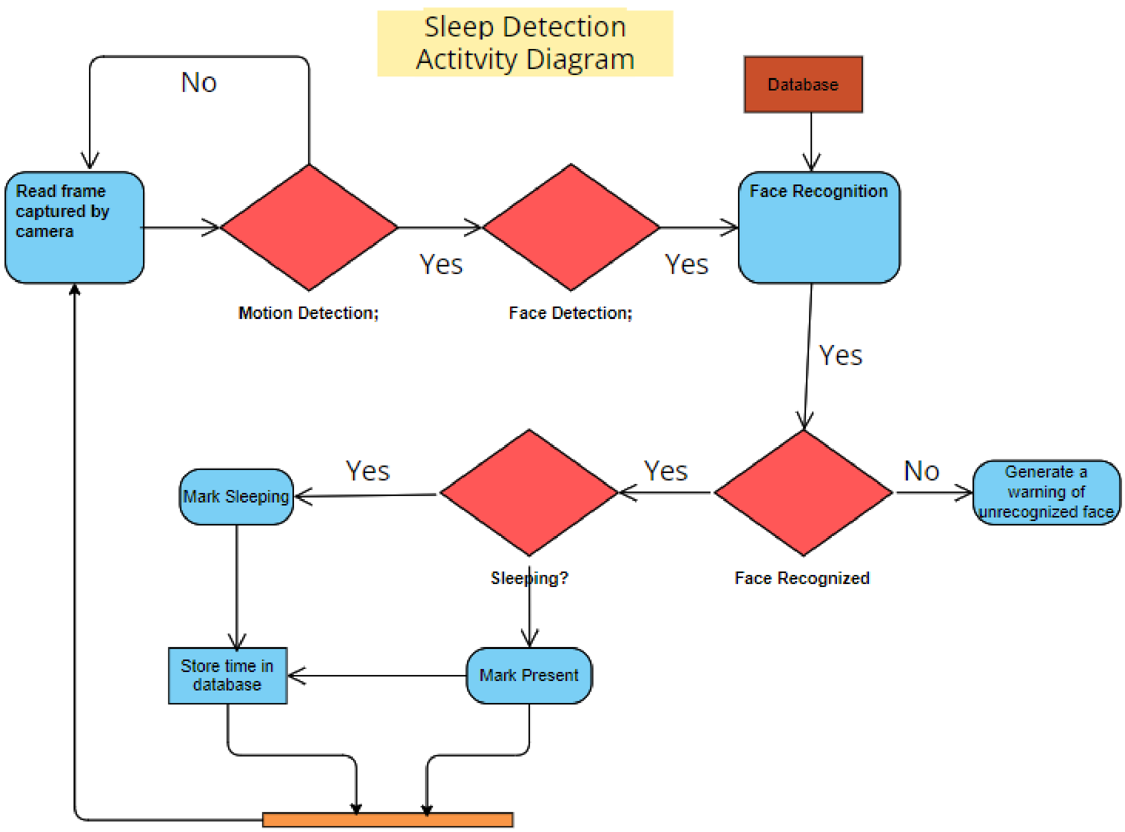

Figure 24. Sleep Detection Activity Diagram

2.10.2.6 Smoke Detection Activity Diagram

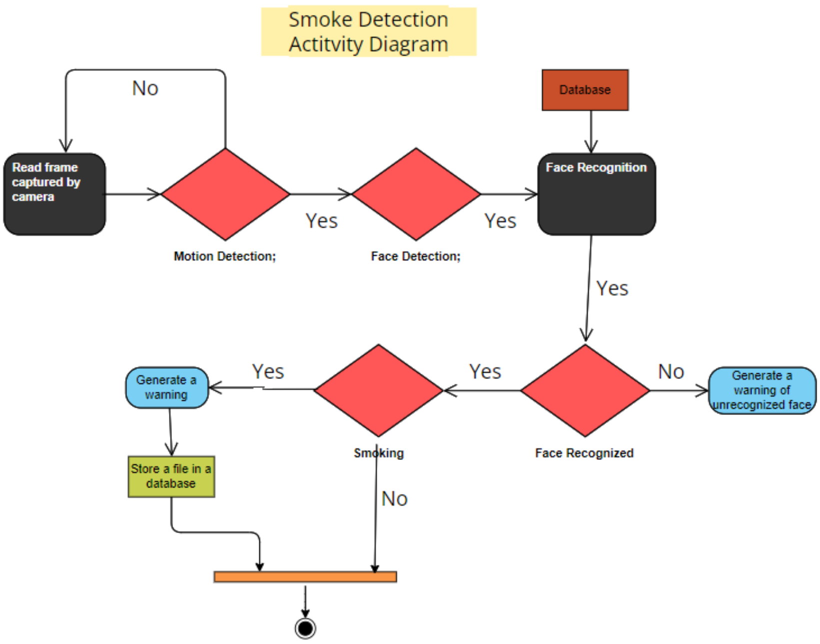

Figure 25. Smoke Detection Activity Diagram

2.10.2.7 People Tracking Activity Diagram

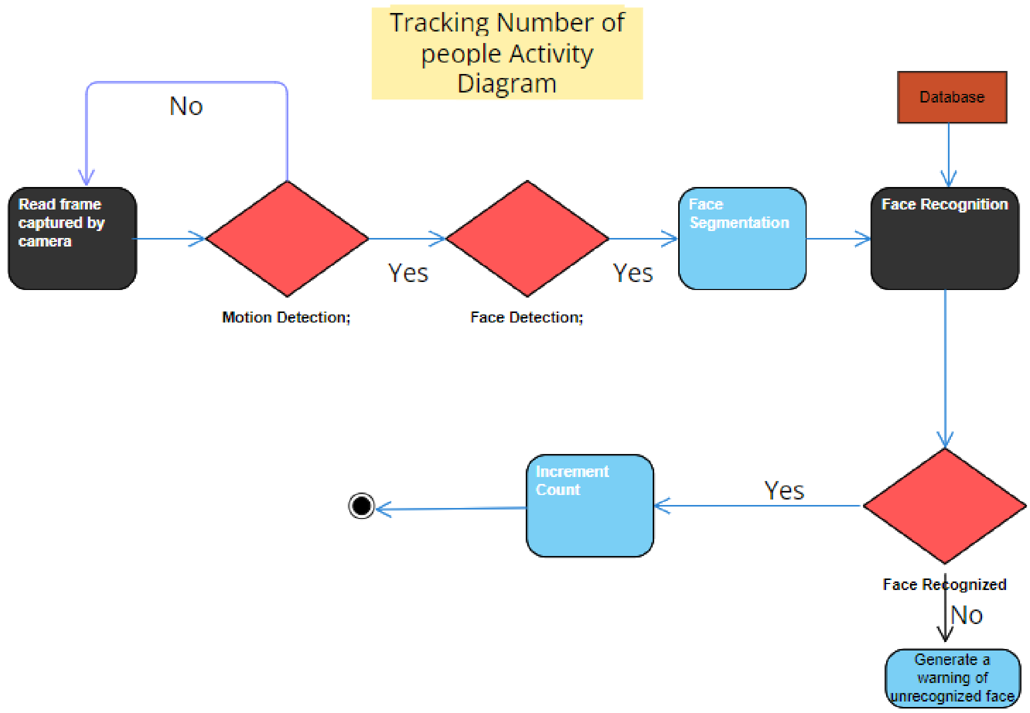

Figure 26. People Tracking Activity Diagram

2.10.2.8 Parking Management Activity Diagram

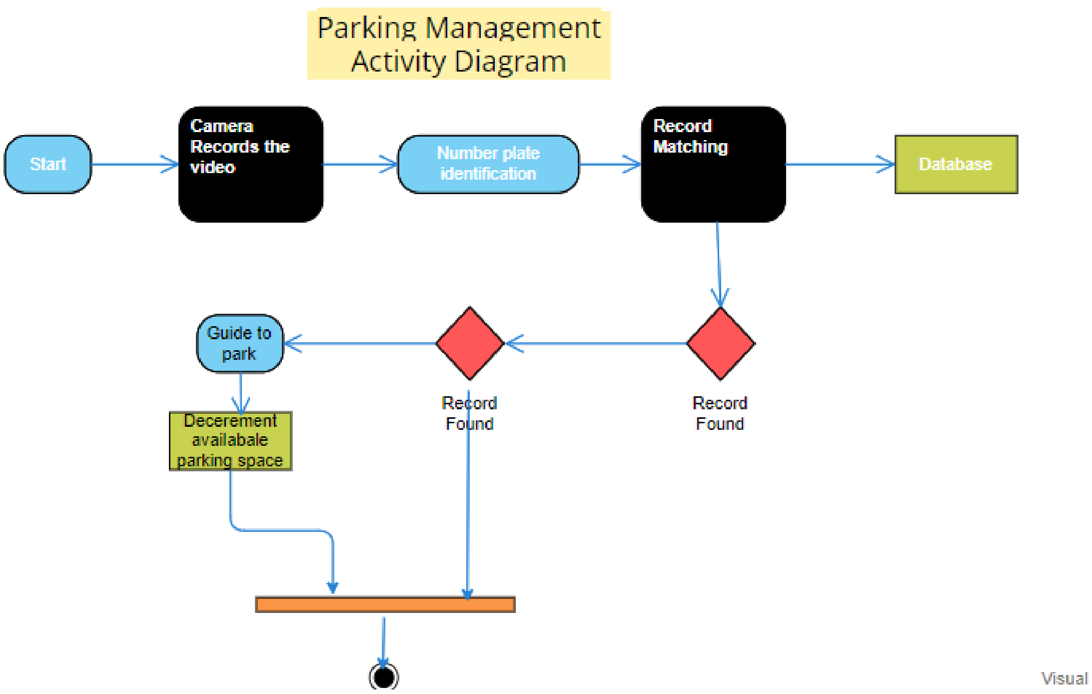

Figure 27. Parking Management Activity Diagram

2.10.3Logical View (State Diagram)

Following are the state diagrams of SMART Campus, showing all the states the system will have during the course of action.

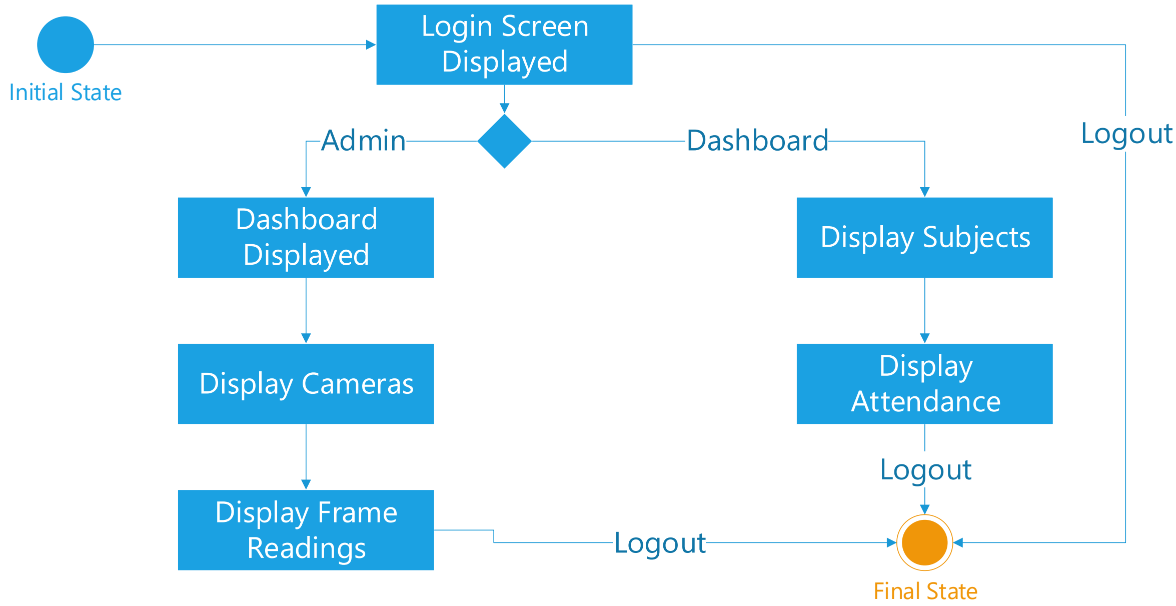

Figure 28. Application State Diagram

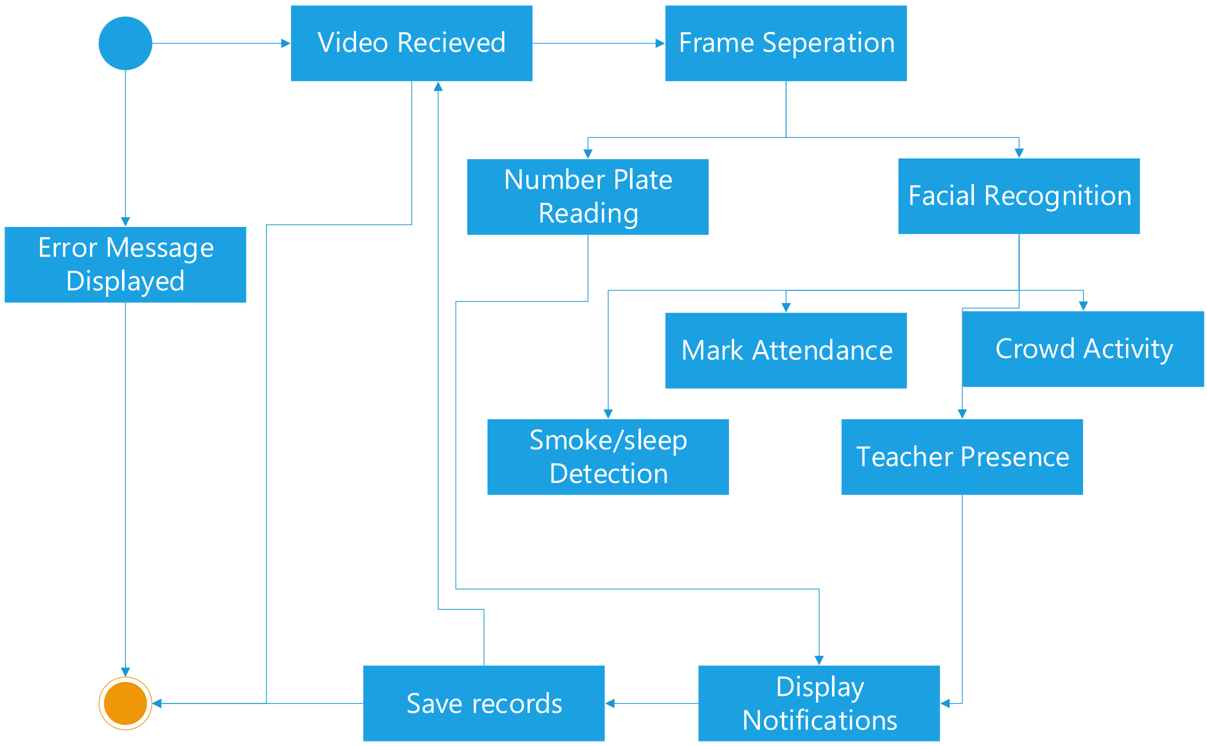

Figure 29. Server State Diagram

2.10.4Structure Chart.

The structure diagram illustrates the components of the SMART Campus system. These elements covered in detail in Section 5 (Component Design).

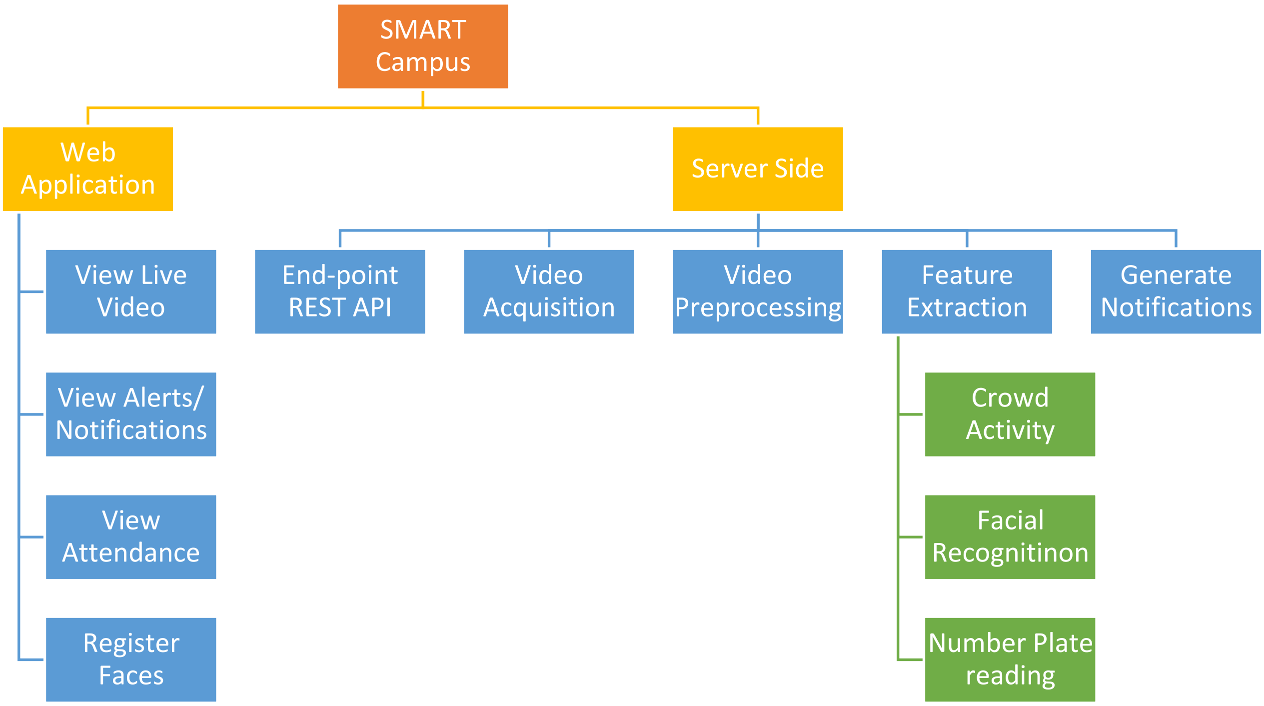

Figure 30. Structure Chart

2.11Design Rationale

SMART Campus is an interactive program that delivers user service via a web application. Each component has been tasked with performing a specific function. The two primary modules are web application and server-side. Web application is responsible for user interaction, whereas server-side is responsible for the back-end.

Therefore, Model View Control (MVC) is an appropriate design pattern to utilise. Model is essentially the machine learning model that accepts a video as input and outputs its classification. It contains parameters that specify the structure of the prediction model, as well as model-specific functions such as model construction and data preparation.

View is used mostly to respond to the browser's request. The Controller is the primary component that links the browser to the model and the model to the view.

Clearly, components can do their duties separately, but a specific flow is required for optimal performance (data as well as control). This resulted in high cohesiveness.

In addition, there is minimal interaction between components; after a component has done its task, it will communicate its state to the other component. As a result, this component will be activated. This resulted in low coupling.

2.12Data Design.

2.12.1Data Description.

The training process of SMART Campus requires an extensive amount of videos for each model to recognize images, number plate reading, crowd activity analysis. First we have to categorize videos and than split the datasets into training, testing and validation sets with ration of 60.10.30. We have collected these videos from multiple online repositories, below is the summary of resources.

Dataset	Details	Source
Tuftus face databse	over 10,000 images, 74 females + 38 males, from more than 15 countries with an age range between 4 to 70 years old	https.//www.kaggle.com/datasets/kpvisionlab/tufts-face-database
UTK Face Database	consists of 20k+ face images in the wild (only single face in one image)	https.//susanqq.github.io/UTKFace/
Google Facial Expression Dataset	includes 500K triplets and 156K face images.	https.//www.kaggle.com/datasets/mahmoudima/mma-facial-expression
Crowd 11	over 6000 video sequences with an average length of 100 frames per sequence	https.//paperswithcode.com/dataset/crowd11
Car Number plate reading	Contains 931 images of, the front and rear side of the car	https.//www.kaggle.com/datasets/elysian01/car-number-plate-detection
Car Number Plate Reading	Contains 100,000 images of cars	https.//www.kaggle.com/code/aslanahmedov/automatic-number-plate-recognition/data

2.12.2Data Dictionary.

2.12.2.1 Server-side.

1. extract_frames(fps, vid.video).images => this function is a part of the **Frame_Extraction** class and is used to divide the video into video based on fps parameter.

2. get_features(frames.images).string => this function is a part of the **Feature_Extraction** class and is used to extract the required features from the frames.
3. recognize _face(featuredata.frames).video=> this function is a part of the **Facial Recognition** and it feeds the feature data into the model which processes it and classifies the face.
4. analyze_activity(featuredata.frames).video=>this function will analyze activity such violence detection, number of people etc.
5. recognize_number_plate(featuredata.frames).video=> this function will detect cars and will read their number plates at entrance and exit.
6. notify(processedvideo.video).string, video => this function takes in the processed video and returns the relevant notification and resultant video.
7. rcvvideo().video => this function is a part of the **rest_api** class and is used to receive video from the application.
8. sendresults(notification, processedvideo). string => this function is part of the **rest_api** class and is used to transmit the results.

2.12.2.2 Application side.

1. main().void => this function is used to start the web application.
2. get_video(address.string).boolean => this function is a part of the Video_Input class which is basically tasked with taking in the user's input and returns true/false if video input is successful or not.
3. send_error_notification().string => the function returns an error notification in case video input fails.
4. rcv_notifcations().string, video => this function is a part of the Application class and it is tasked with receiving the results from the server.

2.12.2.3 Entity Relationship Diagram.

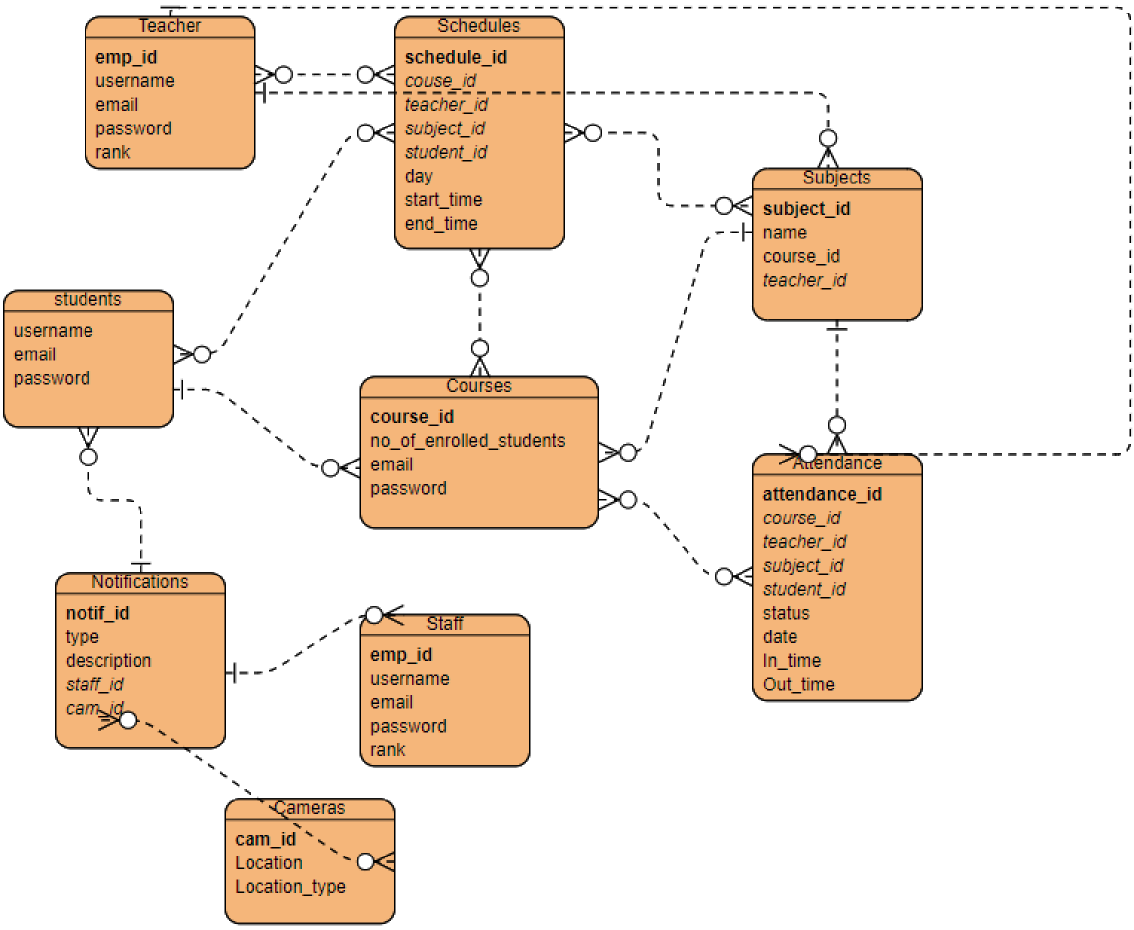

Figure 31. ER Diagram

2.13Component Diagram.

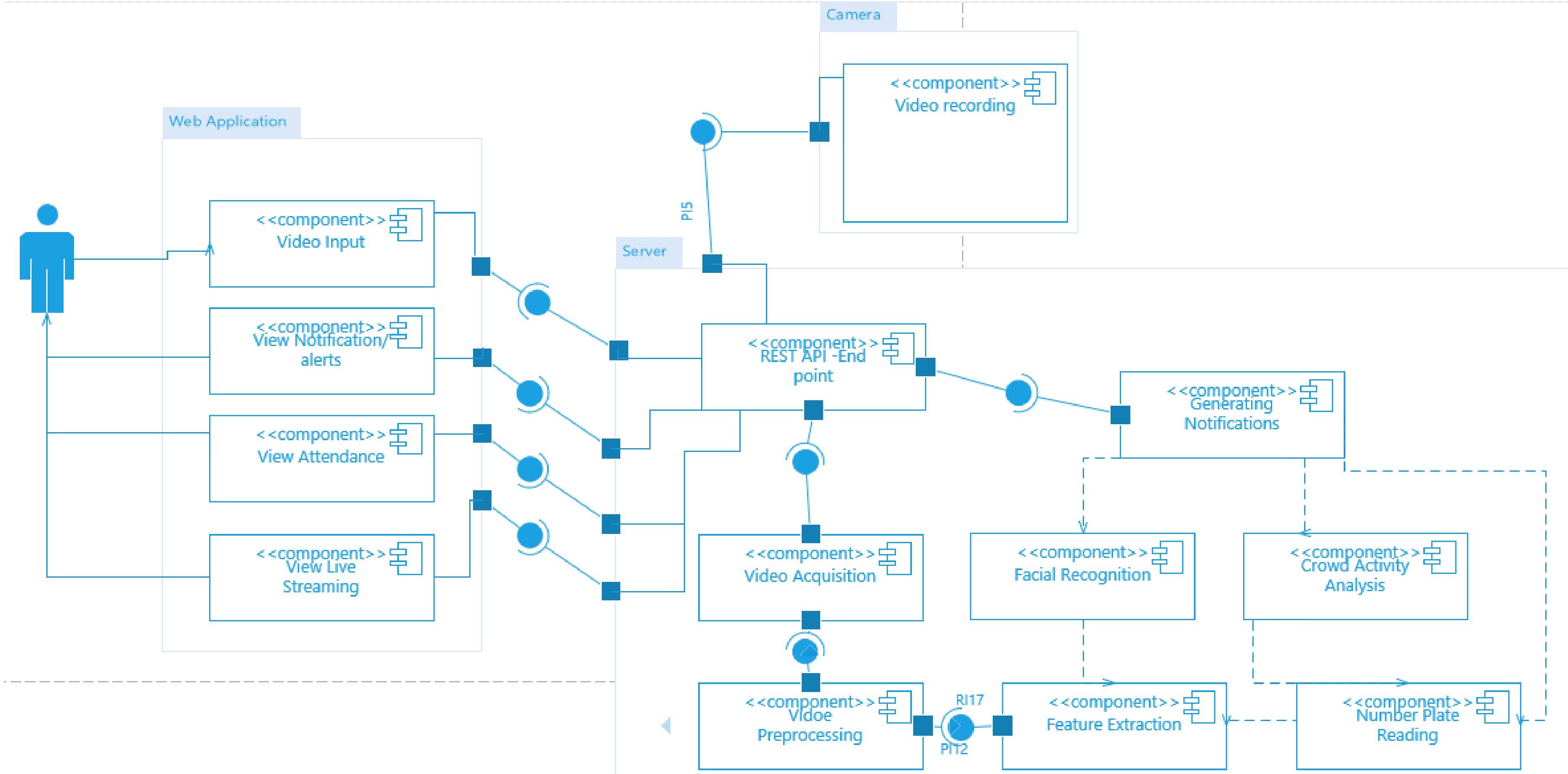

Figure 32. Component Diagram

2.13.1Web Application Module.

This module performs all the front-end tasks for Smart Campus System which includes student Registration, notification alerts/results and live processed video acquisition from server(admin/staff side) ,attendance view and notification view(student side). This module provides the base for successful working of the server module.

2.13.1.1 Staff/Admin Side.

2.13.1.1.1 Student Registration.

Identification	Name. Student Registration Location. Web Application Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Registration** **Requirement** The system shall be able to register new students.. **Description** This feature enables the system to register new students and add their images to database. This input will form the basis for student detection for attendance, sleep and smoke detection.
Function	This component of system interfaces with registration of students.
Subordinates	No subordinates
Dependencies	This component is independent module and runs in parallel for registration of new students.

Interfaces	No interfaces as registration process don't need to interface with any other module.
Resources	**Hardware.** Desktop Computer, LAN/WiFi connection. **Software.** Web Browser (Chrome, Firefox, IE), database for storing student`s images.
Processing	Images will further process after registration for attendance, sleep and smoking detection.
Data	This component uses students` data for fetching registration which includes name, registration number and images.

2.13.1.1.2 Video Acquisition from server

Identification	Name. Video Acquisition from server Location. Web Application Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Acquire video** **from server** **Requirement** The system shall be able to successfully fetch processed video from server. **Description**

	This feature will basically fetch processed video from server at backend. It is responsible for interacting with the REST API to fetch information.
Function	This component of system interfaces with the REST API to revceive information from the server.
Subordinates	None.
Dependencies	This component is dependent on.- Server for live time processed video Database images
Interfaces	None.
Resources	**Hardware.** Server, Monitoring Screen, LAN/WiFi connection. **Software.** Web Browser (Chrome, Firefox, IE), Database at backend.
Processing	video Acquisition component would be responsible for transmitting processed video to admin monitor.
Data	This component uses following information of the application. - Time of video acquisition from server.

2.13.1.1.3 View Notification Alerts/Results

Identification	Name. View Notification Alerts/Result. Location. Web Application Module
Type	Component

Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Display results** **View Notification** **alerts** **Requirement** The web application shall be able to receive the alerts/results from server and display them to admin/control panel. **Description** This feature is responsible for receiving the classification result and the video (after it has undergone detection) from the server and display them.
Function	This component of system interfaces with the REST API to receive the results from the server.
Subordinates	None.
Dependencies	This component is dependent on.- Video Acquisition Component.
Interfaces	None.
Resources	**Hardware.** monitoring screen, server, Desktop Computer, LAN/WiFi connection. **Software.** Web Browser (Chrome, Firefox, IE)
Processing	Display Alerts/Results component would be in charge of displaying the video classification note and videos.

Data	This component uses following information of the application. - Time on which results received.

2.13.1.1.4 Students side.

2.13.1.1.5 View Attendance

Identification	Name. View Attendance. Location. Web Application Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **View Attendance** **Requirement** The web application shall be able to receive the attendance from server and display them to students` desktop. **Description** This feature is responsible for receiving the attendance after processing from the server and display them.
Function	This component of system interfaces with the REST API to receive the attendance from the server.
Subordinates	None.
Dependencies	This component is dependent on.- Server processing after facial recognition.
Interfaces	None.
Resources	**Hardware.** Desktop Computer, LAN/WiFi connection. **Software.** Web Browser (Chrome, Firefox, IE)

Processing	Display attendance after server processing on the basis of facial recoginition.
Data	This component uses following information of the application. - Time to fetch attendance results from server.

2.13.1.1.6 View Notifications

Identification	Name. View Notification. Location. Web Application Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Show** **Notifications** **Requirement** The web application shall be able to receive the Notifications from server and display them to Students` Desktop. **Description** This feature is responsible for receiving Notification from the server and display them.
Function	This component of system interfaces with the REST API to receive the Notifications from the server.
Subordinates	None.
Dependencies	This component is dependent on.-

	Server Processing.
Interfaces	None.
Resources	**Hardware.** Desktop Computer, LAN/WiFi connection. **Software.** Web Browser (Chrome, Firefox, IE)
Processing	Display Notification component would be in charge of displaying notification.
Data	This component uses following information of the application. - Time to fetch notifications from server.

2.13.2Server-Side Module

This module performs all the back-end functionalities related to pre-processing of videos, feature extraction, classification, ML algorithms implementation and decision making. Feature extraction is the main input for Algorithm implementation component.

2.13.2.1 Endpoint – REST API

Identification	Name. Endpoint – REST API Location. Server Side Module
Type	Component

Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Endpoint – REST** **Requirement** This component is responsible for communicating to and fro with the web application. **Description** This component is the main contact point between the web application and the server-side. It receives the video from the cameras and send them to desktop after processing.
Function	This component of system interfaces with Video fetched and Display Results components of the application.
Subordinates	None.
Dependencies	None.
Interfaces	None.
Resources	**Hardware.** Server, LAN/WiFi connection. **Software.** Flask (Web microframework)
Processing	Endpoint – REST API component would receive videos from cameras and send results to the web application.
Data	This component uses following information of the application. - Video from live cameras, images from database, Classification Note and Processed Video.

2.13.2.2 Acquire Video

Identification	Name. Acquire Video Location. Server-Side Module
Type	Component
Purpose	This component fulfills following requirement from Software Requirements Specification Document. **Acquire Video** **Requirement** The server-side should be able to receive or access the live cameras. **Description** This feature enables the server-side to interpret the input received at the endpoint. video is processed at live time for detection.
Function	This component of system interfaces with the Endpoint – REST API to obtain the information received at the endpoint.
Subordinates	None.
Dependencies	This component is independent module, however, it depends on input from the Endpoint.
Interfaces	None.
Resources	**Hardware.** server, cameras, LAN/WiFi connection. **Software.** Docker, Python
Processing	Acquire video component would receive live video which would sent for preprocessing.

Data	This component uses following information of the application. - Video

2.13.2.3 Pre-Processing Video

Identification	Name. Pre-Processing Video Location. Server side Module
Type	Component
Purpose	This component fulfills following requirement from Software Requirements Specification Document. **Video Pre-Process** **Requirement** System shall be able to effectively divide video into frames as per the specified fps parameter. **Description** This component is responsible for the dividing the video into required number of frames so that they undergo feature extraction.
Function	This component of system interfaces with the Video Acquisition module to obtain the video.
Subordinates	None.
Dependencies	This component is independent module, however, it depends on input from Video Acquisition component.
Interfaces	None.

Resources	**Hardware.** server, LAN/WiFi connection. **Software.** Docker, Python
Processing	Video Pre-Processing component is responsible for preparing the video for feature extraction stage.
Data	This component uses following information of the application. - Video/Live Stream, fps.

2.13.2.4 Feature Extraction

Identification	Name. Feature Extraction Location. Server side Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Feature Extraction** **Requirement** The model shall be able to extract feature data for each frame. **Description** This component would basically perform feature extraction on frames.
Function	This component of system interfaces with the Video Pre-Processing module to obtain the frames.
Subordinates	Detection
Dependencies	This component is independent module, however, it depends on input from Video Preprocessing component.

Interfaces	None.
Resources	**Hardware.** server, Desktop Computer, LAN/WiFi connection. **Software.** Docker, Python
Processing	Feature Extraction component would process the incoming frames and store the feature data for each frame. This is then fed to the Smart Campus System model.
Data	This component uses following information of the application. - Frames of video.

2.13.2.5 Facial Recognition

Identification	Name. Facial Recognition Location. Server side Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Facial Recognition** **Requirement** The model shall be able to receive and load the feature data for each frame. The model shall be able to assess the features data and based on it, give a final classification that will Recognize the face. **Description** This major component covers two requirements. One is that the model loads the feature data for each of the frames into the neural networks. And assess that data to classify the facial recognition.

Function	This component of system interfaces with the Feature Extraction component to obtain the feature data.
Subordinates	Generating Attendance/Alerts
Dependencies	This component is dependent on the Registration component which stores the data and images of students.
Interfaces	None.
Resources	**Hardware.** server, database, LAN/Wi-Fi connection. **Software.** Docker, Python, TensorFlow.
Processing	Recognition is performed by database which is stored at the backend that perform inference based on the feature data.
Data	This component uses following information of the application. - Databases images and data.

2.13.2.6 Smoke/sleep detection

Identification	Name. Smoke/sleep detection Location. Server side Module
Type	Component

Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Smoke/sleep detection** **Requirement** The model shall be able to receive and load the feature data for each frame. The model shall be able to assess the features data and based on it, give a final classification that will detect whether the person is sleeping or smoking. **Description** This major component covers two requirements. One is that the model loads the feature data for each of the frames into the neural networks. And assess that data to classify the detection.
Function	This component of system interfaces with the Feature Extraction component to obtain the feature data.
Subordinates	Generating Alerts
Dependencies	This component is dependent on the Registration component which stores the data and images of students to tell who is smoking or sleeping.
Interfaces	None.
Resources	**Hardware.** server, database, LAN/Wi-Fi connection. **Software.** Docker, Python, TensorFlow.
Processing	Detection is performed by database which is stored at the backend that perform inference based on the feature data.
Data	This component uses following information of the application. - Databases images and data.

2.13.2.7 Teacher presence

Identification	Name. teacher presence Location. Server side Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Teacher detection** **Requirement** The model shall be able to receive and load the feature data for each frame. The model shall be able to assess the features data and based on it, give a final classification that will detect whether the teacher is present in class or not. **Description** This major component covers two requirements. One is that the model loads the feature data for each of the frames into the neural networks. And assess that data to classify the detection.
Function	This component of system interfaces with the Feature Extraction component to obtain the feature data.
Subordinates	Generating Notification/Alerts
Dependencies	This component is dependent on the Registration component which stores the data and images of teachers to detect the presence.
Interfaces	None.

Resources	**Hardware.** server, cameras, database, LAN/Wi-Fi connection. **Software.** Docker, Python, TensorFlow.
Processing	Detection is performed by database which is stored at the backend that perform inference based on the feature data.
Data	This component uses following information of the application. - Databases images and data.

2.13.2.8 Crowd activity

Identification	Name. Crowd activity Location. Server side Module
Type	Component
Purpose	This component fulfills following requirement from Software Requirements Specification Document. **Crowd Activity Detection** **Requirement** The model shall be able to receive and load the feature data for each frame. The model shall be able to assess the features data and based on it, give a final classification of crowd gathering and activity. **Description** This major component covers two requirements. One is that the model loads the feature data for each of the frames into the neural networks. And assess that data to classify the crowd activity.
Function	This component of system interfaces with the Feature Extraction component to obtain the feature data.
Subordinates	Generating Notification/Alerts

Dependencies	This component is independent.
Interfaces	None.
Resources	**Hardware.** server, cameras, LAN/Wi-Fi connection. **Software.** Docker, Python, TensorFlow.
Processing	Detection is performed based on the feature data.
Data	This component uses following information of the application. - crowd analysis at a specific place where crowd is not expected.

2.13.2.9 Generating Alerts/Results

Identification	Name. Generating Alerts/Results Location. Server side Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document. **Detection** **Requirement** The model shall be able to transmit the alerts/results back to the web application. **Description** This last component is responsible for producing the relevant alerts/results based on the classification done by the model.

	And then send that notification to the application through the Endpoint.
Function	This component of system interfaces with the Detection components to obtain the classification data.
Subordinates	None.
Dependencies	This component is dependent on the Detection components for the classification data.
Interfaces	None.
Resources	**Hardware.** server, Desktop Computer, LAN/Wi-Fi connection. **Software.** Docker, Python.
Processing	Generating Alerts/Results properly formats the response which includes the classification note and the resultant video and sends it to the Endpoint for it to be transmitted to web application.
Data	This component uses following information of the application. - Classification data, Resultant video.

2.14Human Interfaces.

2.14.1Use Case Design.

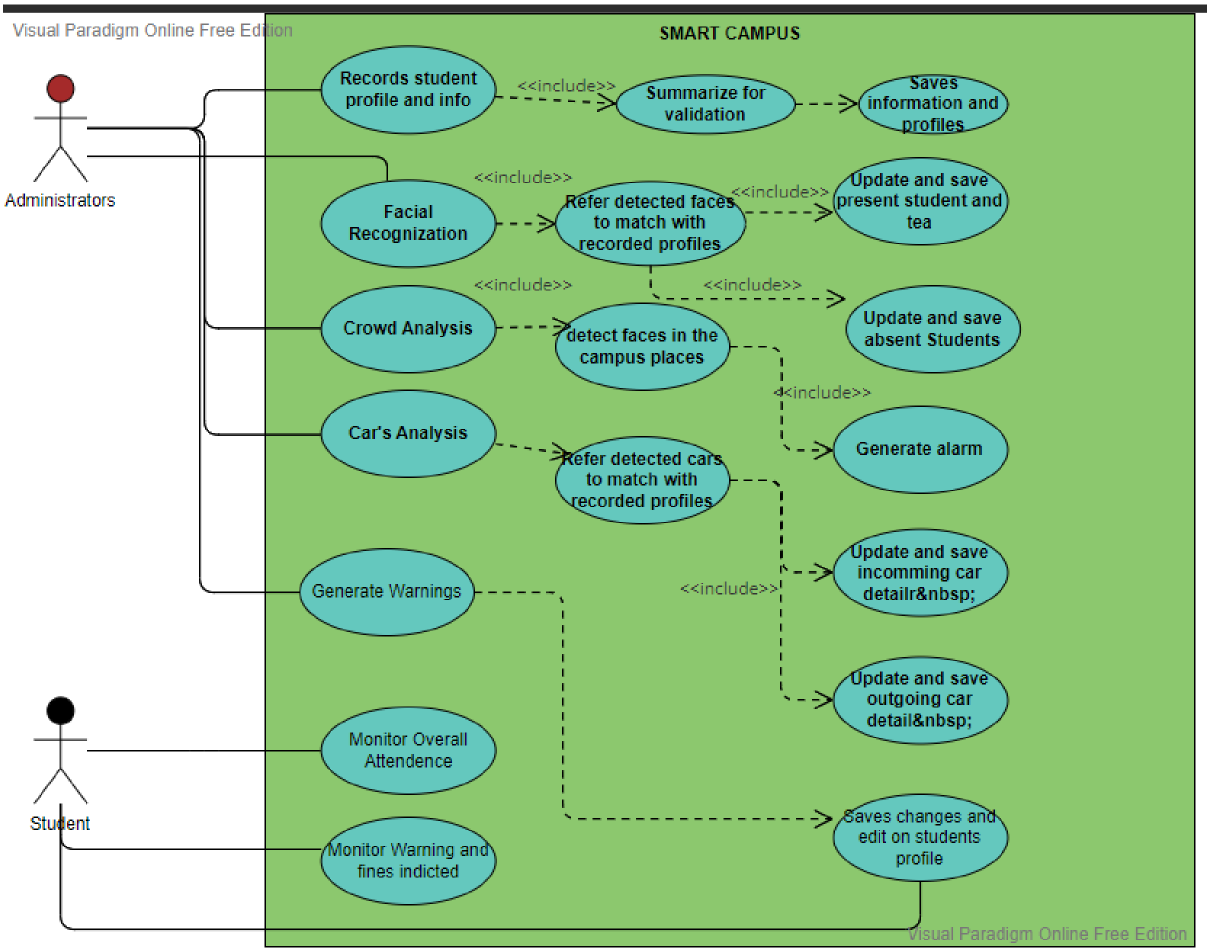

Figure 33. Use Case Diagram

The use case in the above figure is described in detail below.

2.14.2Video Acquisition.

Use case name	Video Acquisition
Primary actor	Server
Secondary actor	Camera
Normal course	- Server will continuously gather data from the cameras.

	- Data will be continuously uploaded on the server. - On the request by the user, server will display the data stream in runtime - Video is successfully acquired from the provided path.
Pre-Condition	User must ensure that the link to the live stream is valid, the specified video is present in the local directory, the application has access to a functional cam (whichever applicable).
Post-Condition	Connection is available to connect to the server.
Alternate course	The application was unable to acquire video from the input information provided by the user. Resulting data not found.
Pre-condition	The link to live stream was not valid or the specified video was not present in the local directory.
Post-condition	Error notification is displayed.
Extends	Upload Video From Local Directory, Provide Link To Live Stream.
Assumptions	- Videos being inputted are of the minimum supported resolution i.e. 480p. - The video which the user is uploading must not be more than the supported length.

2.14.3Face Recognition.

Use case name	Face Registration
Primary actor	User
Secondary actor	Server, Camera, Database

Normal course	- Camera will register students and faculty in the database.
	- Database will save all the faces. - On the request by the user, the specific person will be displayed
Pre-Condition	User must ensure that the camera is integrated with the application.
Post-Condition	Connection is available by the camera to the application.
Alternate course	The face features were not recognized by the application.
Pre-condition	Face features not recognized , loss of data from the video stream sent.
Post-condition	Error notification is displayed.
Extends	Re send video stream to analyze face features.
Assumptions	- Videos being inputted are of the minimum supported resolution i.e. 480p. - The video which the user is uploading must not be more than the supported length.

2.14.4View Attendance

Use case name	View Attendance
Primary actor	User
Secondary actor	Database

Normal course	- A table of Attendance will be displayed for a student to check his/her attendance.
	- Database will show the present / absent entries. - On the request by the user, the specific person attendance will be displayed
Pre-Condition	User must ensure that the latest data is saved in the database.
Post-Condition	Connection is available by the database to the application.
Alternate course	The latest data was not saved
Pre-condition	Loss of data, Not recognized student.
Post-condition	Error notification is displayed.
Extends	Check database connectivity.
Assumptions	- Face not recognised.

2.14.5View Results/Notifications

Use case name	View Results/Notifications
Primary actor	User
Secondary actor	Server

Normal course	- After the application has transmitted information about the target video, the user and the application wait for the response.
	- Application receives the processed video which has undergone the violence detection procedure at the server end. - Application also receives a notification mentioning that 'violence was detected'. - Application displays the resultant video along with the notification for the user to see.
Pre-Condition	The application has a working connection.
Post-Condition	Resultant video and 'violent' notification displayed.
Alternate course	The application receives the processed video in which no violence was detected.
Pre-condition	The application has a working connection.
Post-condition	Resultant video and 'non-violent' notification displayed.
Extends	N/A
Assumptions	- Videos being inputted are of the minimum supported resolution i.e. 480p. - The live stream is currently running, that is, it has not finished. - The user is waiting for the results.

2.14.6Load Models

Use case name	Load Models
Primary actor	Docker Image
Secondary actor	N/A

Normal course	- Docker images load the latest version of the Smart Campus System Models, which has been set by the developer.
	- Dependencies for the running the model, including TensorFlow, Django, Python/karas and its libraries, are loaded.
Pre-Condition	The developer provided the correct path to the model.
Post-Condition	The Docker image gains access to the specific version of the model.
Alternate course	Docker image was not able to gain access to the Smart Campus System model.
Pre-condition	The developer did not provide the correct path.
Post-condition	The Smart Campus System model could not be accessed and consequently, violence detection could not take place.
Extends/Includes	N/A
Assumptions	- Developer provides path to the Smart Campus System model which is most accurate in its performance.

2.14.7Pre-processing Video

Use case name	Pre Processing Video
Primary actor	Docker Image
Secondary actor	N/A

Normal course	- The video/link provided has been successfully accessed.
	- The video is successfully divided into frames based on the pre-defined fps parameter. - In case of live videos, a buffer storage is used to store some part of the video and perform pre-processing on it.
Pre-Condition	The Docker image successfully gained access to the video/live stream.
Post-Condition	The video has been divided into frames.
Alternate course	Docker image was not able to divide the video/link into frames.
Pre-condition	The Docker image was unable to gain access to the videos as it maybe in an unsupported format or the live stream has gone offline.
Post-condition	Docker image could not divide the videos into frames.
Extends/Includes	N/A
Assumptions	- Video/Link has been received at the endpoint.

2.14.8Run Predictions

Use case name	Run Predictions
Primary actor	Docker Image
Secondary actor	Smart Campus System Models

Normal course	- The Smart Campus System model receives and loads the frames.
	- Global and spatio-temporal features are generated. - These features are then fed into the neural network for classification.
Pre-Condition	The Smart Campus System model and its dependencies were successfully loaded by the Docker image. The video was successfully divided into frames.
Post-Condition	The video/live stream is undergoing violence detection.
Alternate course	The Smart Campus System model was unable to run prediction on the provided frames due to poor quality/noise.
Pre-condition	The Smart Campus System model and its dependencies were successfully loaded by the Docker image. The video was successfully divided into frames.
Post-condition	The target video/live stream could not undergo violence detection.
Extends/Includes	Receive Frames, Analyze Frames.
Assumptions	- The environment for running the model was set-up. - The server does not run out of RAM/VRAM while running predictions.

2.14.9Return Results

Use case name	Return Results
Primary actor	Docker Image
Secondary actor	N/A

Normal course	- The Smart Campus System model is running and checking for violence in the video/live stream.
	- The Docker image waits until the video is being checked. - The model returns the 'violent' classification notification and resultant video. - These results are then transmitted to the Endpoint.
Pre-Condition	The video is undergoing violence detection by the model.
Post-Condition	The video/live stream is classified as violent and the subsequent results are transmitted to the endpoint.
Alternate course	The Smart Campus System model was unable to find violence in the videos/live stream.
Pre-condition	The video is undergoing violence detection by the model.
Post-condition	The target video/live stream did not contain any violence therefore, a 'non-violent' classification note and video were sent to the endpoint.
Extends/Includes	Generates Final Classification
Assumptions	- The environment for running the model was set-up. - The server does not run out of RAM/VRAM while running predictions.

2.14.10 Receive Frames

Use case name	Receive Frames
Primary actor	Smart Campus System Model
Secondary actor	Docker Image

Normal course	- The Smart Campus System model is successfully loaded and waiting for input.
	- The Docker image inputs the frames according to the pre-defined batch size. - Smart Campus System model checks if RAM/VRAM enough to start processing.
Pre-Condition	The Smart Campus System model has been successfully loaded.
Post-Condition	The frames were received successfully.
Alternate course	The Smart Campus System model was unable to accept all of the frames.
Pre-condition	The Smart Campus System model has been successfully loaded.
Post-condition	The model runs prediction on a part of the video, and not the entire video.
Extends/Includes	N/A
Assumptions	N/A

2.14.11 Analyze Frames

Use case name	Analyze Frames
Primary actor	Smart Campus System Model
Secondary actor	Docker Image

Normal course	- The Smart Campus System model accepts the number of frames as per the batch size.
	- All the frames are analyzed and feature data for each frame is generated and stored. - This feature data is then fed into the neural network.
Pre-Condition	The model successfully receives the frames.
Post-Condition	The feature data for all of the frames is generated and fed to the model.
Alternate course	The Smart Campus System model was unable to generate features data for the frames due to poor resolution of frame images.
Pre-condition	The model successfully receives the frames.
Post-condition	Feature data could not be generated for the frames.
Extends/Includes	N/A
Assumptions	- The environment for running the model was set-up. - The server does not run out of RAM/VRAM while running predictions.

2.14.12 Give final Prediction.

Use case name	Give Final Classification
Primary actor	Smart Campus System Model
Secondary actor	Docker Image
Normal course	- Model has performed its inference on the feature data provided.
	- Each frames has been individually classified as violent/non-violent. - The number of violent frames outnumbered non-violent frames. Or the number of violent frames crossed a set threshold (not set as of now).

	- The video is classified as ‘violent’.
Pre-Condition	The model has successfully performed inference on the feature data.
Post-Condition	‘Violent’ classification generated.
Alternate course	The number of non-violent frames outnumbered violent frames. Or the number of violent frames was below the set threshold.
Pre-condition	The model has successfully performed inference on the feature data.
Post-condition	‘Non - Violent’ classification generated.
Extends/Includes	N/A
Assumptions	- The environment for running the model was set-up. - The server does not run out of RAM/VRAM while running predictions.

2.15Requirements Matrix.

REQUIREMENT	PRIORITY (1-9)
Req-1: The system shall be able to recognize student faces and mark attendance.	**Facial Recognition + camera:** This component will take the pre-processed frames from video and will recognize face.
Req-2: The system will able to check the teacher’s presence in classroom	**Facial Recognition + camera:** This component will recognize the face and will match face with teacher’s faces in database to check teacher’s presence
Req-3: Person Tracking	**Facial Recognition + camera:** This component will recognize faces and will store the location using camera location for 2 hours to track people.
Req-4: Sleep Detection	**Facial Recognition:** This Component will recognize facial Expressions and check whether student is sleeping or not.

Req-5: The system will be able to get video from camera and preprocess the video	**Video Acquisition:** This feature is responsible to get the video from camera. **Video preprocessing:** This feature extract frames from video and apply preprocessing techniques
Req-6: The system shall be able to manage car parking in campus	**Number Plate Reading:** This feature is responsible for reading car number plates and maintaining in/out record.
Req-7: The system shall be able to analyze crowd's activity	**Crowd Activity:** This feature will manage to analyze crowd's activity.

Chapter 3. Implementation And Testing

3.1 System Overview.

The SMART Campus is able to detect and recognize faces to mark attendance and tell teacher presence, analyze crowd activity by counting no of people, detecting tobacco smoking, reading license plates. We have used various machine and deep learning algorithms and architectures like YOLO, arcface, Resnet, MobileNet etc. We have trained 4 different models for the mentioned functionalities.

3.2 Facial Recognition.

Additive Angular Margin Loss, often known as ArcFace, is a loss function used in facial recognition applications. These tasks typically involve the softmax. A performance gap for deep face recognition under significant intra-class appearance fluctuations is caused by the softmax loss function's failure to explicitly optimize the feature embedding to enforce more similarity for intraclass samples and diversity for inter-class samples.

By substituting Angular Margin Loss for Softmax Loss in the similarity learning process used by ArcFace, the classification task can be addressed using distance metric learning.

The inner product of two normalized vectors can be used to calculate the cosine distance, which is a technique used by search engines to determine the separation between faces. If the two vectors are identical, will equal 0 and cos will equal 1. If they are orthogonal, cos will equal zero and will be /2. As a result, it can be applied as a similarity metric.

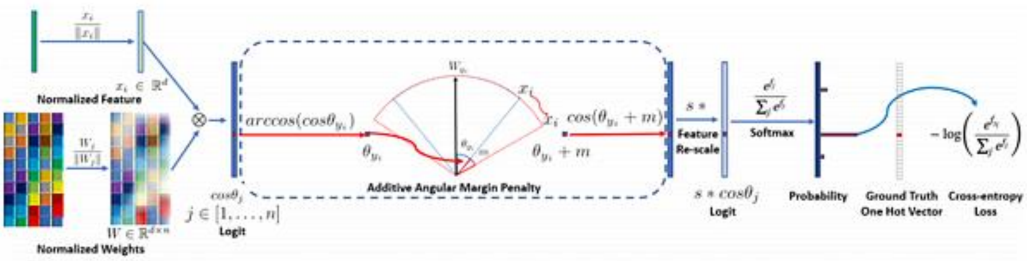

Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature x_i and weight W normalisation, we get the $\cos\theta_j$ (logit) for each class as $W_j^T x_i$. We calculate the $arccos\theta_{y_i}$ and get the angle between the feature x_i and the ground truth weight W_{y_i}. In fact, W_j provides a kind of centre for each class. Then, we add an angular margin penalty m on the target (ground truth) angle θ_{y_i}. After that, we calculate $\cos(\theta_{y_i} + m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross entropy loss.

Algorithm 1 The Pseudo-code of ArcFace on MxNet

Input: Feature Scale s, Margin Parameter m in Eq. 3, Class Number n, Ground-Truth ID gt.

1. x = mx.symbol.L2Normalization (x, mode = 'instance')
2. W = mx.symbol.L2Normalization (W, mode = 'instance')
3. fc7 = mx.sym.FullyConnected (data = x, weight = W, no_bias = True, num_hidden = n)
4. original_target_logit = mx.sym.pick (fc7, gt, axis = 1)
5. theta = mx.sym.arccos (original_target_logit)
6. marginal_target_logit = mx.sym.cos (theta + m)
7. one_hot = mx.sym.one_hot (gt, depth = n, on_value = 1.0, off_value = 0.0)
8. fc7 = fc7 + mx.sym.broadcast_mul (one_hot, mx.sym.expand_dims (marginal_target_logit - original_target_logit, 1))
9. fc7 = fc7 * s

Output: Class-wise affinity score $fc7$.

Figure 34. Arcface （Source ：https://arxiv.org/abs/1801.07698）

The Fully Connected (FC) layer of a typical classification task takes the inner product of the features and weights and applies Softmax on the output.

In ArcFace, cos is calculated by calculating the inner product after normalizing the weights of the FC layer and features. Softmax is used to cos to determine the loss. After calculating the inner product and applying arccos to the cos values, we now only add an angle margin of +m for the labels that are valid. By doing this, we stop the FC layer's weight from being unduly dependent on the input data set.

3.2.1 Task of Face Recognition

Let's first study about the Face Recognition task's operation and the reasons behind its necessity before delving deeply into the ArcFace technique.

The process of recognizing or validating one or more faces in a picture is known as face recognition. There are many reasons why it may be beneficial to identify a face in an image: Facebook recognizes and tags people in their photographs, and airport security might check that a passenger's face matches the face in his passport. In businesses, we wish to allow admission only for authorized personnel.

Comparing two faces to see if they belong to the same person is known as a verification task.

Comparing and recognizing a given face with a database of previously stored faces is known as an identification task.

Identifying a face in an image, feature extraction, and face matching are the three basic processes in the face recognition process.

3.2.2 Face Matching

3.2.2.1 Feature Embeddings

Feature extraction and classification make up a conventional CNN for classification. The model acquires the distinctive facial features during training and creates feature embeddings during the feature extraction procedure. After training is finished, you can bypass the classification step and create feature embeddings—essentially, a digital "fingerprint"—for each face image. An alternative approach to see an embedding vector is as the transformation of data with high dimensions into those with relatively low dimensions.

By utilizing either the cosine similarity or the square distance between two vectors, these embeddings assist us in determining whether there is a similarity between two individuals.

Two separate photographs of the same person will have vectors with high similarity and low distance, while two different images of the same person will have vectors with low similarity and a significant square distance.

After faces have been identified and cropped in two face photos, we may analyze the images using the ArcFace model to create two feature embeddings. Once we have two embedding vectors, we may compute the square distance or cosine similarity between them to ascertain whether the two photos were of the same subject.

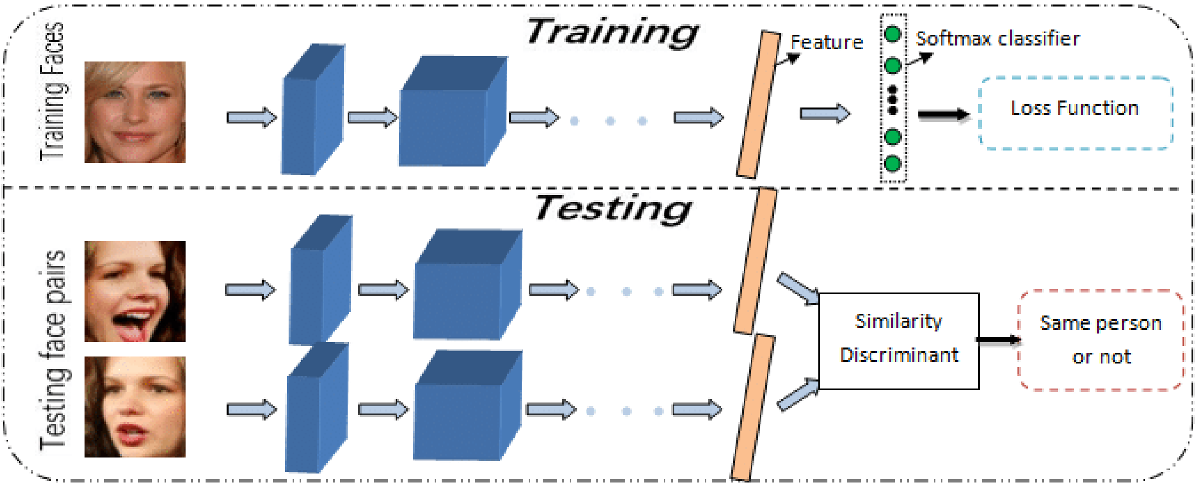

Figure 35. General Pipeline for face verification

3.2.3 Cosine Similarity.

The cosine similarity has a range of -1 to 1. This is a measurement of the similarity in direction (and not size) between two vectors that are not the zero vector in the inner product space. The cosine similarity is the same as an inner product between two normalized vectors and is equivalent to the cosine between two vectors.

Cosine similarity is 1 when two vectors are pointing in the same direction.

When two vectors are 90 degrees apart, cosine similarity is 0, and when they are opposite, it is -1.

3.2.4 Softmax Vs Arcface

The end of a typical classification network is often where SoftMax and Categorical Cross-Entropy loss are applied. Numbers are converted into probabilities using SoftMax. It provides a probability for each class that adds up to 1 for each object. The class with the highest likelihood is picked once training is complete. The difference between two probability distributions is calculated using the categorical cross-entropy loss, which is minimized during training by using back-propagation.

SoftMax's disadvantage is that it doesn't create a safety buffer, which causes the borders to be a little hazy. As much as possible, we want the vectors of two photographs of the same person to be similar, and as much as feasible, we want the vectors of two images of two different persons to be unlike. This means that, like SVM, we wish to generate a margin.

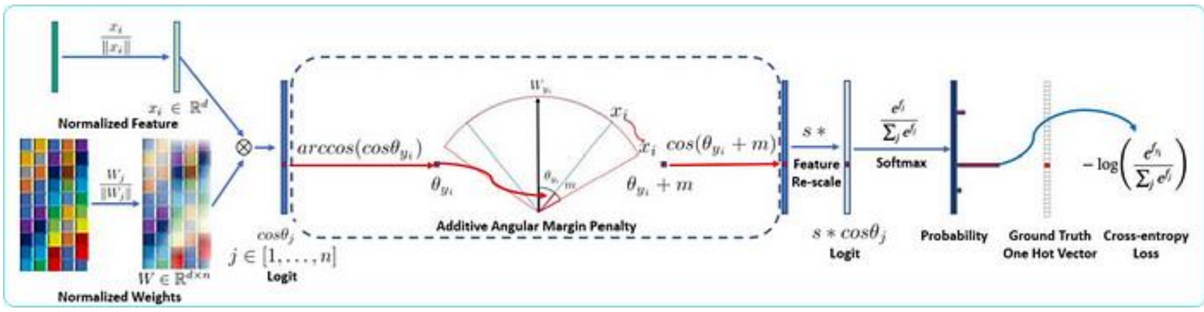

Figure 36. Cmposition of Arcface

3.2.5 Architecture

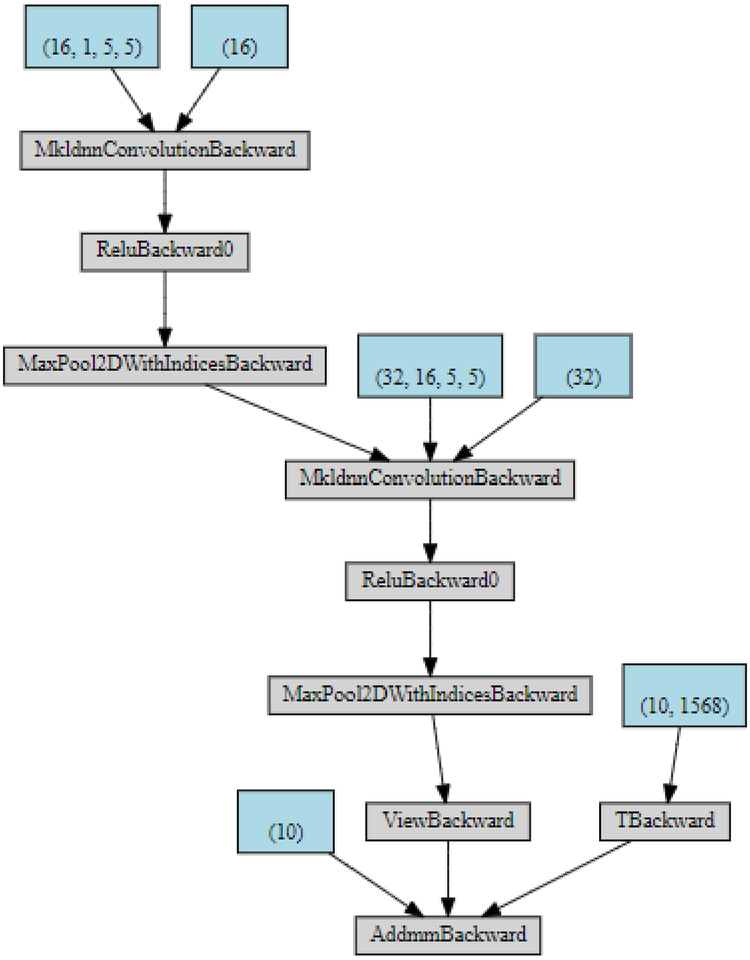

Figure 37. Our Architecture

3.2.6 Results and Conclusions

```
Evaluate Model: 100% ████████████ 8/8 [00:02<00:00, 3.31it/s]

[loss]=1.7830 - [acc]=0.6154 - [val_loss]=1.9209 - [val_acc]=0.9760
[INFO] 3 lives left!
Time taken to execute code : 397.972295889
```

Figure 38. Training Results

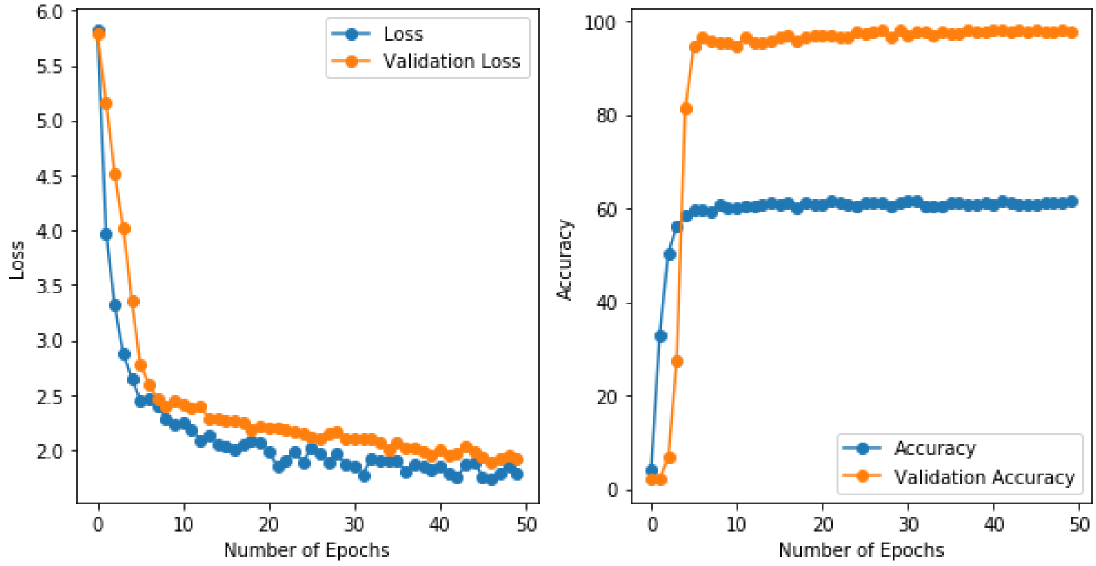

Figure 39. Accuracy and Loss

```
Evaluate Model: 100%  8/8 [00:02<00:00, 3.26it/s]

loss: 1.949543297290802 - acc: 0.968
```

Figure 40. Testing Results

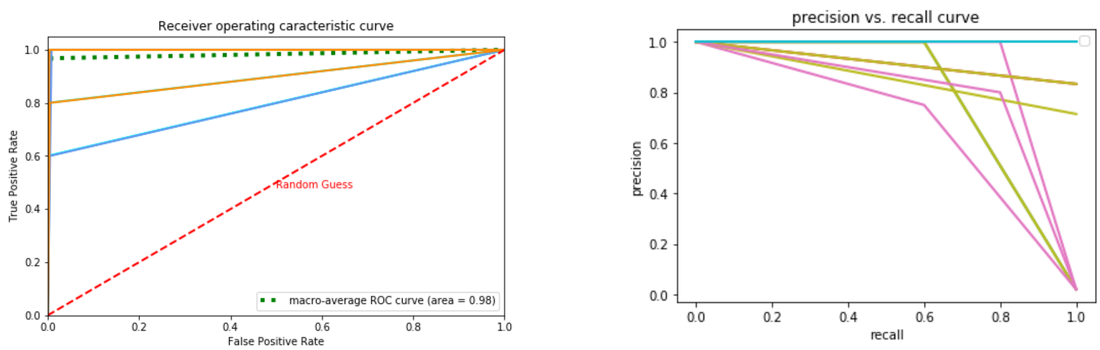

Figure 41. Evaluation of Facial Recognition Model

3.3 Smoking Detection.

Data collection: For the machine learning model to be accurate, training data must be of high quality and quantity. Different lighting conditions, population density, camera angles, and other elements that may have an impact on the model's accuracy should all be covered by the data. The information can be gathered from a variety of sources, including social media platforms, drones, and security cameras.

Pre-processing: To enhance the quality and relevance of the data, pre-processing must be done on the image data before it is fed into the machine learning model. Images may need to be resized, filtered, cropped, and normalized during the pre-processing stage.

Feature Extraction: From the pre-processed picture data, pertinent features are extracted using feature extraction methods. Edge identification, color histograms, and texture analysis are a few examples of these features.

Machine Learning Model: A multitude of machine learning techniques, including support vector machines, decision trees, and deep neural networks, can be used to count the number of individuals in a crowd. Convolutional neural networks (CNN) are a type of deep learning model that is particularly good at image-based tasks and can attain high accuracy with a lot of training data.

Evaluation: Metrics like precision, recall, and F1 score are used to gauge how accurate the machine learning model is. The evaluation's findings are applied to the model's adjustment and performance enhancement.

Implementation: To count the number of people in a crowd, a machine learning model is implemented in a practical context. The model can be used in conjunction with security cameras or drones to give real-time crowd analysis and counting.

Limitations: A number of variables, like changes in illumination, occlusions, and fluctuations in crowd density, can impact how accurate the machine learning model is. Additionally, the model could have trouble with crowded or overlapping scenarios, which might make it challenging to count certain individuals precisely.

In general, crowd analysis and crowd size estimation can both benefit from the use of machine learning. Machine learning models can be trained to precisely count the number of individuals in a crowd and offer insightful data on crowd behavior with the correct data and methods.

The model was fine-tuned on pretrained weights of ResNet50.

```
Epoch 100/100
130/130 [==============================] - 144s 1s/step - loss: 0.1816 - accuracy: 0.9325 - val_
loss: 0.2618 - val_accuracy: 0.8989
```

Figure 42. Training Results

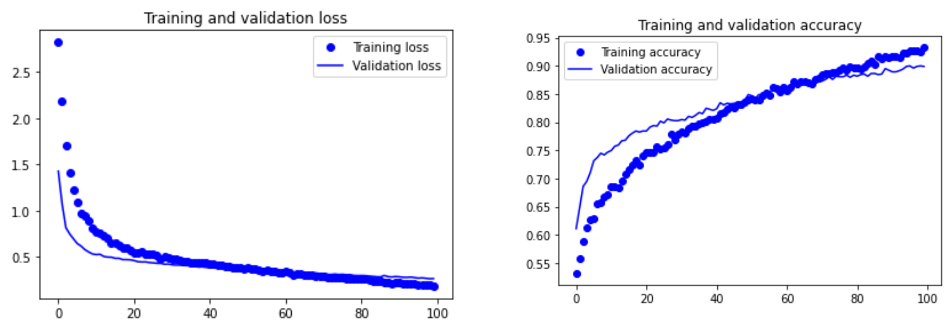

Figure 43. Training and Validation Accuracy and Loss

3.3.1 Model Evaluation

The test dataset will be used to evaluate the performance of the model.One of the metrics that will be tested would be accuracy which measures the fraction of predictions the model got right. Other metrics are as follows:

3.3.1.1 Precision(P):

The fraction of true positives (TP, correct predictions) from the total amount of relevant results, i.e., the sum of TP and false positives (FP). For multi-class classification problems, P is averaged among the classes. The following is the formula for precision.

$P=TP/(TP+FP)$

3.3.1.2 Recall(R):

The fraction of TP from the total amount of TP and false negatives (FN). For multi-class classification problems, R gets averaged among all the classes. The following is the formula for recall.

$R=TP/(TP+FN)$

3.3.1.3 F1 score(F1):

The harmonic mean of precision and recall. For multi-class classification problems, F1 gets averaged among all the classes. The following is the formula for F1 score.

$F1=2*(TP*FP)/(TP+FP)$

	precision	recall	f1-score	support
not_smoking	0.48	0.43	0.45	521
smoking	0.64	0.69	0.66	776
accuracy			0.58	1297
macro avg	0.56	0.56	0.56	1297
weighted avg	0.58	0.58	0.58	1297

Figure 44. Tobacco Smoking Detection Model Evaluation

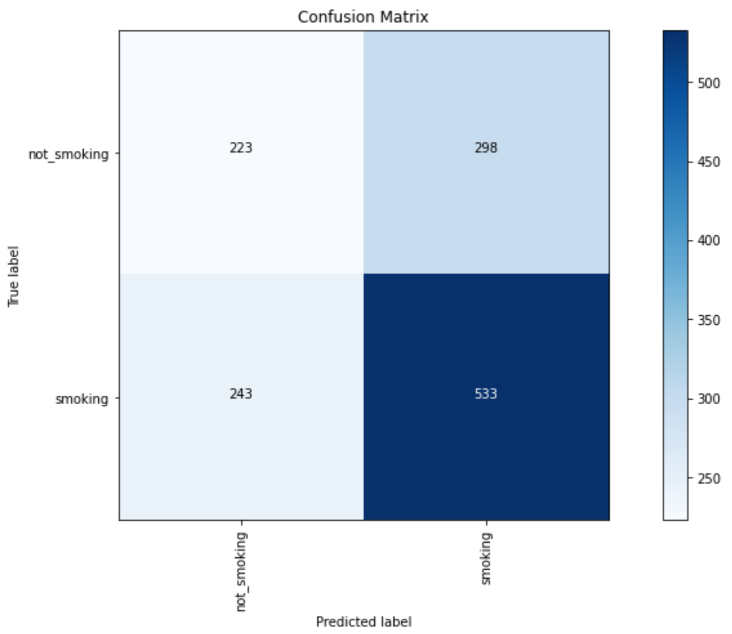

Figure 45. Tobacco Smoking Detection Confusion Matrix

3.4 Car Number Plate Reading.

The number plate can be detected anywhere in the vehicle. The image is devised for robust plate detection. ANPR performed under constraints as brightness, angle of the plate, and resolution can use basic image processing techniques discussed in [23, 28–30]. Contouring is a method used for shape scanning and entity localization and identification but it would not give us accurate results in all situations; for example, the image is tilted. For the identification of number plates, a deep neural network-based YOLO (You Only Look Once) model is recommended. To analyze the performance of the YOLO family on proposed ANPR methods, we have trained YOLOv3 and YOLOv4 to localize the number plate region.

The number plate can be detected anywhere in the vehicle. The image is devised for robust plate detection. ANPR performed under constraints as brightness, angle of the plate, and resolution can use basic image processing techniques discussed in [23, 28–30]. Contouring is a method used for shape scanning and entity localization and identification but it would not give us accurate results in all situations; for example, the image is tilted. For the identification of number plates, a deep neural network-based YOLO (You Only Look Once) model is recommended. To analyze the performance of the YOLO family on proposed ANPR methods, we have trained YOLOv3 and YOLOv4 to localize the number plate region.

In the recognition stage, the characters from the image are recognized by using different approaches. A commonly used technique behind character recognition is the Artificial Neural Network (ANN). ANN is considered a script since the neural network comprises interconnecting artificial neurons and a mathematical definition by using feedforward backpropagation (BP). In [25], BP is proposed as a neural network-based method. The most important neural network for character recognition is the Convolutional Neural Network

(CNN) which is used in Section 4 as an implemented method of the previous ANPR system. Our proposed ANPR method for recognition involves OCR Tesseract, which is the time-efficient method.

At this stage, the number plate's labels are stored in a file to keep a record. We have reconstructed the recognized characters into plate label strings and stored them in the excel file which is further utilized for scoring.

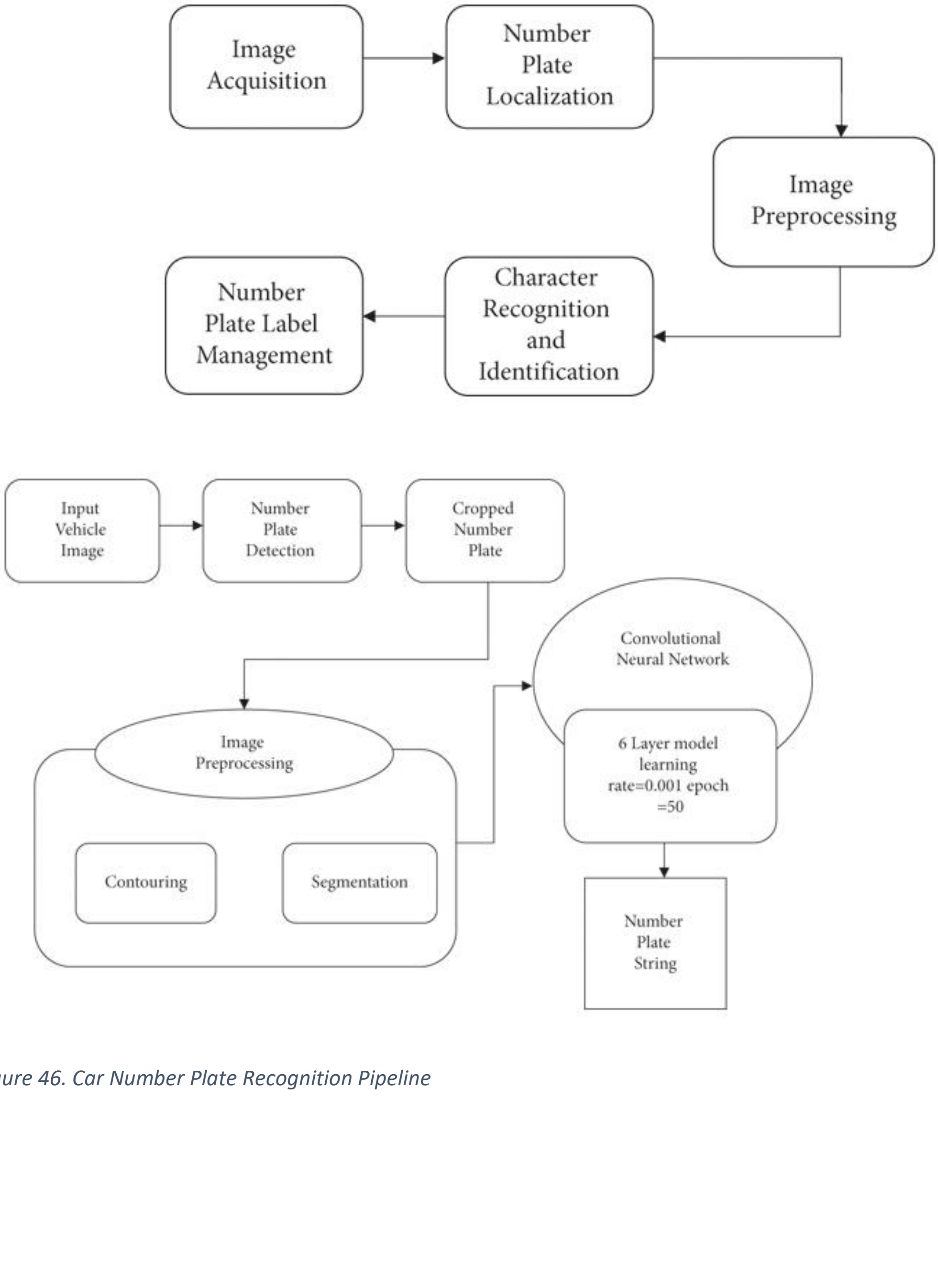

Figure 46. Car Number Plate Recognition Pipeline

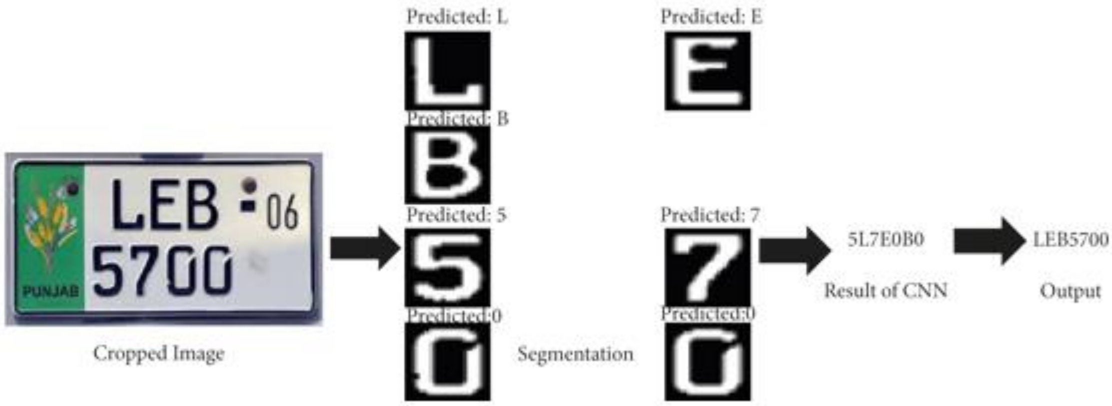

Figure 47. ANPR Pipeline Results

Chapter 4. Implementation And Testing

Computer vision and machine learning have been applied to many fields including security, surveillance, and traffic management. In this article, we will discuss four applications of computer vision and machine learning: tobacco smoke detection, facial recognition, people counting, and number plate detection.

4.1 Tobacco Smoke Detection

Tobacco smoke detection is a critical application in public places where smoking is prohibited. The computer vision-based system can detect the smoke and alert the authorities. The machine learning algorithms can be trained to identify the shape and pattern of the smoke, and the system can be integrated with a security system to notify the authorities.

4.2 Facial Recognition

Facial recognition is a widely used application of computer vision and machine learning in security systems. Facial recognition systems can be used to identify individuals in real-time, and they are used in various settings, such as airports, banks, and border control. Facial recognition algorithms use various techniques such as deep learning to extract features from facial images and match them with a database of known faces. The accuracy of facial recognition systems has improved significantly in recent years, and they are becoming increasingly popular in security systems.

4.3 People Counting

People counting is a useful application of computer vision in traffic management and crowd control. The system uses cameras to capture images of the crowd and uses machine learning algorithms to detect and track people. The system can be used to count the number of people in a specific area and predict the flow of people. The data collected from the system can be used to optimize crowd control measures and improve public safety.

4.4 Number Plate Detection

Number plate detection is a widely used application of computer vision and machine learning in traffic management. The system uses cameras to capture images of vehicles and uses machine learning algorithms to extract the number plates from the images. The system can be used to identify stolen or suspicious vehicles and alert the authorities. The data collected from the system can be used to optimize traffic flow and improve road safety.

In conclusion, computer vision and machine learning have become an integral part of many security and surveillance systems. These technologies have improved public safety and are becoming increasingly popular in traffic management and crowd control. As these technologies continue to evolve, we can expect more advanced applications of computer vision and machine learning in various fields.

Chapter 5. Conclusion and Future Works

SMART Campus is a very comprehensive project to be done that includes as many feature as you can think to make a campus more safer and technologically advanced.

Here are some of the key feature that may be developed in future.

Theme	Smart City Theme	Smart Campus Theme	Remarks	
1	Social sustainability	People	Community/Smart People	Includes students, staff, faculty, parents and other stakeholders
2	Environmental sustainability	Planet	Campus Infrastructure	Includes university building, streets, safety, environment
3	Economic sustainability	Prosperity	Sustainability/Employability	Includes jobs opportunity, research, budget
4	Governance	Governance	Administration/Management	Includes strategic planning.
5	Propagation	Propagation	Replicability, Innovation	Scalability, Social compatibility

A detailed view of the modules is available at
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7207122/table/tbl0010/?report=objectonly

References

a) IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

b) Real-time video fire/smoke detection based on CNN in antifire surveillance systems by Sergio Saponara, Abdussalam Elhanashi & Alessio Gagliardi2- https://link.springer.com/article/10.1007/s11554-021-01094-y

c) [Human face recognition based on convolutional neural network and augmented dataset by Peng Lu, Baoye Song & Lin Xu2

d) https://arxiv.org/abs/2106.03078v1

e) A People Counting System Based on Face Detection and Tracking in a Video by Zhiwen Yu, Hau-San Wong & Hong Va Leong1

f) https://ieeexplore.ieee.org/document/5279466

g) Automatic Number Plate Recognition System Based on Deep Learning by T. Damak, O. Kriaa, A. Baccar et al.2

h) https://publications.waset.org/10011141/automatic-number-plate-recognition-system-based-on-deep-learning

i) Use Case Modeling Guidelines, which documents the guidelines used to develop the use case model specifying the functional requirements in this specification. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=787548

j) MVC architecture from HCI book Building Interactive Systems- Principles of HCI- Dan Olsen

k) http://en.wikipedia.org/wiki/Component_diagram

l) https://github.com/AzureWoods/faceRecognition-yolo-facenet

m) https://www.hindawi.com/journals/js/2021/4796768/?utm_source=google&utm_medium=cpc&utm_campaign=HDW_MRKT_GBL_SUB_ADWO_PAI_DYNA_SPEC_X_X0000_Feb2023&gclid=CjwKCAjw9J2iBhBPEiwAErwpeYDK4TOI0T-H0FFwQWyOF6PkS3-XCUsatG3pQcVV7MHVigmuRRsbtBoCieYQAvD_BwE

n) https://www.hindawi.com/journals/complexity/2021/5597337/

o) https://github.com/xuexingyu24/MobileFaceNet_Tutorial_Pytorch

p) https://sefiks.com/2020/12/14/deep-face-recognition-with-arcface-in-keras-and-python/

q) https://www.hindawi.com/journals/js/2021/4796768/?utm_source=google&utm_medium=cpc&utm_campaign=HDW_MRKT_GBL_SUB_ADWO_PAI_DYNA_SPEC_X_X0000_Feb2023&gclid=CjwKCAjw9J2iBhBPEiwAErwpeYDK4TOI0T-H0FFwQWyOF6PkS3-XCUsatG3pQcVV7MHVigmuRRsbtBoCieYQAvD_BwE