RAVEN

Final Year Project

Session 2018-2022

A project submitted in partial fulfilment of the
COMSATS University Degree
of
BS in Computer Science / Software Engineering (CUI)

Department of Computer Science
COMSATS University Islamabad, Lahore Campus

19 January 2022

Project Detail

Type (Nature of project)	[ ] **Development** [ ] **R**esearch [✓] **R&D**
Area of specialization	Video Analytics

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	SP18-BCS-031	MUHAMMAD BARAK ULLAH	Sp18-bcs-031@cuilahore.edu.pk	
(ii)	SP18-BCS-015	BURHAN SABIR	Sp18-bcs-015@cuilahore.edu.pk	
(iii)	SP18-BCS-115	MUHAMMAD SAIM ASHFAQ	Sp18-bcs-115@cuilahore.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am MUHAMMAD BARAK ULLAH S/O KHALID QAYYUM, group leader of FYP under registration no CIIT/SP18-BCS-031/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is _9%_ that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: 30/12/2021 Name of Group Leader: MUHAMMAD BARAK ULLAH Signature: ________________

Name of Supervisor: DR. USAMA IJAZ BAJWA Co-Supervisor (if any): ______________________

Designation: ASSISTANT PROFESSOR Designation: ______________________

Signature: ______________________ Signature: ______________________

HOD: ______________________

Signature: ______________________

Abstract

Tracking is usually done by following an object or simply by attaching some microcontroller-devices like GPS or RFID to that object so that their track could be kept. These are the methods which we used previously for tracking, but these have some limitations. For say, if we want to find the location of a lost child and he doesn't have any GPS devices with him with which we can locate his position, it will be quite hard to find him. But in the modern day, we have surveillance cameras located at almost every park and mall. These surveillance cameras help us in monitoring people by integrating some microcontrollers onto these cameras as we can detect and track people through these cameras. The lost child whose location was not known before could now easily be found by using these surveillance cameras and by using a method called person re-identification. This sort of visual tracking is an important addition in the field of tracking as it is computationally less expensive because it will use the already installed surveillance cameras on that area and no external devices will be used (i.e., Wi-Fi, GPS, RFID). Our system will be focussed on the person re-identification process, and it will have low computational cost so that we can implement it on low-end devices as well.

Acknowledgement

We are highly grateful to our institute COMSATS University Islamabad, Lahore Campus for providing us with such great environment to work in with. It would be truly unfair not to acknowledge the efforts of Dr. Usama Ijaz Bajwa for sharing his guidance, wisdom and intelligence with us. He has been our constant support throughout this project. Whenever we felt down, he showed confidence in us and taught us to work hard with determination which helped us in overcoming all the hurdles that came in our way. We are highly indebted to Dr. Usama because it would not have been possible to work on this project without his continuous support and help.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1 Introduction

1.1 Introduction

Video is the sequence of frames or recording of moving visual images. Video streams are now generally used in many surveillance systems. For say, if we want to monitor an object or if there is need of detecting an abnormal or unusual activity at some place, we want video streams for that purpose which are directly being monitored. Through these video streams, we can also detect the location of a particular entity or object. Through these detected regions, we can track that particular entity later so that if we want to know its current location or the track it has used to reach some destination can easily be retrieved. This work of tracking is known as video tracking in which we continuously detect and locate an object or multiple objects over time using video streams from cameras. This whole work of detecting activities and tracking of these objects come under the domain of video analytics. Video analytics has become an emerging topic in the modern era. A lot of modification is being done on daily basis in Computer Vision. As a subfield of computer vision and video processing, the process of tracking is very demanding yet challenging. We will be particularly focusing on tracking side in which we will re-identify the person and then will track that person.

Tracking through video streams i.e., Visual Tracking is new to some extent as a lot of research is yet to be done in that field but tracking itself is not a newly discovered thing. Tracking is also being done by using technologies like (Wi-Fi, GPS, RFID) which use sensors and tags to track the project related entities. In systems which use GPS, one needs to have continuous internet connection and location of its device needs to be enabled with the help of which the end user will track that person/object. In case of RFID, there are two components tags and readers. Tags are attached with the objects which needs to be tracked and the other component (reader) has antennas which receive signals back from that RFID tag which was attached to the object. These techniques are not cost effective as they need tags and readers to be attached with the object that needs to be tracked or in the case of GPS, one needs to have internet connection through which his location would be tracked.

To overcome these issues, we are using the newly emerging tracking technique which is Visual Tracking / Multiple Object Tracking. MOT is the process of locating multiple objects over a sequence of frames (video). The MOT problem can be viewed as a data association problem where the goal is to associate detections across frames in a video sequence. [1]. We have 2 types of datasets provided for training purposes, one is video streams and the other one is sequence of frames. If we have data in the form of video streams, first we convert that stream into frames and then pass these frames to the model because no model is available which can be trained directly on video streams. Video streams are then converted to frames according to the directions given by the dataset providers.

The problem which we will be focusing on will be person re-identification. Re-identification generally is a very tough task due to its complexity which is caused due to the presence of low image resolutions, illumination changes, occlusions, complex camera environments, background clutter etc. This process of re-identification is widely used for specific object retrieval from multiple cameras. Re-identification basically is the process in which we associate images of the same object in different cameras on different occasions. At first, when the person is detected in the camera, that person will be labelled and then its relevant information will be stored in the database. Next time when that person will appear in the frame, our system checks whether that person has appeared before by identifying him from the database. If it has been appeared before, it assigns that same label again to that person who was assigned before and track its path by using different tracking technologies. Whereas, if that person has appeared for the first time in the frame, he/she will be assigned an ID and a new label to it and its information will be stored in the database. This cycle of labelling and tracking continues throughout the whole re-identification phase.

1.2 Objectives

The objectives for our proposed project will be as follows:

- To build a system that will be able to detect, track and re-identify multiple people as they appear in the video stream.
- To save the record of a person at which time he/she has passed through our camera view so that we would be able to retrieve it later if we want to check his/her previous appearance on site.
- To build a system that will have low computational cost so that it would be easy to implement it on low-end devices as well.

1.3 Problem statement

The world has advanced quite rapidly but security is still one of the main concerns for people when they go out in public places because it is quite often that we hear news about burglary, theft and street crimes. In order to limit and control these activities, there is a strong need of automated surveillance systems to be installed in those places. It will help the law enforcement and will bring fear to the criminal as he would know that his activities would be monitored which will refrain him from committing crime to some extent. The problem that we are dealing with is detecting the person and then tracking him afterwards, but the problem arises when the person disappears from one camera and then appears in another camera, his previous track would be lost.

To solve this problem, we have proposed a system which aims to detect and re-identify a person who has previously been seen in a network of cameras that come under our domain. It is a solution to the problem where we need overlapping cameras so that the record of a person would not be lost if he goes out of the range of the camera and then appears again. The process of re-identification counters that problem and even on large space with multiple non overlapping cameras, it recognises the individual in different locations.

1.4 Assumptions & Constraints

The proposed project's assumptions and constraints will be as follows:

1.4.1 Assumptions

The proposed project is assumed to be a user-friendly application that helps to handle surveillance tasks by detecting, tracking and re-identifying multiple people which in return helps in improving the security system as a result. It is believed that the product benefits through:

- It will be serving as a surveillance system which will help the security personnel.
- Detection and tracking of multiple people appearing in a stream.
- Re-identification of people appearing in the video stream which in return will help the security personnel in case of any anomalous event.
- It will help in reducing crime rates as people will know that their activities are being recorded.
- We have built a resource-efficient system that works on a simple low end machines.

1.4.2 Constraints

Although there are a lot of pros in our proposed application, but there are some technical issues that we might face in some situations and those issues are described below:

- The angle from which the video stream is coming must be stable because we know that the stability of camera directly affects the quality of the video stream.
- Illumination variation might challenge our system because at some places where there is low light the system will be unable to identify the person.
- Occlusion in some cases may also affect our system because it occurs in crowded places.
- In some cases, person's different body parts might have different poses in different camera angles and maybe his movement speed would be different. Our system might find it difficult to re-identify that person due to the change of pose and speed because it will be that same person but will look different in consecutive frames due to the continuous change in pose and body movement.

1.5 Project scope

Security always has been a major concern for every person, either at home or in public places like malls, parks, universities etc. To improve the security, nowadays public places are equipped with the surveillance cameras. Security personnel are hired who sit behind these surveillance cameras all day long to inspect and monitor the area.

We have proposed a person re-identification system which makes it easier for the security personnel to overcome the hectic routine of 24/7 presence in front of the surveillance cameras and if somehow, they are unable to detect the person involved in some false activity either due to the negligence of a security personnel or due to any other reason, our system will assist them in tracking that person and will help them by showing his previous appearances in that area.

We will be focused on developing a lightweight application which could easily be deployed on low-end devices because it requires a lot of spending to have access to heavy GPU's which the end user is not likely to spend. We will be using re-identification in place of other techniques as it is profitable due to its simplicity and low cost associated to it as it will be used with standard cameras. As said earlier, our focus it to develop a lightweight application which could easily be run at low-end devices as it decreases the computational cost and eliminate other unnecessary expenditures associated with other tracking devices.

Chapter 2

Requirements Analysis

2 Requirements Analysis

2.1 Literature review

A lot of work has been done in the field of visual tracking and re-identification through various approaches. A few research and application based related systems are discussed below.

2.1.1 Research-Based Related work

1. **Vision based 3-D People Tracking in Smart Room**

In this paper, Dirk et al. proposed an idea of smart room is used which includes cameras tor track people in an indoor environment [1]. Every camera has its own dedicated computer on which the tracking is done. The process of tracking is done after background subtraction where each camera has a background subtraction module which helps in extracting the foreground region.

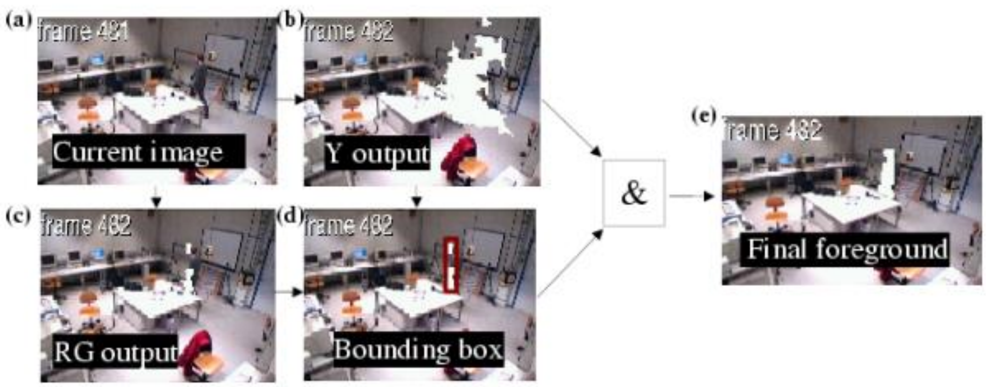

Figure 1: Extracting Foreground [1]

This tracking system was able to track two people using three cameras at 3-7 fps. Fifty seconds sequences of two people walking in the room were recorded at 5 fps.

Until the subjects were 1m apart this system was working fine, because when subjects came very close the background subtraction merged them in foreground region and one of the subject's track was lost for a second. So, this system is limited to only track two people which are at least 1m apart from each other at one time which do not fulfil the demands of multi camera multi objects tracking systems. And also, the dedicated computer for each camera is going to be very costly.

2. Pedestrian Detection and Tracking in Video Surveillance System

In this paper, Ujwalla et al. presents us with the challenges and issues one faces while dealing with the detection and tracking in surveillance systems [2]. It states that there were systems before which were performing that task, but they were compromising on the speed and accuracy. Moreover, there were some trackers which were good but due to their high computational cost they were impractical. Their paper has provided the solution which could overcome these problems.

They have used following algorithms for tracking.

- Kernel-based tracking
- Point based tracking
- Silhouette based tracking.

Some of the major challenges that they faced includes noise which worsened the quality of camera stream, deformation of object due to different movement in different frames, blockage of sight (occlusion) by pedestrian (intentionally/unintentionally), change in pose but the combined approach of shape analysis and body tracking help them to overcome these challenges.

In this proposed system tracking has done by tracking the human head and other body parts by means of hierarchical tracking method. This method was good enough to detect or track the people even in the case of partial blockage of sight (occlusion). The drawbacks of this system were high computational cost as it takes millions of parameters as an input and less efficiency of algorithms as compared to other deep learning models.

3. Automated Vision Tracking of Project Related Entities

In this paper, Ioannis et al. discussed the tracking of project related entities like construction site related entities such as materials, equipment and personnel that how to do that with minimal of effort and without any privacy concern [3]. That project related entities can be the personnel working in that site or different type of material or machinery used on that site. The automated system would prove out to be useful in monitoring the progress of the entities and inventory control. It would help in materials management, collision/accident prevention and for security purposes as well.

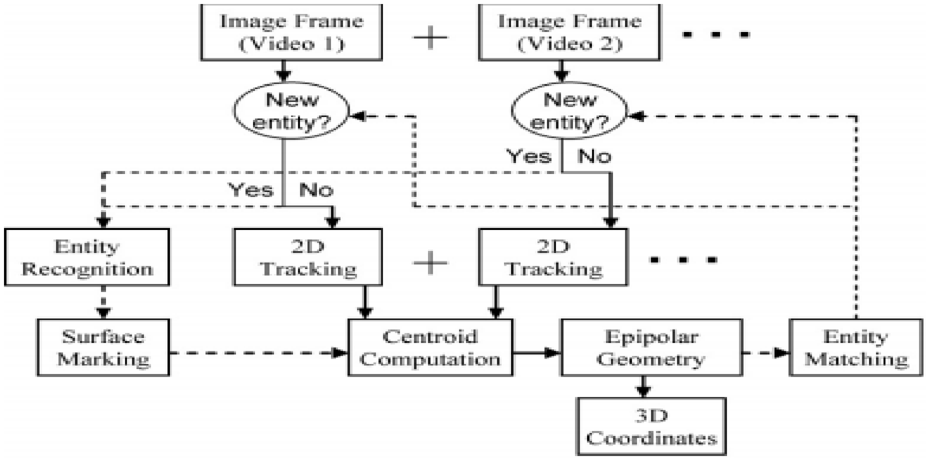

Figure 2: Overview of their Tracking Methodology [3]

They are mainly doing this task to get their project related entities time spent on site and cost of their activities. They used this technique because it proved out to be less costly for them as no additional cost was required as they used standard cameras for their tracking which were already installed on site. For recognition of personnel and equipment, different detection algorithms were used to match their characteristics as the system would extract the unique characteristics of each entity i.e., for personnel they used their uniforms for identification as shown in Figure 3. But that technique does not prove out to be useful in case of equipment because each equipment has distinct features such as structure, shape and texture. For that purpose, their centroid 2D coordinates were then computed and 2D projections of those entities were then tracked using 2D vision tracking technologies as shown in Figure 4.

Figure 3: Personnel Tracking. [3]

Figure 4: Equipment Tracking. [3]

4. Light-weight Multi branch Network for Person Re-identification

In this paper, Fabian et al. proposed a light weight multi branch network based on multi-camera multi-target person re-identification [4]. The network consists of three branches that optimize the global, partial, and channel-wise representations using simple computations respectively. Despite this branching, this model succeeded in keeping the number of parameters low using the OS-Net model, a lightweight feature extractor that has recently proven to be more efficient and accurate than other backbones for RE-ID tasks. This proposed paper uses deep neural network which achieves state-of-the-art results on two important benchmark datasets, Market-1501 and CUHK03. This model gives great efficiency outperforming standard ResNet50 backbones despite a much lower number of parameters. This model uses almost 9 million parameters, but it can be reduced too much almost 3 million parameters to make our model more efficient for low end devices.

5. Spatial-temporal Person Re-identification

Most of current person re-identification (Re-ID) methods neglect a spatial-temporal constraint. When we query an image the feature distance is computed by conventional method between gallery image and our query image and then return a similarity ranked table. When the gallery database is very large in practice, these approaches fail to obtain a good performance due to appearance ambiguity across different camera views.

In this paper, Guangcong et al. proposed a novel two-stream spatial temporal person Re-ID (St-Re-ID) framework that mines both visual semantic information and spatial-temporal information [5]. They introduced similarity metric integrated with Logistic Smoothing (LS) and create a unified framework which can unify two different types of information. St-Re-ID aims to exploit both the visual feature similarity and the spatial-temporal constraint in a unified framework. To this end, we propose a two-stream architecture which consists of three sub-modules as follows

- Visual feature stream.
- Spatial-temporal stream.
- Joint metric sub-module.

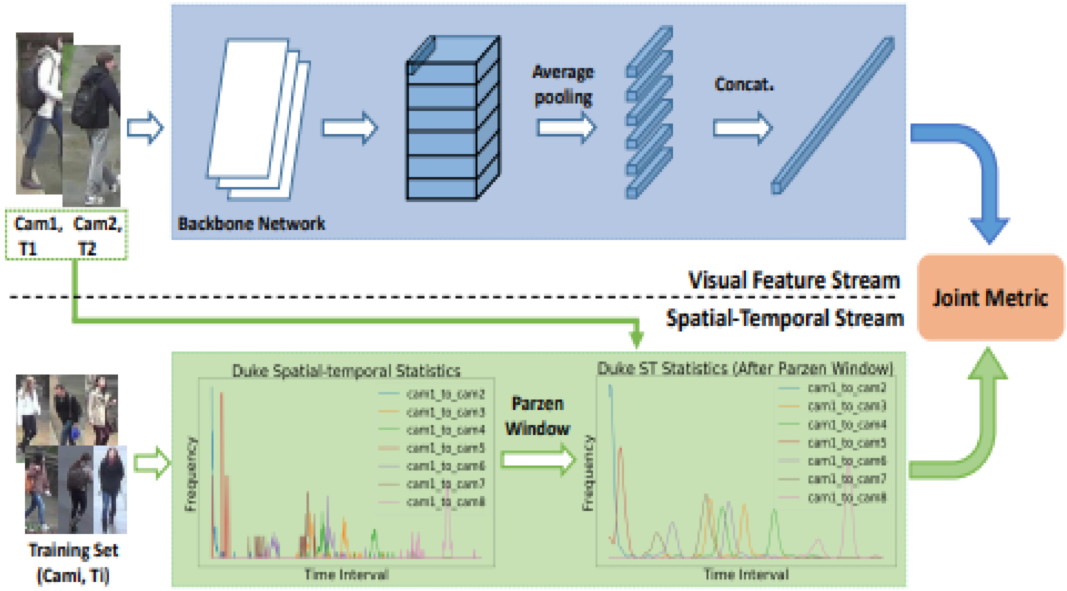

Figure 5: The proposed two-stream architecture. It consists of a visual feature stream, a spatial-temporal stream, and a joint metric sub-module. [5]

This model offers high efficiency but the only drawback in this model is the computational cost, as we cannot run this model on the low-end devices.

6. Top DropBlock for Activation Enhancement in Person Re-Identification

In this paper, Rodolfo et al. introduced the Top Drop-Block Network (Top-DB-Net) for the Re-ID problem [6]. BDB Network is used as baseline because it is quite similar with the approach that we are going to use. The method they used proved to be more worthy as it dropped the features with top activations only whereas in BDB network random features get dropped. Another major difference it has with BDB Network is that it creates an independent drop mask whenever there is an input based on its top activations.

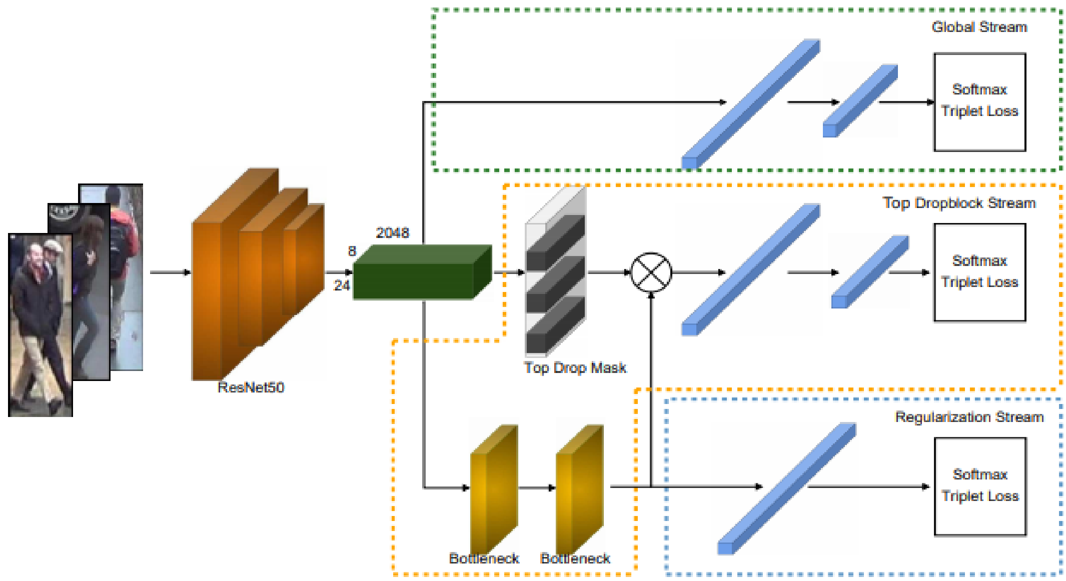

Figure 6: Proposed Top DropBlock Network (Top DB-Net). [6]

Top-DB-Net is designed to further push networks to focus on task-relevant regions and encode low informative regions with discriminative features. This method is based on three streams consisting of

- A classic global stream which is one of the state-of-the-art methods that follows the technique or standard feature encoding.
- Second stream is Top DropBlock which is the reason due to which the network can maintain its performance. It drops the most activated horizontal stripes of feature tensors to enhance activation in task discriminative regions and improve encoding of low informative regions.
- A third stream regularizes the second stream avoiding that noise which was previously generated by dropping the features.

The results obtained from that system suggested that the method which they proposed for interpretability of activation maps could help in the process of re-identification, but they want to further expand their systems domain as Top DropBlock is defined here by dropping ratios and using horizontal stripes only. They want to encode more high-level features in their system and would want to see its impact on the re-identification process.

2.1.2 Application-Based Related Work

1. Person Re-identification using Pose-Normalized and Occlusion Independent Training

In this project, Aditya et al. aimed at developing a model using Deep Neural Networks (DNN) which will be able to handle input images with variable input size [7]. They used two pre-processing techniques to reduce the possibility of overfitting. One of the major challenges one faces during person re-identification is pose variation and occlusion. When a person is moving away or towards the camera, his body movement and pose is not static and is mostly unconstrained which in return affects the re-identification process to a larger extent.

Figure 7: Pose and Camera Viewpoint variation of the person. [7]

They have used two pre-processing techniques, random erasing and pose normalization. After pre-processing they generated the training set for generative adversarial network (GAN). They used ResNet-50 model as backbone of their system. They have built a Pose-Normalized GAN based re-identification model and used open-pose to draw the pose of the training images from Market-1501 image dataset.

Figure 8: GAN generated image after 12 epochs. [7]

After experimenting, it was seen that increasing the number of epochs doesn't improve the performance of the network. The highest performance was met by inclusion of random erasing along with Part-Based Convolutional Baseline (PCB).

2. Person Re-identification using Appearance

In this project, Prerna has tried to build a person re-identification algorithm which will use three appearance cues out of which one will be the dominant colours like same colour or textures in sports teams which help in re-identifying the person [8]. Second and third method will be the colour histograms and colour invariants with the help of which they tried to achieve good results without even training on different datasets. They applied the above methods on the three different datasets, one was of volleyball game, one from soccer and one pedestrian dataset. After experimenting, it was concluded that the colour invariants method was the more accurate among the three methods. The other two methods were not very good as compared to the colour invariants approach as they both use the holistic approach. Every method has its own condition where it performs better. Normally, colour histograms approach gives better results than the dominant colour approach but in case of person wearing same uniform i.e., players of some team, the dominant colour approach would prove better. They have moved to the re-identification process using appearance cues because it is not easy to re-identify person based on their jersey numbers and facial recognition as it is itself a very challenging problem.

Figure 9: Snapshot from EPFL volleyball dataset. [8]

They tested their system on the three datasets to evaluate their performance and the method which gave the best accuracy was colour invariants as it was quite hard to differentiate between teams and referees in sports dataset and then re-identification was also quite challenging in other two methods because of the occlusion and closed system as they had to perform re-identification in the court or soccer field area only. Maximum accuracy achieved in volleyball dataset was 90% and in soccer dataset it was 89.921% and both were achieved by colour invariant using its hist signature. In case of pedestrian dataset, the maximum accuracy achieved was 79.444% by colour invariants using comb signature. These results are calculated after 10 frames.

2.2 Stakeholders List

Stakeholders of our project will be

1. Project Team
   - Muhammad Barak Ullah
   - Muhammad Saim Ashfaq
   - Burhan Sabir
2. Project Supervisor
   - Dr. Usama Ijaz Bajwa
3. CUI Lahore

2.3 Requirement elicitation

2.3.1 Functional requirements

Functional requirements define the scope and working of system components in a way that is feasible for both the user and the team developing it. It defines how system's components are going to operate. It is a description of the facility that the software would perform.

1. **Functional Requirement-01: Login**

Functional requirements of login are mentioned in Table 1.

Table 1: Functional Requirement-01: Login

ID	Functional Requirements
FR11	The application asks the user to enter his username and password.
FR12	The application authenticate username and password after user enters his login information and clicks on the login button.
FR13	The application grant access to the user after verification of login credentials.
FR14	The application returns an error message in case of wrong input of login credentials.

2. Functional Requirement-02: Logout

Functional requirement of logout is mentioned in Table 2.

Table 2: Functional Requirement-02: Logout

ID	Functional Requirements
FR21	User redirected to the login page after clicking on the logout button.

3. Functional Requirement-03: Save person's record into Database

In Table 3, functional requirements of saving record into database are mentioned.

Table 3: Functional Requirement-03: Save person's record into Database

ID	Functional Requirements
FR31	The application saves the record of every person that comes in our system domain into the database including his ID and appearance time
FR32	The application allocates new ID to every new person appearing in the stream if his/her data is not present in our database.

4. **Functional Requirement-04: Save Frames**

In Table 4, functional requirement of saving frames into database are mentioned.

Table 4: Functional Requirement-04: Save Frames

ID	Functional Requirements
FR41	The application allows the user to save frames of the stream into the database along with other data which is already being saved.

5. **Functional Requirement-05: View Report**

In Table 5, functional requirements of viewing the report are mentioned.

Table 5: Functional Requirement-05: View Report

ID	Functional Requirements
FR51	The application allows the user to view report of person appearing in the video stream.
FR52	The report contains that person's ID, time slots when he previously appeared and through which cameras, he/she has passed.
FR53	The application returns an error message and blank report if the person data is not present in our system

2.3.2 Non-functional requirements

Non-functional requirements describe how the system should behave and evaluate the performance of the system. These requirements cover up all the leftovers from the functional requirements.

1. **Non-functional Requirement-01: Performance**

Non-functional requirements regarding performance of the application are mentioned in Table 6.

Table 6: Non-functional Requirement-01: Performance

ID	Non-Functional Requirements
NFR11	Application's response time is not affected irrespective of the mass interacting with it.
NFR12	The application's response time is minimum when user interacts with the app.

2. Non-functional Requirement-02: Usability

Non-functional requirements regarding ease in usability of the application are mentioned in Table 7.

Table 7: Non-functional Requirement-02: Usability

ID	Non-Functional Requirements
NFR21	The interface of the application is user friendly.

3. Non-functional Requirement-03: Security

Non-functional requirements regarding security of the application are mentioned in Table 8.

Table 8: Non-functional Requirement-03: Security

ID	Non-Functional Requirements
NFR31	Only authenticated users would be allowed to access the applications functionalities.
NFR32	The application ensures both data integrity and privacy of the user.

4. Non-functional Requirement-04: Maintainability

Non-functional requirements regarding maintainability of the application are mentioned in Table 9.

Table 9: Non-functional Requirement-04: Maintainability

ID	Non-Functional Requirements
NFR41	The application is maintained properly. If a problem occurs in the application, should be fixed as soon as possible.
NFR42	The application should be updated and repaired whenever need occurs.

5. Non-functional Requirement-05: Compatibility

Non-functional requirements regarding compatibility of the application with other devices are mentioned in Table 10.

Table 10: Non-functional Requirement-05: Compatibility

ID	Non-Functional Requirements
NFR51	The application is compatible with all the desktop devices.
NFR52	The application is not undergoing any major changes when shifting to another operating system.

2.3.3 Requirement traceability matrix

The requirements traceability matrix is displayed in Table 11.

Table 11: Requirements Traceability Matrix

FR_ID	Requirement description	Objective	Priority
FR-01	The system facilitates the user to login into the system.	User account logged in.	High
FR-02	The system allows the user to logout of the system.	User logs out of the system.	High
FR-03	The system allows the user to upload person's data into the	Upload person's record into database.	High

	database.		
FR-04	The system allows the user to save frames of the person into the directory.	Save person's frames into directory.	High
FR-05	The system allows the user to view report of the suspected person.	View Report	High

2.4 Use case descriptions

2.4.1 Use case-01: Login

The use case description for the login process is shown in Table 12.

Table 12: Use Case Description-01: Login

ID:	1
Name of Use Case:	Login
Actors:	User, Database
Description:	The user can log in to the application.
Pre-Condition:	User should already have an account. User must know his/her login credentials of his/her account.
Post-Condition:	User has been signed into the application, successfully.
Events:	1. User fills the username and password fields. 2. On authentication from the database, the user will be logged in.
Alternatives Flow:	In case of forgetting the password, the user must click on the forget password and reset his password through his email.
Exceptions:	None

2.4.2 Use case-02: Re-training model

The use case description for the re-training of anomaly is shown in Table 13.

Table 13: Use Case Description-02: Re-training model

ID:	2
Name of Use Case:	Re-training model
Actors:	Directory, Trained Model
Description:	The application shall retrain the model on the trained model itself or by choosing dataset from the directory.
Pre-Condition:	Trained model should exist within the system.
Post-Condition:	The trained model is re-trained.
Events:	1. The application gets the live stream from the cameras or choose the dataset from the Directory. 2. The application passes it through the trained model. 3. The model is retrained by new training examples.
Alternatives Flow:	No incoming stream from camera or no dataset available in the directory.
Exceptions:	None

2.4.3 Use case-03: Detection and Tracking

The description of the use case of the detection and tracking process is shown in Table 14.

Table 14: Use Case Description-03: Detection and Tracking

ID:	3
Name of Use Case:	Detection and Tracking

Actors:	Directory, Trained model
Description:	The application shall save the frames as well after the process of Detection and Tracking.
Pre-Condition:	User should select video for detecting and tracking.
Post-Condition:	Every person in video is having bounding box and being tracked.
Events:	1. The trained model detects the person. 2. The trained model will create bounding boxes around that detected person. 1. That detected person will be then tracked.
Alternatives Flow:	No stream available whose frames would be saved.
Exceptions:	None

2.4.4 Use case-04: Re-identification

The use case description for re-identification the model is shown in Table 15.

Table 15: Use Case Description-04: Re-identification

ID:	4
Name of Use Case:	Re-identification
Actors:	Database, Trained Model
Description:	The application shall get video stream from the camera/directory and send preprocessed videos to the trained model for re-identification and then save record into the database.
Pre-Condition:	There should be a tacked video stream coming in from the directory.
Post-Condition:	The person is re-identified, and its record is being saved.
Events:	2. The tracked video will be coming from

	directory. 3. That detected and tracked person will then be re-identified if his/her data is already present in the database. 3. After re-identification, the record of that person will be saved into the database.
Alternatives Flow:	If the detected person is not present in the database, a new ID will be generated against that person and its record will be saved into the database.
Exceptions:	None

2.5 Use case design

2.5.1 Use case-01: Login

The use case for the login process for the user and admin is shown in Figure 10.

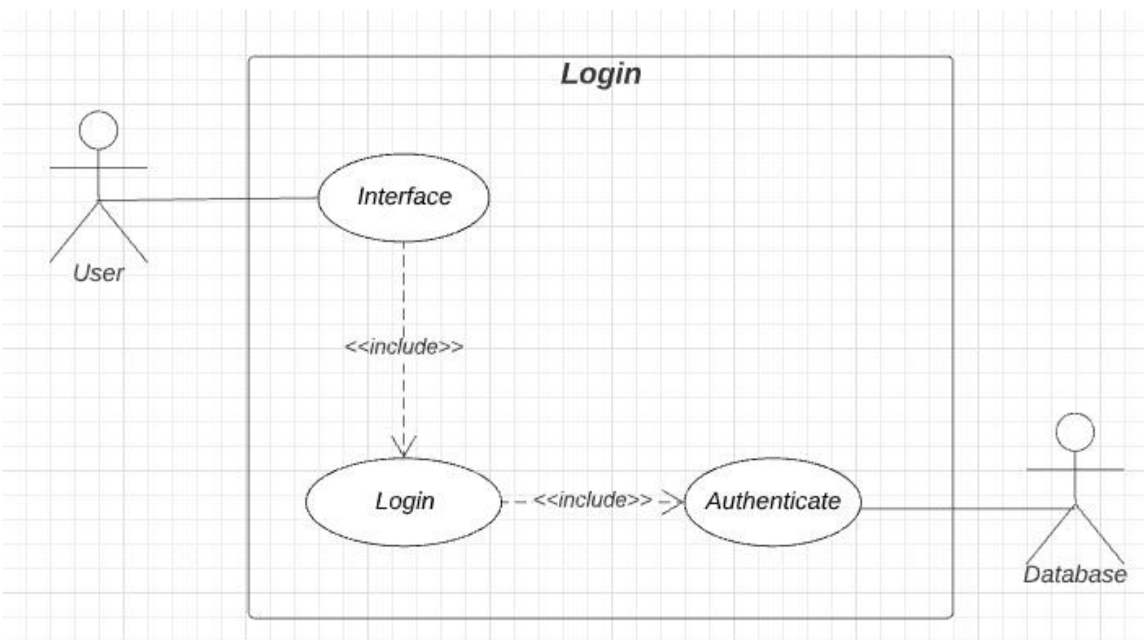

Figure 10: Use case-01: Login

2.5.2 Use case-02: Re-training model

The use case of re-training of model is shown in Figure 11.

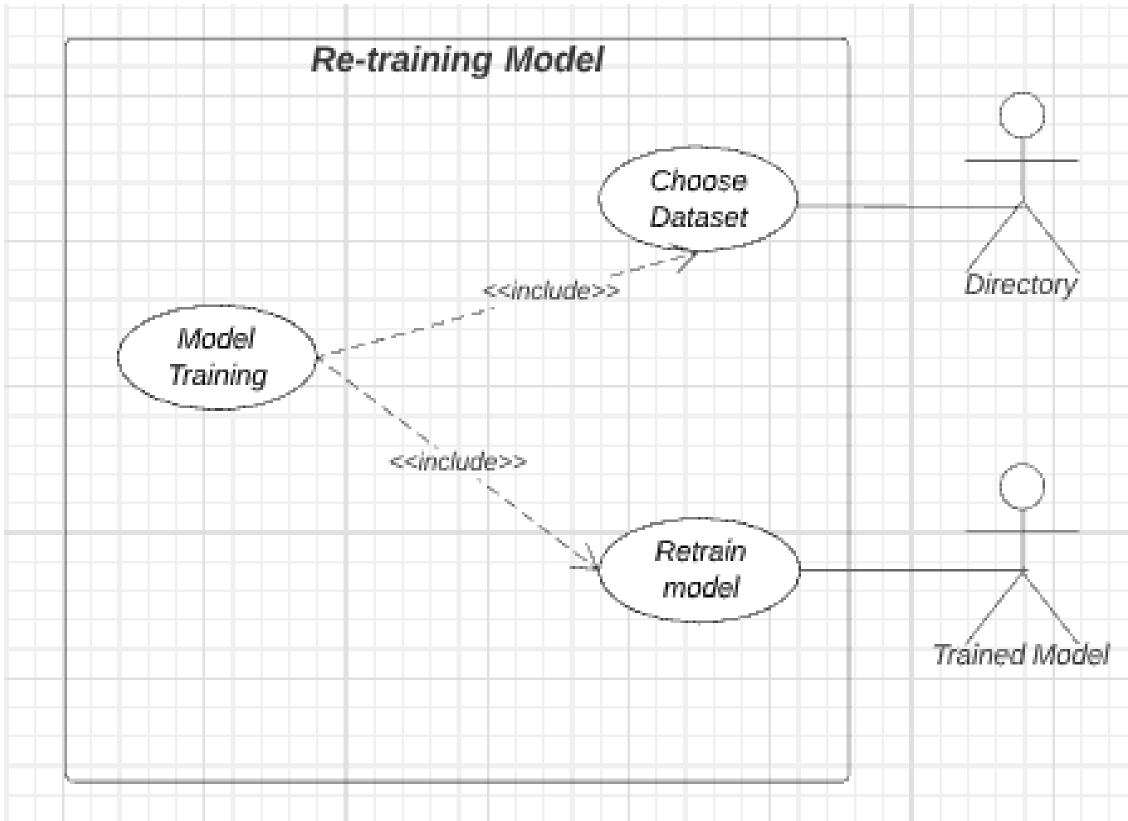

Figure 11: Use case-02: Re-training model

2.5.3 Use case-03: Detection and Tracking

The use case for the detection and tracking process is displayed in Figure 12.

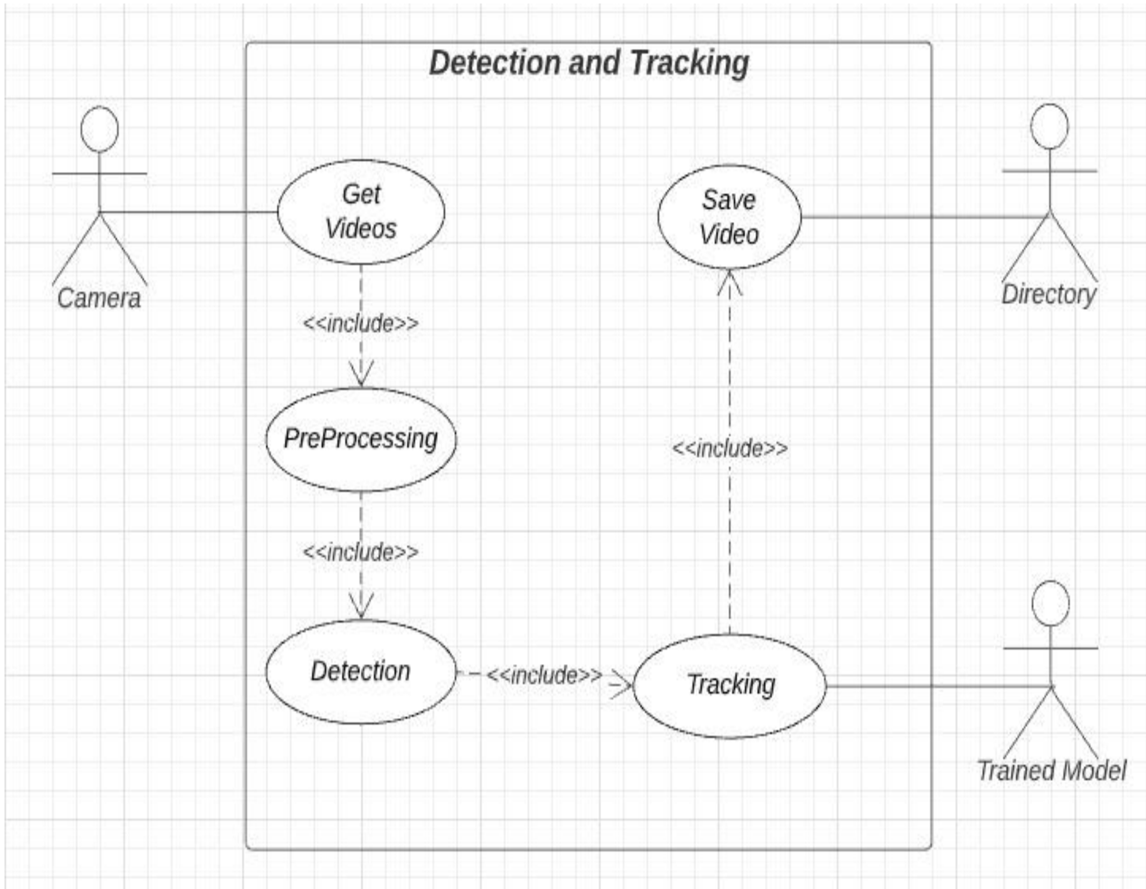

Figure 12: Use case-03: Detection and Tracking

2.5.4 Use case-04: Re-identification

The use case for the process of re-identification is displayed in Figure 13.

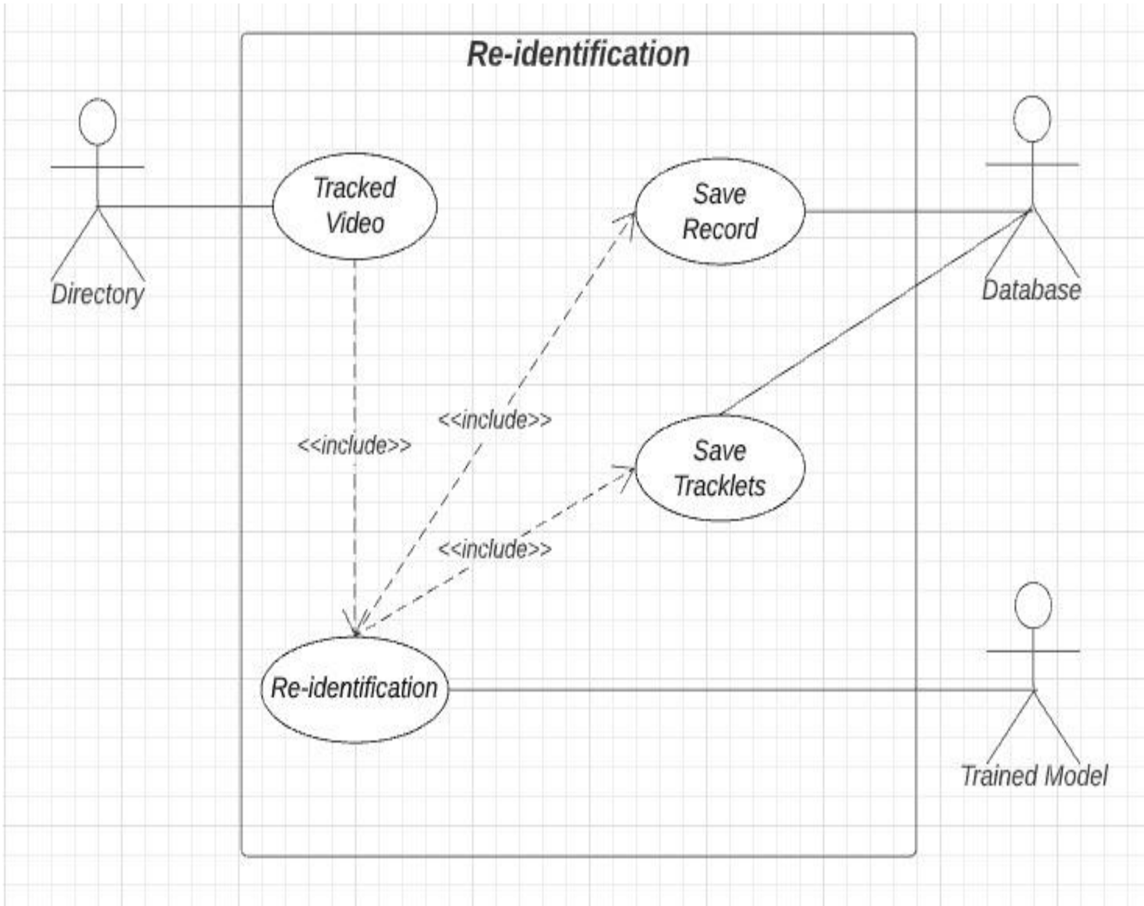

Figure 13: Use case-04: Re-identification

2.5.5 Use case-05: RAVEN (Complete System)

The use case in this Figure 14 describes the overview of the entire system by combining all modules in one diagram for a better understanding of the reader.

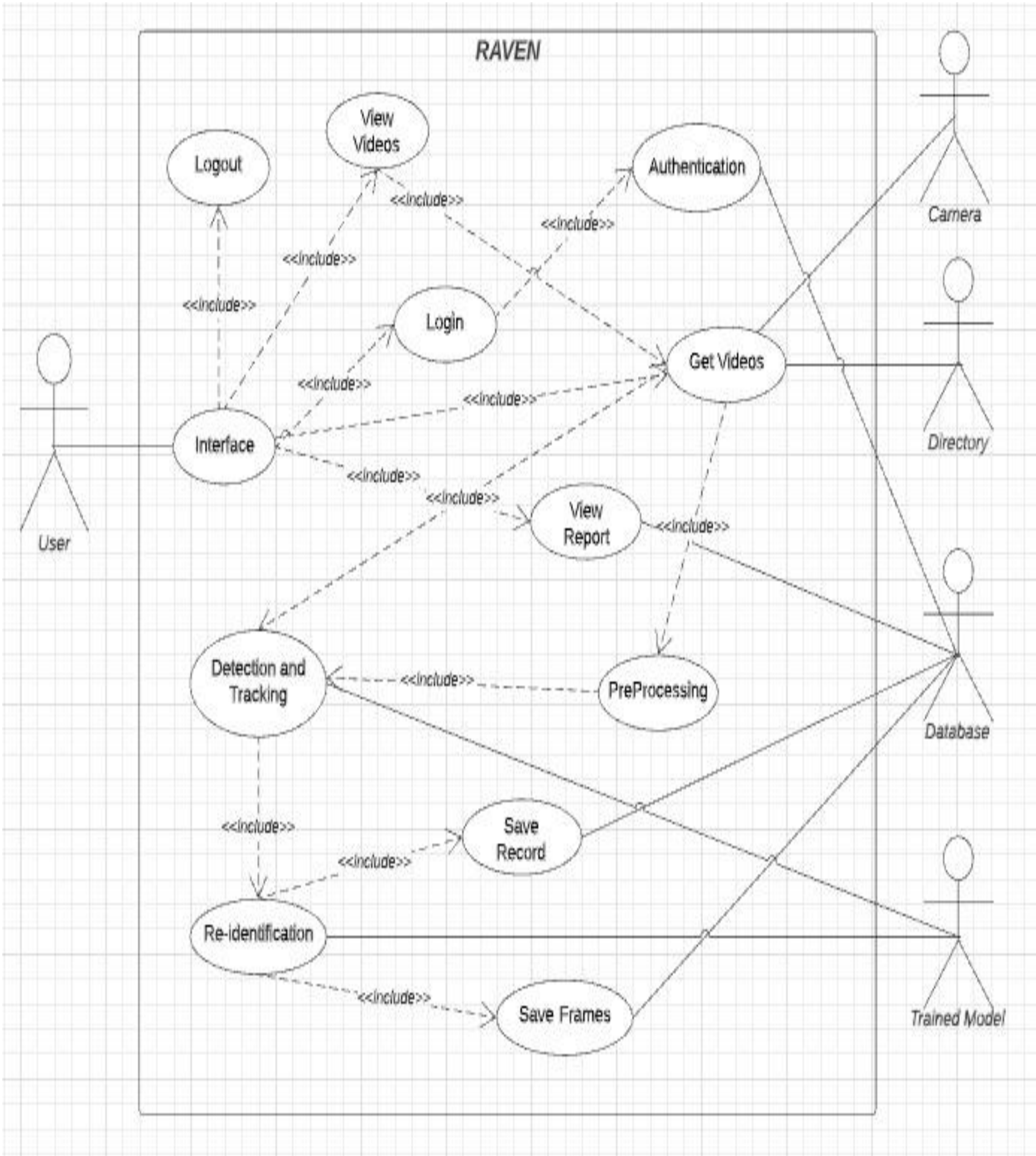

Figure 14: Use case-05: RAVEN

2.6 Software development lifecycle model

The software development life cycle is a conceptual outline in which all the activities that come during the software life cycle are represented including planning, designing, building, coding, testing, deployment and maintenance. It is an activity where we identify our problem requirements, their solution and implementation and the development of the project.

2.6.1 Model Used in our project:

We are using **Incremental Model** for the development process of our system. In this model, requirements are broken down into multiple modules of the cycle which helps in breaking the bigger problem into multiple smaller easily managed modules. It is an iterative model, and each module is built one by one and is then integrated with the other modules.

2.6.2 Why?

We have chosen this model because this model helps in debugging process and reduce risks that come during development process as it handles all the modules separately in iterations. It will be easier for us to alter anything in our project because of the standalone existence of all the modules. As after every iteration, one module would be integrated with other modules which would help in revising our system until we reach our goal. It would be an easier option to test and debug during smaller iterations.

Chapter 3

System Design

3 System Design

3.1 Work Breakdown Structure

The work breakdown structure for our project is shown in Figure 15. It describes different phases of the project including research, design, training, development and testing.

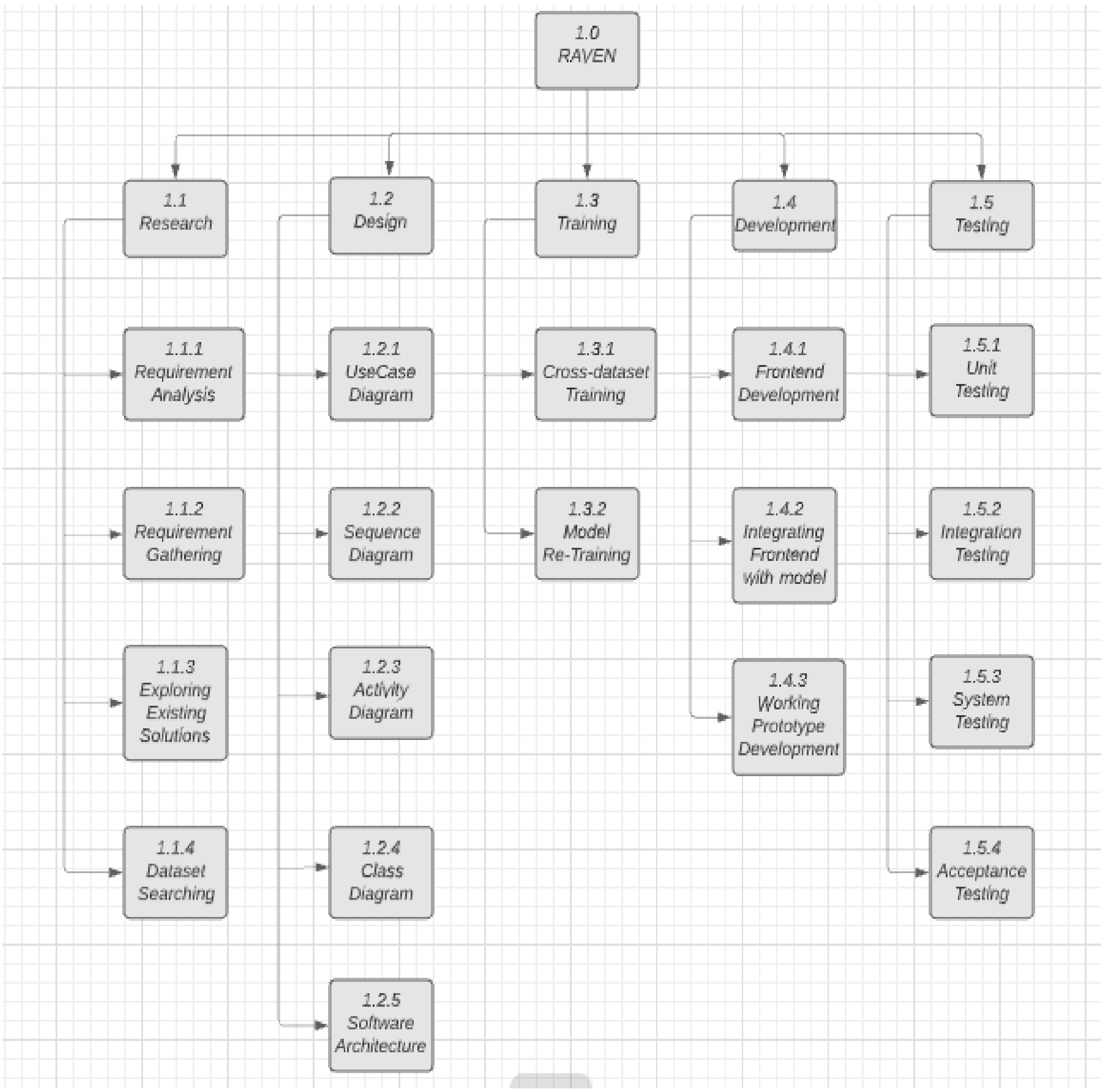

Figure 15: Work Breakdown Structure

3.2 Activity diagram

3.2.1 Login

The activity diagram of login process is shown in Figure 16.

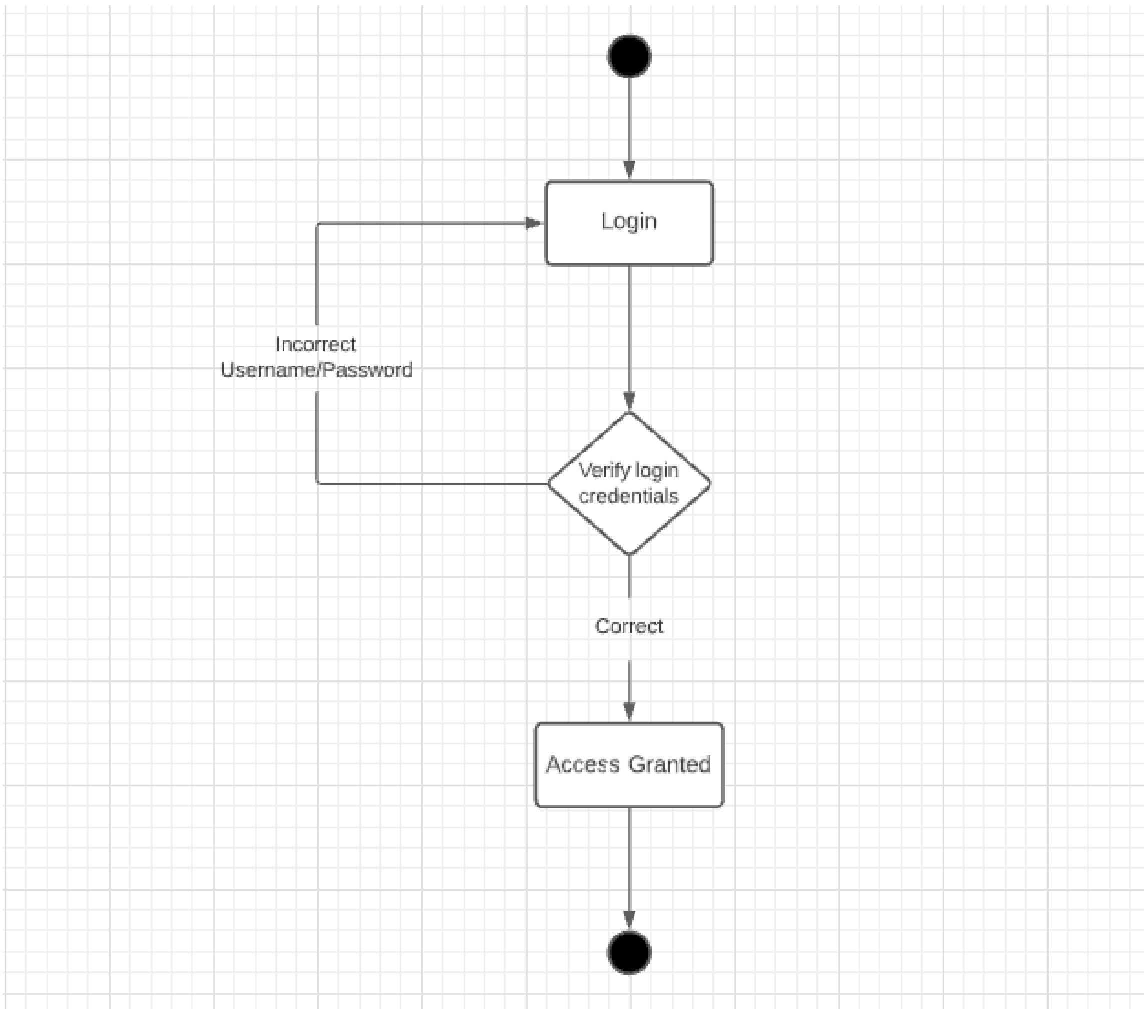

Figure 16: Activity Diagram-01: Login

3.2.2 Re-identification

The activity diagram for the process of re-identification is shown in Figure 17.

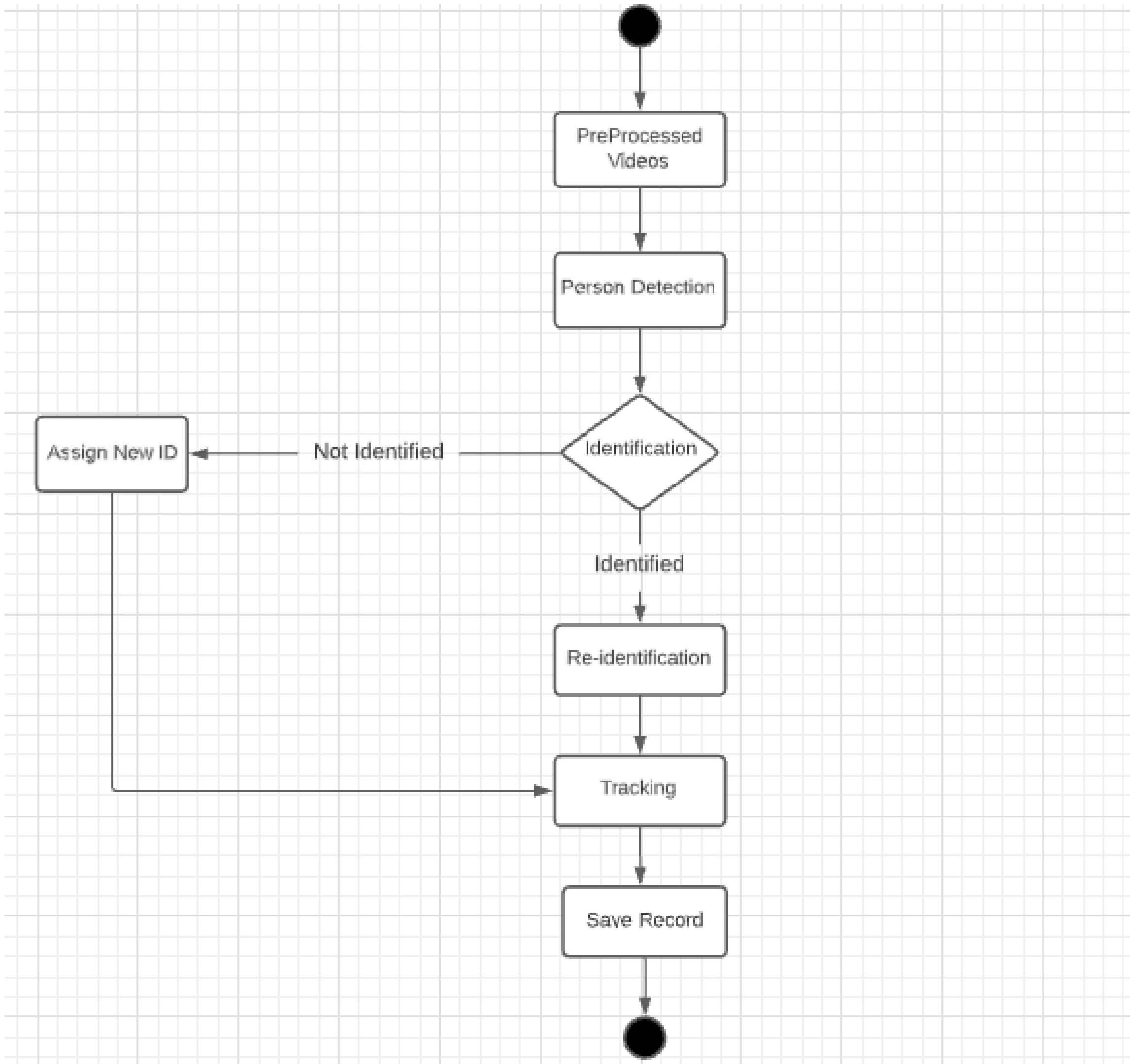

Figure 17: Activity Diagram-02: Re-identification

3.3 Sequence diagram

3.3.1 Sequence Diagram: Login

The sequence diagram for the login process is shown in Figure 18. It shows the sequence of events that occur during the login process.

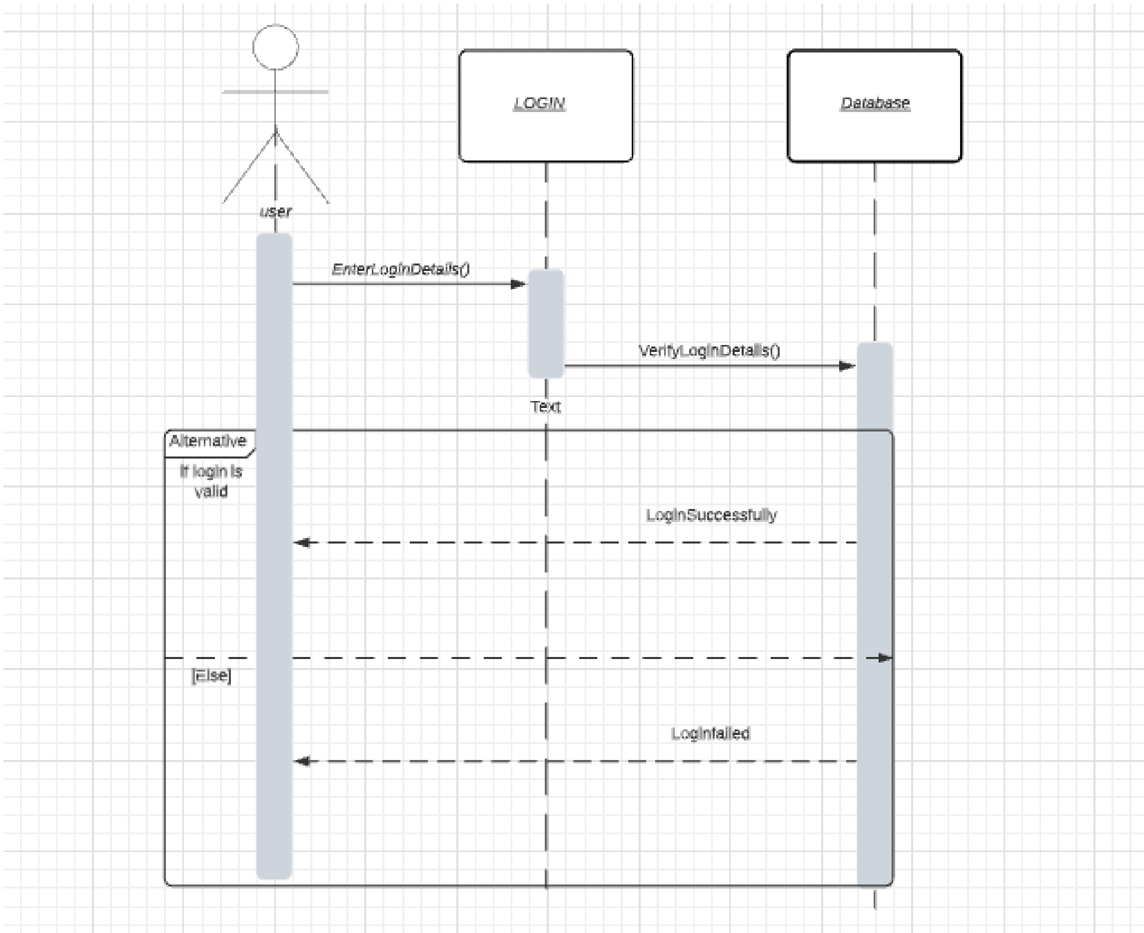

Figure 18: Sequence Diagram-01: Login

3.3.2 Sequence Diagram: Re-identification

In Figure 19, sequence diagram for the process of re-identification is presented. It displays the sequence of how the trained model extract features and then after that the process of re-identification happens.

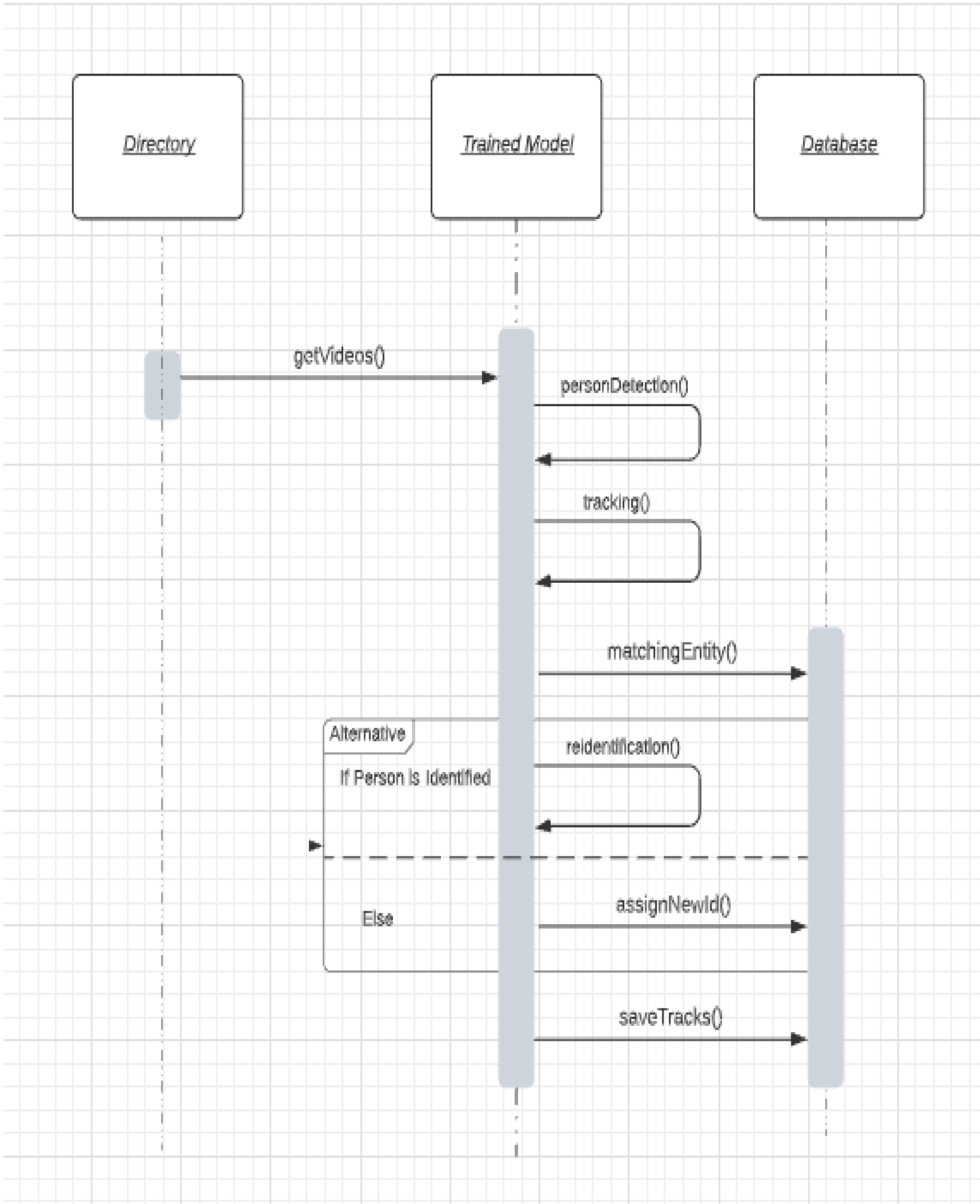

Figure 19: Sequence Diagram-02: Re-identification

3.4 Software Architecture

The application will get live stream from the CCTV cameras and will pass these streams through the trained model. The application will detect, track and re-identify the person. If the person appearing in our system has appeared for the first time, he will be assigned with a new ID otherwise the person will be re-identified, and his track will be saved in our local database by the user/client. Figure 20 represents the software architecture of our system.

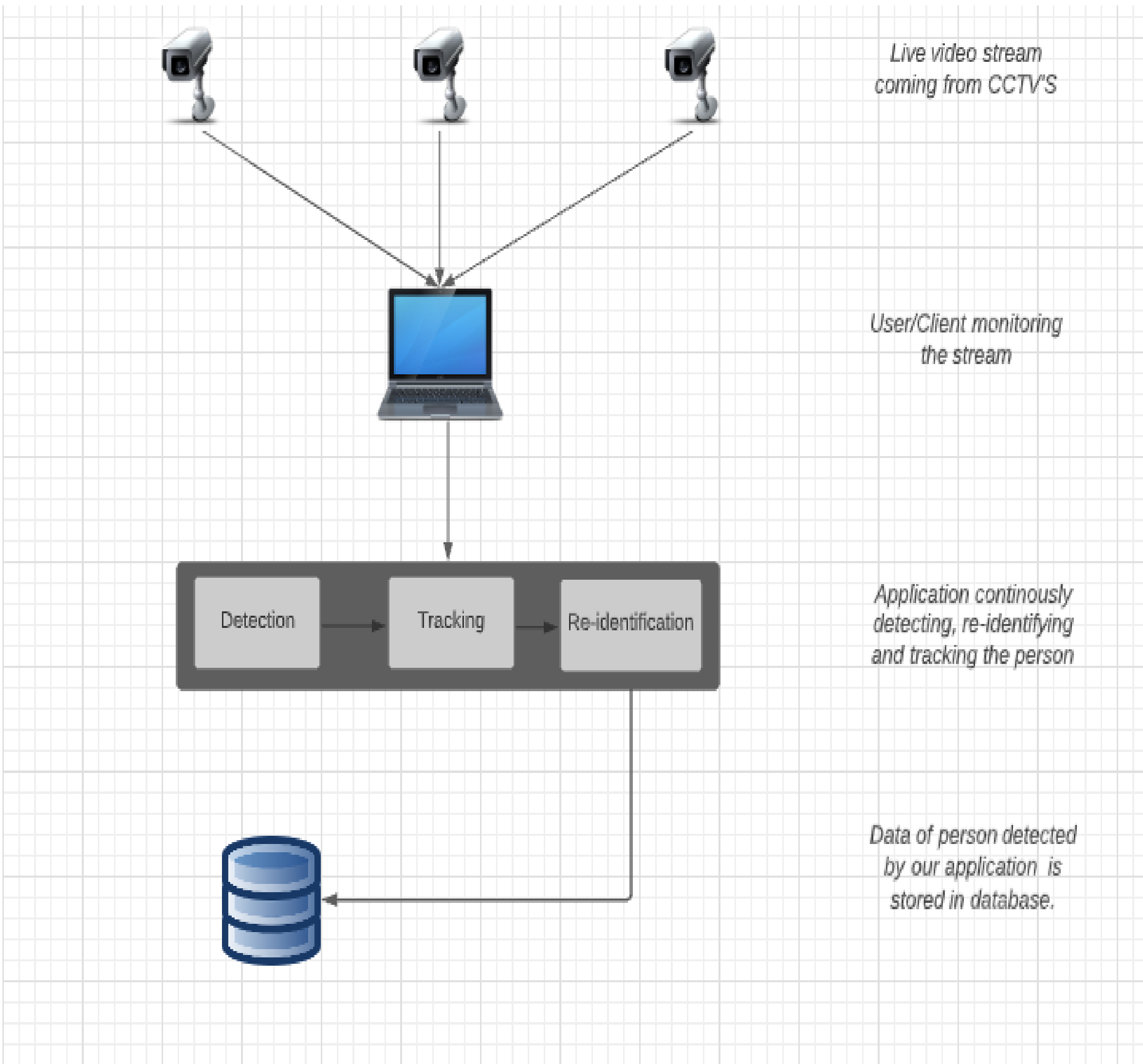

Figure 20: Software Architecture

3.5 Class Diagram

In Figure 21, class diagram of our project is presented which represents the static view of our application. It describes the role of each class that which class is responsible for what functionality in our system.

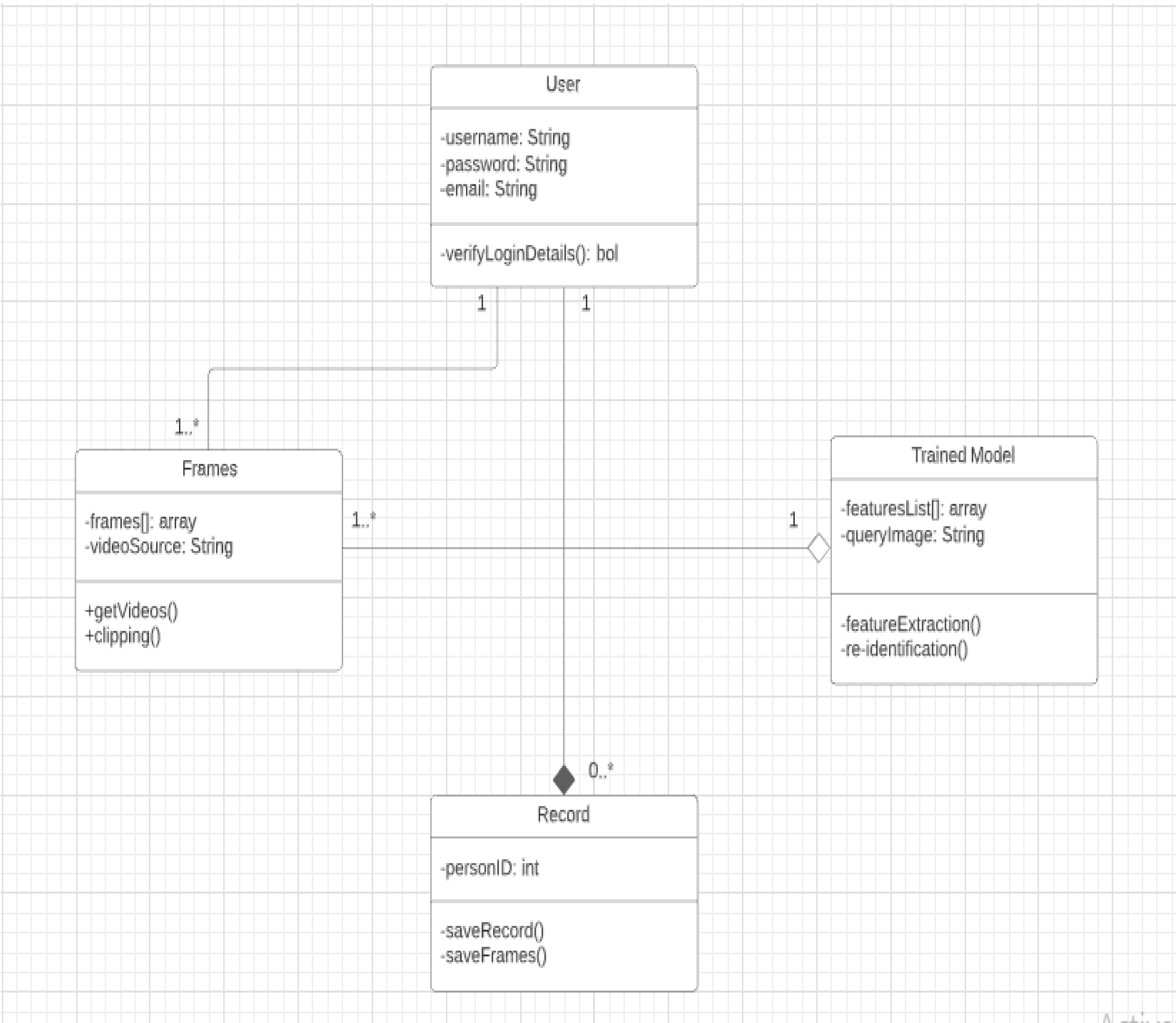

Figure 21: Class Diagram

3.6 Network Diagram

In Figure 22, Gantt chart is presented which represents which tasks would be done in which time frame and it shows in what sequence our project tasks would be done. It helps us in scheduling our tasks which be done in order to complete our project.

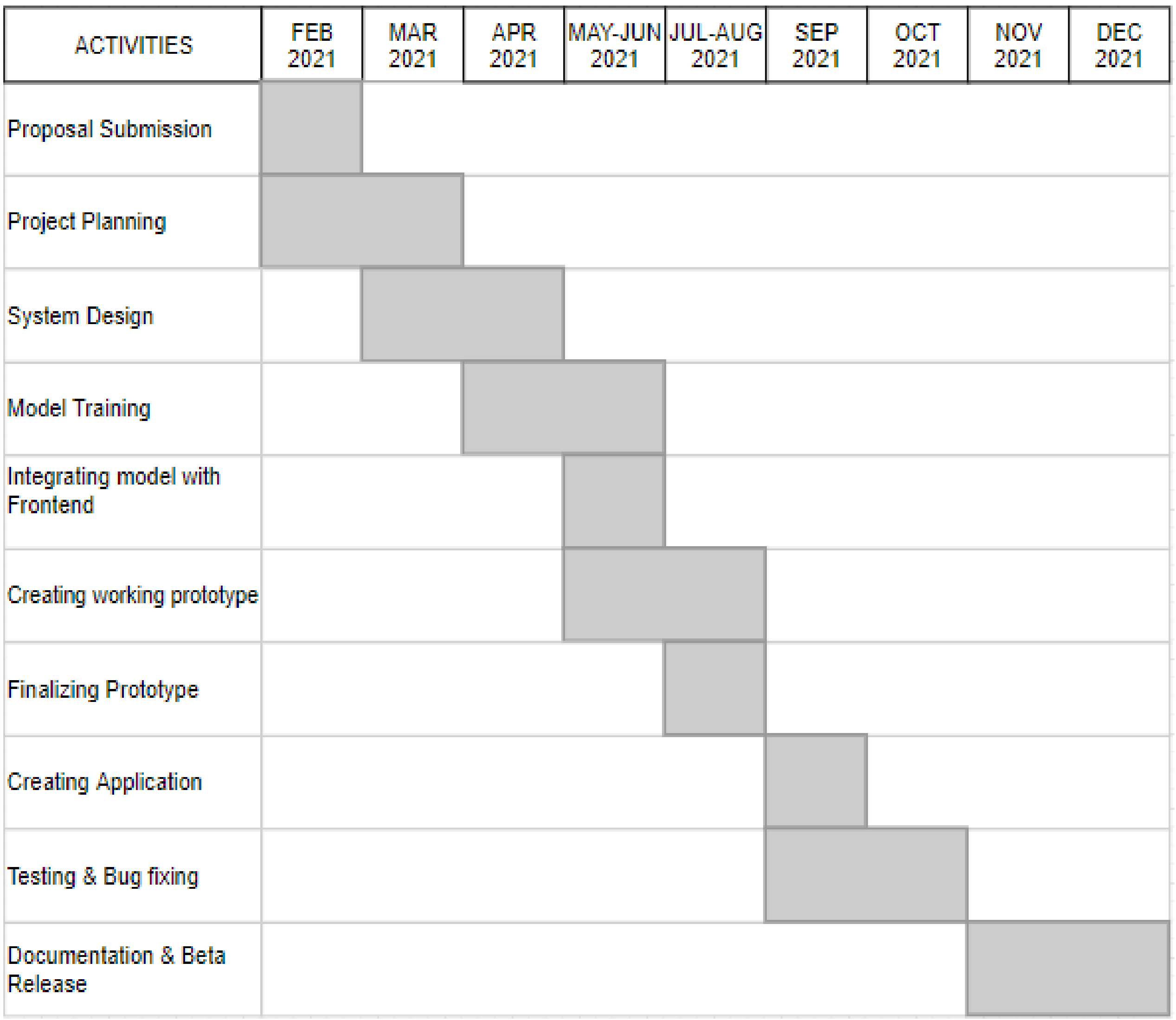

Figure 22: Network Diagram (Gantt Chart)

3.7 Collaboration Diagram

In Figure 23, collaboration diagram of our project is presented which shows the flow of our system in which it is going to work.

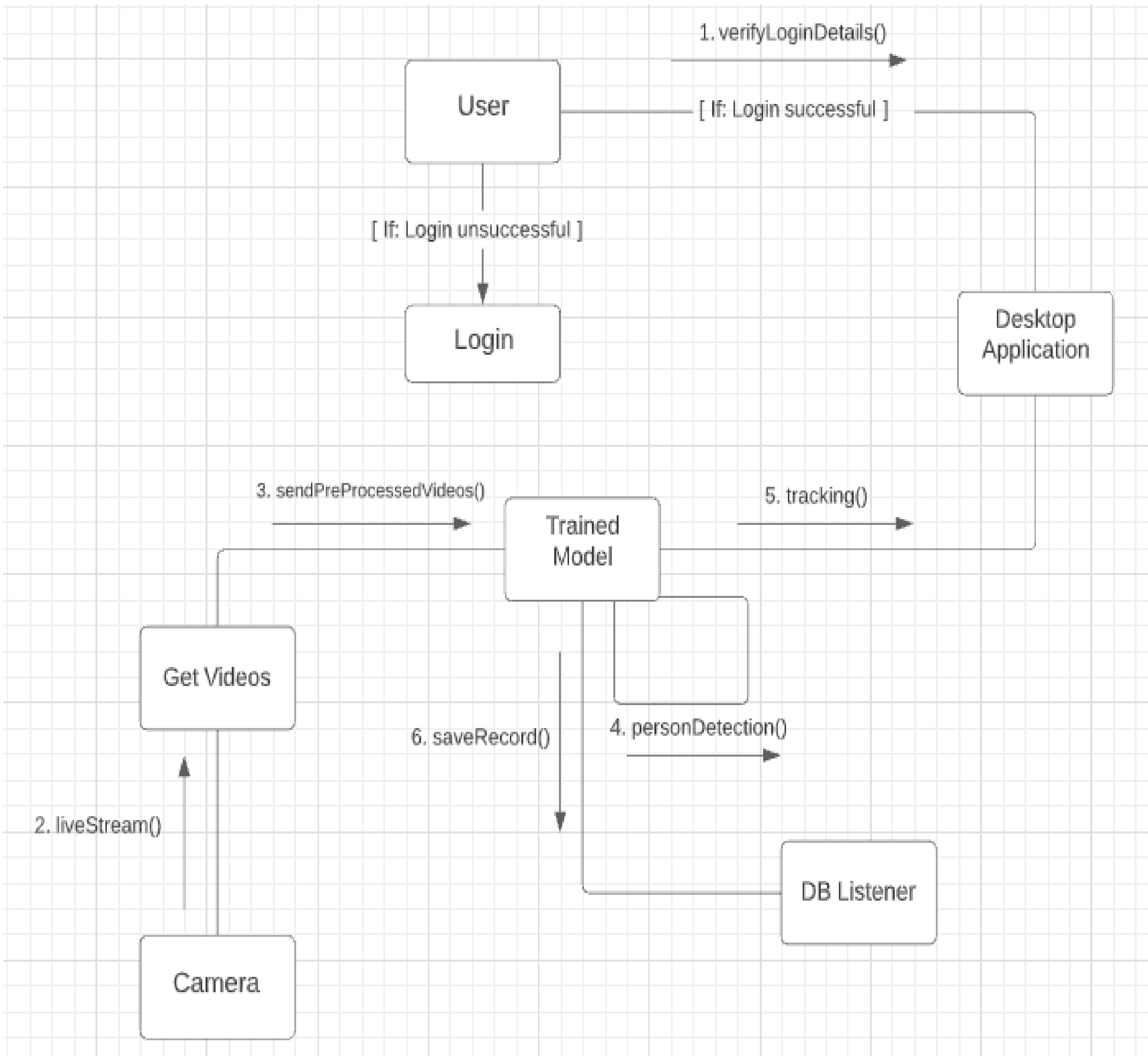

Figure 23: Collaboration Diagram

Chapter 4

System Testing

4 System Testing

4.1 Unit Testing

Unit testing is the sort of software testing in which the components and individual units of the software are tested. It is done to check whether the components of the system are performing as they were expected to perform. In unit tests, the whole system is divided into multiple section and modules and these sections are then tested to check for their correctness. We haven't performed the unit testing yet but for future work, the test cases for the unit testing are:

4.1.1 Test Case-01

In Table 16, test case for the Functional Requirement FR-01: Login is presented.

Table 16: Test Case-01

Test case ID	TC-01
Application Name	RAVEN
Use case	Login
Input Summary	User clicks the login button for logging into the application.
Output Summary	Success: The user has been logged in.
Pre-Conditions	Users must enter correct login information.
Post-Conditions	User has been logged in successfully and is now navigated to the home screen.

4.1.2 Test Case-02

In Table 17, test case for the Functional Requirement FR-02: Logout is presented.

Table 17: Test Case-02

Test case ID	TC-02

Application Name	RAVEN
Input Summary	User clicks the logout button to logout from the application.
Output Summary	Success: User has been logged out.
Pre-Conditions	User must be logged in.
Post-Conditions	User successfully logged out from the application and is redirected to the login page.

4.1.3 Test Case-03

In Table 18, test case for the Functional Requirement FR-03: Save person's record into Database is presented.

Table 18: Test Case-03

Test case ID	TC-03
Application Name	RAVEN
Use Case	Re-identification
Input Summary	The application will get live stream from the camera.
Output Summary	Success: Person's record including person ID, his appearance time and camera no. will be saved into database.
Pre-Conditions	There must be live stream coming from the camera.
Post-Conditions	Person's record has been successfully saved into the database.

4.1.4 Test Case-04

In Table 19, test case for the Functional Requirement FR-04: Save Tracklets is presented.

Table 19: Test Case-04

Test case ID	TC-04
Application Name	RAVEN
Use Case	Detection and Tracking
Input Summary	The application will get live stream from the camera.
Output Summary	Success: Person's tracklets are saved into the directory.
Pre-Conditions	There must be live stream coming from the camera.
Post-Conditions	The frames of the person appearing in the camera are successfully saved into database.

4.1.5 Test Case-05

In Table 20, test case for re-training of the model is presented.

Table 20: Test Case-05

Test case ID	TC-05
Application Name	RAVEN
Use Case	Re-training model
Input Summary	The application will get live stream from camera or choose dataset from directory.
Output Summary	Success: Model is re-trained using new training examples.

Pre-Conditions	There must be live stream coming from the camera or there must be dataset present in the directory.
Post-Conditions	The model is re-trained and will now generate better results with new training examples.

4.2 Integration Testing

In integration testing, we integrate the individual units and components of the system and then test them as a group to see if they are working as they were intended to be. It is done to make sure that the components are integrated without any complications and to check if any problem has occurred after integration of the system components.

The following functionalities were integrated with application:

- Adding video from the directory to detect, track and re-identify a person.
- Save video after detection and tracking and then passing tracked video to the model for re-identification.
- Integration of application with database so that person's appearance time could be saved.

The application was tested and checked. Our application passed the integration testing phase.

4.3 Acceptance Testing

Acceptance testing is the final step in the software testing procedure. At this final level of testing, we check whether our system has met the requirement specifications. In this step, we make sure that the system under test is complete with the basics and necessities required.

We tested our application against the following requirements:

- Authorized user login.
- Detection and Tracking of person.
- Re-identification of person from the tracked video.
- Saving record of person in the database.
- Saving tracklets of person in the directory.

After successive stages of development, our application now fulfils the above listed requirements.

Chapter 5

Application End

5 Application End

5.1 User Interface

This section contains screenshots of the various screens of our desktop application Raven.

5.1.1 Welcome Screen

Figure 24 depicts the welcome screen of our desktop application which is developed in python's library using PyQt5. The user will be able to log in and create new account.

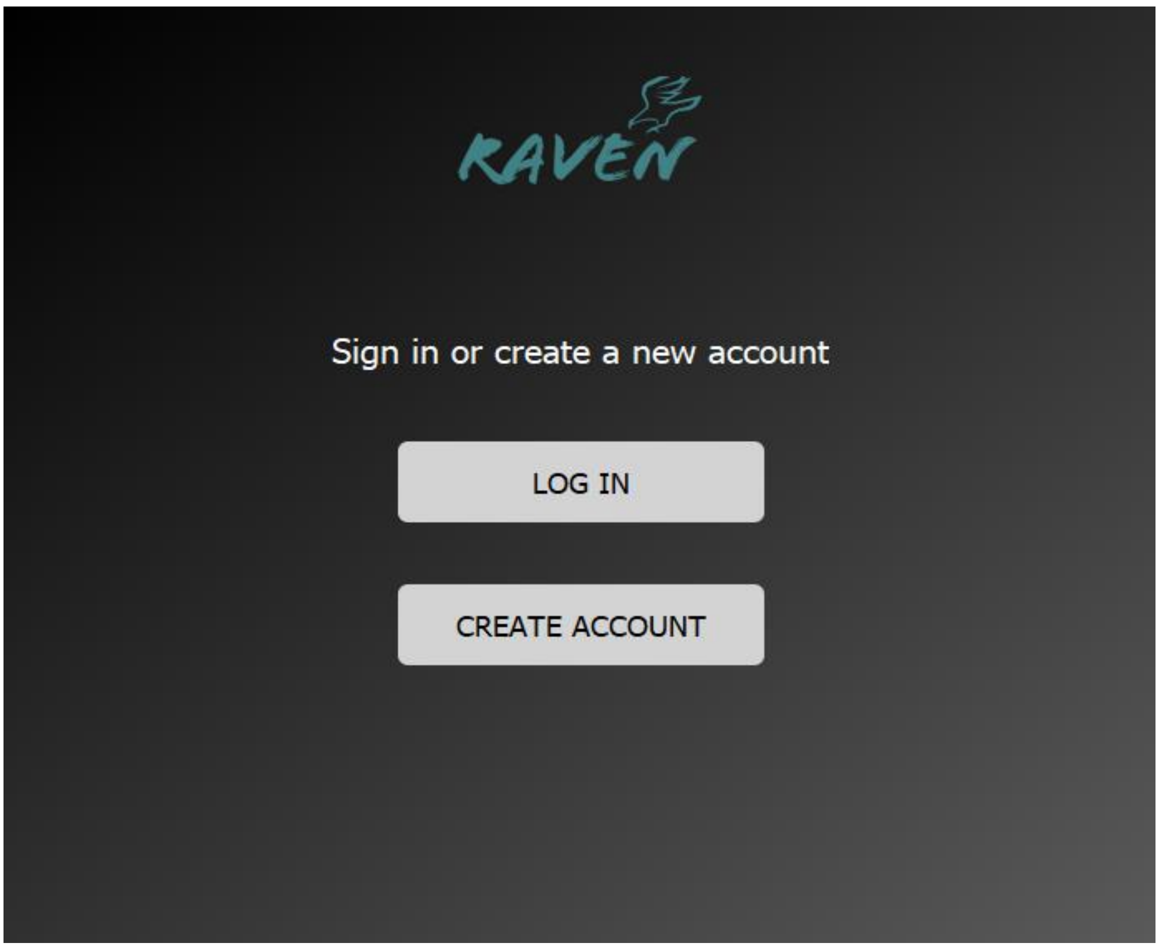

Figure 24: Welcome Screen of Raven

5.1.2 Sign up

Figure 25 depicts the sign-up screen of our desktop application. The user will be able to register into our system by filling in the required fields.

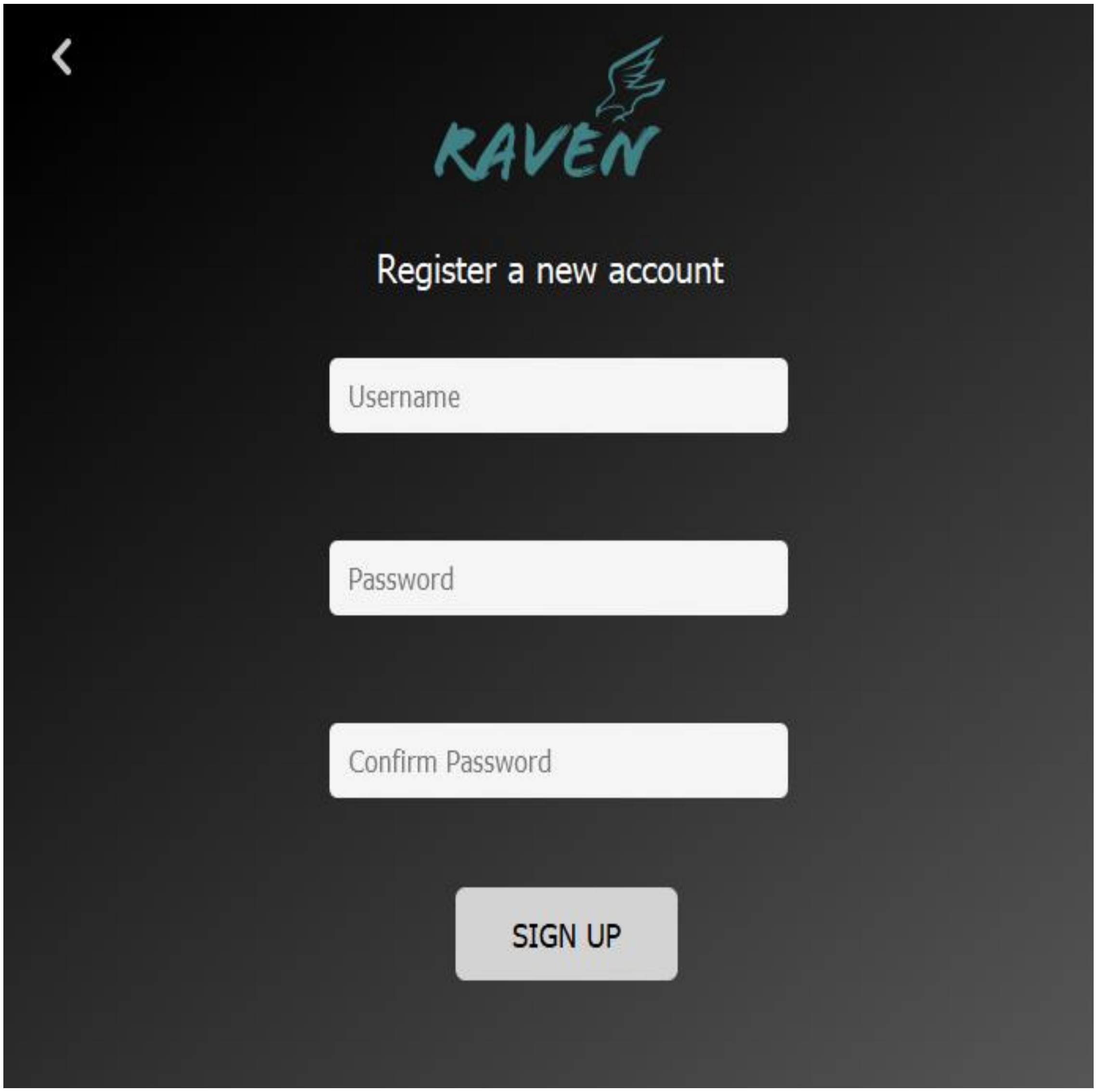

Figure 25: Sign up Screen of Raven

5.1.3 Login

Figure 26 depicts the login screen of our desktop application. The user will enter his username, and password which will be validated from the database and in case of correct credentials, user will be logged into the application.

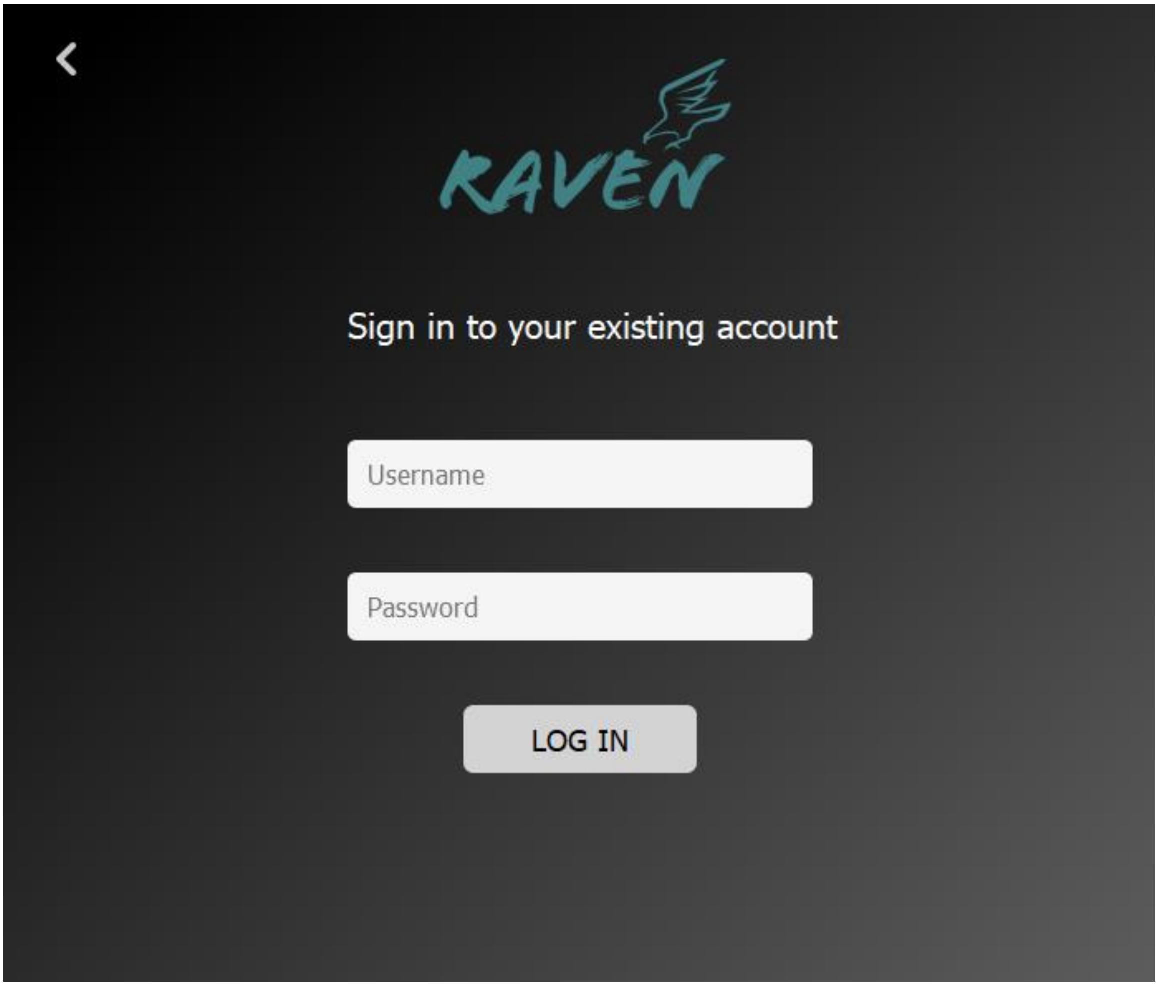

Figure 26: Login Screen of Raven

5.1.4 Home Screen

Figure 27 depicts the home screen of our application Raven. On the left side, we have a vertical menu bar which shows different modules of our application. We can select video from gallery or open video from webcam and perform re-identification on that. Other two modules named view video and view report shows the appearances of person that appeared in our system (which the user can search by passing id in the search bar. On the top right corner, logout button is displayed which will logout the user from the application.

Figure 27: Home Screen of Raven

5.1.5 View Report

Figure 28 depicts the view report module of our application. User will be able to view report against a specific id and against all the id's that are present in our database. In report, appearance time against each id will be displayed.

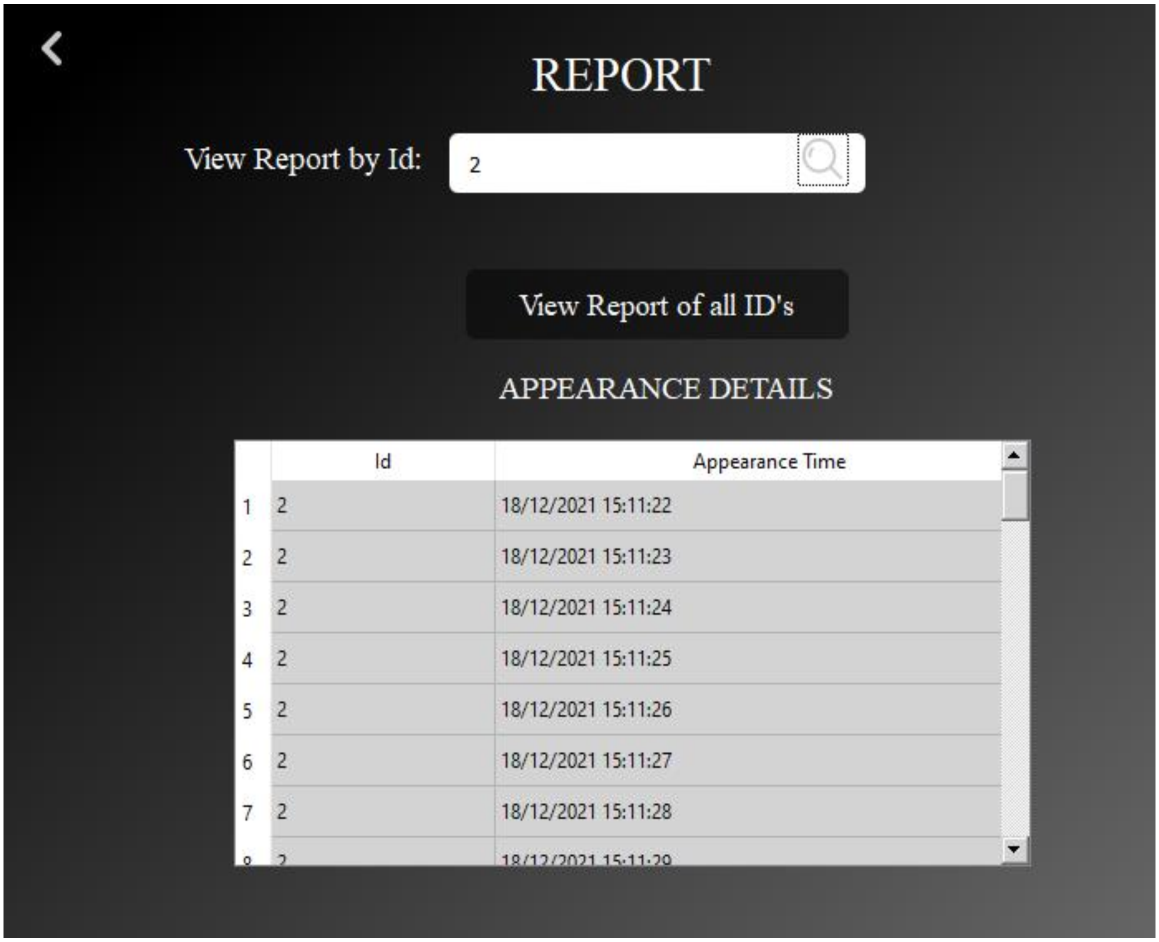

Figure 28: View Report Module

Chapter 6

Project Code

6 Project Code

Our project's code is written in Python (version 3.7.6) which can be divided into the following sections:

- YOLO v3 Model
- OSNet Model ('osnet_ain_x1_0')
- Retrieving saved data from database.

The code is built with the following libraries:

- PyQt5 version 5.15.4
- PyTorch version 1.6
- OpenCV version 4.5.3.56
- Tensorflow
- NumPy version 1.14.5
- SQLite version 3.12.2

6.1 YOLO v3

YOLO v3 is used for detection and tracking purposes. At first, person appearing in the frame will be detected and a bounding box will be generated against that person and an id will be assigned to that bounding box. After detection, it will be tracked afterwards.

```
def main(yolo):
  testvideo,_ = QFileDialog.getOpenFileName(self, 'Select Video', QD
ir.currentPath(), 'Videos (*.mp4 *.avi)')
  videolist = [testvideo]
  max_cosine_distance = 0.2
  nn_budget = None
  nms_max_overlap = 0.4

  # deep_sort
  model_filename = 'model_data/models/mars-small128.pb'
  encoder = gdet.create_box_encoder(model_filename,batch_size=1) # u
se to get feature

  metric = nn_matching.NearestNeighborDistanceMetric("cosine", max_c
osine_distance, nn_budget)
  tracker = Tracker(metric, max_age=100)

  is_vis = True
  out_dir = 'videos/output/'
```

```
  print('The output folder is',out_dir)
  if not os.path.exists(out_dir):
    os.mkdir(out_dir)
    all_frames = []
    for video in videolist:
      print("This is video path",video)
      loadvideo = LoadVideo(video)
      video_capture, frame_rate, w, h = loadvideo.get_VideoLabels()
      while True:
        ret, frame = video_capture.read()
        if ret != True:
          video_capture.release()
          break
          all_frames.append(frame)

      frame_nums = len(all_frames)
      tracking_path = out_dir+'tracking'+'.avi'
      combined_path = out_dir+'allVideos'+'.avi'
      if is_vis:
        fourcc = cv2.VideoWriter_fourcc(*'MJPG')
        out = cv2.VideoWriter(tracking_path, fourcc, frame_rate, (w,
 h))
        out2 = cv2.VideoWriter(combined_path, fourcc, frame_rate, (w
, h))
        #Combine all videos
        for frame in all_frames:
          out2.write(frame)
        out2.release()

      #Initialize tracking file
      filename = out_dir+'/tracking.txt'
      open(filename, 'w')

      #Initializing reid file
      reid_filename = out_dir+'/reid.txt'
      open(reid_filename, 'w')

      # fps = 0.0
      frame_cnt = 0
      t1 = time.time()
      track_cnt = dict()
      images_by_id = dict()
      ids_per_frame = []
      for frame in all_frames:
        image = Image.fromarray(frame[...,::-1]) #bgr to rgb
```

```
        boxs = yolo.detect_image(image) # n * [topleft_x, topleft_y,
 w, h]
        features = encoder(frame,boxs) # n * 128
        detections = [Detection(bbox, 1.0, feature) for bbox, featur
e in zip(boxs, features)] # length = n
        text_scale, text_thickness, line_thickness = get_FrameLabels
(frame)

        # Run non-maxima suppression.
        boxes = np.array([d.tlwh for d in detections])
        scores = np.array([d.confidence for d in detections])
        indices = preprocessing.delete_overlap_box(boxes, nms_max_ov
erlap, scores) #preprocessing.non_max_suppression(boxes, nms_max_ove
rlap, scores)
        detections = [detections[i] for i in indices] # length = len
(indices)

        # Call the tracker
        tracker.predict()
        tracker.update(detections)
        tmp_ids = []
        for track in tracker.tracks:
          if not track.is_confirmed() or track.time_since_update > 1
:
            continue

          bbox = track.to_tlbr()
          area = (int(bbox[2]) - int(bbox[0])) * (int(bbox[3]) - int
(bbox[1]))
          if bbox[0] >= 0 and bbox[1] >= 0 and bbox[3] < h and bbox[
2] < w:
            tmp_ids.append(track.track_id)
            if track.track_id not in track_cnt:
              track_cnt[track.track_id] = [[frame_cnt, int(bbox[0]),
 int(bbox[1]), int(bbox[2]), int(bbox[3]), area]]
              images_by_id[track.track_id] = [frame[int(bbox[1]):int
(bbox[3]), int(bbox[0]):int(bbox[2])]]
            else:
              track_cnt[track.track_id].append([frame_cnt, int(bbox[
0]), int(bbox[1]), int(bbox[2]), int(bbox[3]), area])
              images_by_id[track.track_id].append(frame[int(bbox[1])
:int(bbox[3]), int(bbox[0]):int(bbox[2])])

          cv2_addBox(track.track_id,frame,int(bbox[0]),int(bbox[1]),
int(bbox[2]),int(bbox[3]),line_thickness,text_thickness,text_scale)
```

```
            curr_time = datetime.now()
            dt = curr_time.strftime("%d/%m/%Y %H:%M:%S")
            print("date and time =", dt)
            write_results(filename,'mot',frame_cnt+1,str(track.track_i
d), dt, int(bbox[0]),int(bbox[1]),int(bbox[2]),int(bbox[3]),w,h)
        ids_per_frame.append(set(tmp_ids))
```

6.2 OSNet Model

OsNet model will be responsible for performing re-identification after detection and tracking. The model will extract features at first and will then compute distance. The distance metric that we have used is cosine. That computed distance will then be compared with a threshold value for re-identification.

```
def __init__(self):
  self.model = torchreid.models.build_model(
      #name='resnet50',
      #name='osnet_x0_25',
      name='osnet_ain_x1_0',
      num_classes=1,#human
      loss='softmax',
      pretrained=True,
      use_gpu = False
      )
  torchreid.utils.load_pretrained_weights(self.model, 'model_data/mo
dels/osnet_ain_x1_0_msmt17_256x128_amsgrad_ep50_lr0.0015_coslr_b64_f
b10_softmax_labsmth_flip_jitter.pth')
  self.optimizer = torchreid.optim.build_optimizer(
      self.model,
      optim='adam',
      lr=0.0003
    )
  self.scheduler = torchreid.optim.build_lr_scheduler(
      self.optimizer,
      lr_scheduler='single_step',
      stepsize=20
      )
  _, self.transform_te = build_transforms(
      height=256, width=128,
      random_erase=False,
      color_jitter=False,
      color_aug=False
```

```
    )
  #self.dist_metric = 'euclidean'
  self.dist_metric = 'cosine'
  self.model.eval()

  def _extract_features(self, input):
    self.model.eval()
    return self.model(input)

  def _features(self, imgs):
    f = []
    for img in imgs:
      img = Image.fromarray(img.astype('uint8')).convert('RGB')
      img = self.transform_te(img)
      img = torch.unsqueeze(img, 0)
      features = self._extract_features(img)
      features = features.data.cpu() #tensor shape=1x2048
      f.append(features)
    f = torch.cat(f, 0)
    return f

  def compute_distance(self, qf, gf):
    distmat = metrics.compute_distance_matrix(qf, gf, self.dist_metr
ic)
    # print(distmat.shape)
    return distmat.numpy()
```

6.3 Retrieving saved data from database

Following functions will help the user in retrieving data from the database.

6.3.1 Search Video by Id

The function search_video_byid() will run the video of the person (against ID) in the application after retrieving video's saved path from the database.

```
def search_video_byid(self):
        input_id = self.SearchEdit.text()
        print(input_id)
        self.imgLabel.setText("")
        conn = sqlite3.connect("Raven_DB.db")
        cur = conn.cursor()
        print("Successfully Connected to SQLite")
```

```
        outp_dir = '/videos/output/tracklets/{s}.avi'.format(s=str(i
nput_id))
        print(outp_dir)
        #print(idx)
        query = "SELECT IdPath from Tracklets where Id={i}".format(i
=input_id)
        print('Query: ', query)
        cur.execute(query)
        row = cur.fetchone()
        print(row)
        if row is None:
            self.search_error.setText("No person with that ID exists
 in our system.")
            self.imgLabel.setText("hh")
        else:
            self.search_error.setText("")
            l = list(row)
            final_path = "".join(l)
            print('Final result', final_path)
            init_path = 'C:/Users/DELL/Desktop/Multi-Camera-Person-
Tracking-and-Re-Identification'
            complete_path = init_path + final_path
            print(complete_path)

            cap = cv2.VideoCapture(complete_path)
            if (cap.isOpened()== False):
                print("Error opening video stream or file")
            while(cap.isOpened()):
                ret, frame = cap.read()
                if ret == True:
                    self.displayImage(frame,1)
                if cv2.waitKey(60) == 27:
                        break
            capture.release()
            cv2.destroyAllWindows()
            cv2.destroyAllWindows()


            conn.commit()
            cur.close()
            conn.close()
```

6.3.2 View Report by Id

The function View_Report_byid() will display person's appearance time that is saved against its ID in the database. It will retrieve its data and display it in the table in the view report module.

```
def View_Report_byid(self):
        inp_id = self.SearchEdit2.text()
        print(inp_id)
        conn = sqlite3.connect("Raven_DB.db")
        cur = conn.cursor()
        print("Successfully Connected to SQLite")
        query = "SELECT DISTINCT Id,Time FROM Report where Id={i}".f
ormat(i=inp_id)
        query2 = "SELECT Id FROM Report where Id={i}".format(i=inp_i
d)
        cur.execute(query2)
        row1 = cur.fetchone()
        if row1 is None:
            self.error_2.setText("No person with that ID exists in o
ur system.")
        else:
            self.tableWidget.setRowCount(50)
            tableRow = 0
            for row in cur.execute(query):
            #if row is None:
             #   self.error_2.setText("No person with that ID exists
 in our system.")
                self.error_2.setText("")
                print(row)
                self.tableWidget.setItem(tableRow, 0, QtWidgets.QTab
leWidgetItem(str(row[0])))
                self.tableWidget.setItem(tableRow, 1, QtWidgets.QTab
leWidgetItem(row[1]))
            #self.tableWidget.setItem(tableRow, 2, QtWidgets.QTableW
idgetItem(row[2]))
                tableRow=tableRow+1
        #cur.execute(query)
        #row_check = cur.fetchone()
        conn.commit()
        cur.close()
        conn.close()
```

6.3.3 View Full Report

The function View_Full_Report() will display every person's appearance time that is saved against their ID in the database. It will retrieve their data and display it in the table in the view report module against the button view report of all id's.

```
def View_Full_Report(self):
        conn = sqlite3.connect("Raven_DB.db")
        cur = conn.cursor()
        query = "SELECT DISTINCT Id,Time FROM Report where Id in (SE
LECT DISTINCT Id from Report)"

        self.tableWidget.setRowCount(50)
        tableRow = 0

        for row in cur.execute(query):
            print(row)
            self.tableWidget.setItem(tableRow, 0, QtWidgets.QTableWi
dgetItem(str(row[0])))
            self.tableWidget.setItem(tableRow, 1, QtWidgets.QTableWi
dgetItem(row[1]))
            tableRow=tableRow+1
```

Chapter 7

Conclusion

7 Conclusion

7.1 Problems faced and lessons learned

7.1.1 Google Colab Limitations:

One of the problems we faced was limitations of RAM provided by Google Colab. Google Colab provide us 12GB RAM while using free resources. In order to use 25GB RAM we must move on Colab pro which isn’t available for free. Colab has a window of 90minutes timeout of inactivity and 12 hrs maximum timeout which is short for training purposes because training process itself is very time taking.

7.1.2 Use of OSNet model:

The model we choose for our project is OSNet [9] model. No sufficient work has been done on this model so far due to which we faced a lot of trouble while searching for solutions to the problems related to model as no application work has been done on this model. So, using OSNet model without any background work in our project was an uphill task.

7.1.3 Efficiency Issue:

In the start, the efficiency of OSNet model was not up to the mark. Then we have to fine tune the parameters which help us in improving the efficiency of model. We change the image size, batch size and train our model on those parameters so that if query data tweaks, our model would be able to show good efficiency.

7.1.4 Size of Dataset:

Size of dataset was one of the main problems we faced during the model testing phase. The size of MSMT [10] dataset was very large as it had more than 80,000 images. When we tried to access those images, RAM size exceeds its limit, and the testing gets halted. One of the solutions that we got was to divide the dataset into two halves and then train the dataset on both files and then compute the result collectively.

7.2 Project summary

We know that security is the necessity of every person and for that reason surveillance cameras are installed at almost every public place. These surveillance cameras help the security and law enforcement agencies in reducing crime rate as these cameras are integrated with intelligent systems which help them in monitoring purposes. The basic purpose of installing these cameras is to keep the track of human activity, the cameras are integrated with such intelligent systems that they themselves monitor the situation and alert the user if some unwanted activity takes place. A lot of things get in control when one knows that he is being monitored, he has that fear of being caught due to which he restrains himself of committing some false act. In an underdeveloped country like Pakistan, there is a high need to have automated surveillance systems which would help control crime and would help the security agencies in solving cases.

The system that we have proposed is a sort of surveillance system in which we will track an individual using multiple cameras where the identity of the person will be retrieved from camera by matching that person's data with the data present in our database. Once our system will detect the person, it would be possible for us to track the moving person from one point to another and its track would be saved in our system. This saving of record of that person would help the security agencies for tracking the suspect of a crime scene if they are looking for a person who has appeared in the area covered by our system, so our system would help them in reaching that person by providing them with his history details.

7.3 Future work

Some ideas are given below for the possible expansion of our project in the future.

1. Currently, this system would work on desktop as we are intending to make a desktop application only. In future, for more feasibility of users the project could be extended to a mobile base application.
2. Another possible extension would be to provide cloud-based services.

Chapter 8

References

8 References

[1] D. Focken and R. Stiefelhagen, “Vision based 3-D People Tracking in Smart Room,” 2002. [Online]. Available: https://www.ri.cmu.edu/pub_files/pub3/focken_dirk_2002_1/focken_dirk_2002_1.pdf. [Accessed 11 May 2021].

[2] U. Gawande, K. Hajari and Y. Golhara, “Pedestrian Detection and Tracking in Video Surveillance System,” in *Recent Trends in Computational Intelligence*, 2020.

[3] I. Brilakis, M.-W. Park and G. Jog, “Automated vision tracking of project related entities,” *Advanced Engineering Informatics,* p. 10, 2011.

[4] F. Herzog, X. Ji, T. Teepe, S. Hurnman, J. Gilg and G. Rigoll, “LIGHTWEIGHT MULTI-BRANCH NETWORK FOR PERSON RE-IDENTIFICATION,” Technical University of Munich, Munich.

[5] G. Wang, J.-H. Lai, P. Huang and X. Xie , “Spatial-Temporal Person Re-identification,” Guangdong Key Laboratory of Information Security Technology, China.

[6] R. Quispe and H. Pedrini, “Top-DB-Net: Top DropBlock for Activation,” Institue of Computing, University of Campinas, Campinas.

[7] A. Kumar, K. A. Mangipudi and P. Vihang, “Person Re-Identification Using Pose-Normalized And Occlusion Independent Training,” Texas A&M University.

[8] P. Chikersal, “Person Re-identification using Appearance: Final Report,” Nanyang Technological University, Singapore.

[9] K. Zhou, “torchreid,” 2019. [Online]. Available: https://kaiyangzhou.github.io/deep-person-reid/pkg/models.html#reid-specific-models. [Accessed 25 February 2021].

[10] K. Zhou, “torchreid,” 2019. [Online]. Available: https://kaiyangzhou.github.io/deep-person-reid/datasets.html#msmt17-msmt17. [Accessed 14 May 2021].

Turnitin Originality Report

CS33_RAVEN.docx by Anonymous

From ABC (Assignments and papers)

- Processed on 31-Dec-2021 5:17 AM PST
- ID: 1736647175
- Word Count: 7755

Similarity Index
9%
Similarity by Source

Internet Sources:
4%
Publications:
6%
Student Papers:
3%

sources:

1 1% match (publications)
Rodolfo Quispe, Helio Pedrini. "Top-DB-Net: Top DropBlock for Activation Enhancement in Person Re-Identification", 2020 25th International Conference on Pattern Recognition (ICPR), 2021

2 1% match (publications)
Fabian Herzog, Xunbo Ji, Torben Teepe, Stefan Hormann, Johannes Gilg, Gerhard Rigoll. "Lightweight Multi-Branch Network For Person Re-Identification", 2021 IEEE International Conference on Image Processing (ICIP), 2021

3 1% match (Internet from 09-Oct-2021)
https://paperswithcode.com/paper/spatial-temporal-person-re-identification

4 1% match (student papers from 12-Jul-2021)
Submitted to Innopolis University on 2021-07-12

5 < 1% match (Internet from 19-Jul-2020)
https://medium.com/datadriveninvestor/multiple-object-tracking-using-person-re-identification-f9b7360cda1a