Software Requirement Specification Document For Edge-Fog Intelligent Surveillance System for smart cities

Nour Ahmed, Nour ElHoda Hisham, Mariam Hesham, Samiha Hesham, Sandra Fares
Supervised by: Dr. Islam Tharwat, Eng. Lobna Mostafa

June 4, 2021

Table 1: Document version history

Version	Date	Reason for Change
1.0	12-Dec-2020	SRS First version's specifications are defined.
1.1	20-Dec-2020	Added Non-Functional Requirements Identified remaining functional requirements. Identified hardware constraints Added end user Interface
1.3	28-Dec-2020	Non-Functional Requirements are updated. Added class diagram and data base schema Added use case for actors in the system and operational scenarios
2.0	1-Jun-2021	Diagrams are updated Project plan is updated

GitHub: https://github.com/SandraFW/intelligent-Surveillance-for-smart-cities

Abstract

Surveillance systems are of vital importance for the development of smart cities as mentioned by K. Zhang et al in [1]. These systems can be considered vision organs of such cities. It is expected that a huge amount of data (Big Data) will be generated in smart cities. Therefore, to ensure the safety of its citizens, it is important to provide an efficient and real-time analysis of these data to get real-time responses, when catastrophic events occur. Accordingly, transmitting this massive data to the cloud, to be processed, is relatively slow as proved by Mluleki Sinqadu et al in [2]. Therefore, the purpose of this project is to implement an edge computing- based surveillance system to offer real-time data processing. When surveillance videos capture an incident, the data get transferred to the edge for processing. Moreover, a rapid response is then provided to properly handle the occasion. Furthermore, despite tackling scalability obstacles, the system should handle privacy-sensitive data to overcome the privacy challenges in smart cities discussed by Lei Cui et el. in [3].

1 Introduction

1.1 Purpose of this document

The purpose of this document is to present a detailed description of how to use an edge-fog model to handle criminal activities (fighting, kidnapping, robbery) on the streets. It will figure both functional and non-functional requirements of the system in addition to the interfaces, GUI, and constraints that will cover the system. Moreover, this document will define both stakeholders and developers of the system view.

1.2 Scope of this document

The scope of this document is to show the basic outlines of our system requirements [4] to understand our system for any further updates later on and the issues we may face. Also, it states the main objective of the system with a well-studied timeline. Furthermore, the objectives that we will reach in our system. Requirements outlined in this document are subject to be changed.

1.3 System Overview

The system aims to prevent any criminal acts (fighting, kidnapping, robbery) in smart cities by fast prediction and detection of those incidents, using a framework composed of three computing layers as shown in Figure 1, as some authors proposed in their papers [5], [6], [7] and [8]. First, the cameras located at the potential crime scenes act as the edge nodes; Light-weighted algorithms get deployed on them for any possible failure in the device, crime detection, and face comparison algorithm to recognize criminals. Once a crime gets detected, the video streams drive to the second layer, a two-tier fog nodes layer. In the first tier, captured video streams get further processed by applying a video summarizing algorithm, and a face blurring algorithm which ensures safety and privacy measures for the citizens by blurring their exposed identities. In the second tier, a deep learning algorithm gets applied to data to confirm the occurrence of the crime and specify its type. Moreover, deblurring and face detection gets implemented to detect criminals' faces. Then data

get transmitted to the third layer, the cloud layer, where it triggers an alarm to notify police units of the incident. Moreover, data backup and system updates are done in this layer.

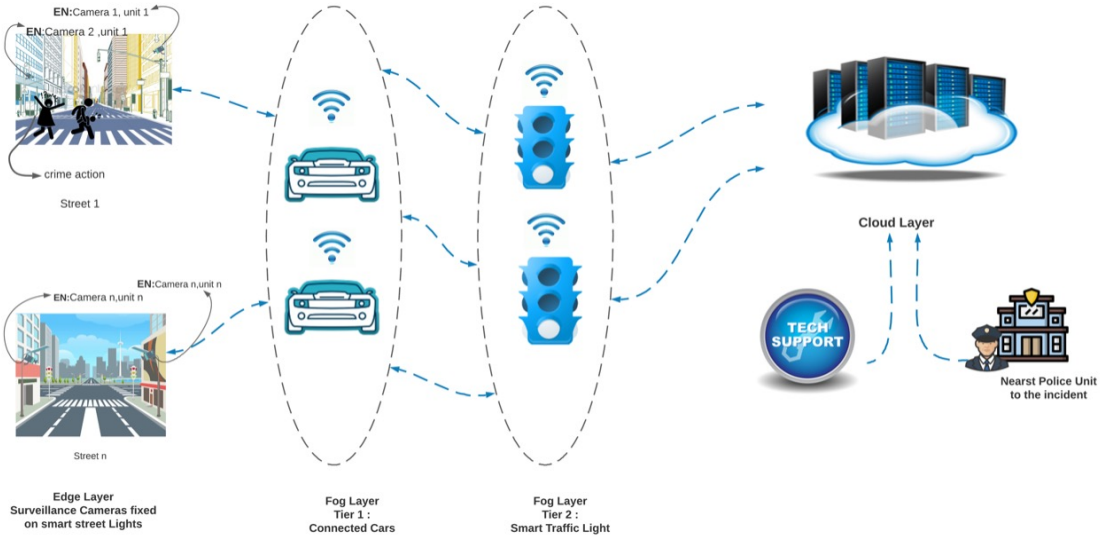


Figure 1: Illustration of high-level view of the system.

1.4 System Scope

- Fog/Edge computing-based surveillance system to monitor the whole city 24/7.
- Efficiently detect any criminal activity (fighting, kidnapping, robbery) and response immediately.
- Privacy assurance by blurring people's faces to secure their exposed identity in the video.
- Scalable system to handle massive data generated from surveillance cameras and sensors.
- Applying a prediction function to predict criminal activities.
- Dispatch data and location to the nearest police station.
- Camera failure detection (black screen, green screen, unfocused capturing) in order to successfully capture data.
- Apply face recognition on criminals when an incident occurs.

1.5 Business Context

The intelligent surveillance system monitors smart city streets to assists the police, or the competent authority units, to get notified of any incident that occurs to take quick action against it using the three computing layers edge, fog, and cloud. Our proposed system aims to provide a high level of safety to the citizens in smart cities. Therefore, the system captures the pedestrians to collect data throughout the day. The collected data are processed to detect any failure in the surveillance cameras and if an incident occurred. If a crime got detected, the system identifies the criminal and forwards the collected information to the police unit to take the proper action and helps them track the criminal.

2 Similar Systems

2.1 Academic

- In the hopes of decreasing and preventing crimes in a smart home environment (SHE), Tanin Sultana and Khan A. Wahid [8] proposed a framework called IoT guard. The authors designed an edge-fog architecture to assist in detecting crimes, predicting them, and sending alerts to police units to take immediate actions and prevent any catastrophic event from occurring. As shown in Figure 2, Initially, the edge nodes, or the cameras, detect if an event took place. Whenever an action is detected, the edge nodes will forward the surveillance data to the fog nodes to get processed. The fog nodes will then apply object detection algorithms, predict possible events, send alerts to the police unit, and finally send data to the cloud server for system updates or any further analysis. Results show the system's performance is better than transitional IoT surveillance systems. It is efficient and scalable. Other systems that target the safety of the citizens include [9], which ensures the public's safety using an application for Smart Transportation Safety (STS), and [10] which detects any violent act that occurs on social media platforms.

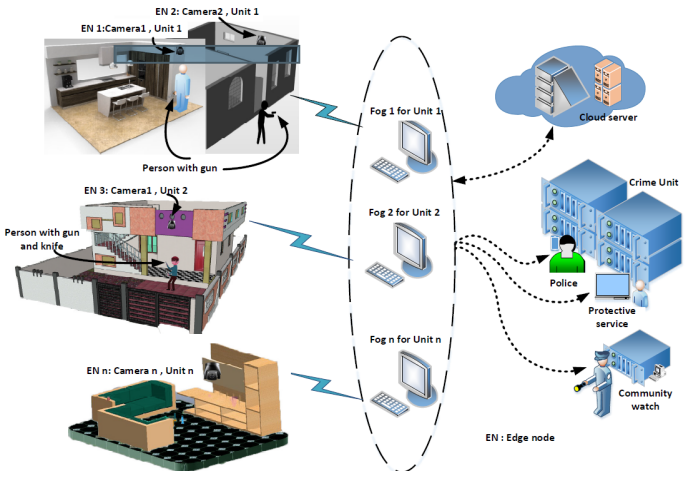


Figure 2: IoT-guard security management system

- Hui Sun et al. [11] presented an edge computing-enabled video usefulness model (VU) in order to handle failures that might occur in surveillance videos, using failure detection approaches, as it usually takes a great amount of time for maintenance teams to determine what causes the failure. Moreover, the system provides bandwidth improvement, as useless video surveillance data might consume bandwidth that is needed by useful and informative videos. Experimental results show that the system detects failures efficiently. Various systems' target, like this one, is to make use of useful information. In [12] a multi-view video summarization is presented to extract convenient data. Moreover, a survey paper [13] is published to discuss this system's objectives, advantages, and drawbacks.

- Due to the large number of data collected by surveillance cameras, many challenges appeared in analyzing, storing, and retrieving data. Santamaria et al. [14] introduced the basics needed to build a framework for people's abnormal behaviors, after investigating the current techniques and technologies. The authors proposed a distributed architecture as shown in Figure 3. Distributed architectures assist in reducing the computation of the end devices, which help in improving the performance and avoiding waste of bandwidth and memory. In addition, they mentioned the responsibility of each layer in the system, the possible computer vision techniques that can be used with their pros and cons as Neural networks, and the KNN model for object detection and tracking. Results show that an alarm is triggered when detecting an anomaly in order to track it. Other systems that aimed to improve the performance include [15]

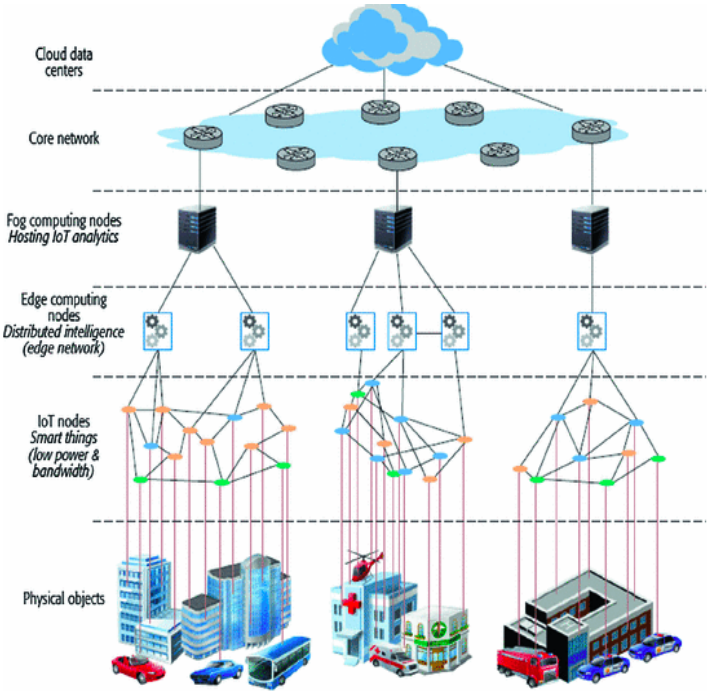


Figure 3: the distributed ecosystem in cloud domain

- Jianyu Wang et al. [16] designed fog computing architecture for urban surveillance systems to provide low latency and decrease the workload of cloud centers. The authors implemented Network Functions Virtualization and Software-Defined Networking on the cloud computing platform OpenStack, resulting in a real-time video processing system. Experimental results are promising. They show that the system can be developed on a larger scale and that fog computing can be potentially used for smart urban surveillance systems. Likewise, Yutong Liu et al. [17] also proposed a system that improves the latency, as well as the accuracy.

- In the hopes of exploiting fog computing and tackling the cost challenges faced when building surveillance systems, Mansoor Nasir et al [18] presented a cost-effective fog computing based surveillance system. The system does not only reduce the energy consumed and the bandwidth, but also reduces the cost. The authors proposed a video summarization technique to reach their goal. By using Raspberry Pi devices as the system's fog nodes, and by distributing data onto these nodes to provide a summary, the system is scalable and delivers satisfactory outcomes. Other systems that provide cost-effective surveillance solutions using fog computing include [19] and [20]

2.2 Business Applications

- For helping smart cities to achieve a safe and secure environment, Logipix [21] Technical Development Ltd is a privately held organization founded in 1996 in Hungary that provides intelligent video monitoring solutions for wide public areas in the cities. They highlighted their solutions as a scalable city-wide system that is designed and developed to achieve a video surveillance system and traffic violation management tasks at the same time. The main goal of Logipix solutions is to provide a real-time system that is able to surveil the whole city 24/7 efficiently, the system also can respond immediately and take proper action in case of detecting any abnormal event. Moreover, the system captures all the events with high resolution even at miles away, face recognition is still possible. For helping the competent authorities, these recordings can be used as unappealable evidence on demand.

- BT Business [22] is a retail division of the United Kingdom telecommunications company that provides many solutions that build smart cities. For securing smart cities and in order to help the administrators make swift decisions and help them to fast respond to what's going on and to prevent suspicious events before they happen, BT provides surveillance solutions for local councils, streets, public transport , police stations and more. Moreover, they helped many United Kingdom cities to turn to smart cities by the installation of public WiFi and cameras to help monitor traffics, give real-time alerts for trams, buses and more. For cost-effective and best-of-breed surveillance solutions, BT company partners with industry-leading suppliers as CISCO and BOSCH and many other huge suppliers.

3 System Description

3.1 User Problem Statement

One of the challenges that face surveillance systems in smart cities is Big data. Each camera records a massive amount of video and audio data to be processed. To handle such massive data, it is essential to figure out a way to provide real-time data processing analysis efficiently.

3.2 User Objectives

Our system is crucial, as its fundamental goal is to ensure the citizens' safety and privacy. For our users, the system should provide a rapid response to take appropriate action in time. When an anomaly gets captured on a surveillance camera, the system should detect it and determine the type of crime. Whether it is kidnapping, fighting, or robbery. Moreover, the system should track the criminal, identify his face, and trigger an alarm with the current location of the incident for the users, in our case, to the nearest police or competent authorities unit.

3.3 User Characteristics

- Police: Must have the basic knowledge to deal with the system in following up on any notification that will be sent to catch the criminal.
- Admin: Must have the basic knowledge to deal with the whole system in adding a new account for the competent authorities, updating and deleting their accounts. Also, for receiving the camera failure alerts.

3.4 System Context

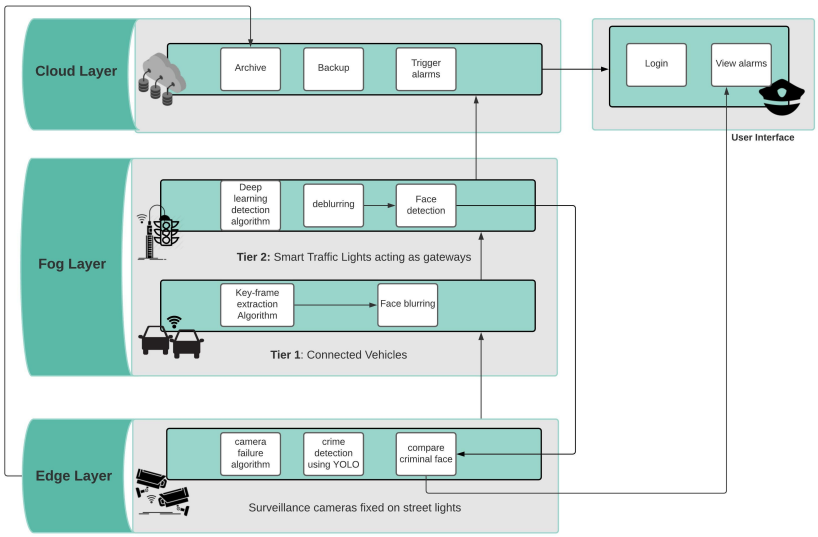


Figure 4: Architecture of the system.

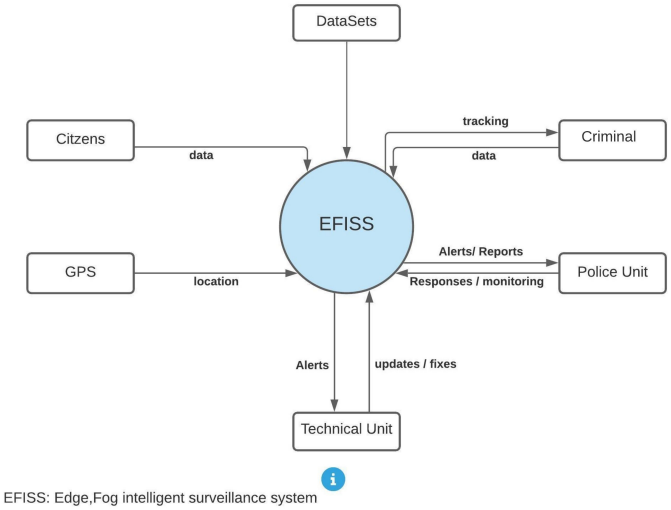


Figure 5: Context Diagram of the system.

4 Functional Requirements

4.1 System Functions

1. The system must be able to detect the criminal action accurately.
2. The system should be able to track the criminal from the location of the incident.
3. The system must apply face blurring for citizens in order to enhance privacy in surveillance system.
4. The system should send alert if any failure is detected in the cameras occurred in order to help the maintenance team to solve the problem.
5. The system should send an alert message when criminal behaviour is detected.
6. The system should provide a face recognition function to recognize criminals.
7. The system must filter the incident using video splitting technique in order to be saved and sent to the cloud to save storage.
8. The system should be able to predict the event before occurring in order to prevent it.

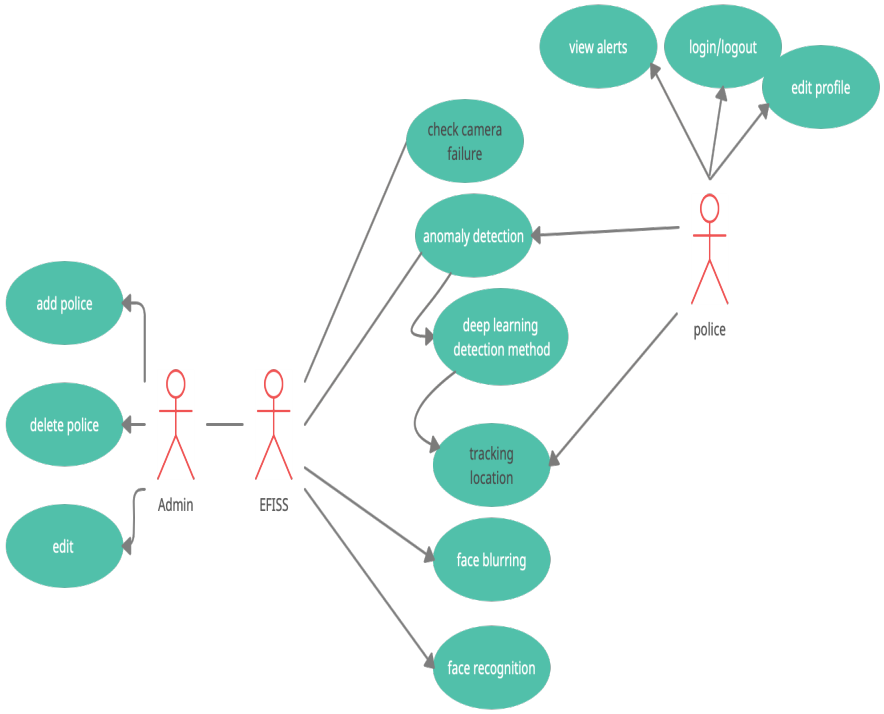


Figure 6: Use Case Diagram

4.2 Detailed Functional Specification

Table 2: Check failure of the camera

Name	Check failure of the camera
Code	FR01
Priority	High
Critical	It's essential to detect if any failure happens to the video streams and enclose it to efficiently detect an event, avoid network overload, and improve the cloud storage usage
Description	The system detects if any failure that impacts the usefulness of the video streams (i.e., natural occlusion, disruption) happens. Then, sends an alert of the failure to the maintenance team to recover it
Input	Video stream
Output	Boolean if it detects failure or not
Pre-condition	A failure happens in the video streams.
Post-condition	A failure is detected and enclosed with an alert to the maintenance team to recover it
Dependency	-
Risk	-

Table 3: Face blurring

Name	Face blurring
Code	FR02
Priority	Extreme
Critical	It's extremely essential to ensure the privacy and identity of the citizens that exposed to privacy violation in the video surveillance
Description	In the real-time video surveillance after detecting the people's faces, the system uses a blur algorithm to blur the face to be indiscernible.
Input	Video stream
Output	Video stream
Pre-condition	Detecting people's faces in real-time video surveillance to blur it
Post-condition	The detected faces are been blurred
Dependency	FR01 – FR03
Risk	If there is any failure that impacts the usefulness of the video stream

Table 4: Crime detection

Name	Crime detection function
Code	FR03
Priority	Extreme
Critical	It is really essential; An action needs to be detected in order to take the appropriate action to predict or prevent it.
Description	when any action is captured from a surveillance camera, crime detection algorithms, as well as image processing techniques, are applied to detect it.
Input	video stream
Output	video stream
Pre-condition	surveillance camera captures an event.
Post-condition	the event gets detected.
Dependency	FR01. This function depends on FR01, because if any failure took place on surveillance cameras, the event will not be detected.
Risk	If any failure happens when capturing video streams.

Table 5: Crime Dispatch Function

Name	crime detail dispatch
Code	FR04
Priority	Extreme
Critical	It is very essential function for the system
Description	On criminal action detection , it gathers location information and dispatches crime image and sends it for the further processing.
Input	video stream
Output	video stream
Pre-condition	crime is detected in function FR03
Post-condition	Dispatch the stream to other processing
Dependency	FR01-FR03
Risk	Camera has any failure in capturing the video streams and can be fixed with FR01

Table 6: action prediction

Name	action prediction function
Code	FR05
Priority	High
Critical	It is important to predict an event before occurring in order to prevent it and avoid tragic situations.
Description	after dispatching the needed video stream, an event prediction function can be applied to determine the possible events that could happen.
Input	video stream
Output	Boolean if action is criminal.
Pre-condition	Dispatched video stream is processed.
Post-condition	action is predicted.
Dependency	FR01, FR03, FR04. FR01: if any failure took place on surveillance cameras, motion will not be detected in order to predict it. FR03: if crime is not detected, surveillance data cannot be processed further more. So, prediction will not occur. FR04: video stream needs to be dispatched before forwarding it for further processing.
Risk	If any failure happens when capturing video streams, if crime was not detected, or if the needed video stream was not dispatched.

Table 7: Alert Function Description

Name	Alert
Code	FR06
Priority	Extreme
Critical	It is important to alert police by criminal incident.
Description	To maintain the situation, there must an alert send to the police to come urgently and catch the criminal..
Input	Video stream , face recognition ,location
Output	Alert
Pre-condition	After detection of criminal incident in FR05 and face recognition in FR08, an alert must be sent to police by Fog computing.
Post-condition	Alert to police.
Dependency	FR05: By detection of the criminal incident, FR08: by Face recognition FR07: By detecting the location of the criminal, Fog will send an alert to the police to rescue the victim.
Risk	Failure to send an alert if there is data is missing or there was a difficulty to send an alert.

Table 8: criminal Tracking Function

Name	criminal Tracking Function
Code	FR07
Priority	low
Critical	it is a supplementary function in the system
Description	After detection of a criminal incident their should be a tracking for the criminal from the location of the incident to help the authorities catch him
Input	location
Output	location
Pre-condition	crime detected FR03, location detetcted and video dispatched in FR04 and predection FR05
Post-condition	tracing criminal location and report the authorities
Dependency	FR01-FR03-FR04-FR05
Risk	Camera has any failure in capturing the video streams and can be fixed with FR01. Or criminal runs to a place can't be reached by the system

Table 9: Face Recognition Function Description

Name	Face Recognition
Code	FR08
Priority	Extreme
Critical	It is important to recognize the criminal's face to make it easy for the police to catch him.
Description	During the crime scene, there should face recognition from a camera sensor to detect the features of the criminal.
Input	Video stream
Output	Criminal's face
Pre-condition	After detection of the criminal incident in FR05, the camera must capture a criminal's face.
Post-condition	Face recognition
Dependency	FR05: By detection of the criminal incident, the camera will capture or recognize the features of the criminal. Failure in camera when capturing the criminal.
Risk	Failure in camera when capturing criminal.

5 Interface Requirements

5.1 User Interfaces

The system is designed to be more simple and facilitate to use and to seem more clear to the user that he must understand the steps that he must do. Moreover, system provides user by implementing all necessary interactions to make it easy to use.

5.1.1 GUI

The proposed system's application focuses on two main ends, the user and the admin end. The application allows the admin to log in and create accounts for the police officer or security guards. Also, the admin can update and delete the user's account. The user will log in to the application with an email and password set by the admin. Also, the user can update his information and change his password. Moreover, the user will view the alerts sent by the system with images, location, and time.

Figure 7: Home Page

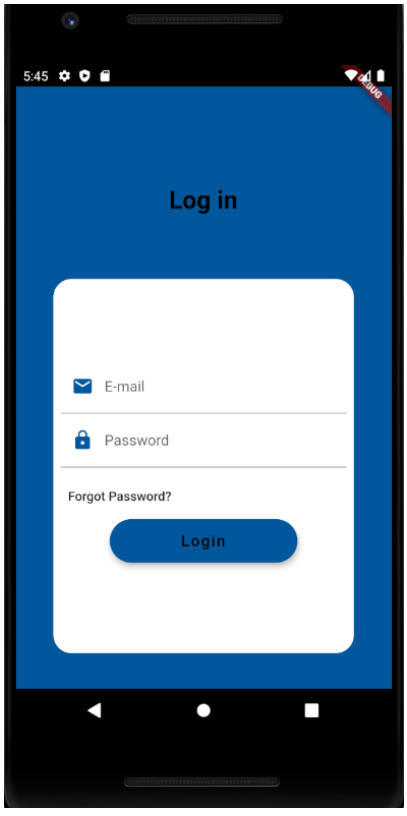


Figure 8: Login Page

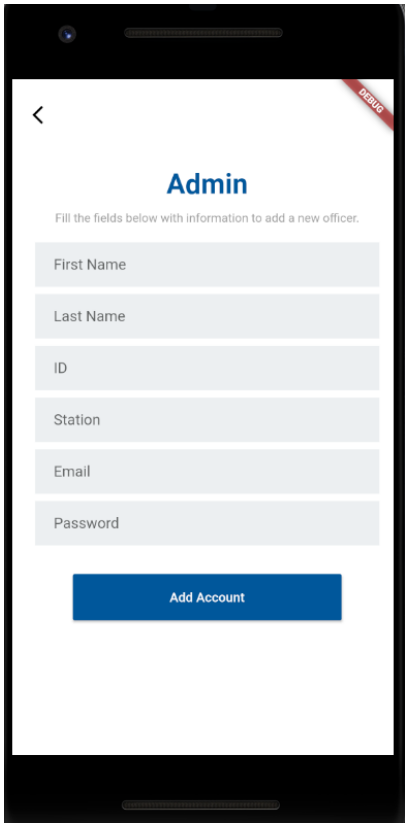


Figure 9: Add new account

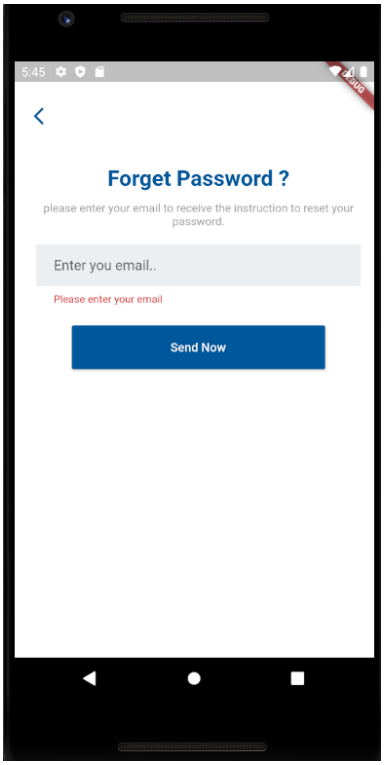


Figure 10: Forget Password

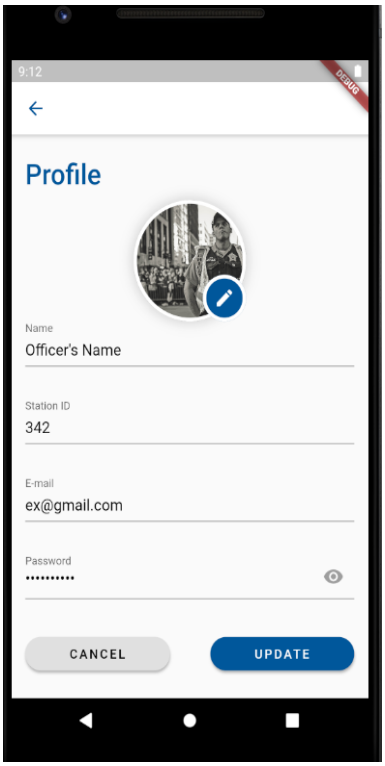


Figure 11: Profile

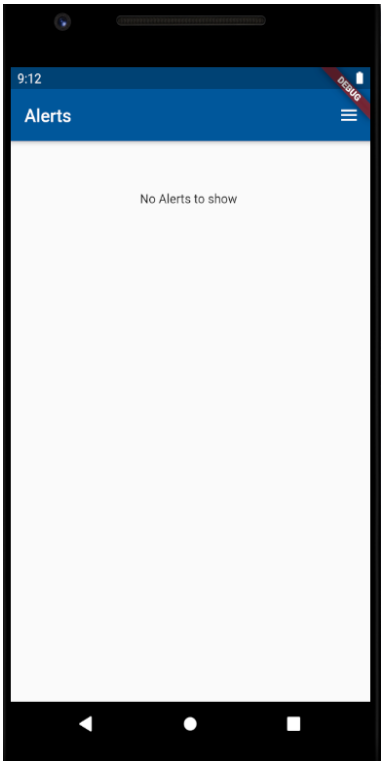


Figure 12: Alerts Page

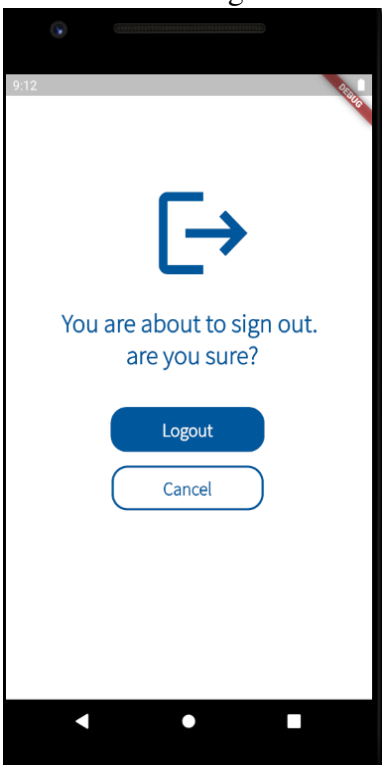


Figure 13: Logout Page

5.1.2 CLI

N/A

5.2 Hardware Interfaces

At this stage, there is no need for hardware as we can use simulator as ifogsim toolkit developed by Harshit [23] and mentioned by Redowan Mahmud in [24] . But on deploying this project in reality, we need IP surveillance cameras to capture the incidents and personal computers at the end-users sites to help them monitor the system or receive alerts and make records for the crimes.

5.3 Communications Interfaces

As our project highly depends on communication among entities, architectures such as fog networking, edge networking, and cloud networking are dominant. To reduce the overload on the network, the processing happens on both the edge and fog nodes to provide real-time response. Moreover, the data then gets transferred to the cloud for further processing.

5.4 API

- Google Maps API, to get the actual location of the event and dispatch it. Also, in tracking the criminal.
- Face Recognition API, for easily real-time detection to detect people's faces.
- Face Blur API for blurring people's faces, as it's essential to ensure privacy for people's identity in public places.

6 Design Constraints

6.1 Standards Compliance

- European Data Protection Board ('EDPB') and the General Data Protection Regulation ('GDPR') settled a guidelines on how to process data from CCTV devices without sharing citizen's bio metrics and personal data including car plates,person's face and many other private things that should be avoided from being shared. [25][26]

6.2 Hardware Limitations

When this project gets applied for real-life use, surveillance cameras can be costly. This project requires various cameras to get installed in multiple areas across a smart city. Moreover, an additional cost gets added for the maintenance of these cameras.

6.3 Other Constraints

Safety and Security considerations: the data transferred through the network and the databases used should be secured against any malicious deformations, Ismagilova et al. discussed it in [27].
Reliability and Fault tolerance: Data should not be corrupted or lost in any case, especially while transferring data through the system layers, N. Mohamed and et al. discussed it in [28].

7 Non-functional Requirements

7.1 Security

The proposed system ensures security and provides data privacy to prevent anonymous and malicious users from using the transmitted data. As mentioned in [29], fog and edge computing deal with privacy issues by reducing the amount of data propagation as it processes the data locally which increases the data security. Also, network isolation mechanisms or the physical infrastructure access policies can be applied to provide a more secure and privacy system.

7.2 Scalability

The proposed system ensures scalability to handle such massive generated data, it must be constructed to be a real-time scalable up and down system. It's expected for the fog networks to deal with more nodes. So, it is needed for them to become acclimated to the system scalability. Otherwise, due to the increment of developing IoT devices, edge services guaranteeing to scale-out accordingly to the enormous amount of devices that being connected to the edge of the network.

7.3 Performance

Fog and edge models guaranteeing the best performance real-time system. They have proven that in the bandwidth and the response time especially. For edge computing, nodes need to be placed near the end of the devices. So, it will reduce the amount of receiving data, which leads to low latency and high bandwidth. Also, due to the small amount of transmitted data as it preprocessing locally. Fog computing achieves low latency and high performance efficiently.

7.4 Interoperability

It's expected for fog, edge, and cloud to be managed by various providers. So, the system ensures interoperability between them, as it's essential when utilizing services from various cloud providers to have the ability to transfer workloads among them. Also, for fog and edge computing, the components are managed by different providers.

7.5 Reliability

With large numbers of edge computing devices and edge data centers associated with the network, it increases the difficulty for any failure to shut down service entirely. In the proposed architecture cloud layer offers backup and archive services in order to guarantee clients retain admittance to

the data they need later on. Effectively incorporating IoT edge computing devices and edge data centers into an extensive edge architecture can accordingly give reliability.

8 Data Design

8.1 Data Description

We have acquired a data set of surveillance videos published by the **'UCF Center for research in computer vision'** called **UCF-Crime**; It is a large-scale dataset of 128 hours of videos; it has 1900 video of many classes as Robbery, Stealing, Shooting ... etc. And it is widely used in any research or project that targets public safety and surveillance. Therefore we took some of those videos from the classes that meet our requirements. 8.2 illustrates the dataset.

8.2 Database design description

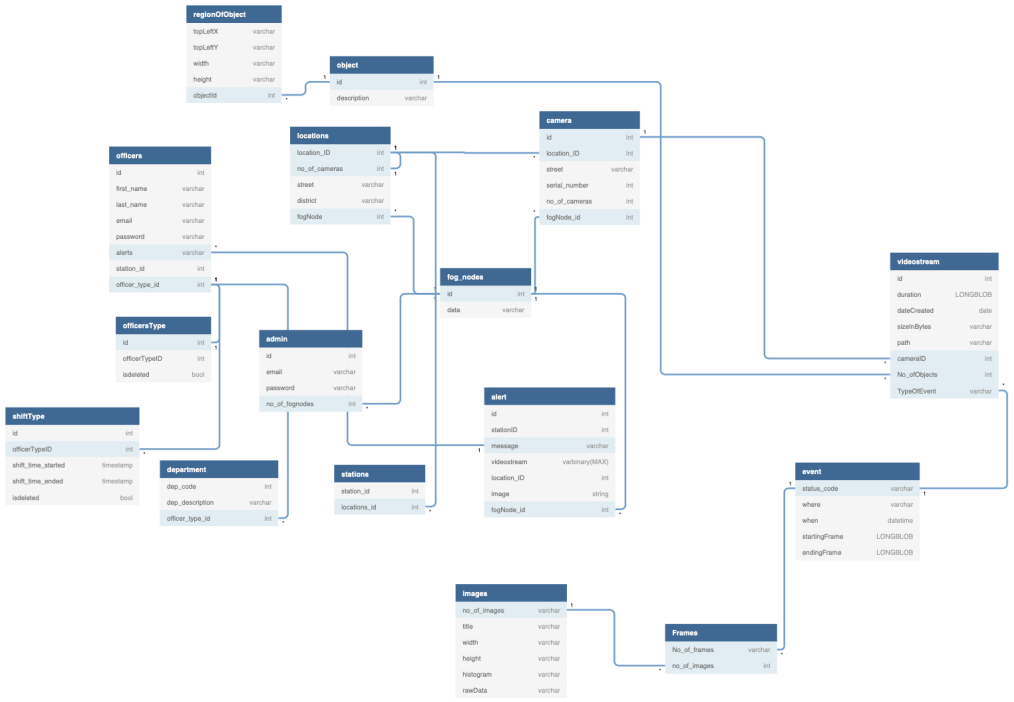


Figure 14: Database of the software

This database in Figure 14 is for the users' software; it also interacts with the criminals' database provided by the social security, and data sets updated frequently to help the performance of detection algorithms used by the system.

Table 10: Dataset Description

Dataset Name	UCF-Crime Dataset
Link	https://www.crcv.ucf.edu/projects/real-world/
Size	95.9 GB
Number of classes	The dataset has 13 classes for anomaly events and normal class; we used 3 of them
Average Video frames	1000-1500 frames per video
Number of videos used	we used for fighting 10 videos,Robbery 10 videos,normal events 10 videos
comments	This data set has many videos in many places; we chose the suitable videos in streets. Also it didn't include kidnapping events; so we gathered 6 surveillance videos from Youtube with same video length and specifications as UCF-crime videos

9 Preliminary Object-Oriented Domain Analysis

9.1 Inheritance Relationships

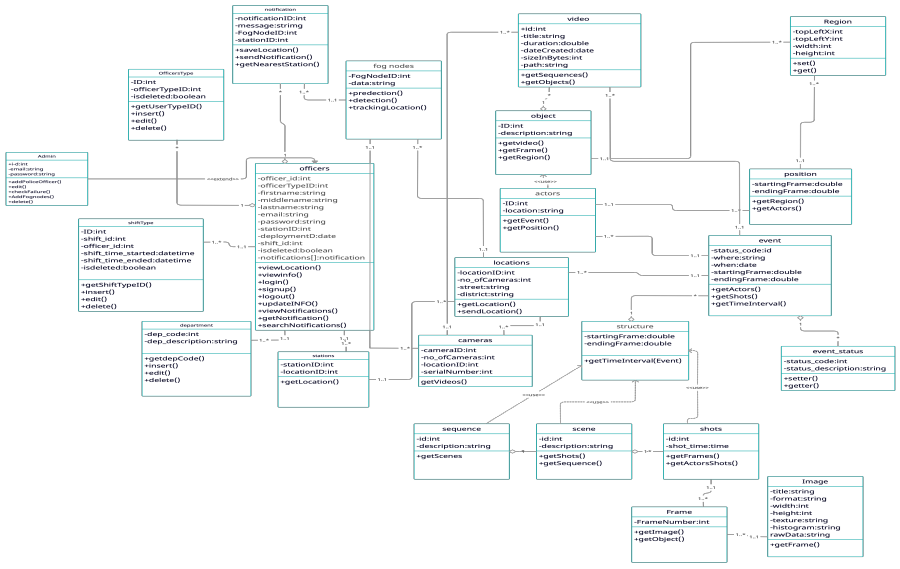


Figure 15: Class Diagram

9.2 Class descriptions

Table 11: Class Name - officers

List of Superclasses	none
List of Subclasses	none
Purpose	The class represents police officer'data
Collaborations	associated with shift type,stations,department and aggregated with officers type and notifications.
Attributes	officerid:int,officerTypeID:int,firstname:string,last name:string,email:string, password:string,stationID:int,deployment date:date,shiftID:int,isdeleted:boolean, notifications[]:notification.
Operations	login(email,password),logout(),signUP(firstname,lastname,email,password, officersTypeID),updateINFO(officersID,firstname,lastname,email,password), viewNotifications(notificationID),searchNotifications().
Constraints	officer is one of the core classes of the application

Table 12: Class Name - Officers type

List of Superclasses	none
List of Subclasses	none
Purpose	the class represents officer's types.
Collaborations	aggregated with officers.
Attributes	ID:int,officerTypeID:int,isdeleted:boolean
Operations	getusertypeID(),insert(),edit(),delete()
Constraints	cannot work without officers class

Table 13: Class Name - Admin

List of Super-classes	officers
List of Sub-classes	none
Purpose	a class represent admin in the system
Collaborations	extends with officers.
Attributes	ID:int,email:string,password:string
Operations	createpoliceofficer(),edit(),delete(),checkFailure(),AddFognodes()
Constraints	Admin is the main controller in the application.

Table 14: Class Name - shift Type

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents shift types for the police officers.
Collaborations	associated with officers.
Attributes	ID:int,shiftID:int,officerID:int,shiftTimeStarted:datetime,shiftTimeEnded :date-time,isdeleted:boolean
Operations	getShiftTypeID(),insert(),edit(),delete()
Constraints	can't work without officers.

Table 15: Class Name - department

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents departments of the police officers.
Collaborations	associated with officers.
Attributes	depCode:int,depDescription:string
Operations	getDepCode(),insert(),edit(),delete()
Constraints	cannot work without officers.

Table 16: Class Name - stations

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents stations locations of each police officer.
Collaborations	associated with officers and locations.
Attributes	stationID:int,locationID:int
Operations	getLocation()
Constraints	cannot work without officers and locations.

Table 17: Class Name - notifications

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the notifications that views to the police officer.
Collaborations	associated with fog nodes and aggregated with officers.
Attributes	ID:int,message:string,FogNodeID:int
Operations	saveLocation(),sendNotification(),getNearestStation() getLocation()
Constraints	cannot work without officers and fognodes.

Table 18: Class Name - fog nodes

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the fognodes that sends the alerts that came from the edge node to the officers through class notifications
Collaborations	associated with notifications,locations and cameras.
Attributes	FogNodeID:int,data:string
Operations	prediction(),detection(),trackingLocation()
Constraints	cannot work without locations and cameras.

Table 19: Class Name - cameras

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the edge nodes that carries the video data and ready to be sent to fog nodes.
Collaborations	associated with locations,fognodes and videos.
Attributes	cameraID:int,noOfcameras:int,locationID:int ,serialNumber:int
Operations	getVideos(),sendToFognodes(),caameraFailure(),CrimeDetection()
Constraints	cannot work without videos and locations.

Table 20: Class Name - locations

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the locations of the surveillance camera that the kidnapping event will occur.
Collaborations	associated with cameras,stations,fogNodes and event.
Attributes	LocationID:int,noOfcameras:int,street:string,district:string
Operations	getLocation(),sendLocation()
Constraints	cannot work without event class

Table 21: Class Name - video

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents video data.
Collaborations	associated with cameras,event and aggregated with object. and event.
Attributes	id:int,title:string,duration:double,dateCreated:date,sizeInBytes:int,path:string
Operations	getSequences(),sendObjects()
Constraints	cannot work without object class.

Table 22: Class Name - object

List of Super-classes	none
List of Sub-classes	actors
Purpose	the class represents objects in the video frame.
Collaborations	associated with region and aggregated with video class.
Attributes	id:int,description:string
Operations	getVideo(),getFrame(),getRegion(),faceDetection(),faceBlurring
Constraints	cannot work without video class.

Table 23: Class Name - actors

List of Super-classes	object
List of Sub-classes	none
Purpose	the class represents actors which is mainly the kidnapper or the victim in the video frame.
Collaborations	associated with position and event.
Attributes	id:int,location:string
Operations	getEvent(),getPosition()
Constraints	cannot work without object class.

Table 24: Class Name - Region

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the region of the objects that are in the frame video.
Collaborations	associated with position and object.
Attributes	topLeftX:int,topLeftY:int,width:int,height:int
Operations	getter(),setter()
Constraints	cannot work without object and position classes.

Table 25: Class Name - position

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the exact video frame that the event occur at be saved and detection will be processed on it.
Collaborations	associated with region and actors.
Attributes	startingFrame:double,endingFrame:double
Operations	getRegion(),getActors()
Constraints	cannot work without actors class.

Table 26: Class Name - event

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the event of kidnapping.
Collaborations	associated with video,actors,locations and aggregated with event status and structure
Attributes	statusCode:id,where:string,when:date,startingFrame:double,endingFrame:double.
Operations	getshots(),getActors(),getTimeInterval()
Constraints	cannot work without actors and locations classes.

Table 27: Class Name - event status

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the status of the event occurred.
Collaborations	aggregated with event class.
Attributes	statusCode:id,statusDescription:string
Operations	getter(),setter()
Constraints	cannot work without event class.

Table 28: Class Name - structure

List of Super-classes	none
List of Sub-classes	scene,sequence,shot
Purpose	the class represents the structure of the event occurred that is divided into shots,scenes and sequence.
Collaborations	aggregated with event class.
Attributes	startingFrame:double,endingFrame:double
Operations	getTimeInterval(Event)
Constraints	cannot work without event class.

Table 29: Class Name - sequence

List of Super-classes	structure
List of Sub-classes	none
Purpose	the class represents the sequence of the structure the video surveillance captured when an event occured.
Collaborations	aggregated with scene class.
Attributes	id:int,description:string
Operations	getScenes()
Constraints	cannot work without structure class.

Table 30: Class Name - scene

List of Super-classes	structure
List of Sub-classes	none
Purpose	the class represents the scenes of the kidnapping event that occurred.
Collaborations	aggregated with shots class.
Attributes	id:int,description:string
Operations	getSequence(),getShots()
Constraints	cannot work without structure class.

Table 31: Class Name - shots

List of Super-classes	structure
List of Sub-classes	none
Purpose	the class represents the shots of the actors in the event that occurred.
Collaborations	associated with frame class.
Attributes	id:int,shotTime:time
Operations	getFrames(),getActorShots()
Constraints	cannot work without structure class.

Table 32: Class Name - Frame

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the frame of the actors that occurred in an event.
Collaborations	associated with shot and image classes.
Attributes	FrameNumber:int
Operations	getImage(),getObject()
Constraints	cannot work without shot class.

Table 33: Class Name - Image

List of Super-classes	none
List of Sub-classes	none
Purpose	the class represents the image data of the frame captured.
Collaborations	associated with frame class.
Attributes	title:string,format:string,width:int,height:int,texture:string, histogram:string,rawData:string
Operations	getFrame(), deepLearningDetection(),videoSummarization()
Constraints	cannot work without Frame class.

10 Operational Scenarios

10.1 Scenario 1: Intelligent Surveillance System

As shown in Figure 16 and Figure 17, the system should go through various stages to successfully propose to users the required information to take the appropriate action to stop the kidnapping. First, the system should check for any failures in the surveillance cameras. If camera failure gets detected, the system should determine its cause and manage to fix it. If not, face blurring gets applied to protect people's privacy while the surveillance data get processed. Moreover, a motion detection algorithm gets used to identify if any abnormal behavior occurred. When motion gets observed, the video stream containing the occasion gets dispatched for further processing. A prediction algorithm gets then implemented to determine if the event is a kidnapping case. If yes, a face recognition algorithm gets used to identify the kidnapper. Moreover, an alarm gets triggered to notify the users, and the kidnapper gets tracked.

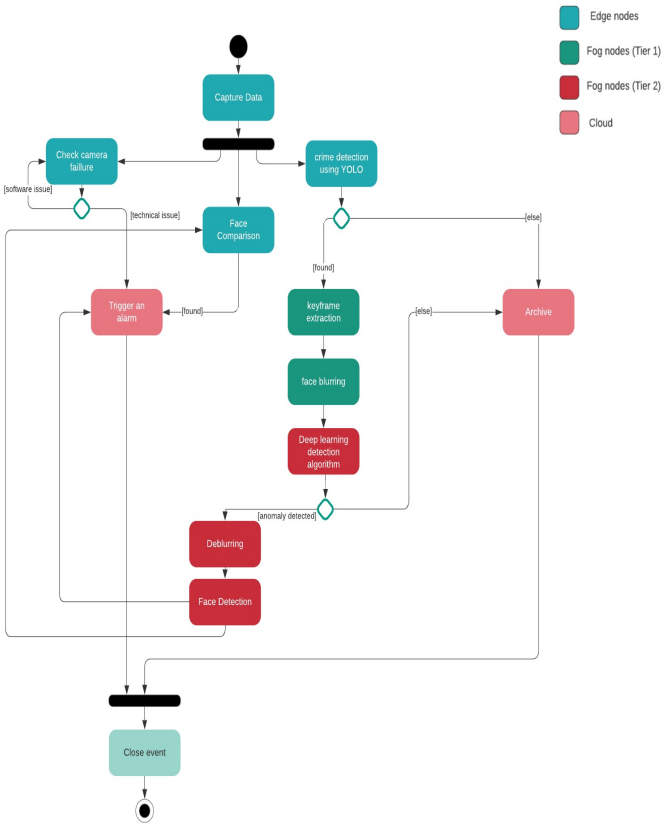


Figure 16: Activity Diagram

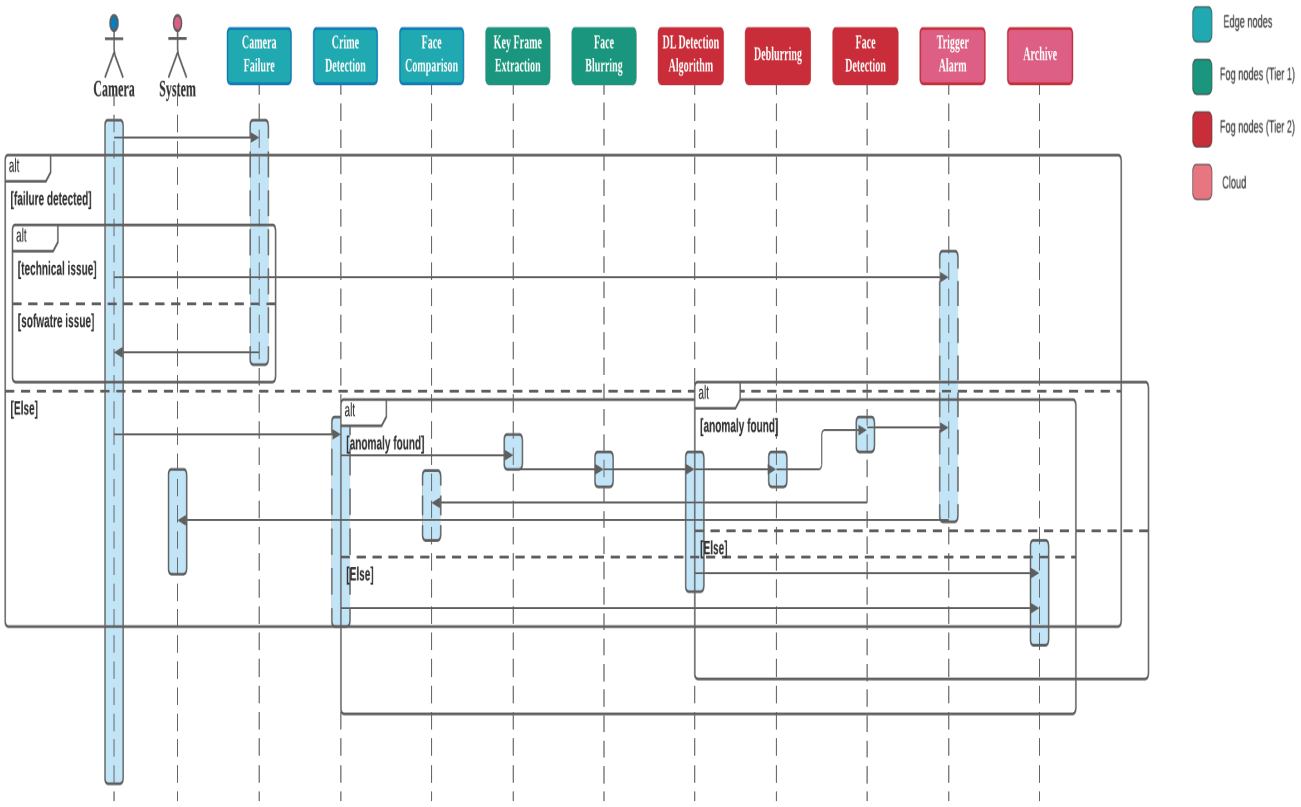


Figure 17: Sequence Diagram

10.2 Scenario 2: Admin

The system's admin controls all the users within the system. He has the ability to:

- Add new police officers.
- Edit police officers' information.
- Delete police officers.

10.3 Scenario 3: Police Units

Police officers can sign up and log into the system. Moreover, when an incident occurs, they get notified via notifications. When they view notifications, they can then go through video streams, identify the kidnappers' faces, and track the location of incidents.

11 Project Plan

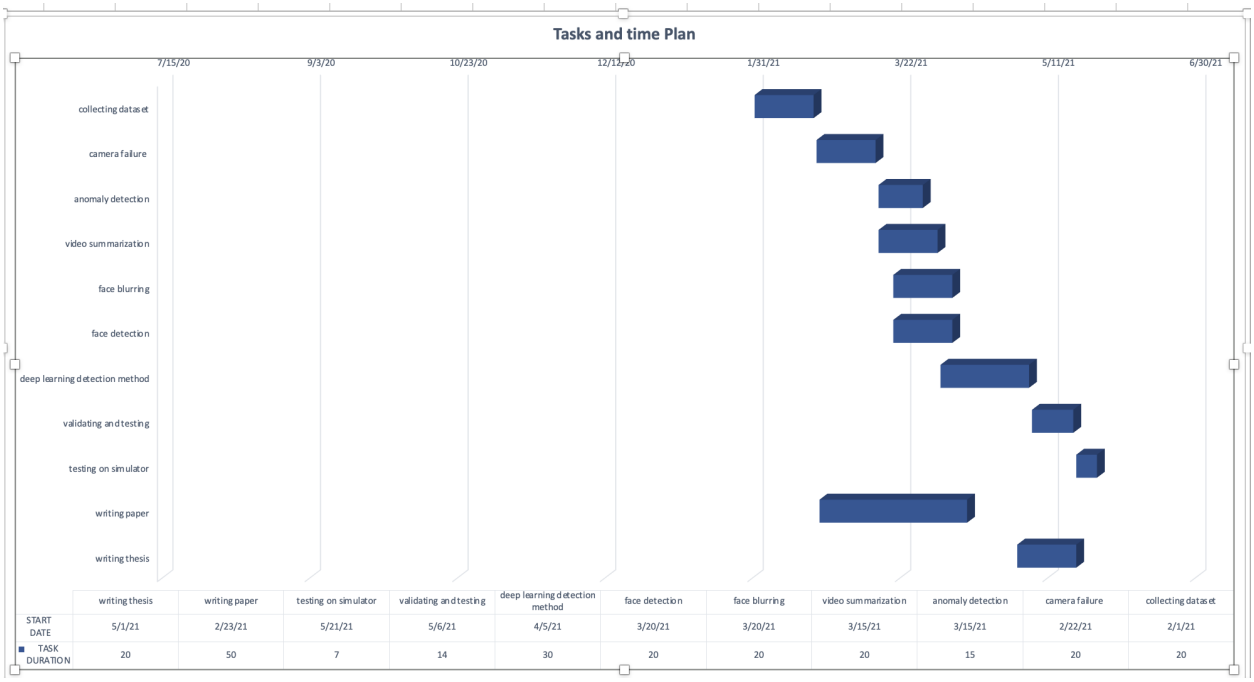


Table 34: ISS time plan

Id	Task	Start Date	Number of Days	Team Member
1	Collecting Dataset	2/1/2021	5	nour ahmed, sandra
2	Work on GUI	3/1/2021	15	samiha, mariam
3	camera failure	3/5/2021	10	nour elhoda
4	video summarization	3/1/2021	10	nour ahmed
5	crime detection	3/1/2021	10	sandra
6	face blurring technique	3/20/2021	20	mariam
7	face detection	3/20/2021	20	samiha
8	deep learning detection classifier	4/5/2021	35	nour ahmed,sandra
9	mobile application	4/5/2021	40	samiha,mariam
10	enhancing video quality	4/5/2021	30	nour elhoda

12 Appendices

12.1 Definitions, Acronyms, Abbreviations

Fog computing: A mid-layer between edge devices and the cloud, at which data processing occurs.

Edge computing: An architecture at which data gets processed on devices rather than the cloud to provide low latency.

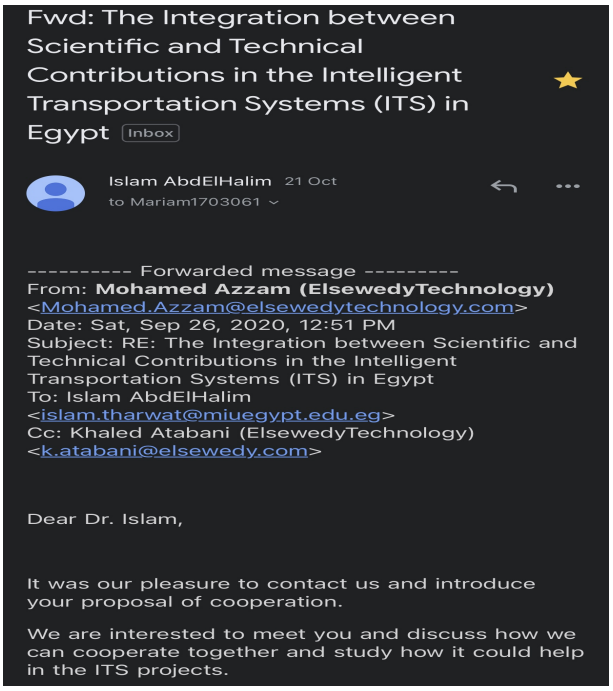
SQL : Structured Query Language used to communicate with database.

EFISS:Edge Fog Intelligent Surveillance System.

12.2 Supportive Documents

We have done a systematic review, as well as a review paper, which was sent to SSIC 2021(3rd International Conference on Smart Systems: Innovations in Computing). The paper is accepted now and we are waiting to publish it.The paper is available on github.

Figure 18: Contacting ELSEWEDY technology company



Fwd: The Integration between Scientific and Technical Contributions in the Intelligent Transportation Systems (ITS) in Egypt Inbox

Islam AbdElHalim 21 Oct
to Mariam1703061

---------- Forwarded message ---------
From: **Mohamed Azzam (ElsewedyTechnology)** <Mohamed.Azzam@elsewedytechnology.com>
Date: Sat, Sep 26, 2020, 12:51 PM
Subject: RE: The Integration between Scientific and Technical Contributions in the Intelligent Transportation Systems (ITS) in Egypt
To: Islam AbdElHalim <islam.tharwat@miuegypt.edu.eg>
Cc: Khaled Atabani (ElsewedyTechnology) <k.atabani@elsewedy.com>

Dear Dr. Islam,

It was our pleasure to contact us and introduce your proposal of cooperation.

We are interested to meet you and discuss how we can cooperate together and study how it could help in the ITS projects.

Figure 19: The company's response

Intelligent Surveillance Systems for Smart Cities: A Systematic Literature Review

Nour Ahmed Ghoniem, Samiha Hesham, Sandra Fares, Mariam Hesham, Lobna Shaheen, and Islam Tharwat Abdel Halim

Abstract The concept of smart cities is emerging rapidly with the objectives of using the technology to enhance the efficiency and quality of services provided to the population in such cities. Due to the importance of having continuous and uninterrupted urban space monitoring, many studies have been carried out to propose an intelligent surveillance systems for smart cities. Several research topics have already been explored in such domain. However, given the variety of problems and solutions, it is complex to identify research trends and gaps related to intelligent surveillance systems for smart cities without a systematic review. Therefore, this paper presents a systematic literature review of such a topic. The review revealed current research trends, challenges, and future directions in the area of intelligent surveillance systems. This work provides a list of references to find the mapped papers, thus paving the way for enhancing the efficiency, reliability, and robustness of such systems.

Keywords Internet of things (IoT), Computer Vision, Cloud and Edge Computing, Digital Image and Video Processing, Security and Privacy.

1. Introduction

Nowadays, "Smart City" is a development goal for many cities around the world. The concept of a smart city has been developed by "the implementation of digital technologies in different aspects of urban management and is considered as one of the pillars of the Industry 4.0" [1]. The ultimate goal of a smart city is to create a new urban management perspective that focuses on all aspects of urban real life. In recent times, there has been a lot of research done showing that the surveillance systems are of vital importance for such smart cities. This is justified because the surveillance systems can be considered vision organs of the smart city [2]. Thus, the design of the surveillance systems for a smart city must itself be smart.

In traditional surveillance systems, human operators are the main responsible to manipulate the processing of captured video [3]. Certainly, future smart cities will generate and process an immense volume of data [4]. Hence, various forms of sensing devices (smart vehicles, traffic lights, smartphones, etc.) that are present almost everywhere nowadays can be utilized to develop efficient surveillance systems and urban space monitoring for smart cities. Accordingly, intelligent surveillance systems have been gaining a lot of attention from the scientific community for promising advances in smart cities.

N.A. Ghoniem (✉), S. Hesham, S. Fares, M. Hesham, L. Shaheen, and I.T.A. Halim
e-mail: nour1704021@miuegypt.edu.eg
Faculty of Computer Science, Misr International University (MIU), Cairo, Egypt

S. Hesham
e-mail: samiha1707594@miuegypt.edu.eg

S. Fares
e-mail: sandra1701421@miuegypt.edu.eg

M. Hesham
e-mail: mariam1703061@miuegypt.edu.eg

L. Shaheen
e-mail: lobna.mostafa@miuegypt.edu.eg

I.T.A. Halim
e-mail: Islam.tharwat@miuegypt.edu.eg

Intelligent adoption of surveillance systems could protect, direct, manage, and influence human beings and property [5]. Nevertheless, several problems such as data quality, privacy, government support, security, and computing paradigm. Hence, to understand the state of the art in studies of such systems, we conduct a systematic literature review that aims to help researchers to identify trends and saturated themes in scientific works published in the last 5 years. We use the insights from the review to understand and outline the perspective for the current research trends, challenges and limitations, and future directions of intelligent surveillance systems and urban space monitoring in smart cities.

The layout of the rest of the paper is as follows. Section 2 delineates the research methodology of this work. The perspective for the current research is discussed in Section 3. Section 4 outlines the current challenges and future areas of exploration on this topic. Finally, Sections 5 provides concluding remarks.

Figure 20: Review Paper

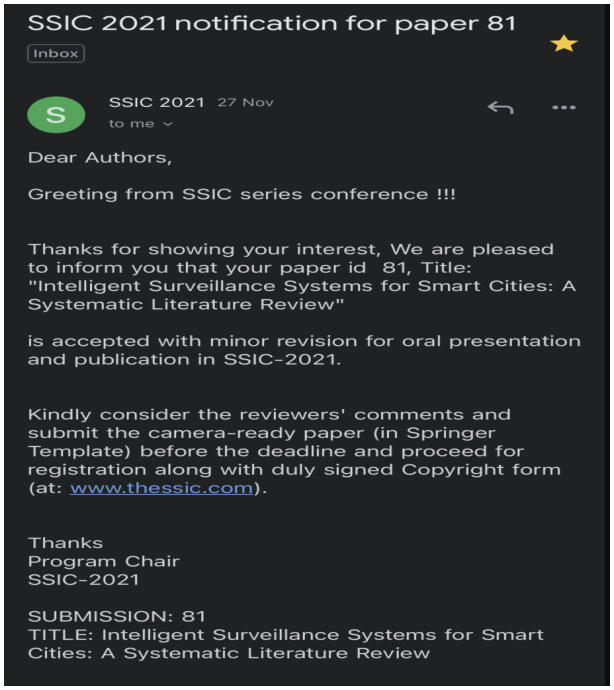


Figure 21: Acceptance email from SSIC

13 References

References

[1] K. Zhang, J. Ni, K. Yang, et al. "Security and Privacy in Smart City Applications: Challenges and Solutions". In: *IEEE Communications Magazine* 55.1 (2017), pp. 122–129. DOI: 10.1109/MCOM.2017.1600267CM.

[2] Mluleki Sinqadu and Zelalem Sintayehu Shibeshi. "Performance Evaluation of a Traffic Surveillance Application Using iFogSim". In: *3rd International Conference on Wireless, Intelligent and Distributed Environment for Communication*. Ed. by Isaac Woungang and Sanjay Kumar Dhurandher. Cham: Springer International Publishing, 2020, pp. 51–64.

[3] L. Cui, G. Xie, Y. Qu, et al. "Security and Privacy in Smart Cities: Challenges and Opportunities". In: *IEEE Access* 6 (2018), pp. 46134–46145. DOI: 10.1109/ACCESS.2018.2853985.

[4] template. "Software Requirement Specification Document". In: *IEEE Communications Magazine* (2020), p. 9.

[5] I. Ledakis, T. Bouras, G. Kioumourtzis, et al. "Adaptive Edge and Fog Computing Paradigm for Wide Area Video and Audio Surveillance". In: *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 2018, pp. 1–5. DOI: 10.1109/IISA.2018.8633626.

[6] Jianyu Wang, Jianli Pan, and Flavio Esposito. "Elastic Urban Video Surveillance System Using Edge Computing". In: New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450355285. DOI: 10.1145/3132479.3132490. URL: https://doi.org/10.1145/3132479.3132490.

[7] X. Xu, Q. Wu, L. Qi, et al. "Trust-Aware Service Offloading for Video Surveillance in Edge Computing Enabled Internet of Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–10. DOI: 10.1109/TITS.2020.2995622.

[8] Tanin Sultana and Khan A Wahid. "IoT-Guard: Event-Driven Fog-Based Video Surveillance System for Real-Time Security Management". In: *IEEE Access* 7 (2019), pp. 134881–134894.

[9] Augusto JV Neto, Zhongliang Zhao, Joel JPC Rodrigues, et al. "Fog-based crime-assistance in smart iot transportation system". In: *IEEE access* 6 (2018), pp. 11101–11111.

[10] Francisco A Pujol, Higinio Mora, and Maria Luisa Pertegal. "A soft computing approach to violence detection in social media for smart cities". In: *Soft Computing* (2019), pp. 1–11.

[11] Hui Sun, Weisong Shi, Xu Liang, et al. "VU: Edge Computing-Enabled Video Usefulness Detection and its Application in Large-Scale Video Surveillance Systems". In: *IEEE Internet of Things Journal* 7.2 (2019), pp. 800–817.

[12] Tanveer Hussain, Khan Muhammad, Amin Ullah, et al. "Multi-View Summarization and Activity Recognition Meet Edge Computing in IoT Environments". In: *IEEE Internet of Things Journal* (2020).

[13] Tanveer Hussain, Khan Muhammad, Weiping Ding, et al. "A comprehensive survey of multi-view video summarization". In: *Pattern Recognition* 109 (2020), p. 107567.

[14] Amilcare Francesco Santamaria, Pierfrancesco Raimondo, Nunzia Palmieri, et al. "Cooperative video-surveillance framework in internet of things (IoT) domain". In: *The Internet of Things for Smart Urban Ecosystems*. Springer, 2019, pp. 305–331.

[15] Gabriele Baldoni, Marcello Melita, Sergio Micalizzi, et al. "A dynamic, plug-and-play and efficient video surveillance platform for smart cities". In: *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2017, pp. 611–612.

[16] Jianyu Wang, Jianli Pan, and Flavio Esposito. "Elastic urban video surveillance system using edge computing". In: *Proceedings of the Workshop on Smart Internet of Things*. 2017, pp. 1–6.

[17] Yutong Liu, Linghe Kong, Muhammad Hassan, et al. "Litedge: towards light-weight edge computing for efficient wireless surveillance system". In: *Proceedings of the International Symposium on Quality of Service*. 2019, pp. 1–10.

[18] Mansoor Nasir, Khan Muhammad, Jaime Lloret, et al. "Fog computing enabled cost-effective distributed summarization of surveillance videos for smart cities". In: *Journal of Parallel and Distributed Computing* 126 (2019), pp. 161–170.

[19] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, et al. "UAVFog: A UAV-based fog computing for Internet of Things". In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBL* IEEE. 2017, pp. 1–8.

[20] Ruimiao Ding, Xuejun Li, Xiao Liu, et al. "A cost-effective time-constrained multi-workflow scheduling strategy in fog computing". In: *International Conference on Service-Oriented Computing*. Springer. 2018, pp. 194–207.

[21] *LOGIPIX SAFE SMART CITY SOLUTION*. URL: `https://www.logipix.com/safe-and-smart-city-solution/`.

[22] *Surveillance solutions that build smart cities*. URL: `https://business.bt.com/solutions/surveillance/smart-cities/`.

[23] Harshit Gupta, Amir Dastjerdi, Soumya Ghosh, et al. "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments". In: *Software: Practice and Experience* 47 (June 2016). DOI: `10.1002/spe.2509`.

[24] R. Buyya and S. N. Srirama. "Modeling and Simulation of Fog and Edge Computing Environments Using iFogSim Toolkit". In: *Fog and Edge Computing: Principles and Paradigms*. 2019, pp. 433–465. DOI: `10.1002/9781119525080.ch17`.

[25] European Data Protection Board. "Guidelines 3/2019 on processing of personal data through video devices". In: EDPB Plenary meeting. 2019, pp. 1–29.

[26] General Data Protection Regulation. "Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is "likely to result in a high risk" for the purposes of Regulation 2016/679". In: GDPR meeting. 2017, pp. 1–22.

[27] Elvira Ismagilova, Laurie Hughes, Nripendra P. Rana, et al. “Security, Privacy and Risks Within Smart Cities: Literature Review and Development of a Smart City Interaction Framework”. In: *Information Systems Frontiers* (July 2020). ISSN: 1572-9419. DOI: 10.1007/s10796-020-10044-1. URL: https://doi.org/10.1007/s10796-020-10044-1.

[28] N. Mohamed, J. Al-Jaroodi, and I. Jawhar. “Towards Fault Tolerant Fog Computing for IoT-Based Smart City Applications”. In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 2019, pp. 0752–0757. DOI: 10.1109/CCWC.2019.8666447.

[29] Amal Al-Qamash, Iten Soliman, Rawan Abulibdeh, et al. “Cloud, Fog, and Edge Computing: A Software Engineering Perspective”. In: *2018 International Conference on Computer and Applications (ICCA)*. IEEE. 2018, pp. 276–284.