Software Proposal Document for project Intelligent Surveillance for smart cities

Mariam Hesham,Nour Ahmed ,Samiha Hesham,Sandra Fares

October 24, 2020

Proposal Version	Date	Reason for Change
1.0	24-October-2020	Proposal First version's specifications are defined

Table 1: Document version history

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

Abstract

Surveillance systems are of vital importance for the development of smart cities. 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. 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.

1 Introduction

1.1 Background

The video surveillance system is a crucial component in the development of smart cities. Where they are useful for crime prevention,terrorist or forensic evidence detection,and traffic monitoring. But the old traditional surveillance cannot accomplish it,that's why smart surveillance systems with smart detection modules were introduced and always in enhancements till now.

However, implementing a citywide smart video surveillance system involves many challenges. A large number of cameras have to be deployed across the city, producing large amounts of data and information every day. Besides, storing and enabling access to video devices and data at this large scale, in both real-time and on-demand manner, demand more computing and storage resources than traditional video surveillance systems. These challenges have to be addressed to maximize the effectiveness of smart video surveillance[7].

Cloud computing was proposed then to solve some of those issues as mentioned in[8] and [1], and it proved to be efficient but with the growth of data, it could not accomplish the desired results, and fast response due to the possible delay of transmitting the data to and from the cloud server, and also it could not solve the privacy and security concerns. So, fog computing technology is recently introduced as a promising one to handle the current challenges of the surveillance systems development for smart cities.

1.2 Motivation

1.2.1 Academic

Since smart cities utilize IoT devices in recent times, in order to improve their infrastructure, numerous amounts of data get generated at a rapid rate. Passing data to the cloud is unfavorable when dealing with surveillance systems, as these systems require urgent and instant responses to avoid misfortune events. Therefore, it is necessary to procure a strategy that will assist in managing and processing data effectively in a short period of time. Edge computing is a recently proposed solution for this problem. Instead of transferring data to the cloud, it gets transferred to the edge for real-time data processing. Various techniques are used to exploit this new technology and use it efficiently to solve different barriers. B. V. Natesha and Ram Mohana Reddy Guddeti [19] proposed a fog computing based surveillance system, for smart cities, that processes data using deep learning methods in order to provide low latency and minimize the cost of the services. Likewise, G. Baldoni et al. [4] present edge computing to reduce the network traffic and offer scalability.

1.2.2 Business

With the expansion of IoT devices in smart cities, there is a massive amount of data generated from the cameras. The edge-based surveillance system provides businesses and organizations with significant benefits. It provides real-time data processing analysis which helps the organizations to take proper action and make an immediate decision efficiently in case of an accident or abnormal events. Furthermore, huge organizations as smart cities need to keep their information and their citizen's data secure. Edge computing provides the system with security and privacy while working on lower bandwidth. Moreover, Edge computing needs limited storage to keep the data. So, there is no need for a central server for action-taking. Therefore it provides the business and the organizations with lower costs.

1.3 Problem Statement

One of the challenges that face surveillance systems in smart cities is Big data, as the smart city involves a huge amount of cameras in 24/7 mode to cover the whole city. Each camera records a massive amount of video and audio data to be processed. To handle such massive data, it is essential to provide a way to get real-time data processing analysis efficiently. Cloud computing has been utilized to perform data processing analysis leading to slower performance. Therefore, edge computing is essential for real-time data processing analysis with high performance and low latency.

2 Project Description

2.1 Objectives

Our main objective is to build a fog computing based framework for surveillance data in order to detect abnormal human behaviors in the streets. We will improve an algorithm for anomaly detection with remarkable stability and accuracy. The system is also developed to work on enhancing the latency by reducing the amount of data. Moreover, we will work on developing the privacy protection scheme for data security.

- The system will provide real time data processing analysis.
- The system will response immediately and take a proper action when an event is detected.
- The system will provide privacy assurance for the citizens.
- The system will provide scalability with unlimited storage.

2.2 Scope

- Fog/Edge computing-based surveillance system to monitoring roads, streets, and squares in smart cities.
- Scalability system to handle such massive data generated from surveillance cameras and sensors.
- Efficiently detect any abnormal actions caused by human behavior and response immediately.
- Privacy assurance by blur people's faces to secure their exposed identity in the video.

2.3 Project Overview

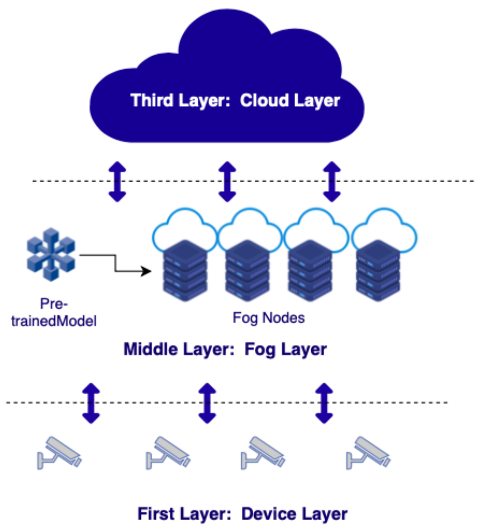


Figure 1: three-layer fog computing architecture

Our system will be classified into three layers as shown in the figure. The first layer is the device layer where the surveillance cameras will capture the data and send those video streams over the network, then second which is the fog layer will do most of the work needed as on it the videos will be processed using a pre-trained classification model for anomaly detection with the help of our fog nodes; this layer can also be divided into two layers in order to facilitate and reduce the overhead and after detection, if no abnormality in the streams, those streams will be summarized to reduce the huge amount of data that can be sent, then we will apply a securing algorithm to provide privacy while transferring the data.And finally send them to the cloud layer to be stored and retrieved easily.

2.4 Stakeholder

2.4.1 Internal

Team members:

- Nour Ahmed (team leader)
- Mariam Hesham
- Sandra Fares
- Samiha Hesham
- Dr.Islam Tharwat
- Eng.Lobna Shahen

General supervisor:

- Dr.Ashraf AbdelRaouf.

2.4.2 External

Ministry of communication, Elsewedy technology, Government companies, Smart cities and Grand city shopping centers as they need our system to monitor their places 24/7 with unlimited storage.Also detecting any abnormal activity efficiently.

3 Similar System

3.1 Academic

- Jianyu Wang et al. [6] 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.

- In the hopes of exploiting fog computing and tackling the cost challenges faced when building surveillance systems, Mansoor Nasir et al [13] 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.

- Ning Chen et al. [5] discussed in their chapter (Smart City Surveillance in Fog Computing) fog computing and its impact on smart city surveillance systems. Moreover, they stated the major challenges faced in smart cities, the advantages of fog computing compared to cloud computing, and introduced a traffic detection case study. The system architecture is made up of three layers demonstrated in Figure 2: data collection, fog, and cloud center. In the system, drones capture video streams and send them to get processed, using target tracking and speed calculation algorithms, in the fog layer. Moreover, data can be transmitted to the cloud center for further long-term analysis. This approach provides real-time processing and covers a wide range of areas.

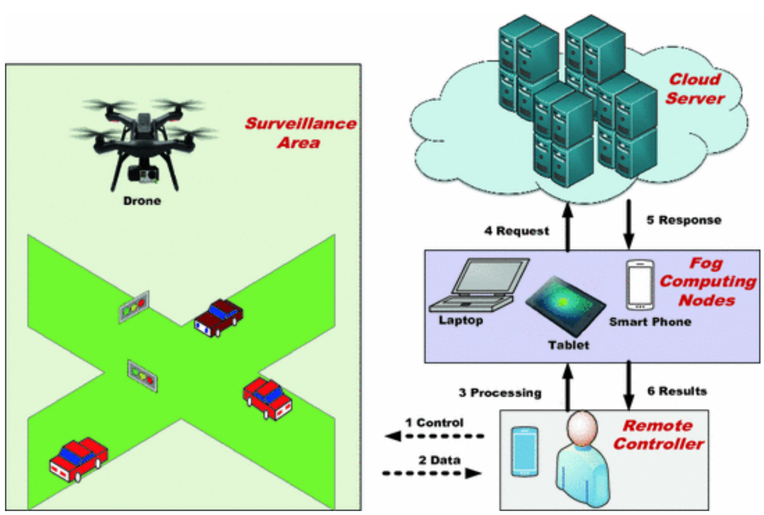


Figure 2: Fog-based Traffic Detection in Smart Cities

- Yutong Liu et al [11] address the obstacles faced when working with wireless surveillance systems, since wireless systems improve the performance and are easier to manage. As the amount of data to be transmitted gets larger, the challenges become visible and the burden on the network increases. This system reduces the pressure on the network by filtering and applying video compression. Moreover, the system provides real-time processing and monitoring using edge computing approach. Experimental results show that the system accomplishes 82 percent latency reduction. As well as, 91.26 percent transmission workload reduction as the redundant data get filtered out.

- Natesha et al[19] proposed two fog architecture as an experiment to reduce latency and cloud resource consumption. The first architecture shown in Figure 3 was the Local Fog Node using SSN. They used the fog nodes at the lower level to analyze and extract the data to reduce data size and to help the low computational powered devices, an Intel Neural Compute Stick(NCS) which provides the pre-trained neural network model for video analysis and real-time object classification had been installed. The second architecture in Figure 4 was Fog Server video processing using DNN. Where a data center is used as the fog server as a layer between the cloud layer and devices. Applying those two architectures on the vehicle data set proved that using fog computing for processing real-time videos is a successful approach while the other architecture is more efficient. However, they proved using one of those architectures or both of them creating a hybrid architecture can solve many challenges.

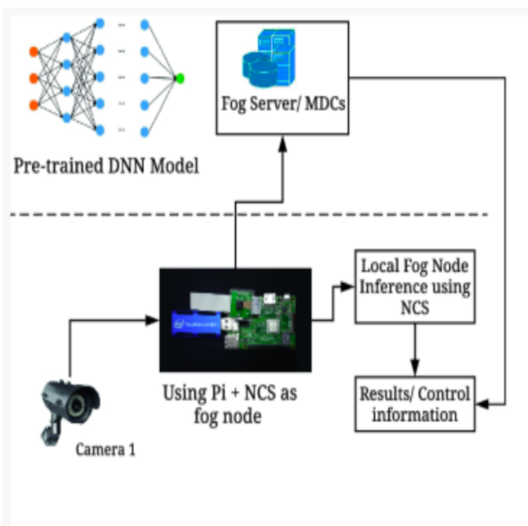


Figure 3: the Local Fog Node using SSN

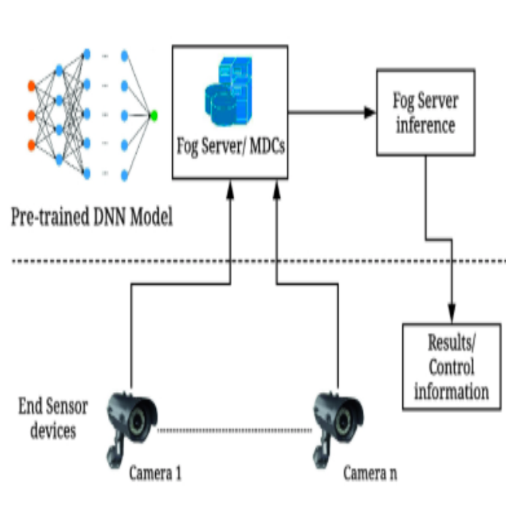


Figure 4: Fog Server video processing using DNN

- Sinqadu et al [16] mentioned that using cloud-based architectures faces low latency while dealing with sensitive real-time applications; Therefore they proposed a new fog-based model for traffic surveillance was proposed. This model is based on three phases, inputting the IoT video streams, recognizing and tracking. Then, They used the ifogSim tool to simulate their approach using FogDevice, Sensor, Actuator, Tuple class, and Application class. The results compared to cloud-based systems proved to be efficient in the average latency, energy consumption, data transfer rate, and more. This paper also proved that fog-based applications do better than cloud ones.

- Santamaria et al [14] mentioned that due to the large number of data collected by surveillance cameras, many challenges appeared to analyze, store, and retrieve data. In this chapter they are introducing the basics needs to build a framework for people's abnormal behaviors. After investigating the current techniques and technologies. Using distributed architecture Figure 5. was the main solution as it helps the performance of the system, then 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 KNN for Object detection and tracking.

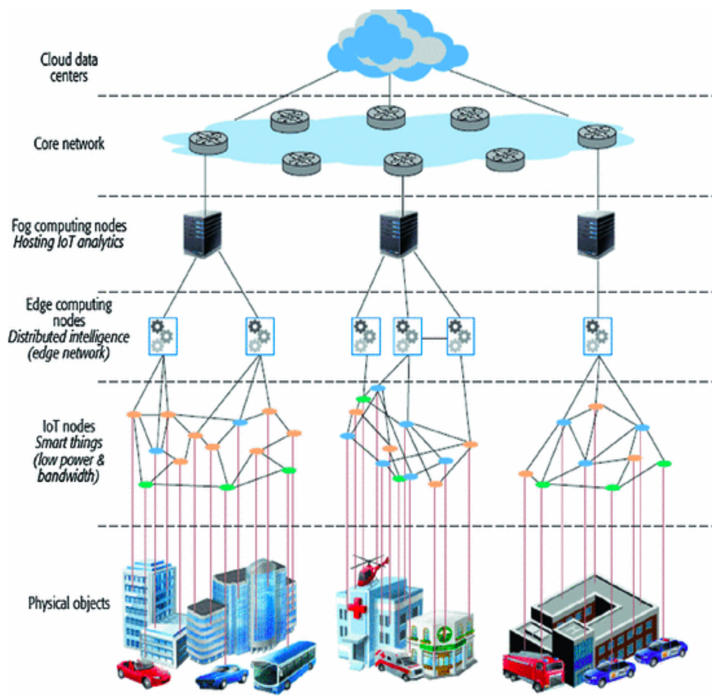


Figure 5: the Distributed eccosystem in cloud domain

- Ioannis Ledakis et al [9] aim to create a real-time surveillance system for smart cities. One of the challenges that face the system is big data. The surveillance system utilized many cameras in 24/7 mode to cover the whole city. So, a massive amount of video and audio data is been generated and needs to be processed. The data must be transmitted from the IoT devices to the central unit for processing analysis. The authors found that this way will lead to late latency. In addition to its complexity when other various types of data are found. In this article, the authors merge the cyber foraging approach for unloading of calculation tasks to machines in the nearest place to the devices. The aim is to create a Fog-Edge based surveillance system that can handle such massive data and provide the system with real-time data processing analysis and decision-making. The result shows the system act with high performance and low latency.

- Ning Chen et al [2] proposed a Fog-based traffic surveillance system to ensure security for the citizens and reduce the crime rate in smart cities. They use drones to monitor the vehicles on the whole road. Also, the system offers various tracking methods for detecting and tracking the suspicious vehicle with high accuracy. The proposed system aims to solve the high latency challenge caused by transferring the massive amount of data generated by the devices to the cloud. They utilized Fog computing to provide the system with real-time analysis for data processing efficiently and rapidly decision-taking in case of detecting a suspicious vehicle or any abnormal events. The proposed experiment result is encouraging. It shows that the system can handle multi-target effectively with high performance and low latency.

- Jianguo Chen et al [10] proposed a deep learning-based distributed intelligent video surveillance system and deploy it in edge computing. Due to the massive data generated from the IoT devices and massive network communication, the system aims to solve such challenges and provide real-time and accurate data analysis with low latency. Therefore, the proposed system will transfer the computing data from the central network to the edge network. The proposed approach shows that the system performs well and can handle such a huge video data and provide accurate and accelerated analysis. Moreover, it shows that edge computing provides the system with scalability and flexibility.

- Baladoni et al[3] Proposed a video surveillance platform for smart cities as shown in Figure 3 that presents the following main features: Opening to different video acquiring technologies,Plug-and-play,Flexibility and scalability with the number of transmitting and receiving devices.The proposed technology is achieved by using SAN and NAN prototypes that are recognized according to the SDN/NFV standard paradigm by using general purpose x86-based hardware equipped with a software SDN switch and a virtualization environment as shown in Figure 4.The proposed platform presents a lot of advantages like reduction of network traffic,Low end to end latency,OpEx and CapEx reduction,Scalability and platform add-ons.

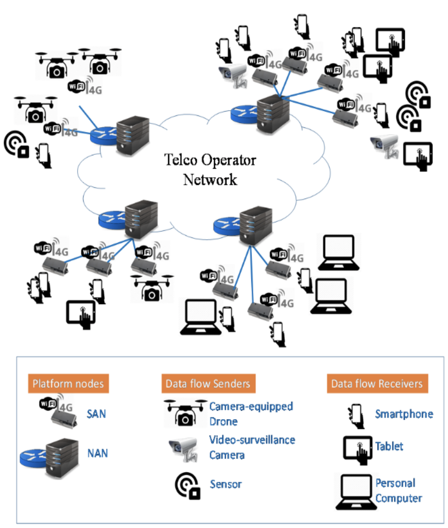


Figure 6: Platfrom architecture

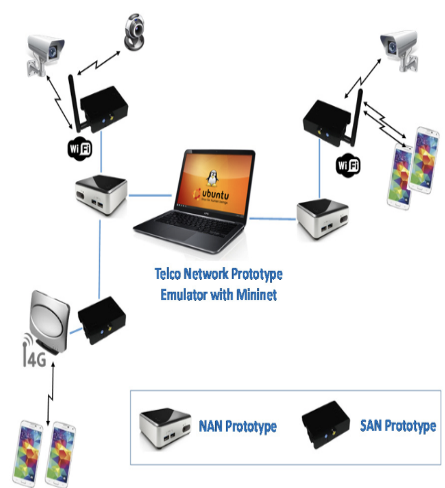


Figure 7: Prototype description

- Xu et al[18] Presented a task offloading method for the video surveillance using the edge computing as shown in Figure 5 that enables IoV to shorten the time cost,Remain the load balance of edge nodes and to maximize the privacy entropy.They presented a new algorithm design called TOM that solves multi-objective optimization problem by SPEA2(improving the strength Pareto evolutionary algorithm).Moreover,Implementing the normalization processing by TOPSIS(Technique for Order Preference by Similarity to Ideal Solution)and MCDM(Multiple Criteria Decision Making)methods.The final results shows that when the number of tasks is huge,TOM has stronger improvement on privacy protection than FFD algorithm which proofs that the experimental simulation performs efficient and trust.

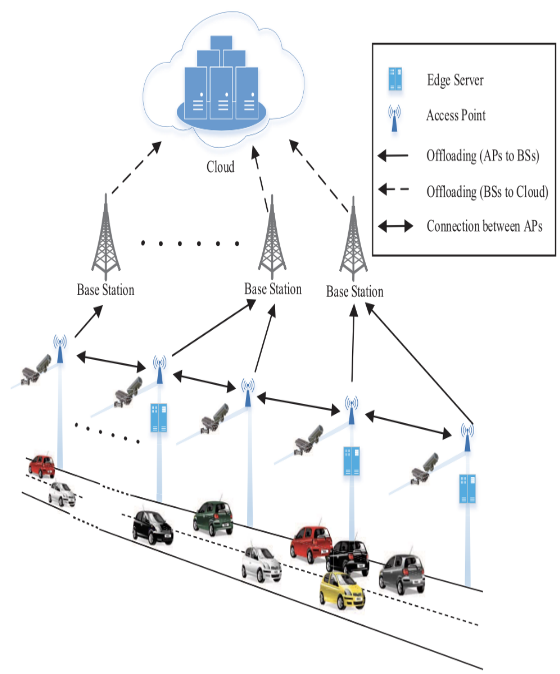


Figure 8: A video surveillance architecture with the edge computing.

- Sun et al [17] Mentioned that due to the large number of data collected by video surveillance many challenges appears later on in the pre-processing stage. As a result, The authors proposed a video usefulness model the(VU) model for large scale video surveillance systems to detect failures in data set to reduce the network bandwidth overload and improve the storage usage in the cloud.Which is based on edge computing and cloud computing. Efficiently and accurately analyzing useful video streams using suitable resources. Moreover,They mentioned ten different types of failures and categorized them into three main domains.They used a data set from a realistic video surveillance system that has approximately 4,433 IP cameras. In addition to 4,232 online cameras that contains different types of failures to examine there proposed model.The types of failures they collected where conducted into two main groups basic evaluation and scalability evaluation.The basic evolution is measured by two metrics which are TP and TN.The results shows that their proposed model helps to reduce the bandwidth of the network and the amount of useless data that are stored in the cloud.

4 What is new in the Proposed Project?

Nowadays, with the expansion of video surveillance system in public places, the system can expose the identity of the people and transfer the private data to server-based video analysis, which make the data insecure and it is considered as violating privacy rights. The proposed system will design Fog-Edge based surveillance system for smart cities. The system provides a real-time data processing analysis effectively and rapidly decision taking with high performance. Moreover, the system ensures privacy protection and secure citizen's data by balancing between the video analysis and unnecessary data that appear the user's identity in the video.

5 Proof of concept

Our system should proceed over several processes for it to perform effectively. As the first step is to model and simulate the system, we initiated the procedure by searching for an appropriate tool to use for building the system's architecture. As shown in Figure 9, iFogSim is a widely used simulation tool. Moreover, it is designed specifically for the modeling and simulation of fog computing, which makes it suitable to use in our project.

Table 3. Simulator tools for fog computing.

Year	Simulator	Implementation Technologies	Metrics	Objective	Citations
2016	Edge-Fog [23]	Python		Distribute task processing on the participating cloud resources	73
2017	FogTorch [64]	Java	QoS, reliability of links and nodes, power consumption, security, monetary costs	Find eligible deployments of an application over a fog infrastructure	173
2017	FogTorchΠ [65]	Extension of FogTorch	Resource utilization and QoS accuracy	Same as FogTorch and many QoS profile according to a probability distribution	51
2017	iFogSim [35]	Extension of CloudSim, Java, JSON	Energy consumption, network congestion, and operational costs	Performance of resource management policies	509
2017	MyiFogSim [34]	Extension of iFogSim	Latency	Resource allocation	38
2018	FogDirSim [70]	RESTful API, Python	Performce in terms of uptime, energy consumption, resource usage	Compare application management and infrastructure management policies	4
2018	FogNetSim++ [63]	based on OMNeT++	Energy module, scheduling algorithms, pricing model	General simulation of fog environments	25
2019	FogWorkFlowSim [72]	Java	Performance (time, energy and processing cost	Evaluate resource and task management strategies	
2019	OPNET [73]	Visual studio, based on OPNET	Processing time	Cope with the massive amount of confidential and security-sensitive data	3
2019	YAFS [68]	Python, JSON	Energy models, network utilization, response time, network delay	Analyze the design and deployment of applications	7
2019	FogDirMime [69]	Python		Support FogDirector [77]	9
2019	Fogbus [71]		Latency, energy, network and CPU usage		27
2020	MobFogSim [5]	Extends iFogSim	Same as iFogSim	Evaluate application behaviour and performance	0

Figure 9: Simulator tools comparison

Additionally, we installed iFogSim and examined the features, packages, and the libraries introduced in it as a set up for our simulation process. As well as the components needed to perform simulation that are displayed in Figure 10: Physical Components, Logical Components, and Management Components.

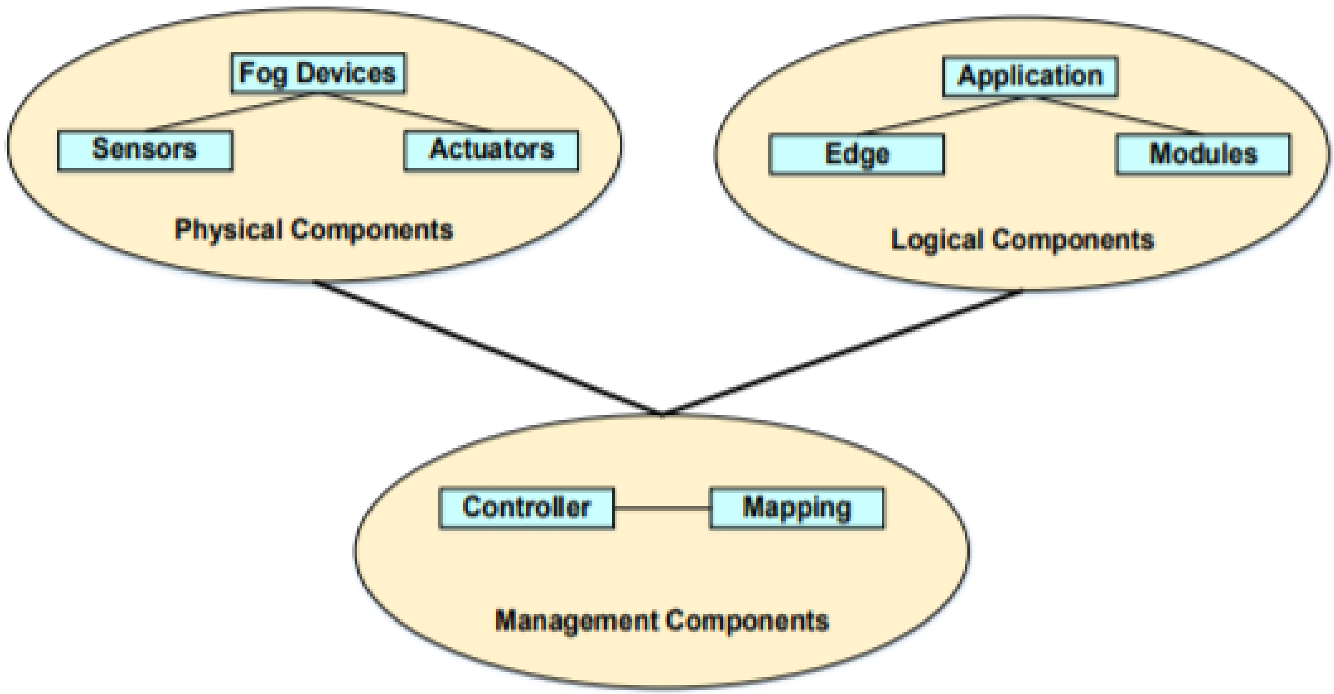
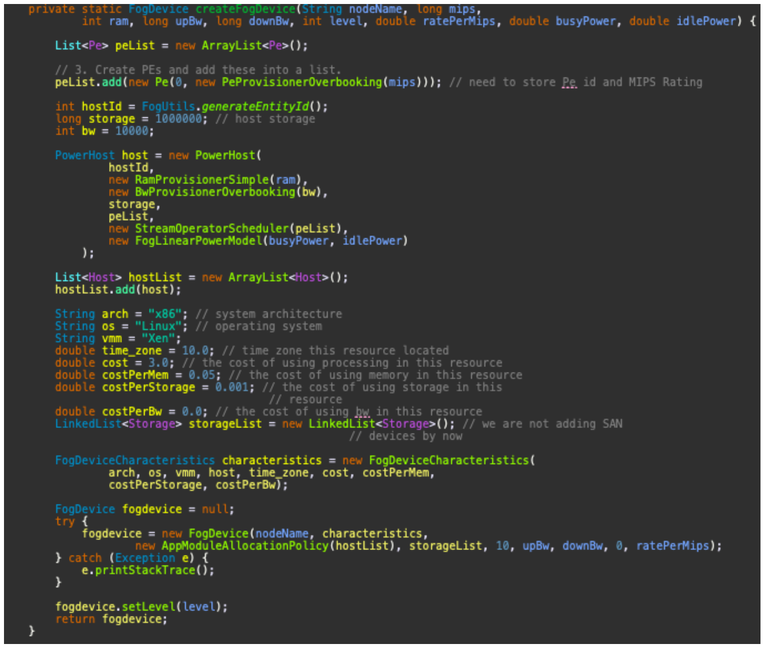


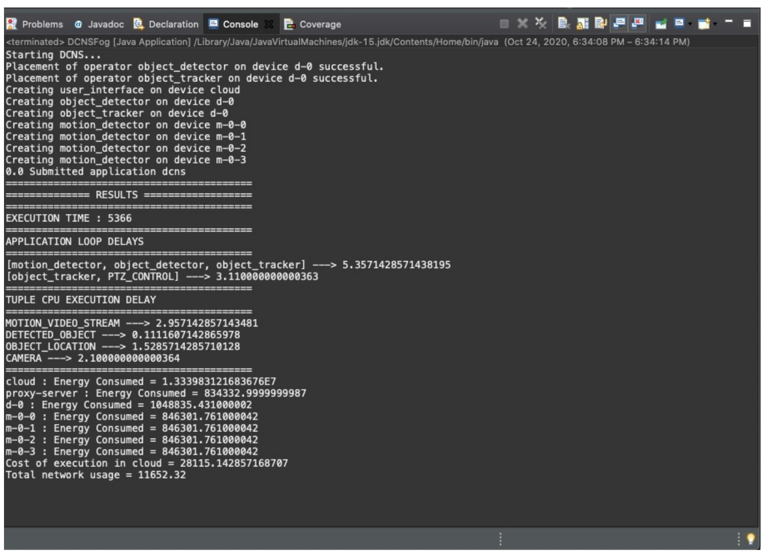
Figure 10: iFogSim simulation components

As Redowan Mahmud and Rajkumar Buyya [12] illustrated, fog devices are required to be classified. After arranging them, we can then move to the logical components which are classified into three classes: AppEdge, AppModule, and AppLoop. These components are needed for the structure of the application. By means, the application module class identifies different modules within the application. Moreover, the Application edge connects modules with each other if any dependencies are visible. Lastly, the management components include the Controller and the Mapping class, which manage both the physical and logical components. The mapping class associates each accessible resource to the fog devices, according to the AppModule demands. Finally, the Controller collects all the results (cost, energy consumption, etc.). We inspected the DCNS.fog class in Figure11 which contains a case study for intelligent surveillance and examined the results presented in Figure12 .



```java
private static FogDevice createFogDevice(String nodeName, long mips,
        int ram, long upBw, long downBw, int level, double ratePerMips, double busyPower, double idlePower) {

    List<Pe> peList = new ArrayList<Pe>();

    // 3. Create PEs and add these into a list.
    peList.add(new Pe(0, new PeProvisionerOverbooking(mips))); // need to store Pe id and MIPS Rating

    int hostId = FogUtils.generateEntityId();
    long storage = 1000000; // host storage
    int bw = 10000;

    PowerHost host = new PowerHost(
            hostId,
            new RamProvisionerSimple(ram),
            new BwProvisionerOverbooking(bw),
            storage,
            peList,
            new StreamOperatorScheduler(peList),
            new FogLinearPowerModel(busyPower, idlePower)
        );

    List<Host> hostList = new ArrayList<Host>();
    hostList.add(host);

    String arch = "x86"; // system architecture
    String os = "Linux"; // operating system
    String vmm = "Xen";
    double time_zone = 10.0; // time zone this resource located
    double cost = 3.0; // the cost of using processing in this resource
    double costPerMem = 0.05; // the cost of using memory in this resource
    double costPerStorage = 0.001; // the cost of using storage in this
                                    // resource
    double costPerBw = 0.0; // the cost of using bw in this resource
    LinkedList<Storage> storageList = new LinkedList<Storage>(); // we are not adding SAN
                                                    // devices by now

    FogDeviceCharacteristics characteristics = new FogDeviceCharacteristics(
            arch, os, vmm, host, time_zone, cost, costPerMem,
            costPerStorage, costPerBw);

    FogDevice fogdevice = null;
    try {
        fogdevice = new FogDevice(nodeName, characteristics,
                new AppModuleAllocationPolicy(hostList), storageList, 10, upBw, downBw, 0, ratePerMips);
    } catch (Exception e) {
        e.printStackTrace();
    }

    fogdevice.setLevel(level);
    return fogdevice;
}
```

Figure 11: DCNS code



```
<terminated> DCNSFog [Java Application] /Library/Java/JavaVirtualMachines/jdk-15.jdk/Contents/Home/bin/java (Oct 24, 2020, 6:34:08 PM – 6:34:14 PM)
Starting DCNS...
Placement of operator object_detector on device d-0 successful.
Placement of operator object_tracker on device d-0 successful.
Creating user_interface on device cloud
Creating object_detector on device d-0
Creating object_tracker on device d-0
Creating motion_detector on device m-0-0
Creating motion_detector on device m-0-1
Creating motion_detector on device m-0-2
Creating motion_detector on device m-0-3
0.0 Submitted application dcns
=========================================
============== RESULTS ==================
=========================================
EXECUTION TIME : 5366
=========================================
APPLICATION LOOP DELAYS
=========================================
[motion_detector, object_detector, object_tracker] ---> 5.3571428571438195
[object_tracker, PTZ_CONTROL] ---> 3.110000000000363
=========================================
TUPLE CPU EXECUTION DELAY
=========================================
MOTION_VIDEO_STREAM ---> 2.957142857143481
DETECTED_OBJECT ---> 0.1111607142865978
OBJECT_LOCATION ---> 1.5285714285710128
CAMERA ---> 2.100000000000364
=========================================
cloud : Energy Consumed = 1.333983121683676E7
proxy-server : Energy Consumed = 834332.9999999987
d-0 : Energy Consumed = 1048835.431000002
m-0-0 : Energy Consumed = 846301.761000042
m-0-1 : Energy Consumed = 846301.761000042
m-0-2 : Energy Consumed = 846301.761000042
m-0-3 : Energy Consumed = 846301.761000042
Cost of execution in cloud = 28115.142857168707
Total network usage = 11652.32
```

Figure 12: case study: intelligent surveillance output

In addition, we created a topology using the fog.GUI package, as shown in Figure13, as an initial step to developing an architectural design for our system, analyzing it, and starting up the implementation for our system.

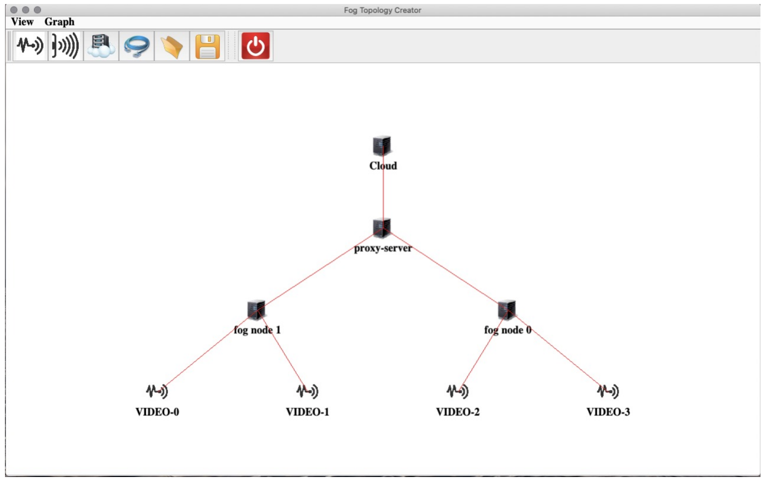


Figure 13: System topology

6 Project Management and Deliverables

6.1 Deliverables

In order to reach our goal and obtain satisfactory results, we need to successfully achieve and deliver the fundamental elements of our system:

- System's architectural model
- Algorithms for anomaly detection
- Framework for video summarization
- Privacy protection scheme

First, the fog computing- based surveillance system architecture should be implemented using a simulation and modeling tool. Then, anomaly detection algorithms should be efficiently implemented to offer real-time data processing. Moreover, a framework should be developed for video summarization, in order to overcome the scalability challenge. In addition, privacy protection scheme will be presented for data security and to overcome privacy concerns in smart cities.

Deliverables	Completion date
System Architecture	December 2020
anomaly detection+ video summarization	February 2021
Privacy protection scheme	April 2021

7 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). we are waiting for their response.The paper is available on github.

From: Islam AbdElHalim <islam.tharwat@miuegypt.edu.eg>
Sent: Wednesday, September 23, 2020 10:46 AM
To: Mohamed Azzam (ElsewedyTechnology) <Mohamed.Azzam@ElsewedyTechnology.com>
Subject: The Integration between Scientific and Technical Contributions in the Intelligent Transportation Systems (ITS) in Egypt

Dear Eng. Mohamed,

I hope this mail finds you well.

First of all, I'd like to express my sincere gratitude for our last discussion.

As you will recall, on 1st Apr 2019, I've completed my Ph.D. thesis at Faculty of Engineering, Ain Shams University. The thesis was awarded the BEST Ph.D. thesis in Computer and Systems Engineering. As well, I've published 3 international research articles (attached) in IEEE and Elsevier Journals, in which one of them was published in the IEEE Intelligent Transportation Systems Magazine (IEEE ITS Magazine).

Besides, I'm currently working as an Assistant Professor at Faculty of Computer Science, Misr International University (MIU). Since I'm supervising, this year, a graduation project that aims to utilize, deploy, and market several outcomes of my thesis in real applications related to the ITS project in Egypt. We are able to contribute in several ITS related applications including Traffic Monitoring and Management, Road Safety, Enhanced Route Guidance and Navigation, Surveillance Systems, Highway Platooning, Electronic Toll Collection, Vehicular Communication, Infotainment, etc.

So,in light of the great efforts and contributions of the ELSEWEDY company in the ITS project in Egypt; I would be grateful if you could let me know whether there is any interest from your side to let us cooperate and engage with the company in the ITS project.

Kind regards,

Islam THarwat, Ph.D.
Assistant Professor
Faculty of Computer science
Misr International University
Mob:01001146444
E-mail: islam.tharwat@miuegypt.edu.eg

Figure 14: Contacting ELSEWEDY technology company

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.

We will contact you to arrange for the date and time of the meeting.

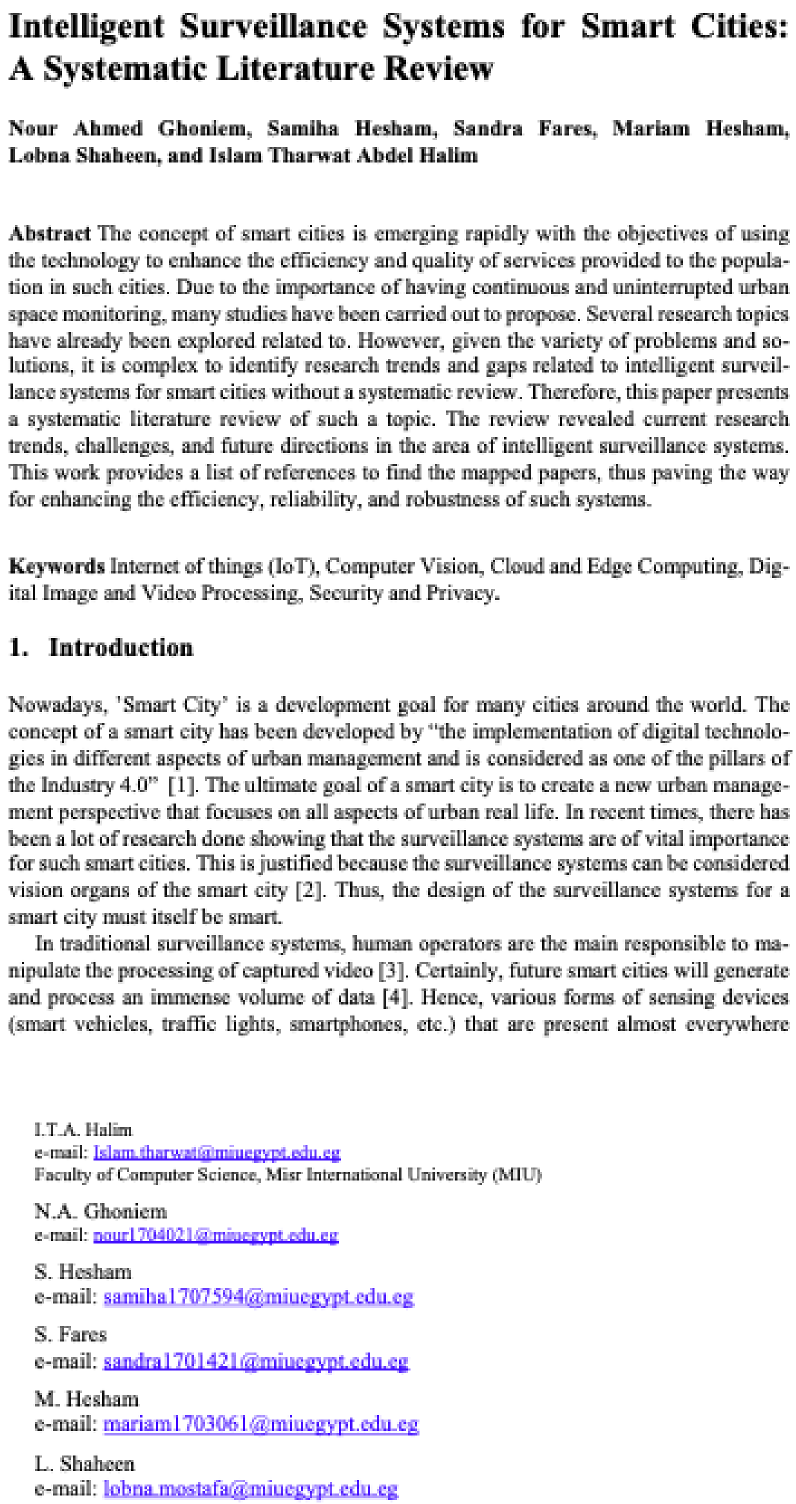
Best Regards,

Mohamed Azzam
Presales Manager

CCIE # 17115

Plot no. 27, 1st District, 5th Settlement, New Cairo, Cairo–Egypt
Mobile: +201220227791
Web: http://www.elsewedytechnology.com/
Email: Mohamed.Azzam@ElsewedyTechnology.com

Figure 15: 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. Several research topics have already been explored related to. 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

I.T.A. Halim
e-mail: Islam.tharwat@miuegypt.edu.eg
Faculty of Computer Science, Misr International University (MIU)

N.A. Ghoniem
e-mail: nour1704021@miuegypt.edu.eg

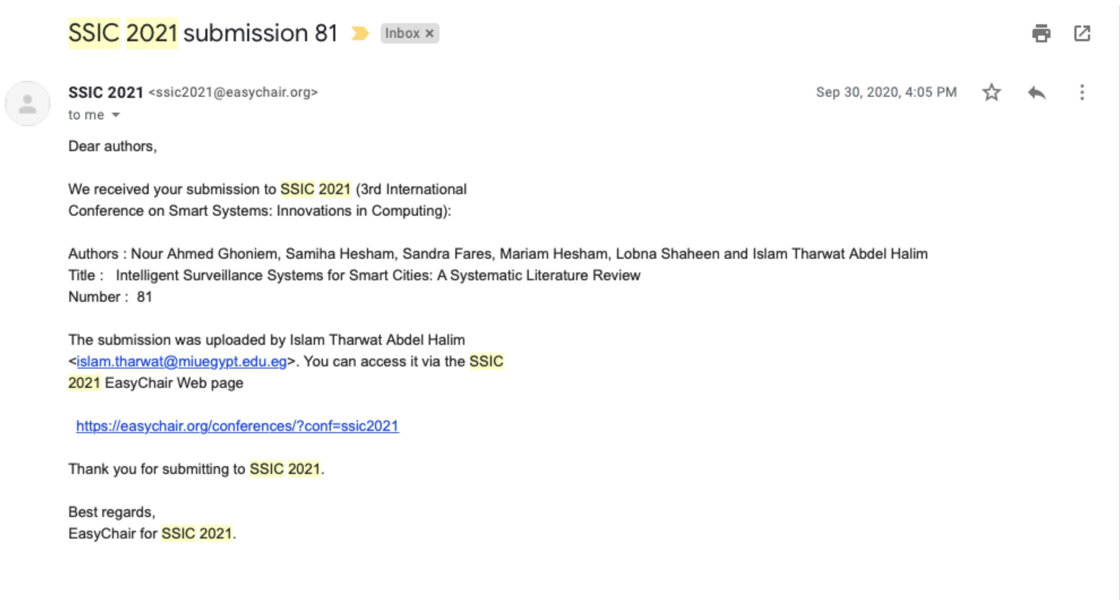
S. Hesham
e-mail: samiha1707594@miuegypt.edu.eg

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

Figure 16: Review Paper



SSIC 2021 submission 81 Inbox ×

SSIC 2021 <ssic2021@easychair.org> Sep 30, 2020, 4:05 PM
to me

Dear authors,

We received your submission to SSIC 2021 (3rd International Conference on Smart Systems: Innovations in Computing):

Authors : Nour Ahmed Ghoniem, Samiha Hesham, Sandra Fares, Mariam Hesham, Lobna Shaheen and Islam Tharwat Abdel Halim
Title : Intelligent Surveillance Systems for Smart Cities: A Systematic Literature Review
Number : 81

The submission was uploaded by Islam Tharwat Abdel Halim <islam.tharwat@miuegypt.edu.eg>. You can access it via the SSIC 2021 EasyChair Web page

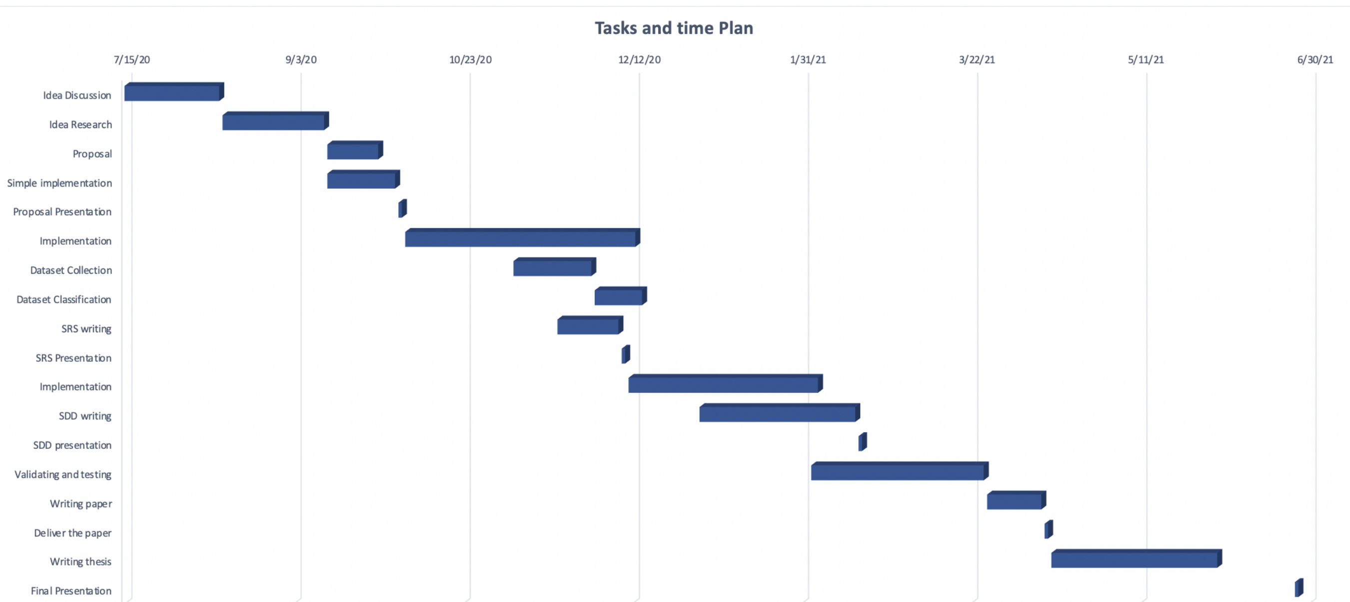
https://easychair.org/conferences/?conf=ssic2021

Thank you for submitting to SSIC 2021.

Best regards,
EasyChair for SSIC 2021.

Figure 17: Submission email from SSIC 2021

7.1 Tasks and Time Plan



	Final Presentation	Writing thesis	Deliver the paper	Writing paper	Validating and testing	SDD presentation	SDD writing	Implementation	SRS Presentation	SRS writing	Dataset Classification	Dataset Collection	Implementation	Proposal Presentation	Simple implementation	Proposal	Idea Research	Idea Discussion
START DATE	6/26/21	4/15/21	4/13/21	3/27/21	2/3/21	2/17/21	1/1/21	12/11/20	12/9/20	11/20/20	12/1/20	11/7/20	10/6/20	10/4/20	9/13/20	9/13/20	8/13/20	7/15/20
TASK DURATION	1	49	1	16	51	1	46	56	1	18	14	23	68	1	20	15	30	28

7.2 Budget and Resource Costs

we may publish articles and research papers in the future. Moreover, we may use additional computing tools.

8 References

References

[1] Alexey Medvedev et al. "Waste Management as an IoT-Enabled Service in Smart Cities". In: *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Ed. by Sergey Balandin, Sergey Andreev, and Yevgeni Koucheryavy. Cham: Springer International Publishing, 2015, pp. 104–115.

[2] Ning Chen et al. "Dynamic urban surveillance video stream processing using fog computing". In: *2016 IEEE second international conference on multimedia big data (BigMM)*. IEEE. 2016, pp. 105–112.

[3] G. Baldoni 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)*. 2017, pp. 611–612. DOI: 10.1109/CCNC.2017.7983190.

[4] Gabriele Baldoni 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.

[5] Ning Chen et al. "Smart city surveillance in fog computing". In: *Advances in mobile cloud computing and big data in the 5G era*. Springer, 2017, pp. 203–226.

[6] 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.

[7] Nhat-Quang Dao et al. "Management of Video Surveillance for Smart Cities". In: *Handbook of Smart Cities: Software Services and Cyber Infrastructure*. Ed. by Muthucumaru Maheswaran and Elarbi Badidi. Cham: Springer International Publishing, 2018, pp. 285–310. ISBN: 978-3-319-97271-8. DOI: 10.1007/978-3-319-97271-8_11. URL: https://doi.org/10.1007/978-3-319-97271-8_11.

[8] S. Durga, S. Surya, and E. Daniel. "SmartMobiCam: Towards a New Paradigm for Leveraging Smartphone Cameras and IaaS Cloud for Smart City Video Surveillance". In: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. 2018, pp. 1035–1038. DOI: 10.1109/ICOEI.2018.8553974.

[9] Ioannis Ledakis 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)*. IEEE. 2018, pp. 1–5.

[10] Jianguo Chen et al. "Distributed deep learning model for intelligent video surveillance systems with edge computing". In: *IEEE Transactions on Industrial Informatics* (2019).

[11] Yutong Liu 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.

[12] Redowan Mahmud and Rajkumar Buyya. "Modelling and simulation of fog and edge computing environments using iFogSim toolkit". In: *Fog and edge computing: Principles and paradigms* (2019), pp. 1–35.

[13] Mansoor Nasir 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.

[14] A. F. Santamaria et al. "Cooperative Video-Surveillance Framework in Internet of Things (IoT) Domain". In: *The Internet of Things for Smart Urban Ecosystems*. Ed. by Franco Cicirelli et al. Cham: Springer International Publishing, 2019, pp. 305–331. ISBN: 978-3-319-96550-5. DOI: 10.1007/978-3-319-96550-5_13. URL: https://doi.org/10.1007/978-3-319-96550-5_13.

[15] G. Sreenu and M. A. Saleem Durai. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis". In: *Journal of Big Data* 6.1 (June 2019), p. 48. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0212-5. URL: https://doi.org/10.1186/s40537-019-0212-5.

[16] 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.

[17] H. Sun 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 (2020), pp. 800–817. DOI: 10.1109/JIOT.2019.2936504.

[18] X. Xu 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.

[19] BV Natesha and Ram Mohana Reddy Guddeti. "Fog-Based Video Surveillance System for Smart City Applications". In: *Evolution in Computational Intelligence*. Springer, pp. 747–754.