ACTIVE VISION USING SELF-RECONFIGURABLE SMART CAMERA NETWORK

A thesis submitted in partial fulfilment of the requirements
for the award of the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

SHASHANK

(2K17/PhD/EC/05)

under the supervision of

Professor S. Indu

Department of Electronics & Communication Engineering

Delhi Technological University

Delhi-110042

November 2023

DELHI TECHNOLOGICAL UNIVERISTY

CERTIFICATE

This is to certify that the thesis entitled "**Active Vision Using Self-Reconfigurable Smart Camera Network**" being submitted by Shashank (Reg. No.: 2K17/PhD/EC/05) for the award of degree of Doctor of Philosophy to the Delhi Technological University is based on the original research work carried out by him. He has worked under our supervision and has fulfilled the requirements, which to our knowledge have reached the requisite standard for the submission of this thesis.

It is further certified that the work embodied in this thesis has neither partially nor fully submitted to any other university or institution for the award of any degree or diploma.

Prof. S. Indu
Supervisor
Professor
Department of ECE
Delhi Technological University

Prof. O.P. Verma
Head of the Department
Department of Electronics and Communication Engineering
Delhi Technological University

Declaration of Authorship

I hereby declare that all information in the thesis entitled "**Active Vision Using Self-Reconfigurable Smart Camera Network**" has been obtained and presented in accordance with the academic rules and ethical conducts as laid out by Delhi Technological University. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

(**Shashank**)
Research Scholar

Acknowledgements

First and foremost, thanks to the Almighty for giving me strength and inspiration to carry out this research work. I owe a deep sense of gratitude to all his comprehensive soul whose divine light has enlightened my path throughout the journey of my research.

I would like to express my sincere and heartfelt thanks to my research supervisor **Prof. S. Indu** for her valuable guidance, enthusiastic encouragement and persistent support. I am truly grateful from the core of my heart for her meticulous approach, wonderful assistance of her perspective and fruitful discussions on the research topic. Her careful supervision and personal attention have given me a lot of confidence and enthusiasm, during the different stages of my doctoral investigations. I invariably fall short of words to express my sincere gratitude for her patience and motivation.

I express my sincere gratitude to the **Prof. O. P. Verma, Head of Electronics and Communication Engineering Department** for his endless support and cooperation. I am also thankful to **Prof. Neeta Pandey, DRC Chairman,** Electronics and Communication Engineering Department for constantly encouraging and supporting me in this endeavour of mine.

My appreciation also goes to all my teachers and colleagues in the Dept. of Electronics and Communication Engineering, Delhi Technological University, Delhi, for their constant encouragement and motivation.

I dedicate this thesis to my wife **Malvika** for her endless love, support, and encouragement throughout my academics. I in a special way, am grateful to my wife for her continuous support, encouragement, and equal efforts throughout the research period of mine, thus so far. My father, **Shri Param Veer** not only raised and nurtured me but also taxed himself dearly over the years for my education and intellectual development. My mother, **Smt. Raj Bala** has been a source of motivation and strength during moments of despair and discouragement. She has blessed me throughout with her continual prayers and blessings.

Shashank.

This thesis is dedicated to my wife "Malvika"

for her endless love, support and encouragement

Abstract

Efficiency of a computer vision system depends on accuracy of information extraction and data processing capabilities of the computer vision system. A camera frame capturing an object of interest in the center of a field of view carries the maximum information of the object. Thus, to enhance the accuracy of the information, the system requires continuous reconfiguration of configuration spaces of one or more camera sensors deployed in the system. However, identification of the object requires processing images, and the position of an object of interest in the frame can change dynamically. Thus, reconfiguration of configuration space is difficult for real time applications. A computer vision system capable of re-configuring its configuration space (*i.e.,* through dynamic calibration of camera sensors) is known as an active vision system.

For a better understanding of an event (or scene), the active vision system further requires association of various activities detected by the camera sensors temporally, which requires high processing capabilities to perform accurate spatiotemporal analysis of various image frames captured temporally by different camera sensors. Such systems rely on high computational complexity models and require enormous resources (such as Artificial Intelligence based systems). Systems deployed in mobile environment have limited resources (*i.e.,* limited power supply, storage and processing capabilities), and thus are incapable of performing tasks with higher computational complexity, and thus lack efficient reconfiguration of camera sensor parameters (*i.e.,* the configuration space) which leads to images (or frames) being captured with very low information of the objects of interest, yielding low performance and accuracy of the system. To address the aforementioned problem, this thesis presents a computer implemented framework (*i.e.,* Spatiotemporal Activity Mapping (SAM) framework) that enables pixel-wise sensitivity allotment based on spatiotemporal activity analysis of frames captured over a flexible time period. The SAM framework presents various filters efficiently

designed with very low computational complexity for accurate detection of areas of interest, for re-configuration of calibration parameters. The SAM framework presents a flexibility of selection of the criticality of activities detected by the system, and thus is effective in a variety of computer vision applications such as road surveillance, sports analysis, ambient living applications, and the like.

Model-based systems only work in known conditions and fail miserably in unforeseen conditions. Systems employing Artificial Intelligence (AI) can manage to tackle unforeseen environments, however, such systems require iterative training to learn and train to develop understanding of new events and activities over a long period of time, and thus are not reliable for real-time applications. Thus, the contemporary systems lack real-time reconfiguration of configuration space for an adequate scene understanding of a new activity or event. To address the aforementioned problem, this thesis presents another computer implemented framework (*i.e.*, Adaptive Self-Reconfiguration (AdapSR)) framework that enables a number of computer vision systems to exchange information and data sharing, and thus learn to tackle an unforeseen condition at a very high rate. The AdapSR framework is fairly efficient in performance for applications employing higher computational and storage capabilities for high levels of accuracy and fast learning such as driverless navigation, adaptive activity analysis, and the like. The AdapSR framework further provides a concept of decentralized network of active vision systems that enables establishment of standardization of protocols for a plurality of computer vision applications associated together over a blockchain network in near future.

Thus, by developing these novel techniques and framework models, all major issues regarding self-reconfiguration of computer vision systems have been addressed. This thesis incorporates the developed techniques and their performance evaluation along with future directions.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Sight is considered as the most powerful sense of a human being. The reason behind sight being the most powerful sense is the content of visual information in each vision (or visual frame) captured through the sight. The visual information is processed by the human brain to derive an understanding of the captured scene. A computer vision system is an artificial system that tries to mimic the functionality of a human vision system. According to Reisslein et al. in [1], computer vision systems are distributed systems configured to extract visual information from data sensed by a plurality of cameras via co-operative sensing in real-time.

With advancement of technology and processing capabilities of artificial systems, computer vision has found new domains of applications in the last two decades. Today, computer vision systems find applications in the areas of surveillance for security and detection of trespassing [2], driverless vehicles [3], sports analysis [4], healthcare robotics, aviation, ambient living [5], tele-immersion, disaster management, situation cognizance [6] and many more. A survey in [2] reports an estimated 7.6% annual growth of computer vision applications (in terms of Compound Annual Growth Rate (CAGR)) from 2020 to 2027 based on the progressive growth of computer vision applications observed in recent years.

Computer vision systems [7] commonly utilize one or more camera sensors coupled to processing circuitry (such as a processing unit or a data processor) having a combination of software-hardware components capable of image processing to derive a desired functionality. The camera sensors of the computer vision system are configured to capture one or more image frames (bearing information of the scene) of a desired environment (*e.g.*, scene) and provide the image frames (hereinafter interchangeably referred to as “frames”) to the processing circuitry for further processing. The processing circuitry of the computer vision system is configured to perform one or more data processing tasks on each frame to derive an understanding of one or more activities in the environment of the captured scene. More specifically, the processing circuitry receives sensed data (or image frames) from the camera sensors, combines the sensed data, and performs data processing to obtain visual understanding of the scene. The computer vision system is generally coupled to a control circuitry (*e.g.*, an actuation unit) that receives the visual understanding of the scene from the processing circuitry and performs one or more actions based on the functionality of the system. The aforementioned process flow of a typical computer vision system is shown in Figure 1.1 hereinbelow.

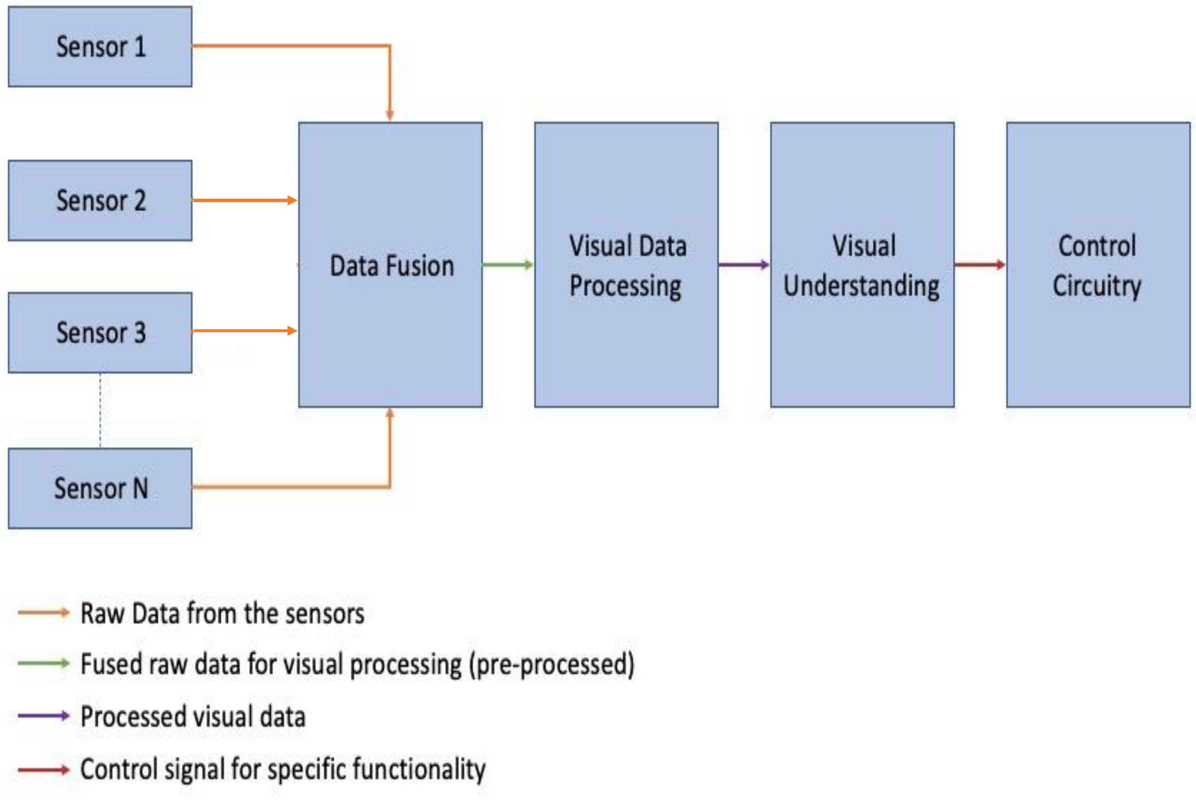


Figure 1.1 Process flow of a Computer Vision application

Performance of the computer vision system depends on accuracy of data extraction (in terms of the information content of objects of interest in the frame) and the computer vision system's data processing capabilities. High accuracy of the captured data demands continuous reconfiguration (*e.g.,* calibration) of the configuration space of the computer vision system's camera sensors in response to the activity detected by the computer vision system. As a person skilled in the art will appreciate, a frame with an object of interest captured in the centre of the field of view (FoV) of the camera sensor bears the maximum information about the object of interest (as the camera sensor captures the object in the centre of its field of view with highest resolution). Thus, a timely re-configuration of the configuration space is required to maximise the information contained in each frame and enhance the efficiency of the computer vision system. To achieve the aforementioned, the computer vision system must be competent enough to derive a configuration state (through configuration space of the camera sensors) based on the scene understanding derived by processing the previous frames captured by the sensors and calibrate its configuration state accordingly for the upcoming frames (*i.e.,* self-reconfiguration).

To address the aforementioned issues concerning, we have conducted our research and developed several techniques for adaptive self-reconfiguration of smart camera networks employed in computer vision applications with low latency.

1.1 Active vision

A human vision system sends impulse responses to the vision system by processing the visual data, and thus in response, the human vision system changes the field of view for the upcoming events accordingly. Similarly, for a computer vision system, it is critical to understand what to look at (*i.e.,* identify the objects of interest) and where to look for (*i.e.,* determine regions of presence of objects of interest in field of views of the camera sensors). Each camera sensor has a configuration space that includes internal calibration parameters (such as aspect ratio, aperture, focal length etc.) and external calibration parameters (such as orientation, rotation, translation, Pan, Tilt, Zoom etc.). To capture scenes with maximum information of the objects of interest (*i.e.*, at highest resolution), the camera's configuration space (*i.e.*, the internal and external calibration parameters) must be adjusted in accordance with the information of the objects of interest derived by processing the frame.

Active vision systems [8] (also known as 'active computer vision systems') are computer vision systems capable of calibrating (or re-configuring) the internal and external parameters (*i.e.*, the configuration space) of their camera sensors to alter viewpoints of the cameras according to the functionality of the system. Operations of active vision systems can be classified into two fundamental levels: a deployment level (*i.e.*, for data extraction and calibration of the camera sensors) and a processing level (*i.e.*, where the visual processing takes place to derive an understanding). Thus, the challenges associated with designing an ideal active vision system is also categorised into two types (*i.e.*, the challenges associated with the deployment of the active vision system [9] and the challenges associated with data processing [2]). The first type of challenges [9] is associated with sensor node reconfiguration in order to obtain sensor data containing optimal information in relation to each camera's resource limitations. The second type of challenges [2] is associated with the data-processing level of operation to maximize the scene understanding of while keeping computational complexity as low as possible. As a result, an effective active vision system necessitates deployment efforts at both the hardware and software levels. Therefore, to design an efficient and adaptive self-reconfigurable active vision system, there is a requirement to understand the taxonomy of operational levels of active vision system, the challenges at each level, and effects of enhancement at one level of deployment to other levels. Figure 1.2 depicts a multi-tier taxonomy of challenges for active vision systems employing camera networks based on the aforementioned operational classification.

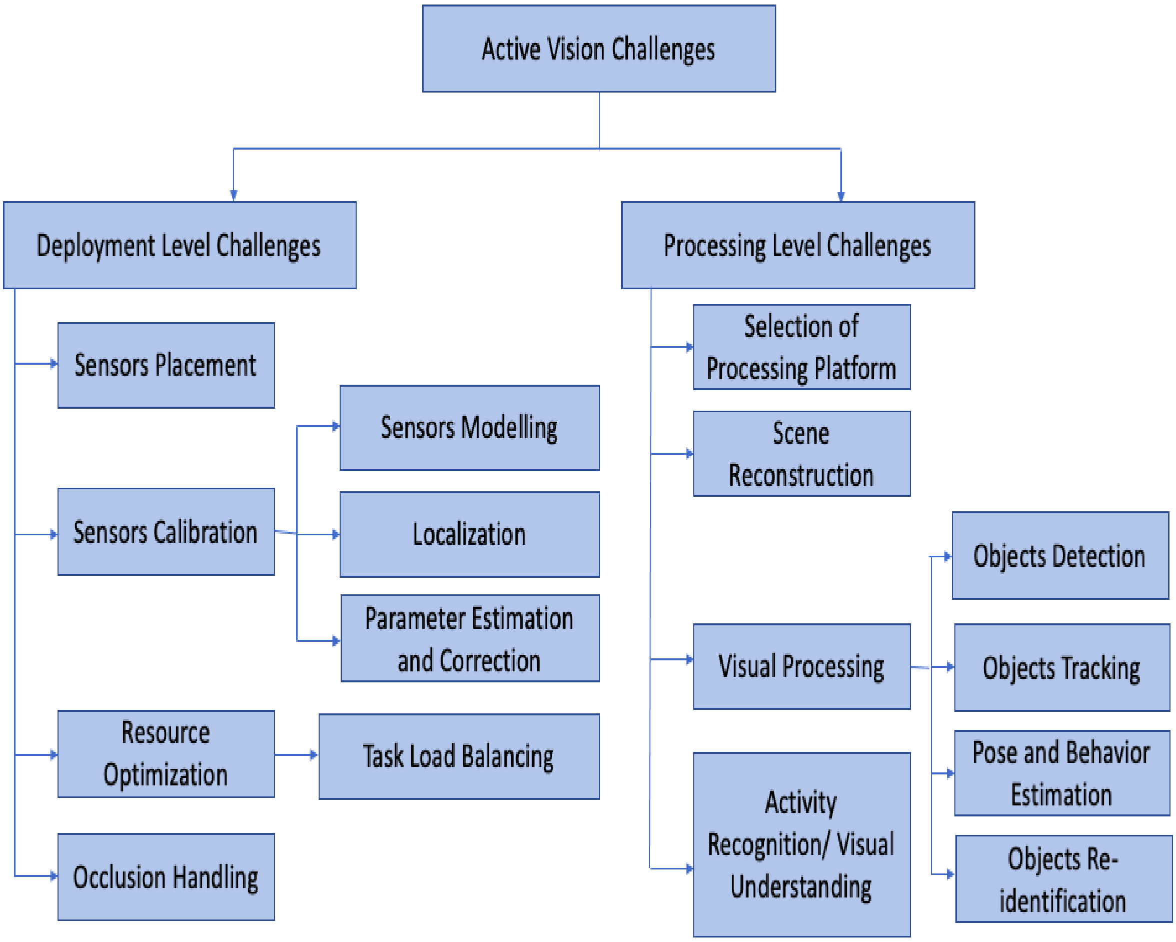


Figure 1.2 Classification of challenges in Active Vision Systems

As shown in Figure 1.2 hereinabove, enhancement of the efficiency of active vision systems demands attention at the deployment level as well as the processing level. Further, the inter-dependence of the calibration of the camera sensor's configuration space and the processing performance of the active vision system makes it more difficult to develop an idol adaptive and self-reconfigurable active vision system. Thus, an objective of this research is to derive an accurate relationship between the calibration of configuration spaces of the camera sensors and the spatiotemporal activity analysis of the scene without high computational complexity.

Another objective of this research is to design an efficient framework for adaptive self-reconfiguration of active vision systems with capabilities of information and data sharing to enhance the learning rate of the systems to tackle an unforeseen condition in near real-time.

Yet another objective of this research is to enable standardization of operational protocols (through a framework) for a plurality of active vision systems associated together over a decentralized network to enhance security of data and information sharing.

Yet another objective of this research is to design a simple autoencoder model to sufficiently compress the size of data to be shared between the plurality of active vision systems in the decentralized network without a high burden of computational resources on the active vision system.

1.2 Research Motivation

An active vision system employing a multi camera network results in an increase in the availability of frames captured by the plurality of camera sensors, thus the possibility of obtaining better information from the captured frames is increased. However, processing enormous data from the plurality of frames also increases the computational load of the active vision system. Further, a higher degree of freedom (*e.g.*, flexibility or movability) of the camera sensor enables capturing frame with enhanced accuracy in terms of the information of the objects of interest in the frame. However, a higher degree of freedom of the camera sensor increases the calibration parameters of the camera sensor and thus increases the computational complexity and load of the system. An active vision system employing a single fixed camera has much lower computational load and thus faces fewer deployment challenges than an active vision system employing a network of mobile Pan-Tilt-Zoom (PTZ) cameras. However, an active vision system with multiple camera sensors with flexible movement can have occlusion-avoidance capabilities that the active vision system with a single camera for sensing may not have.

As for near real-time applications, re-configuration of calibration parameters based on the understanding of scene and the objects of interest is very difficult (especially for systems with limited resources), contemporary active vision systems rely on a user-defined prioritized area of interest. Such systems enable a user to select the prioritized area in the field of view of each camera, either prior to the deployment of the system or while operation of the system. However, such systems require a constant need of a personnel (*i.e.*, the user) to monitor the environment under observation.

As computer vision systems [7] aim to leverage human efforts by developing an understanding of events through the processing of data obtained from a number of sensors, the aforementioned scenarios compromise on the basic fundamental of computer vision system. An ideal active vision system must be capable of deriving information of the scene and calibrate the configuration spaces of the camera sensors based on the derived information. High computational complexity of the active vision system for image processing being the major issue here must be addressed appropriately. Further, to match the human intellect, the active vision system must be adaptive and capable of associating the temporally derived information for re-reconfiguration of camera sensors.

Model-based computer vision systems only work in known environments and fail miserably in unknown environments. Systems that use Artificial Intelligence (AI) can deal with unexpected situations, however such systems require iterative training to learn and train over time to develop understanding of new events and activities, making them unsuitable for real-time applications. For an adequate scene understanding of a new activity or event in real time, modern computer vision systems lack real-time reconfiguration of configuration space. As a result, techniques for near real-time adaptive reconfiguration of sensors for capturing data with optimal information are required to derive optimised information about the objects of interest and enhancement of the system’s quality of service (QoS). In addition, adequate research on optimal resource utilisation for computer vision systems with limited resources is lacking.

1.3 Research Problem

The challenges of designing active vision systems are vastly divided into two broad categories (as shown in Figure 1.2 as deployment level challenges and processing level challenges) that are interdependent. However, the base problem is the computational complexity and load on the processing circuitry for reconfiguration of calibration parameters based on the scene understanding. Each computer vision system is specifically designed to address specific sets of challenges based on its functionality, and thus requires different specifications and capabilities. For example, a surveillance system used in military defense may require a mobile carrier for the computer vision system (such as a drone) with limitations on computational, battery and storage resources, whereas a surveillance system employing computer vision for ambient living installed in a facility may be powered by constant power supply and may not require high computational and/or storage resources based on the limited functionality of the system.

Thus, the broader problem of self-reconfiguration of the camera sensors can be categorized into two classes: (i) Self-reconfiguration of active vision systems with limited resources, and (ii) Adaptive self-reconfiguration of active vision systems with sufficient resources for reduced latency that can be used for near real-time applications.

Self-reconfiguration of active vision systems with limited resources demands reconfiguration framework with low computational complexity and accurate activity detection. Further, there is a need to derive a spatiotemporal relationship between frames captured in past (*i.e.*, past activities) and the present frames (real-time activities) to achieve enhanced understanding of the scene without exploiting the resources of the active vision system.

Adaptive self-reconfiguration of active vision systems with sufficient resources for reduced latency can be used for near real-time applications. Such systems do not deprive of resources, rather demand high learning capabilities to tackle unforeseen conditions efficiently. Adaptive self-reconfiguration of active vision systems requires data and information sharing in a decentralized environment for adaptive learning. Further, such systems demand an efficient compression tool (such as an autoencoder) to compress data while sharing and storing data that can be retrieved without losing essential and critical information.

1.4 Objectives of Research Work

The objectives of this research work are to develop techniques and methods to address the key problems for development of adaptive self-reconfiguration of smart camera networks deployed in active vision systems. These specific objectives are summarized as follows:

Objective 1:

- To review of the existing literature and compare different methodologies proposed to address challenges at various operational levels of active vision system.
- To classify the challenges based on the type, requirements, and functionality of the application employing active vision system, and determine the broad problem areas for each type of application of the active vision systems.

Objective 2:

- To associate past and present activities/events detected by the active vision system by a spatiotemporal relationship without extensively exploiting resources of the active vision system.
- To generate an accurate and efficient spatiotemporal activity map with pixel-wise importance value assigned to each pixel of the field of view of each camera that can be used for self-reconfiguration of the camera sensors of the active vision system.
- To design a reconfiguration framework for improved camera sensor calibration based on the spatiotemporal activity map without exploiting resources of the active vision system.

Objective 3:

- To incorporate self-adaptation in an active vision system.
- To design an adaptive self-reconfiguration framework for improved performance with low reconfiguration latency of the active vision system.
- To develop a distributed network of active vision systems capable of data and information sharing, that can be used for self-reconfiguration of camera sensor parameters.

Objective 4:

- To enable establishment of standardization of protocols for a plurality of active vision applications associated together over the decentralized network of active vision system.
- To design a simple autoencoder model that can sufficiently compress the size of data to be shared between the plurality of active vision systems in the decentralized network without a high burden of computational resources on the active vision system.

1.5 Thesis Contribution

In this thesis, we showcase that the key to enhance performance of an active vision system is to incorporate adaptive self-reconfiguration to the system. The active vision systems are classified based on the functionality and resources available into two categories. The objectives, problems, and type of challenges for reconfiguration of calibration parameters depend on the functionality and application of the active vision system.

The principal contributions of this thesis are:

- A Self-reconfiguration Activity Mapping (SAM) framework is presented for generation of spatiotemporal activity maps with pixel-wise importance value assigned to each pixel that can be used for self-reconfiguration of the camera sensors of the active vision system and showcases improved camera sensor calibration based on the spatiotemporal activity map with very low computational load and complexity.

- An Adaptive Self-Reconfiguration (AdapSR) framework is presented for improved performance and low reconfiguration latency in handling unforeseen conditions. The AdapSR framework enables a distributed network of a number of active vision systems to share data, information, instructions and model to learn from each other's past experiences, and thus learn to tackle unforeseen conditions at a relatively higher rate. The AdapSR framework enables establishment of standardization of protocols for the number of active vision systems in the distributed network.
- Sharing and storing data, models, information and instructions in a distributed network of active vision systems require a large database for storage. Further, the size limitation of the database (or datacenters) is a never-ending challenge for distributed networks. To address the aforementioned, an autoencoder model with low computational load and complexity is presented utilizing basic cryptography principles of Gyrator transform to compress and enhance security of data prior to transfer/storage without losing any important or critical information of the data.

1.6 Thesis Overview

The thesis comprises of seven chapters, and a brief description of the seven chapters of this thesis is given hereinbelow:

Chapter 1 (Introduction): This chapter covers the motivation and purpose of adaptive self-reconfigurable active vision system. This chapter further contain thesis overview, research problem and the objectives of the research work.

Chapter 2 (Literature Review): This chapter provides a detailed study of active vision systems and associated challenges at various taxonomical levels. This chapter covers the state-of-the-art techniques developed in existing research work for development of adaptive smart camera networks. Further, chapter 2 highlights the research gaps in the existing work that has stimulated the development of research objectives. Furthermore, Chapter 2 will showcase the practicability of applications employing SCNs and will provide potential directions for further research in this area.

Chapter 3 (Spatiotemporal Activity Mapping): This chapter will provide a detailed discussion on Spatiotemporal Activity Mapping (SAM) framework for SCN-enabled active vision systems. The framework evaluates the scene spatiotemporally and produces adaptive activity maps for re-configuration of the sensor, such that the region(s) of importance can be captured in the centre of the sensor's field of view. The framework utilizes simple image processing tools such as adaptive background subtraction, binarization, thresholding and federated optical flow for pre-processing the sensor data. Half-width Gaussian distribution is used for temporal relationship between present and past frames. The simple model of the proposed framework results in low computation complexity, and thus low resource utilization.

Chapter 4 (Adaptive Self Reconfiguration): This chapter provides a detailed discussion on Adaptive Self-Reconfiguration (AdapSR) framework for SCN-enabled active vision systems. The framework enables active vision systems to share their derived learning about an activity or an unforeseen environment, which can be utilized by other active vision systems in the network, thus lowering the time needed for learning and adaptation to new conditions. Further, as the learning duration is reduced, the duration of the reconfiguration of the cameras is also reduced, yielding better performance in terms of understanding of a scene. The AdapSR framework enables resource and data sharing in a distributed network of active vision systems and outperforms state-of-the-art active vision systems in terms of accuracy and latency, making it ideal for real-time applications.

Chapter 5 (Autoencoder for AdapSR): This chapter provides a detailed discussion on the Gyrator Transform based data compression and security enhancement. The proposed system enables sharing the compressed datasets and model parameters in the distributed environment, and thus can be utilized as an alternative to highly complex auto-encoder model for AdapSR based systems.

Chapter 6 (Dynamic Speed limit allocation): This chapter provides a detailed discussion on a used case of AdapSR framework (*i.e.*, dynamic speed allocation 'DSA' framework) for dynamic traffic speed limit allocation and effect of the speed limits on accident prediction. The DSA framework utilizes data corresponding to different areas in the form of a plurality of parameters such as traffic density, accident count, static speed limit etc., to predict a most suitable speed limit for each area.

Chapter 7 (Conclusion): This chapter provides a brief summary of all the ideas, observations, and contributions of the results obtained in each objective. Also, the future directions in each field will be sketched in this section.

CHAPTER 2

LITERATURE REVIEW

Applications employing computer vision systems have grown tremendously over the last few years due to advancement in technology and leading research in the area of computer vision. Specifically, the success of any computer vision system highly relies on the data captured by the camera sensors employed in the computer vision system and the processing capabilities of the computer vision system, which are inter-dependent. In this direction, the computer vision systems have taken a next leap as active vision systems are specifically designed to obtain data with higher information by periodic calibration of camera sensor's configuration space and thus performing better in terms of the overall efficiency of the system.

As discussed earlier in Chapter 1, the challenges in active vision systems employing a number of camera sensors (*i.e.*, a camera network) can be categorised into two categories: deployment-level challenges and processing-level challenges. Apparently, the challenges in deployment of the active vision system affect the processing performance of the system and vice-versa. A brief discussion on both categories of challenges is presented hereinbelow.

2.1 Deployment level challenges

Applications of computer vision systems vary in a large domain of areas, and so is the configuration and requirements of the system. Most often, low resource availability and high computational complexity are major set-backs of an active vision system that restrict the calibration capabilities and thus supresses the performance of the system. Additionally, resources optimization by each camera sensor employed in the active vision system is equally important. Further, dealing with unforeseen environments and conditions is another problem for real-time applications of active vision systems, where the systems fail majorly. Designing an accurate sensor architecture with a precise sensor placement to optimize the performance of the active vision system is a further challenge. Lack of visual understanding due to occlusions and false detection due to environmental variations result in inappropriate calibration of the sensors, and thus affects the overall performance of the system. A brief description of the deployment level challenges is as follows.

2.1.1 Sensor Placement

Cameras in SCNs are typically placed with overlapping FOVs to capture the entire operating environment. However, with limited resources, it can be difficult to place cameras with overlapping FOVs for larger operational areas. Camera placement has a direct impact on the amount and quality of data available for processing. For example, if an object is captured in the centre of a camera's field of view (FOV), the quality of the data and thus the visual information available from the data is much higher than if the object is captured at the camera's edges. Further, the camera placement must ensure maximum event coverage, which makes camera placement critical for SCN deployment. Thus, sensor (*e.g.*, camera) placement plays a crucial role in the performance and efficiency of an active vision system. Specifically, most of the research in the area of accurate sensor placement is confined to solving two major problems: (i) maximization of area covered by the camera sensors and (ii) managing non-overlapping camera field of views.

Indu et al. in [10] and Zhang et al. in [11] proposed methods for placement of camera sensors to maximize the surveillance areas covered by a network of camera sensors. Silva et al. in [12] proposed co-ordination between camera sensors employed on a network of unmanned aerial vehicles (UAVs) to improve the efficiency of aerial surveillance. Solutions proposed in [10] - [12] provide camera placement solutions for optimised functionality in a pre-defined configuration of environment, however, they lack architectural flexibility and surveillance space prioritisation. Jamshed et al. in [13] proposed an activity-based prioritization of area under surveillance. In [14], Vejdanparast proposed enhancement of fidelity of camera sensors in a smart camera network for maximization of the surveillance area. In [15], Wang et al. proposed Latin-Hypercube-based Resampling Particle Swarm Optimization (LH-RPSO) based camera placement algorithm for Internet of Things (IoT) devices networks.

Redding et al. in [16], using cross-matching for non-overlapping FOVs, proposed use of a variety of features, such as grey-level co-occurrence matrices, scale-invariant feature transformation, Zernike moments, and colour models, etc., for object handover to manage non-overlapping field of views. Esterele et al. in [17] proposed handover of information for handing non-overlapping field of views of a decentralized network of camera sensors by generating an online real-time vision graph. The information handover for handling non overlapping field of views by the online real-time vision graph in [17] was independent of any a-priory knowledge of the operating environment, and thus provided a flexibility of usage. In [18], Lin et al. proposed a method for active real-time FoV handover control for a single object by captured by a number of Pan-Tilt-Zoom (PTZ) camera sensors. The method proposed in [18] suggested a spatial relation between the PTZ cameras employing a shortest distance rule to determine readiness of each camera sensor prior to the handover. Table 2.1 presents an evolution of techniques and their advantages presented in the prior art for efficient sensor placement.

Table 2.1 Techniques addressing sensor placement problems.

Ref.	Year	Methodology	Advantages
[16]	2008	Online system for tracking multiple people in an SCN with overlapping and non-overlapping views	Development of a larger, more capable, and fully automatic system without prior localization information

[10]	2009	Genetic Algorithm	Maximum coverage of users; Defined priority areas with optimum values of parameters; The proposed algorithm works offline and does not require camera calibration; Minimizes the probability of occlusion due to randomly moving objects
[17]	2011	Ant-colony-inspired mechanism used to grow the vision graph during runtime	Generates a vision graph online; Increased autonomy, robustness, and flexibility in smart camera networks
[18]	2012	Approach to construct the automatic co-operative handover of multiple cameras for real-time tracking	Tracking a moving target quickly and keeping the target within the viewing scope at all times
[11]	2015	Novel model with non-uniformly distributed detection capability (DC)	Orientation of each visual sensor can be optimized through a least-squares problem; More efficient with an averaged relative error of about 3.4%
[13]	2015	Node-level optimal real-time priority-based dynamic scheduling algorithm	Portable system with ease of access in hard-to-access areas.
[12]	2017	Coordination of embedded agents using spatial coordination on strategical positioning and role exchange	Persistent surveillance with dynamic priorities
[14]	2020	Novel decomposition method with an intermediate point of representation	Low computational expense; Higher fidelity of the outcomes
[15]	2020	Latin-Hypercube-based Resampling Particle Swarm Optimization (LH-RPSO)	LH-RPSO has higher performance than the PSO and the RPSO; LH-RPSO is more stable and has a higher probability of obtaining the optimal solution

2.1.2 Camera Sensor Calibration

Each camera sensor of the smart camera network deployed in an active vision system has an individual configuration space that is dependent on the parameters (*i.e.,* the internal and external parameters of the camera sensor). A field of view (FoV) of each camera sensor is dependent on the configuration space of that camera sensor, which impacts the area under observation of the smart camera network. Specifically, calibration of the camera sensors enables the smart camera network to capture images accurately that impacts the performance of the active vision system using the smart camera network for image acquisition. Some systems are designed in such a way that the camera sensors are calibrated to efficiently utilize resources. Some other systems focus on calibration of camera sensors to capture the object(s) of interest at the best possible resolution. For example, when the object(s) of interest change their positions in the camera sensor's field of view, to capture the object(s) of interest with optimized resolution, the configuration space of the camera sensors is required to be adjusted based on an information of position(s) of the object(s) of interest in the camera's field of view.

At some other instances, some smart camera networks having limited resources and restricted functionality (*i.e.,* non-critical applications) rely on dynamic alteration of topology of the smart camera network by temporarily turning off camera sensors when no activity is detected by them. At the operational level, camera sensor calibration can be further divided into three subcategories that are discussed as follows.

- Camera Sensor Modelling: Each camera sensor deployed in active vision application is specifically selected as per the particular requirements (based on the objectives and functionality) of the active vision system, thus the system's overall state of each active vision system is unique. A camera sensor model enables the active vision system to determine the system's overall state in terms of system's state parameters such as power consumption, available resources, bandwidth utilization, calibration parameters, Quality of Service (QoS), and the like. Thus, camera sensor modelling plays an important role to enable self-reconfiguration of smart camera network in an active vision system. Some traditional camera models include thin lens camera model or projection models (such as orthographic projection models, para-perspective projection models, scaled orthographic projection models, linear perspective projection models, and the like). The thin lens camera model (*i.e.*, a linear camera calibration model) accounts for effects of translation and rotation with respect to a view plane. A pin-hole camera model (*i.e.*, a linear perspective projection model) performs better in terms of the performance and QoS, however, it has a high computational complexity as compared to the thin lens model due to a higher number of model parameters. Hall et al. in [19], based on 3D affine transformation with linear perspective projection, proposed a simplified and efficient linear model with reduced computational complexity and a comparatively higher QoS. However, the linear models fail to account for non-linear distortions and thus result in poor QoS. Tsai et al. in [20] by way of a non-linear perspective projection, Toscani in [21] and Wang et al. in [22], by way of non-linear calibration, proposed non-linear camera models for better performance of the system in terms of the overall QoS of the system considering the non-linear distortions.

- Camera Localization: Camera localization facilitates each camera sensor of the smart camera network to have an awareness of a relative position with respect to the other sensors, which plays a vital role while exchanging objects of interest from one camera sensor to the other (*i.e.*, object handover). The camera localization further enables the active vision system to have dynamic topology based on an information of the locations and/or movements of the objects of interest. Furthermore, the camera localization helps in identification of active nodes in the camera network. Points, lines, spheres, cones, circles, and features etc. are some commonly used markers (*i.e.*, identifiers) for camera localization. Simultaneous Localization And Mapping (SLAM) as presented in [23] and [24], respectively, and Structure From Motion (SFM) as presented in [25] are designed for dynamically changing or unknown environmental conditions. The functional architecture of SFM [25] is motivated by human's vision perseverance. SMF [25] combines data of each frame with its motion information to estimate a 3-Dimensional (3D) scene from a 2D image data. In [26], Monte Carlo method proposed using particle filter for sensor (camera) localization. Monte Carlo method in [26] further proposed Recursive Bayesian Estimation based sampling and sorting of samples. Montzel et al. in [27] proposed use of sparse overlapping to design an energy efficient localization method for localization of camera sensors of a distributed camera network. Brachmann and Rotheren in [28] presented an end to end localization pipeline that facilitates 6D pose estimation of objects. Geometric localization proposed in [29] enabled self-calibration of camera sensors through estimated distribution algorithm (EDA) to detect head-to-foot location of pedestrians, which were used for self-calibration of camera sensors. Table 2.2 presents an evolution of techniques and their associated advantages presented in the prior art to address the camera localization problem.

Table 2.2 Techniques addressing camera localization problems.

Ref.	Year	Methodology	Advantages
[26]	1999	Online system for tracking multiple people in an SCN with overlapping and non-overlapping views	Development of a larger, more capable, and fully automatic system without prior localization information
[27]	2004	Sparse overlapping	Better energy efficiency and able to cope with networking dynamics
[23]	2006	SLAM	Locally optimal maps with computational complexity independent of the size of the map
[24]	2006	SLAM	Locally optimal maps with computational complexity independent of the size of the map

[29]	2016	Estimated distribution algorithm (EDA)	Accurate estimation of the features of moving objects (person)
[25]	2017	SFM	Better ambiguity handling in 3D environments
[28]	2018	6D pose estimation using an end-to-end localization pipeline	Efficient, highly accurate, robust in training, and exhibits outstanding generalization capabilities

- Parameter Estimation and Correction: Parameter estimation and correction enables real-time calibration of configuration space of each camera sensor of the smart camera network deployed in the active vision system. Based on the parameter estimation through the camera model, the active vision system enables a correction of parametric values, for calibration of camera sensors of the smart camera network. Zheng et al. in [30], by way of parallel particle swarm optimization (Parallel-PSO) proposed an efficient method with low computational complexity for estimation of focal length of the camera sensors. In [31], Jung and Führ proposed a method for self-calibration of multiple camera sensors deployed in a sensor network. The self-calibration method in [31] used non-linear optimization of a projection matrix for localization of camera sensors. Yao et al. in [32] proposed a field model for self-calibration of a number of multi-view camera sensors deployed in a camera network. The field model in [32] utilized golf and soccer datasets for self-calibration of the multi-view camera sensors. A camera-projector pairs framework based on greedy-descent optimisation was proposed by Li et al. in [33] for parameter estimation and scene reconstruction that facilitated self-calibration of camera sensors. The framework proposed in [33] provided basis for possible enablement of tele-immersion applications with an evolution in resources and technology in future. A self-reconfiguration approach for a camera network with focal-length estimation using homograph from unidentified planar scenes was put forth by Janne and Heikkilä in [34]. Tang et al. in [35] proposed a simultaneous distortion-correction method for self-configuration of parameters specifically for tracking and segmentation of objects of interest. The method in [35] is based on an evolutionary optimisation scheme on an estimated distribution algorithm (EDA). Table 2.3 presents an evolution of techniques and their associated advantages proposed in the prior art to address the parameter estimation and correction problem.

Table 2.3 Techniques addressing parameter estimation and correction problem.

Ref.	**Year**	**Methodology**	**Advantages**
[31]	2015	Projection matrix obtained from non-linear optimization	Better accuracy
[32]	2016	Field model	Automatic estimation of camera parameters with high accuracy
[33]	2017	Greedy descent optimization	Stable and robust automatic geometric projector camera calibration with high accuracy; and Efficient in tele-immersion applications
[34]	2017	Homography from unknown planar scenes	Highly stable
[30]	2018	Parallel particle swarm optimization (PSO)	Low time complexity and efficient performance
[35]	2019	Evolutionary optimization scheme on an EDA	Capability of reliably converting 2D object tracking into 3D space

2.1.3 Resource optimization

The active vision system must be capable of efficiently estimating an overall task load and resources available with each component of the smart camera network to achieve the desired functionality from the active vision system. Further, the active vision system must be capable of determining an optimized task load distribution amongst each component of the smart camera network. Particularly, the resource optimization problem of the active vision system can be divided into two sub categories as discussed hereinbelow.

- Topology Estimation: Sensors switching from active to inactive states cause the network topology to change dynamically. To avoid deviating from the primary goal (*i.e.*, capturing data with high-quality visual information), the overall functionality of the smart camera network is dynamically distributed amongst various camera sensors in the form of task loads. Thus, it is challenging for the Smart Camera Network to compute the dynamic topology, determine the node localizations, and distribute the task load amongst the active nodes for real-time applications. In [36], Marinakis and Dudek proposed a method for generation of weighted directed graph for estimation of topology of a visual sensor network based on statistical Monte Carlo expectation and sampling models. Hangel et al. in [37] proposed a window-occupancy (WO) based method for estimation of camera network topology. The WO based method in [37] had a lot of assumptions, and was insufficient to handle huge amount of visual data captured by the large camera network over a period of time. A topology estimation method presented in [38] by Detmold et al. suggested an exclusion algorithm based on scaling collective stream processing method to handle data from a distributed clusters of nodes in a large network of nodes. The method in [38] provided a decentralized processing scheme for topology estimation of large network of nodes. In [39], Clarot et al. proposed an network topology for distributed networks based on activity matching. Topology estimation using identity and appearance similarity in a distributed network environment was suggested by Zhou et al. [40]. Farrel and Davis in [41] proposed network topology estimation in a decentralized sensor network. A centralized topology estimation for variable lightening conditions (*i.e.*, lightening variations) was presented by Zhu et al. in [42]. Misra and Gautam in [43] proposed a trust-based topology management system for distributed sensor networks. Tan et al. in [44] proposed a topology estimation method based on a blind distance calculation technique. Li et al. in [45] proposed a topology estimation method for a distributed camera network using mean cross-correlation functions and Gaussian functions. Table 2.4 presents an evolution of techniques and their associated advantages proposed in the prior art to address the network topology estimation problem.

Table 2.4 Techniques addressing network topology estimation problem.

Ref.	Year	Methodology	Advantages
[36]	2005	Monte Carlo expectation maximization and sampling	Minimum effects of noise and delay
[37]	2006	Window-occupancy-based method	Efficient and effective way to learn an activity topology for a large network of cameras with a limited number of data
[38]	2007	Exclusion algorithm in distributed clusters	High scalability
[40]	2007	Statistical approach in distributed network environment	Robustness with respect to appearance changes and better estimation in a time varying network
[41]	2008	Decentralized data processing	Robustness with respect to variable appearance and better scalability
[39]	2009	Activity-based multi-camera matching procedure	Flexible and scalable
[42]	2015	Pipeline processing of lightning variations	Automated tracking and reidentification across large camera networks
[43]	2015	Trust-based topology management system	Higher average coverage ratio and average packet delivery ratio
[44]	2018	Blind-area distance estimation	Finer granularity and high accuracy
[45]	2018	Gaussian and mean cross-correlations	Better target tracking under a single region and better interference in multi-view regions

- Task load balancing: In order to achieve an optimized functionality, an effective active vision system must divide the overall functionality of the application into a number of smaller tasks. The task load of each active node of the smart camera network is determined by its local state, orientation, and resource availability. A distributed approach for adaptive task-load assignment based on available energy from the network environment was presented by Kansal et al. [46], which significantly increased the system's lifetime. In [47], Rinner et al. proposed a task allocation framework based on heterogeneous mobile agents for a distributed multi-view camera network. Rinner et al. later in [48] updated the task allocation framework by clustering the areas under observation (*i.e.*, surveillance areas). Karuppiah et al., in [49] proposed a hierarchy-based algorithm for task-load balancing and resource allocation to multiple components of a distributed multi-sensor network. The algorithm in [49] detected fault tolerance using activity density as a parameter for task load balancing and resource allocation in the distributed multi-sensor network. In [50], Dieber et al., using expectation maximization (EM) proposed a task load balancing algorithm deliberately designed to provide efficient resource utilization and optimized monitoring performance. Dieber et al. further extended their work in [51] through market-based handover of objects to efficiently balance task load between multiple camera sensors used for real-time tracking application. A market-based bidding framework was proposed by Christos et al. in [52] for efficient multi-task allocation and task load balancing in a distributed network of camera sensors. Table 2.5 presents an evolution of techniques and their associated advantages proposed in the prior art to address the challenges in task load balancing.

Table 2.5 Techniques addressing task load balancing problem.

Ref.	Year	Methodology	Advantages
[46]	2003	Method for distributed adaptive task-load assignment	Better resource efficiency
[47]	2005	Multiple-mobile-agent-based task-allocation framework	Selective operation of the tracking algorithm to reduce the resource utilization
[48]	2005	Multiple-mobile-agent-based task-allocation framework	Selective operation of the tracking algorithm to reduce the resource utilization
[49]	2010	Hierarchy-based automatic resource allotment	Robust tracking
[50]	2011	Expectation-maximization-based approximation	Efficient approximation method for optimizing the coverage and resource allocation
[51]	2012	Market-based handover	Improved quality of surveillance with optimized resources
[52]	2016	Market-based handover	Improved quality of surveillance with optimized resources

2.1.4 Occlusion handling

Occasionally, an object of interest can be occluded by one or more unwanted objects which results in loss of information of the object of interest at the time of occlusion. Several techniques proposed to handle such a situation involve handing over the object of interest to another camera sensor capable of capturing the objects of interest without occlusion. However, finding a next-best camera sensor with non-occluded object of interest in the FoV in real-time is quite challenging. Occlusion handling becomes more challenging in systems with dynamic topology. Traditionally, solutions proposed for occlusion handling rely on prediction-based approaches to reproduce or predict the portion of the object of interest that is occluded, thus impacts the performance of the system. There is a high possibility of missing critical information (due to prediction), which makes such prediction-based approaches unreliable. In [53], Wang et al., proposed a red-green-blue (RGB) model using patch-match optimization for occlusion detection using smoothness regularization and feature consistency as performance parameters. Quyang et al. in [54] proposed a part-based deep model framework capable of occlusion handling by estimating information loss due to occlusion in the form of an error in detection of visible parts of the occluded object. In [55], Shahzad et al. proposed a statistical approach using K-means technique for occlusion handling in a multi-object tracking environment. Rehman et al. in [56] proposed a social force model to mitigate the effect of occlusion from the occluded images. The model in [56] proposed variational Bayesian method for clustering and concepts of repulsive and attractive forces for multi-object tracking. Chang et al. in [57] proposed a convolution neural network (CNN) based multi-object tracking system capable of handling occlusions by classifying surveillance areas into zones. In [58], Zhao et al. proposed a Gaussian model based adaptive background formulation technique for object tracking and occlusion handling. Liu et al. in [59] proposed a 3-dimensional (3D) mean shift algorithm for handling occlusions based on a derived depth information. Table 2.6 presents an evolution of techniques and their associated advantages proposed in the prior art to address occlusion handling.

Table 2.6 Approaches addressing Occlusion handling problem

Ref.	Year	Methodology	Advantages
[53]	2015	Patch-match optimization	Reduced computational complexity by large displacement motion
[54]	2015	Part-based deep model	Handles illumination changes, appearance change, abnormal deformation, and occlusions effectively
[56]	2015	Social force model	Improved tracking performance in the presence of complex occlusions
[55]	2016	K-means algorithm and statistical approach	Cost-effective in terms of resources (memory and computation)
[58]	2017	Gaussian model for occlusion handling	Handles appearance changes and is capable of dealing with complex occlusions
[57]	2018	CNN	High performance with a limited labelled training dataset
[59]	2018	Distraction-aware tracking system	Effective and computationally efficient occlusion handling

2.2 Processing level challenges

Computer vision systems rely on processing images captured by one or more camera sensors to derive understanding of activities and events in the scene. Performance of active vision systems employing camera network highly rely on the computational capabilities of the system. Real-time applications make it more challenging for the active vision systems as such systems require deriving understanding of scenes in real-time. A brief description of the processing level challenges is presented hereinbelow.

2.2.1 Selection of processing platform

Selection of processing platforms for deployment of an active vision system is as critical as the algorithm used for processing images captured by the camera sensors. Specifically, the selection of a processing platform is based on the functional complexity and the processing time of the processing platform, which is typically dependent on the used case or application of the active vision system. For example, the architecture of a system requiring complex computations in real time may be more complex and thus more expensive than that of a system with a relaxed processing window for applications requiring simpler computations. The processing platforms are typically deployed either on hardware such as Application Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA), or on software such as Graphical Processing Unit (GPU) and Central Processing Unit (CPU). Choice of the processing platform depends on accuracy of result, need of processing capabilities, timeliness, resource use, and adaptability. Hardware based processing platforms are used in deployment of systems with specific and dedicated functionalities. Such systems can have great processing capabilities with a very low latency, however such systems lack flexibility of operations. On the other hand, software based platforms provide high flexibility of operation at the cost of higher latency. Thus, hardware based platforms can be used for real-time active vision applications which require high efficiency, better performance, and faster computations. Software based platforms can be utilized for applications which require higher flexibility in terms of customizable used cases.

Fang et al. in [60] presented a comparative analysis for selection of processing platforms in detail. A comparison of general-purpose computations carried out by CPUs and GPUs in computer vision systems is presented in by Horup et al. in [61]. Guo et al. in [62] proposed a flexible and fast CPU-based computation system for human pose estimation. Tan et al. in [63] proposed a flexible and fast GPU-based deep-learning-based computer vision system. Irmak et al. in [64], Costa et al. in [65], and Carbajal et al. in [66] proposed computer vision systems using Field Programmable Gate Array (FPGA). Xiong et al. in [67] presented computer vision system using Application Specific Integrated Circuit (ASIC) for enhancement of operational flexibility. High processing capabilities, low latency, and flexibility of operation can be derived by hybrid processing platforms (*i.e.*, hardware-software combination) as presented in [68]. The state-of-art research aims to develop stable and efficient hybrid systems with high flexibility of operation and low processing latency.

2.2.2 Scene reconstruction

To obtain useful information from the captured images (*i.e.*, the captured data), the images captured by the active camera sensors must be synchronised. The data captured by each camera sensor must be combined for scene reconstruction to identify the activity of objects of interest over a period of time by analysing temporal frames. Particularly, the reconstruction of scene becomes very challenging when the topology of camera network changes dynamically.

In [69], R. Szeliski proposed a volumetric scene-reconstruction method using a multiple depth maps with layered structure. Martinec et al. in [70] proposed a 3-dimensional (3D) scene reconstruction method. The method in [70] used an uncalibrated image dataset with a pipelining approach to detect regions of interest (ROIs), and match the ROIs by way of a random sample consensus (RANSAC) mechanism. In [71], Peng et al. proposed a network geometry-estimation method for scene reconstruction. The method in [71] suggested two-view geometry estimation by way of an L-2 Estimation Local Structure Constraint (L2E-LSC) algorithm based on local structure constraint.

Effective point matching approaches provide imperative solutions for effective scene reconstruction. In [72], Brito et al. compared different point matching approaches, such as Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented Fast and Rotated Brief (ORB), Fast Retina Key-points (FREAK), and Binary Robust Invariant Scalable Key-points (BRISK). Milani [73] proposed localization-based reconstruction for heterogeneous camera sensor networks. In [74], Ali Akbar et al., using parametric homographs proposed a scene reconstruction method. Ali Akbar et al. in [74] further reviewed various scene-reconstruction approaches. Wang and Guo in [75], using plane primitives of an RGB-D frame, presented an effective scene reconstruction method. In [76], Ma et al., using an adaptive octree division algorithm proposed a mesh-reconstruction system for point-cloud segmentation, mesh re-labelling, and scene reconstruction. In [77], Ichimru et al. utilizing transfer learning, presented 3D scene reconstruction using a CNN under water bubble dataset to avoid distortions.

2.2.3 Data processing

Data processing is one of the major factors that contribute to an overall performance of the active vision system. The accuracy of image processing, deriving information, and understanding of activities in the scenes highly rely on the data processing algorithms used in the active vision system. The selection of an algorithm majorly depends on accuracy and timeliness of the algorithm. Data processing algorithms are specifically selected to fulfil all the requirements of the active vision system based on the used case of the active vision system. Thus, to be used for a variety of applications, the data processing algorithm must be versatile and customizable. Some of the major problem areas in data processing include object detection, object classification and tracking, object re-identification, pose and behaviour estimation, activity recognition, and scene understanding that are discussed hereinbelow.

- Object Detection: The first and primitive step towards deriving an understanding of the scene under observation is to derive the information of the object(s) of interest. To do so, most of the active computer vision systems rely on segregation of foreground from the frame (commonly known as background-foreground segregation) which typically requires analysis of temporal frames captured by the same camera sensor. Detection of multiple objects present in a single frame makes the object detection even more challenging. For real-time active vision applications, multi-object detection becomes very critical as the reconfiguration of camera sensors depend on detection of objects of interest in real-time. Various other factors such as variations in illumination, camera viewpoints, occlusions etc. make real-time object detection even more challenging.

Viola and Jones technique as presented in [78], scale-invariant feature transformation (SIFT) as presented in [79], HOG-based object detection as presented in [80], optical flow based object detection as presented in [81] and [82], and background subtraction technique for object detection as presented in [83] are some of the most commonly used techniques for object detection. Contemporary object detection methods use machine learning based approaches such as neural networks as used in [84], you-only-look-once (YOLO) as used in [85], region proposals (R-CNN) as used in [86], single shot refinement neural networks as used in [87], Retina-Net as used in [88], and single-shot detectors (SSDs) as used in [89]. Advanced machine learning based object detection methods [84]-[89] provide better performance than the traditional model based object detection methods [78]-[83] in terms of object detection accuracy. However, machine learning based methods highly rely on accurate training datasets, which is not a limitation in model based approaches. Progress of object detection techniques used for various computer vision applications from traditional probabilistic prediction based approaches to contemporary and more advanced artificial intelligence (AI) based approaches is presented in [2].

As discussed earlier, occlusions, variations in illumination, and movement of objects of interest are some of the factors of concern for efficient detection of objects of interest. An adaptive background subtraction model, using a single sliding window, by way of a histogram minimum–maximum bucket method was proposed by Roy and Ghosh in [90]. The adaptive background subtraction model in [90] used a median-finding based approach to tackle illumination changes in the evaluating temporal frames. In [5], Bharti et al. proposed an adaptive real-time kernelized correlation framework to handle occlusions. The framework in [5], based on a determined confidence values of object tracker, enabled drones to update their location and boundary information on a distributed network of drones. In [91], Min et al., using pixel lifespan proposed a multi-object detection method to merge pseudo-shadows (specifically ghost shadows) to the image background. Min et al. in [91] proposed state vector machine (SVM) and convolutional neural network (CNN) based classifier to avoid occlusions.

Active vision model for detection of moving objects of interest requires complex computations and relies on estimations and approximations, as the camera sensor is required to follow the objects to capture the objects at highest possible resolution in real time. In [92], Wu et al. proposed a computational model to efficiently address the moving object detection problem with very low latency. The computational model by Wu et al. in [92] suggested evaluating a coarse foreground using singular-value decomposition, and using foreground information to reconstruct the background by way of an in-painting technique. Wu et al. in [92] further used mean shift segmentation refinement of the detected foreground. In [93], Hu et al. used tensor flow to detect moving objects without hampering or altering the scene dynamics. Hu et al. in [93] utilized saliently fused sparse regularization to detect initial foreground and tensor nuclear norms to handle redundancy in the background. To compute spatiotemporal variations, Hu et al. in [93] further proposed a 3D regression kernel with local adaptability, that enabled refinement of the initial foreground. Table 2.7 presents an evolution of techniques and their associated advantages presented in the prior art to address object detection challenges.

Table 2.7 Evolution of techniques addressing object detection problem

Ref.	Year	Methodology	Advantages
[83]	1989	Background subtraction	Low computational complexity
[78]	2001	Viola and Jones technique	Low processing latency with high detection rate
[80]	2005	HOG-based detection	Precise object detection and classification
[79]	2012	Scale-invariant feature transformation	Efficient detection and localization of duplicate objects under extreme occlusion

[81]	2013	Optical flow	Accurate detection of moving objects
[86]	2014	Region proposals (R-CNNs)	High accuracy and precision for object detection
[92]	2015	Background subtraction and mean shift	Refined and precise foreground detection
[85]	2016	You only look once	Low latency multi-object detection
[89]	2016	Deep-neural-network-based SSD	Prediction-based detection for variable shapes of objects
[93]	2016	Tensor flow	Detection of mobile objects in FOVs
[84]	2017	Neural network	Multi-object detection with variable shapes
[90]	2017	Adaptive background subtraction model	Better accuracy as compared to traditional background subtraction
[91]	2017	State-vector machine and CNN-based classifier	Multiple-object-detection approach to detect ghost shadows and avoid occlusions
[82]	2018	Optical flow	Accurate detection of moving objects
[87]	2018	Single-shot refinement neural network	High detection accuracy
[5]	2018	Kernelized correlation framework	Real-time occlusion handling
[88]	2019	Retina-Net	Balanced detection performance in terms of latency, accuracy, and precision of detection

- Object Classification and Tracking: After the detection of object(s) from the scene, the active vision system undergoes object classification to segregate the detected objects into classes, distinguish one object from the other, and determine whether the object is an object of interest or not. Object classification majorly includes three steps: (i) pattern recognition, (ii) clustering of pixels, and (iii) Segregation of pixels. Classification of objects requires selection of appearance parameters (*i.e.*, features) such as shape, colour, texture, temporal pixel motion etc. as discussed in [94], and silhouettes, points, contours, etc. as discussed in [95]. Conventionally, the techniques for object-classification as discussed in [96] can be categorized into the following broad categories: (i) decision based object classification such as decision trees as presented in [97] and [98], and random forest as presented in [99], (ii) statistical-probability based object classification such as Bayesian classification as presented in [100], [101], and [102], discriminant analysis as presented in [103], logical regression as presented in [104], and nearest-neighbour as presented in [105], and (iii) soft-computing based object classification such as SVM presented in [106], multi-layered perceptron as presented in [107], and neural networks as presented in [108] and [109]. Tracking the objects of interest is required to determine one or more activities performed by the objects of interest by analysing consecutive temporal frames. Due to different viewpoints (*i.e.*, difference in perseverance of the objects) in consecutive frames captured by the camera sensors, the computational complexity of the active vision system and chances of error increase drastically as the system encounters distortion of the objects of interest due to different viewpoints. In [110], Villiers et al. proposed a real-time inverse distortion method for correction of distorted images. In [111], use of a number of properties of vanishing properties for calibration of parameters and distortion correction was proposed. In [35], Caprile et al. proposed a distributed algorithm for calibration and correction of radial distortion using vanishing points. Caprile et al. in [35] further proposed tracking a waking human's movement (as poles) for determination of vanishing points. Methods suggested in [112] and [113] used estimation of a centre of distortion for detection and correction of radial distortion. In [114], Huang et al. proposed detection and correction of radial distortion correction based on linear-transformation functions, whereas in [115], Zhao et al. proposed a pipelined process for the same. Methods proposed in [116], [117] and [118] suggested detection and correction of radial and tangential distortions. In [119], Yang et al., using information of depth of the objects proposed estimation and correction of perspective distortion. In [120], detection and correction of optical distortion was addressed by Finlayson et al. using colour-calibration theory. Wong et al. in [121], using a multi-spectral camera model extended the use of colour-calibration theory for distortion estimation. Motion-blur is often a challenge for multi-object tracking systems that impacts the performance of the active vision system. A motion-aware tracker was proposed in [122] by Han et al. to detect and correct motion-blur by estimation and correction of tracking fragments caused by motion-blur or occlusion. A former tracking system proposed by Meinhardit et al. in [123] addressed motion blur estimation and correction in a multi-object tracking environment.

- Pose and Behaviour estimation: Techniques used for pose and behaviour estimation utilize models to associate poses, patterns of postures, and/or shapes of detected objects of interest in consecutive temporal frames to derive a behavioural understanding of activities performed by the objects. Accuracy of pose and behaviour detection depends majorly on selection of models for pose-estimation and deriving a relationship between poses derived from consecutive temporal frames. Particularly, the commonly used techniques for pose and behaviour estimation are either model-based techniques (such as planar models, volumetric models, and kinematic models) or model-free techniques. Planar models use contours as features, volumetric models use volume distributions as features, and kinematic models use pixel motion as features for pose and behavioural estimation. Kinematic models provide pose and behaviour estimation with low computational complexity, however, the performance and efficiency of such models is not reliable, and varies with dynamic changes in the captured scene. Planar models and volumetric models provide better efficiency and performance at a cost of high computational complexity. In [129], Chen et al. proposed an anatomically aware 3D pose-estimation model capable of efficient pose and behaviour estimation with low computational complexity. Staraka et al. in [130] proposed an efficient and accurate kinematic skeletal-model based technique for pose and behaviour estimation in real-time.

- Object Re-identification: To derive an appropriate understanding of a scene, the active vision system is required to associate multiple activities performed by objects of interest by analysing consecutive frames captured by multiple cameras that are captured over different periods of time. Due to several factors such as change in viewpoints, illumination changes, and ghost shadows (as presented in [124]) etc., the possibility of wrong detection (*i.e.*, mis-interpretation or false detection of objects) is very high. To address the challenges of illumination changes that may hamper re-identification of the objects of interest, Zhang et al. in [125], by way of a Fisher vector learning technique proposed an adaptive re-identification framework for the spatiotemporal alignment of frames. Yang et al. in [126] proposed a logical determinant metric learning method to overcome the limitations of variable viewpoints and occlusions in object re-identification. It has been observed that a combination of more than one feature enhances the accuracy of re-identification, if the features and weights of each selected feature are selected appropriately. For the determination of the most efficient features for specific object re-identification problems, Geng et al. in [127] proposed a feature-fusion method based on weighted-center graph theory. The method in [127] proved to be effective in determining an importance value (*i.e.*, efficiency) of each feature of a set of selected features for object re-identification. In [128], Yang et al. proposed a method for using partial information of an occluded object for re-identification of occluded objects.

2.2.4 Activity Recognition and Scene Understanding

There are two paradigms for activity detection and recognition: static and dynamic. Static activity recognition is the process of identifying an activity solely from the spatial analysis of a single frame and thus does not necessities pose estimation. In contrast to static activity recognition, the dynamic activity recognition requires spatiotemporal computations of multiple consecutive frames using scene reconstruction which essentially requires the estimation of the pose and behaviour of the objects of interest over a period of time. Campbell et al. in [131] depicted human motion using phase-space constraints. For activity detection of pedestrians, Oren et al. in [132] proposed a single-frame wavelet templates method. Static activity recognition is commonly used for image captioning [133], manuscript review in the medical field [134], and academia [135]. Nguyen et al. in [136] proposed multi-objective optimisation to monitor real-time activity. Use of dynamic activity recognition for sports analysis is illustrated in [137] and [3], whereas Wu et al. in [4] proposed dynamic activity recognition for smart home applications. Xiang et al. in [138] proposed a multiple-object tracking system capable of making smart decisions based on dynamic activity recognition. Laptev et al. in [139] proposed an abnormal human activity recognition system based on state vector machines.

2.3 Contemporary Research Paradigm

Trained AI/ML models facilitate the active vision systems with fast and accurate data processing capabilities. Thus, most contemporary systems presented in [140] to [156] utilize AI/ML based techniques to address various problems of the active vision system. Over the last few years, a shift has been observed from model-based solutions to Artificial Intelligence (AI) and Machine learning (ML) based solutions to address various challenges of the active vision system. It has further been observed that the state-of-art research majorly focuses on occlusion handling, enhancement of multi-object tracking accuracy, and reducing the re-configuration latency of multi-sensor networks. According to a market report presented in [157], artificial intelligence has an annual growth of over 45% in active computer vision applications. Some recently proposed models presented in [147] and [151] generate highly accurate hybrid systems by combining the concepts of traditional model-based methods with modern AI-based approaches. AI based active vision system as proposed in [153] use the basic principles of traditional model-based methods. However, efficient ML-based operational models for active vision applications require highly accurate and application specific training datasets. Thus, to address active vision challenges for unforeseen conditions, the ML-based systems require intensive training to understand the unforeseen condition, which requires a lot of time and processing capabilities. Thus, active vision systems employing centralised camera networks fail to address various challenges of active vision systems in new unforeseen conditions miserably. Table 2.8 shows some of the most advanced AI-based systems addressing the challenges of self-reconfiguration faced by active vision systems.

Table 2.8 Contemporary Active vision approaches.

Ref.	Challenge Addressed	AI/ML- Based Approach Used
[140]	Camera calibration	Convolutional neural network (CNN)
[141]		Neural network
[142]	Parameter estimation	Convolutional neural network (CNN)
[143]		Deep neural network (DNN)
[144]	Pose estimation	Neural network
[145]	Object detection	Modified CNN
[146]	Object tracking	Residual neural network
[147]		Deep CNN and Kalman filter
[148]		Deep neural network (DNN)
[149]		CNN and deep sort
[150]		Deep-learning-based CNN
[6]	Activity detection	Slow–fast CNN
[151]		Neural network and strider algorithm
[152]	Object re-identification	CNN
[153]		Sparse graph-wavelet-based CNN
[154]	Object re-identification and occlusion handling	Deep-neural-network-based transfer learning
[155]	Localization	CNN
[156]		Neural network

AI/ML based active vision systems are further prone to visual attacks [158] (such as adversarial attacks as presented in [159] and [160]) which affects the efficiency of the active vision system over a period of time. Visual attacks can be targeted if the model predicts their outcome correctly as shown in [158], however in most cases, visual attacks introduce random noise to the model parameters, and are therefore very challenging to undo, resulting in a long-term decline in the model's performance. A study presented in [161] suggests effects of adversarial attacks on the performance of AI/ML based active vision models. A study on different types of visual attacks and methods for mitigation of visual attacks (such as adversarial attacks) is presented in [162] and [163].

2.4 Active vision systems with Limited Resources

Majority of active vision systems, including those suggested in [10], [11], and [12], prioritise the surveillance area based on predefined assumptions about the camera sensor's field of view and manually decided critical areas of surveillance under the camera sensor's field of view. The predefined placement and orientation of the camera sensors results in an inappropriate sensor's pose (as the activities are highly dynamic in their occurrence and can occur anywhere in or outside the camera sensor's field of view), making it difficult for the camera sensors to capture the objects of interest in the centre of its field of view (FOV) for best possible resolution. Such a configuration of the camera network results in inappropriate sensing by the camera sensors that results in inefficient performance of the active vision system.

To optimize the information from the captured frames, the active vision system must locate a region of importance (ROI) in the camera sensors' field of view (FOV), and then reconfigure the configuration space of the camera sensors to align a centre of the region of importance (or interest) with the centre of the camera's field of view. There are a number of ways to determine ROI, activity mapping based detection of region of interest as presented in [164] being one of the most effective ones. The majority of proposed active vision systems for ROI determination ignore the effects of historical events on the activity map and instead concentrate on the activities detected in the present frame for activity mapping. Instead of establishing a spatiotemporal relationship between the current event and earlier events, such systems typically concentrate on reducing frame noise.

The active vision systems used for mobile surveillance applications generally employ a network of camera sensors deployed on mobile drones. Such systems usually have limited resources, which makes it critical to utilize the resources efficiently. The idea of pre-defined prioritization of areas under surveillance as presented in [10] to [12] to capture frames of the scene without activity based reconfiguration fails miserably in such conditions. To deliver an efficient performance, such systems further require relating consecutive frames temporally to derive an effective understanding of the scene and calibrate the configuration space of each camera sensor in real time. The computational complexity of such an active vision system becomes very high and demands high resource utilization, which is a limitation of such mobile systems.

Region of interest (ROI) estimation for vehicle flow detection using morphological operations to filter noise was proposed by Pan et al. [165] and Mehboob et al. [166]. Pan et al. in [165] suggested edge detection for object detection, background subtraction for traffic estimation, and morphological features for noise reduction. Mehboob et al. in [166] used motion vectors for traffic flow estimation and centroid detection using morphological close and erode operation for noise reduction. Although, Pan et al. in [165] and Mehboob et al. in [166] made efforts to reduce the noise in detection of objects of interest, they were unable to stop the scene's undesirable activities from being detected and included in the activity map. Pan et al. in [165] and Mehboob et al. in [166] further failed to tackle unidentified and unforeseen activities efficiently. The ROI detection in [165] and [166] was proposed based on the activity detected from only one frame, and lacked temporal relationship for scene understanding.

Modern methods for activity mapping either use artificial intelligence techniques for ROI detection or extremely complex computational models. A non-parametric spatiotemporal activity model based on Gaussian process regression (GPR) was presented by Marvin and Moritz in [167]. For the purpose of identifying group activity, Sattar et al. in [168] proposed a convolution neural network (CNN) based spatiotemporal activity mapping method. An online feature learning model for spatiotemporal event forecasting was put forth by Zhao and Gao in [169]. Using spatiotemporal activity patterns, Liu and Jing in [170] proposed an artificial intelligence (AI) based activity mapping method for sports analytics. Long short-term memory (LSTM) was used by Yan et al. in [171] to propose an end-to-end position aware spatiotemporal activity analysis.

The methods in [165] and [166] offer straightforward models for activity mapping and ROI detection, however, such methods lack precision of ROI detection. The approaches suggested in [167] to [171] offer effective spatiotemporal activity mapping for ROI detection, but at the expense of high computational complexity, making them unsuitable for systems with scarce or constrained resources. From the aforementioned references, there appears a trade-off between accuracy and computational complexity of the activity mapping approaches.

2.5 Research Gaps

From the review of existing literature as discussed in this chapter, the following research gaps in active vision systems are observed that require exclusive attention:

- From the literature survey, it has been observed that the data captured by camera sensors deployed in active vision systems specifically designed for low resource utilization carry unoptimized information. Such systems rely on either pre-defined prioritization of areas under observation or activity mapping based on only one frame (*i.e.*, most recently captured frame). Thus, such active vision systems results in inefficient activity mapping that results in low efficiency of the systems.
- For efficient calibration of configuration space of each camera sensor deployed in active vision systems, appropriate understanding of the scene is critical. Typically, activities of an object of interest detected over different periods of time, when analysed over a timeline presents an understanding of the events in the scene. From the literature survey,

it has been observed that spatiotemporal relationship between activities and events missing.

- From the literature survey, it has been further observed that the task load for each camera sensor is distributed based on predefined protocols established at the time of deployment of active vision systems which results in inefficient task load balancing between nodes of camera network.
- The state-of-art active vision systems follow a centralized control paradigm for reconfiguration of camera sensors. Such systems fail miserably in unforeseen conditions, and thus lacks security and reliability. From the literature survey, the active vision systems lack adaptiveness which requires a technical solution to incorporate adaptiveness and high data exchange security to the active vision systems.

2.6 Conclusion

A successful computer vision system depends on accurate sensor data collection and a spatiotemporal understanding of the scene. Most sophisticated computer vision systems use a highly complex computation model to address both of the aforementioned issues, which is very problematic for most active vision systems with constrained resources.

As mentioned above, the functionality of active vision systems and the reconfiguration of the camera sensors that feed such active vision systems with data are interdependent. Modern active vision systems struggle miserably to deal with unforeseen conditions as it takes time for the reconfiguration model to be trained and adjust to the new circumstances and develop understanding of the unforeseen event. Thus, it is nearly impossible to reconfigure the configuration spaces of camera sensors deployed in such active vision systems in real-time. Furthermore, to process sensor data and create an understanding of an event, majority of the contemporary active vision systems rely on Artificial Intelligence-based models. However, such models run the risk of data loss because they are vulnerable to visual attacks such as adversarial attacks.

Thus there remains a need to associate the understanding of each activity temporally with a model having low computational complexity such that the model can be used with resource limitations. There further remains a need to incorporate self-adaptation to active vision systems to manage unforeseen conditions with low latency. Furthermore, such a system needs to be highly secure, scalable, and reliable.

The literature review is presented in one of our research papers cited as [172].

Chapter 3

SPATIOTEMPORAL ACTIVITY MAPPING

As there remains a need for a system to derive an understanding of an event by spatiotemporally evaluating a feed from a camera sensor network without exhausting the resources. This chapter presents a Spatiotemporal Activity Mapping (SAM) framework to generate efficient and dynamic spatiotemporal activity maps for a smart camera network with a low computational load and complexity. Thus, the SAM framework provides a solution for efficient and reliable activity based reconfiguration for active vision systems with limited resources. The SAM framework utilizes fundamental computer vision techniques such as frame differencing, binarization, thresholding, and federated optical flow for spatial activity mapping. The SAM framework further presents a temporal relationship function based on half width of normalized Gaussian Distribution to relate past and present frames temporally.

3.1 Foreground detection

As the framework requires low computational complexity and efficient foreground detection, the SAM framework proposes an adaptive background subtraction technique for initial background-foreground segregation. For initial foreground detection, the adaptive background subtraction utilizes frame differencing [173], adaptive thresholding inspired by recursive adaptive thresholding in [174], and binarization in progression to the adaptive thresholding. The frame differencing technique enables the SAM framework to detect pixels in motion in a number of consecutive temporal frames captured by a camera sensor network over a period of time. The adaptive thresholding enables the SAM framework to selectively segregate the pixels of each temporal frame for initial background-foreground segregation. After the adaptive thresholding, the SAM framework generates the initial foreground with original pixel values (*i.e.*, from 0-255). Binarization, in progression to the adaptive thresholding enables the SAM framework to binarize the pixel values of the initial foreground corresponding to each camera sensor of the camera sensor network. The adaptive background subtraction provides an efficient filter for initial foreground detection in each temporal frame at a cost of low resource utilization.

- Frame differencing: frame differencing utilizes differences in pixel values of consecutive frames captured by a stable camera sensor to determine regions in motion captured in the camera sensor's field of view. The ADAPSR framework identifies individual frame differences for each frame by comparing two temporally consecutive frames captured by the same camera sensor. Differences in pixel values of each consecutive frame for a range of 't' from 2nd frame to tth frame is computed using equation 3.1 as presented hereinbelow.

$$F_t\,(i,j) = P_t(i,j) - P_{t-1}(i,j) \qquad (3.1)$$

  Where, i and j represent the pixel width and height, respectively, $F_t\,(i,j)$ denotes the frame difference for t^{th} frame, $P_t(i,j)$ denotes the $(i,j)^{th}$ pixel value of the t^{th} frame, and $P_t(i,j)$ denotes the $(i,j)^{th}$ pixel value of the $(t-1)^{th}$ frame.

- Adaptive Thresholding: Adaptive thresholding provides a selective filtering for pixels of each frame by determining a cumulative mean values of the frame differences in each frame '$F_t\,(i,j)$'. Each frame difference value $F_t\,(i,j)$ below the cumulative mean value (*i.e.,* adaptive threshold value specific to the frame) is assigned a numerical ‘zero’ pixel value, and each pixel value above the cumulative mean value is retained. The cumulative mean value of frame difference corresponding to each frame is determined using equation 3.2 as presented hereinbelow.

$$Th_t = (\textstyle\sum_1^{i.j} F_t\ (i,j))/i * j \qquad (3.2)$$

Where Th_t represents denotes the cumulative mean value of the frame differences (*i.e.,* the adaptive threshold value) for the t^{th} frame.

- Binarization: Binarization enables the SAM framework to represent the initial foreground pixels of each image frame in either of “zero” binary value (*i.e.,* zero pixel value) or “one” binary values (*i.e.,* 255 pixel value) based on the adaptive threshold value, which reduces the computational load for the upcoming steps of activity mapping. The binarization of each pixel of each frame based on the respective adaptive threshold value is determined by using a conditional equation 3.3 as presented hereinbelow.

$$P_t(i,j) = \begin{cases} 0 \text{ (i.e., binary value ‘0’); if } P_t(i,j) < Th_t \\ 255 \text{ (i.e., binary value ‘1’); if } P_t(i,j) > Th_{t.} \end{cases} \qquad (3.3)$$

Upon binarization of the initial foreground, the SAM framework derives an initial foreground of each pixel with reduced dimensions.

- Federated optical flow: The initial foreground detected by the SAM framework includes errors due to unwanted activities such as shadows and undesirable moving portions (such as tree leaves moving in consecutive frames and the like) captured by the camera sensors. To address this problem of detecting undesirable objects in the scene, the SAM framework presents a federated optical flow based activity filter to remove such unwanted portions from the initial foreground. The SAM framework proposes determination of an optical flow in consecutive frames captured by each camera sensor parallel to adaptive background subtraction. The optical flow of each pixel is determined in terms of a pixel heading (*i.e.,* pixel distance moved and a direction of movement of each pixel in each frame with respect to its previous frame). The SAM framework further proposes determining a cumulative optical flow of each pixel in the consecutive frames in terms of the pixel heading. To avoid the effects of atmospheric interference, the direction of motion is detected with a tolerance of 'δ' (*i.e.,* obtained through comparison with the ground truth data over a number of images from various datasets). The cumulative optical flow 'O_c' for each pixel in the frames captured by a camera sensor is determined by equation 3.4 as presented hereinbelow.

$$O_c(i,j)_{dir} = \sum_{1}^{t} O(i,j)_{dir+\delta} \tag{3.4}$$

Where '$O_c(i,j)_{dir}$' represents the cumulative optical flow of *(i,j)* pixel moving in *'dir'* direction, *'t'* represents the number of frames, and '$O(i,j)_{dir+\delta}$' represents the optical flow of $(i,j)^{th}$ pixel in *'dir'* direction with tolerance value 'δ'.

Furthermore, the SAM framework proposes determining a cumulative mean value 'Th_o' from the cumulative optical flow 'O_c' as presented by equation 3.5 hereinbelow.

$$Th_o = \sum \frac{O_c(i,j)_{dir}}{i.j} \tag{3.5}$$

The cumulative mean value 'Th_o' is further used to filter the initial foreground to determine the actual foreground. The filtration of initial foreground using federated optical flow is presented through equation 3.6 hereinbelow.

$$P_t(i,j) = \begin{cases} 255 \text{ (i.e., binary value '1'); If } O_c(i,j) > Th_0 \\ \\ 0 \text{ (i.e., binary value '0'); If } O_c(i,j) < Th_o \end{cases} \tag{3.6}$$

Through the proposed adaptive background subtraction and the federated optical flow, the SAM framework provides an efficient foreground detection for accurate activity mapping without extensive exploitation of computational resources.

3.2 Temporal Relationship Function

Adaptive background subtraction and federated optical flow based filtering enable an efficient foreground detection by processing consecutive frames captured by the camera sensors. However, for efficient activity mapping and scene understanding, there is a need for an accurate yet non-complex temporal frame relationship. To achieve the same, the SAM framework proposes utilization of a normalized Half Width of the Gaussian distribution curve (partially or completely) for temporal association of frame captured by a camera sensor over a period of time. Specifically, for illustration of effect of spatiotemporal relationship in activity mapping, a normalized Half Width Half Maxima (HWHM) Gaussian distribution curve is utilized. Although, the use of the normalized HWHM Gaussian distribution curve is subjected to illustration of the effects of temporal activity mapping for specific circumstances assumed for testing on 300 consecutive frames captured by a camera sensor, it will be apparent to a person skilled in the art that the temporal relationship is not limited to the use of normalized HWHM Gaussian distribution curve as shown in Figure 3.1. Rather, the Half width Gaussian distribution function can be partially or completely utilized for temporal relationship of the consecutive frames based on the criticality and application of the active vision system, it is deployed in.

The normalized Gaussian distribution curve and the normalized HWHM gaussian curves are presented hereinbelow through equations 3.7 and equation 3.8, respectively.

$$G\ (x|\mu,\alpha^2) = (exp^{(x-\mu)/2\alpha})/\sqrt{2\pi\alpha^2} \tag{3.7}$$

$$H(x|\mu,\alpha^2) = (1/(sqrt(2\pi\alpha^2))*\ exp\ ((x\text{-}\mu)/\ 2\alpha^2)) \tag{3.8}$$

Where, $G(x|\mu,\alpha^2)$ *represents the Gaussian distribution function,* $H(x|\mu,\alpha^2)$ represents the normalized HWHM Gaussian function (*i.e.,* temporal relationship function used for association of consecutive frames), μ *and* α represent a mean value and the standard deviation value of the normalized HWHM Gaussian distribution. For normalized Gaussian function being a normalized function, the value of μ is set to '1', that reduces the equation 3.8 to equation 3.9 as presented hereinbelow.

$$H(\alpha) = (sqrt(2*log_n(2))*\alpha)/2 = 1.799\,\alpha \tag{3.9}$$

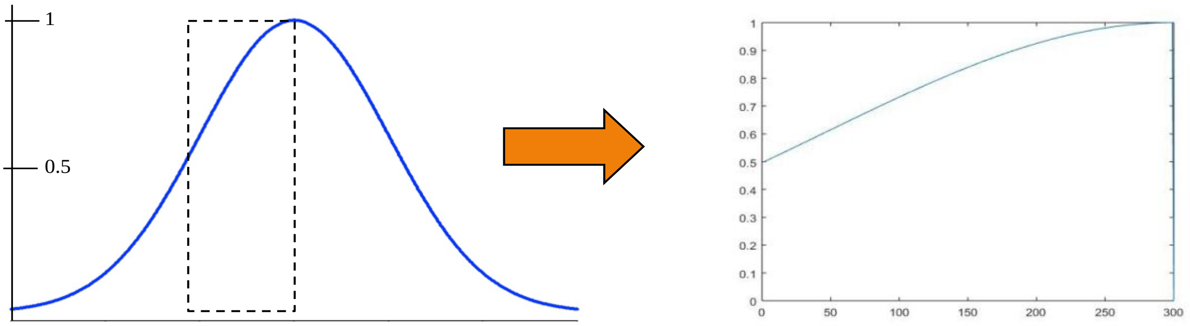


Figure 3.1 Normalized HWHM Gaussian curve for spatiotemporal relationship

3.3 Spatiotemporal activity mapping

The SAM framework proposes determining temporal importance function '*Ht*' from the spatiotemporal relationship through Normalized HWHM Gaussian function '*H(α)*' as presented above. Further, the SAM framework proposes using the temporal importance function for generation of spatiotemporal activity maps corresponding to each camera sensor. The spatiotemporal activity map bears information of Regions of Importance (ROI) corresponding to each camera sensor. Pixel importance '*A(i,j)*' can be derived through spatiotemporal activity map can be derived through equation 3.10 as presented hereinbelow.

$$A(i,j) = \sum_1^t P_t(i,j).H_t \tag{3.10}$$

The pixel importance derived through spatiotemporal activity map cumulatively presents a pixel-wise activity value detected by the camera sensor. However, for analyzing an event that lasts for a very long period of time, the activity map can have very high values which may tend to infinity, specifically for active vision systems deployed for applications such as continuous surveillance/monitoring, analytics of continuous data, and the like. High values corresponding to pixel importance in the activity map further increases the computational load on the system for further computations. Thus, to address the abovementioned challenge, the SAM framework proposes determining normalized importance for each pixel (*i.e.*, pixel sensitivity). The pixel sensitivity can be derived through equation 3.11 as presented hereinbelow.

$$S(i,j) = A(i,j)/\, max(A(i,j)) \tag{3.11}$$

Where, '*S (i,j)*' represents pixel sensitivity value of *(i,j)* pixel, '*A(i,j)*' presents the pixel importance value for *(i,j)* pixel and '*max(A(i,j))*' presents a maximum value of the pixel importance values of the spatiotemporal activity map.

Based on the pixel sensitivity values, the SAM framework proposes generation of a sensitivity map for each camera sensor in the form of a heat map. The SAM framework further proposes segregation of pixels into a number of clusters using the variational Bayesian method presented in [56].

Furthermore, the SAM framework proposes determination of one or more important clusters from the number of clusters based on a sensitivity threshold 'Th_s' derived through mean thresholding of the pixel sensitivity values as presented in equation 3.12 hereinbelow.

$$Th_s = \sum \frac{S(i,j)}{(i.j)} \tag{3.12}$$

For identification of the important clusters, the SAM framework presents determining average sensitivity value of each cluster and comparing the average sensitivity value of each cluster with the sensitivity threshold 'Th_s' as presented in equation 3.13 hereinbelow.

$$S_n(i,j) = \begin{cases} S_n(i,j);\ If\ S_n(avg) > Th_s\ (i.e.,\ important\ clusters) \\ \\ 0;\ If\ S_n(avg) < Th_s\ (i.e.,\ non\text{-}important\ clusters) \end{cases} \tag{3.13}$$

The SAM framework further proposes determining a center point of each cluster of the one or more important clusters. From each center point, the SAM framework proposes determination of a center of Region of Importance (ROI). The SAM framework further proposes reconfiguration of calibration parameters such that a center of the field of view (FOV) is matched to the corresponding center of ROI. As the SAM framework proposes calibration of camera sensors through low computational load, it provides efficient spatiotemporal activity-based calibration for active vision systems with limited resources.

3.4 Model

As discussed earlier, the SAM framework is specifically designed with low computational complexity to enhance the performance of an active vision system by providing activity based reconfiguration of calibration parameters of the camera sensors deployed in the system without extensive utilization of resources. Specifically, the development of an operational model based on the SAM framework requires at least one camera sensor, a processing circuitry, and a local memory unit. The camera sensor is configured to capture consecutive images of a scene (*i.e.*, an environment under observation) over a period of time, which is specific to either of, one or more activities captured in the scene and the criticality of an application for which the model is deployed. For flexibility of operation of the model, the period of time for each application is customizable. The camera sensor, by way of either of, an internal camera processing circuitry (*i.e.*, in case of smart cameras) and the processing circuitry of the model is configured to determine its configuration space in real time such that the configuration space includes internal and external calibration parameters of the camera sensor.

The processing circuitry is configured to perform various image processing tasks as discussed earlier in section 3.1 to section 3.3 such as initial foreground detection using adaptive background subtraction, background filtration through federated optical flow, temporal relationship assignment to consecutive images captured by the camera sensor, spatiotemporal activity mapping, and normalization of spatiotemporal activity map to determine pixel sensitivity map. The processing circuitry is further configured to perform various computational tasks such as clustering of the pixels in the pixel sensitivity map, determining one or more important clusters, determining a centre of each important cluster, determining a centre point of the sensitivity map (*i.e.*, a cumulative centre point of the various individual centre points corresponding to the one or more important clusters), determining a calibration model based on objectives and calibration parameters of the camera sensor, and calibrating the calibration parameters to match the centre point of the sensitivity map with the centre of the camera sensor's field of view.

The local memory unit is configured to store the images captured by the camera sensor, activity maps, pixel sensitivity maps, configuration space values, calibration parameters, camera calibration models etc.

3.5 Process

The SAM framework provides calibration of configuration space of one or more camera sensors deployed in the active vision system based on the spatiotemporal activities detected and analysed in the field of view of each camera in the process shown in Figure 3.2 hereinbelow.

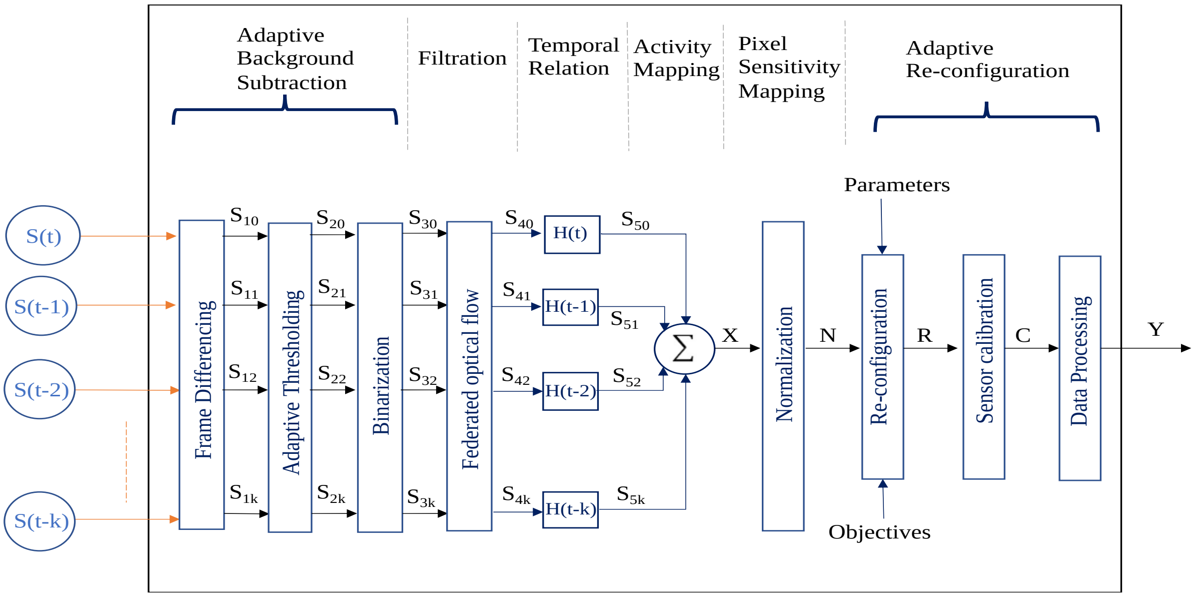


Figure 3.2 Process flow of SAM framework

Notations Used:

k: Number of past frames (images) captured by the camera sensor;

S(t): Image data (in pixel values) sensed by the camera sensor in the present frame;

S(t-i): Image data (in pixel values) sensed by the camera sensor in "i" frames prior to the present frame;

S_{1i}: Initial activity data obtained by object detection on S(t-i) using frame differencing;

S_{2i}: Data obtained by applying adaptive thresholding on S_{1i} for initial filtration of noise from the initial activity data;

S_{3i}: Initial foreground data in binary pixel values obtained by binarization of S_{2i};

S_{4i}: Enhanced foreground Data obtained by filtering S_{3i} using federated optical flow to remove unwanted portions from the initial foreground data;

H(t-k): Temporal function for k^{th} frames prior to the present frame;

S_{5i}: Temporal component of S_{4i} obtained as a product of S_{4i} and H(t-i);

X: Cumulative spatiotemporal activity map for the present frame;

N: Normalized spatiotemporal pixel sensitivity map for the present frame;

R: Reconfiguration parameters for the camera sensor;

C: Data from calibrated camera sensor; and

Y: Activity analysis after processing C.

In operation, the camera sensor captures consecutive images of the environment corresponding to a scene (*i.e.*, one or more activities). The camera sensor further determines and sends the calibration parameters to the processing circuitry. The processing circuitry receives the consecutive images and arranges them according to a respective timestamp attached to each image. The processing circuitry further determines an initial foreground from each image of the consecutive images using adaptive background subtraction that includes frame differencing for determining initial activity data corresponding to each image of the consecutive images to generate an initial activity data, adaptive thresholding for initial filtration of noise from the initial activity data, and binarization for downscaling the pixel values of the filtered initial activity data to binary form to reduce the computational load of proceeding computations. Furthermore, upon determination of initial foreground regions from each image of the consecutive images, the processing circuitry filters unwanted regions from the initial foreground using the federated optical flow technique to determine accurate foreground regions for each image of the consecutive images.

Upon the determination of the accurate foreground regions corresponding to each image, the processing circuitry determines a temporal relationship function value for each image using complete (or partial) half width of Gaussian distribution curve. The portion of the half width Gaussian distribution is selected based on the criticality and type of application and is customizable to provide flexibility of operation for various types of applications. The processing circuitry further determines a corresponding temporal component for each image of the consecutive images by multiplying each image with its corresponding temporal relationship function value. The processing circuitry further determines a cumulative activity map by pixelwise addition of each temporal component of each image of the consecutive images. To avoid high computational complexity due to cumulative pixel values approaching infinite values, the processing circuitry determines normalized values for each pixel (*i.e.*, pixel sensitivity value) for each pixel in the cumulative activity map to generate a sensitivity map (*i.e.*, heat map).

Based on the sensitivity map, the processing circuitry segregates the pixels of the sensitivity map into a number of clusters and determines one or more important clusters using mean thresholding technique. The processing circuitry further determines the centre of each important cluster, and a centre point from all the centres of the important clusters. Furthermore, the processing circuitry calibrates the parameters of the camera sensor such that the centre of the camera sensor's field of view coincides with the centre point.

3.6 Performance Parameters

The performance of the proposed SAM framework is evaluated in terms of multi-object tracking accuracy (MOTA) (*i.e.,* the accuracy of detection of regions having pixels of importance) based on the spatiotemporal pixel sensitivity map derived by processing the number of consecutive images captured by the camera sensors. The MOTA (%) increases when the truly detected importance pixel count increases and the falsely detected importance pixel count reduces. To obtain the ground truth data for sensitivity maps, markers are applied on the consecutive temporal images for evaluation of the performance of SAM framework in terms of MOTA (%). Specifically, the MOTA (%) depends on true positive pixel count (TPC), true positive pixel detection rate (TPR), false positive pixel count (FPC), false positive pixel detection rate (FPR), true negative pixel count (TNC), true negative pixel detection rate (TNR), false negative pixel count (FNC), and false negative pixel detection rate (FNR) as presented in equation 3.14 hereinbelow.

$$MOTA\ (\%) = \{(Pt- Pf)/Pt\} * 100 \qquad (3.14)$$

where, *'Pt'* represents the total pixel count in the pixel sensitivity map, and *'Pf'* represents the count of falsely detected or non-detected pixels in pixel sensitivity map.

The total pixel count *'Pt'* in the pixel sensitivity map is represented by equation 3.15 as:

$$Pt = TPC + FPC + TNC +FNC \qquad (3.15)$$

and the count of falsely detected pixels *'Pf'* in the pixel sensitivity map is represented by equation 3.16 as:

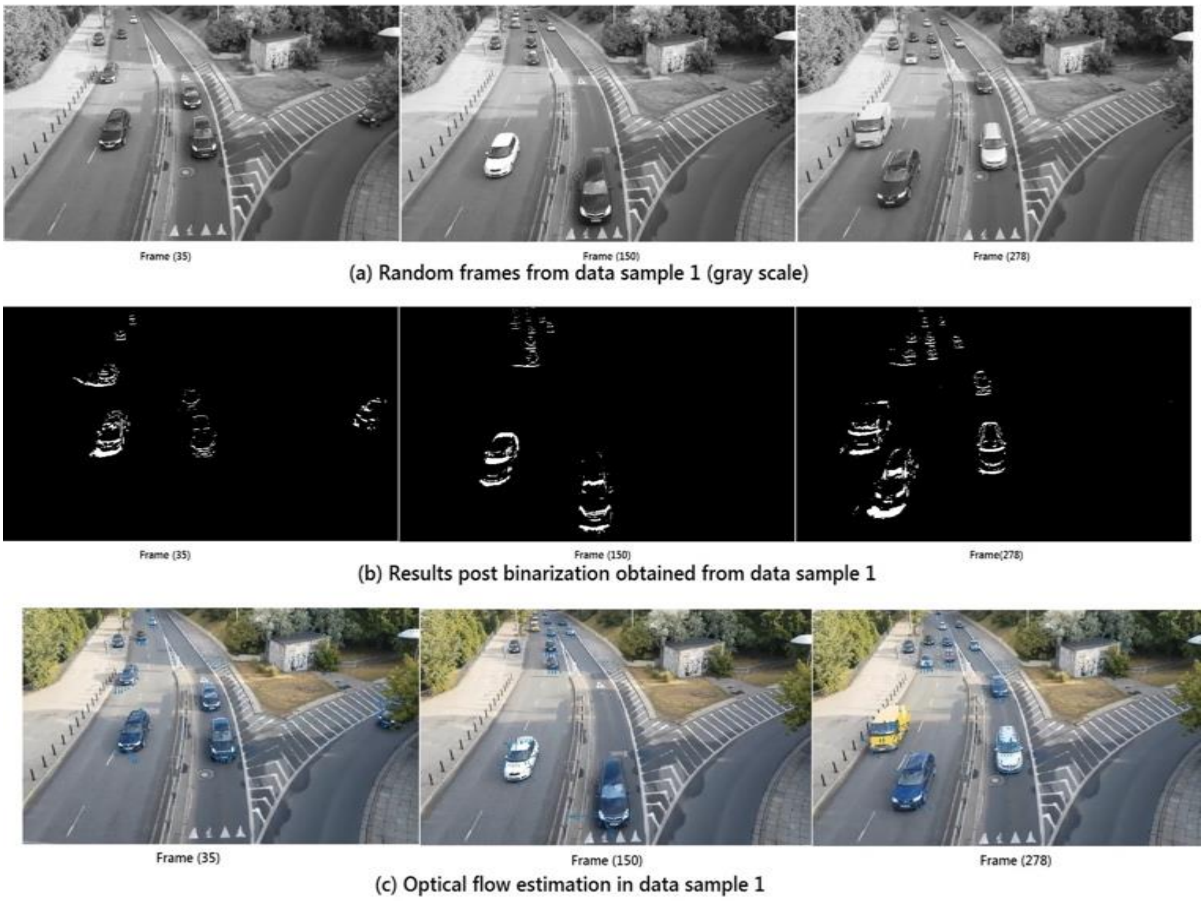
$$Pf = FPC + FNC \qquad (3.16)$$

Based on the MOTA (%), the efficiency of the SAM framework is compared with other state of art systems.

3.7 Simulations

For illustration of the effectiveness of the proposed SAM framework, the model is tested using various video datasets, where each video dataset mimics the images captured by a single stationary camera sensor capturing consecutive images over a predefined period of time. Specifically, the video datasets include surveillance datasets and sports datasets with specifications of 30 frames per second, resolution of 360x640 pixels per frame, and a duration of 10 seconds of each video dataset for multi-object detection and tracking application. As the model is demonstrated for a very short duration of video datasets, the framework is proposed to use HWHM Gaussian with normalized value of mean (μ) and standard deviation (α) of 1 and 0.5, respectively. The performance of the model based on SAM framework is compared with approaches in [165], [166], and [175] in terms of MOTA (%) for traffic flow estimation and multi-object tracking. The simulations and results are derived using MATLAB Image Processing Toolbox on a work station (GPU) with 128 GB of Random-access memory and Intel(R) Xeon(R) Silver 4214 CPU @ 2.19-2.20 GHz. Simulation results of randomly selected frames from the video datasets are shown hereinbelow. Specifically, the extracted frames, the initial foreground after adaptive background subtraction, and the federated optical flow is shown in Figure 3.3 for video data sample 1, Figure 3.5 for video data sample 2, Figure 3.7 for video data sample 3, Figure 3.9 for video data sample 4, Figure 3.11 for video data sample 5, and Figure 3.13 for video data sample 6, respectively. A comparison of activity map and normalized pixel sensitivity derived by different approaches is as shown in Figure 3.4 for video data sample 1, Figure 3.6 for video data sample 2, Figure 3.8 for video data sample 3, Figure 3.10 for video data sample 4, Figure 3.12 for video data sample 5, and Figure 3.14 for video data sample 6, respectively.

<u>Video data sample-1 (Traffic Surveillance dataset)</u>:



(a) Random frames from data sample 1 (gray scale)

(b) Results post binarization obtained from data sample 1

(c) Optical flow estimation in data sample 1

Figure 3.3 Simulation results from video data sample 1 using SAM framework.

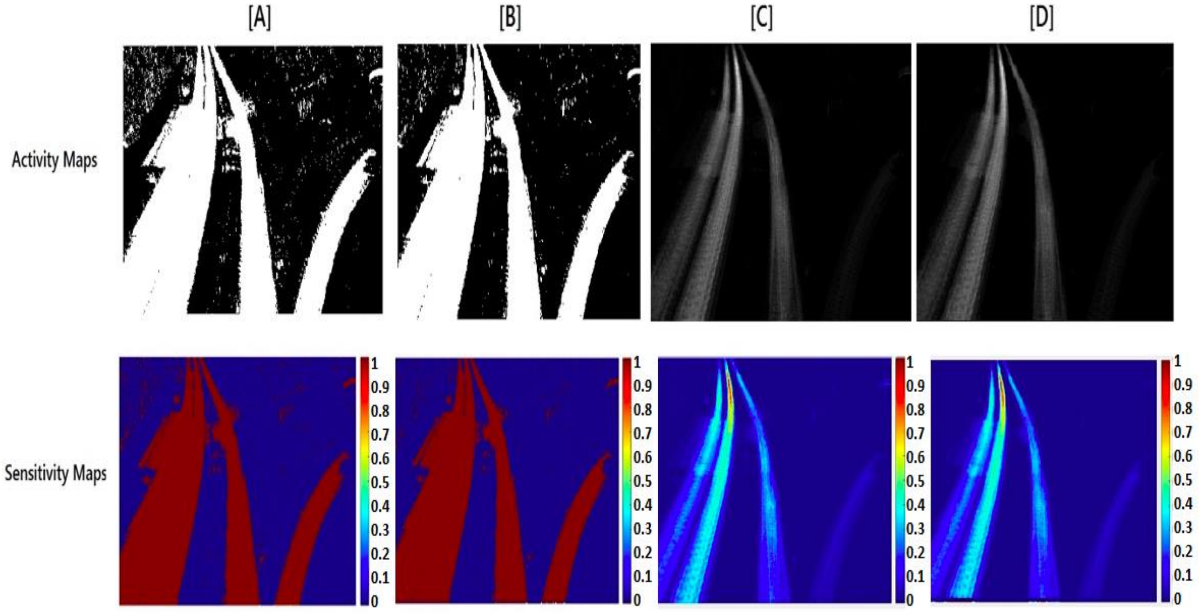
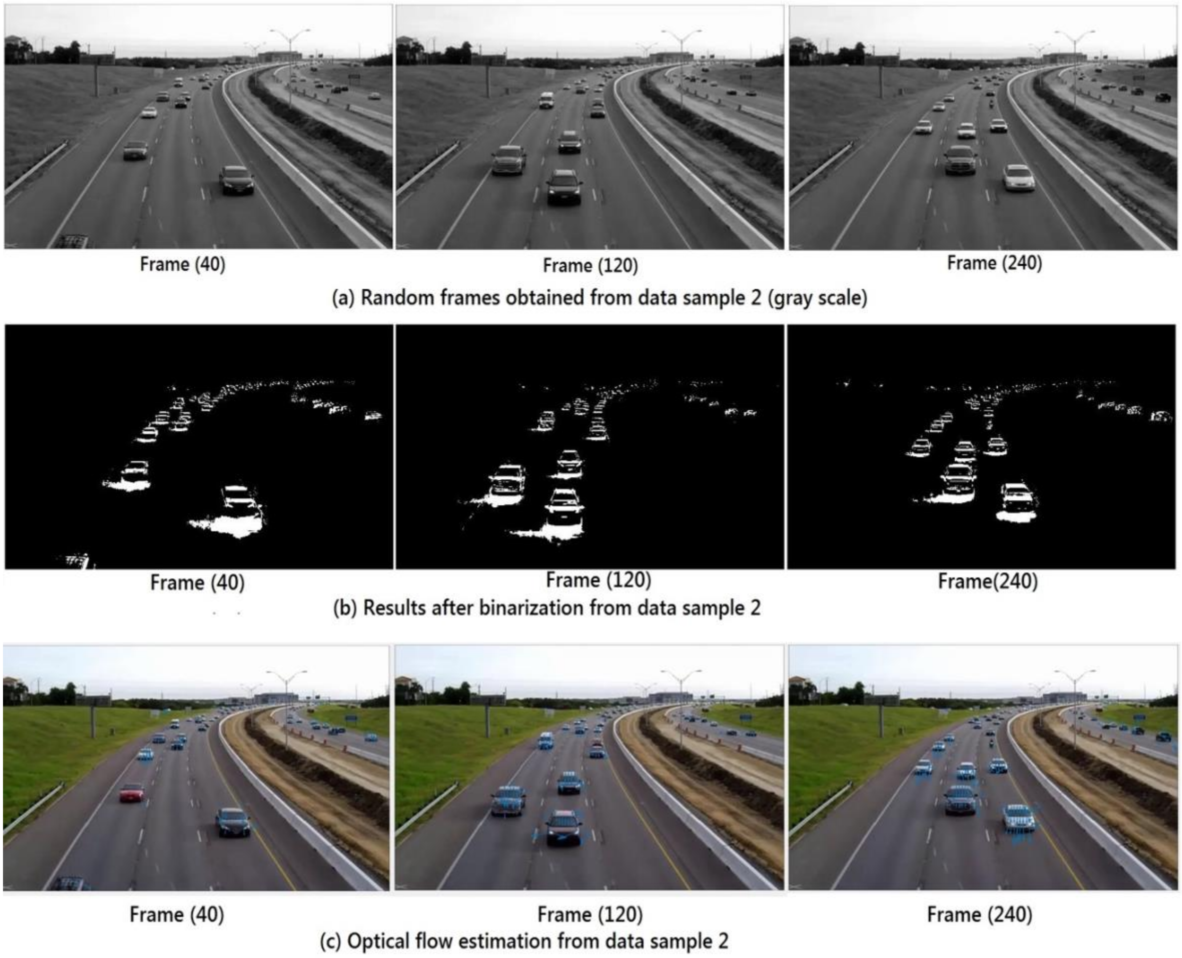


Figure 3.4 Activity maps and pixel sensitivity maps of video data sample 1 by different approaches; [A] by Pan et al. in [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model.

Video data sample-2 (Traffic Surveillance dataset):



(a) Random frames obtained from data sample 2 (gray scale)

(b) Results after binarization from data sample 2

(c) Optical flow estimation from data sample 2

Figure 3.5 Simulation results from video data sample 2 using SAM framework.

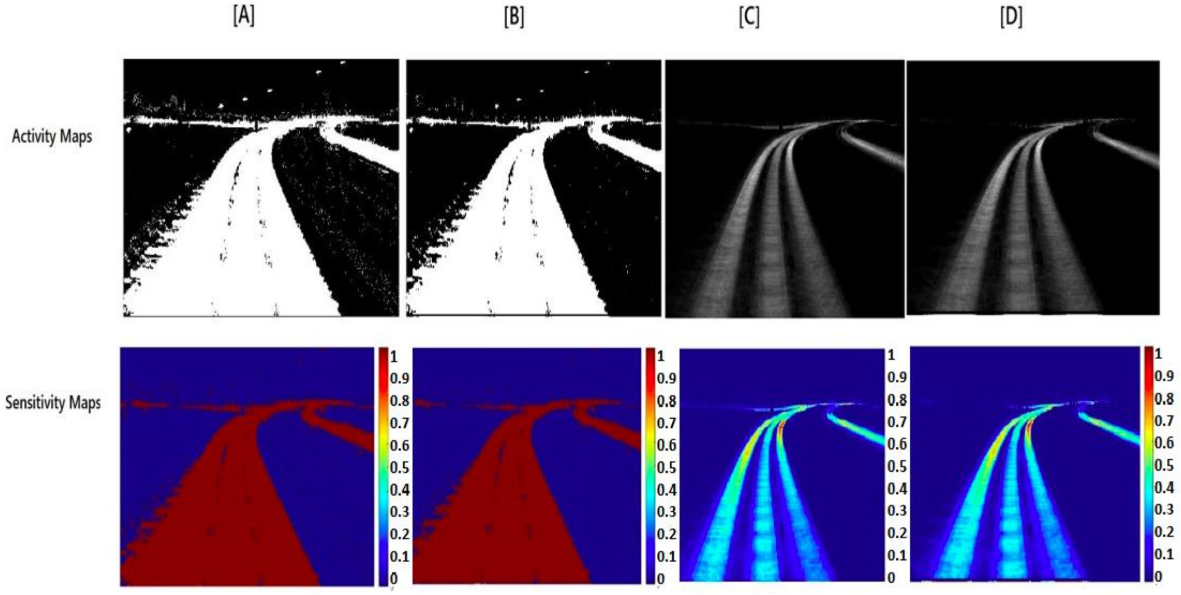
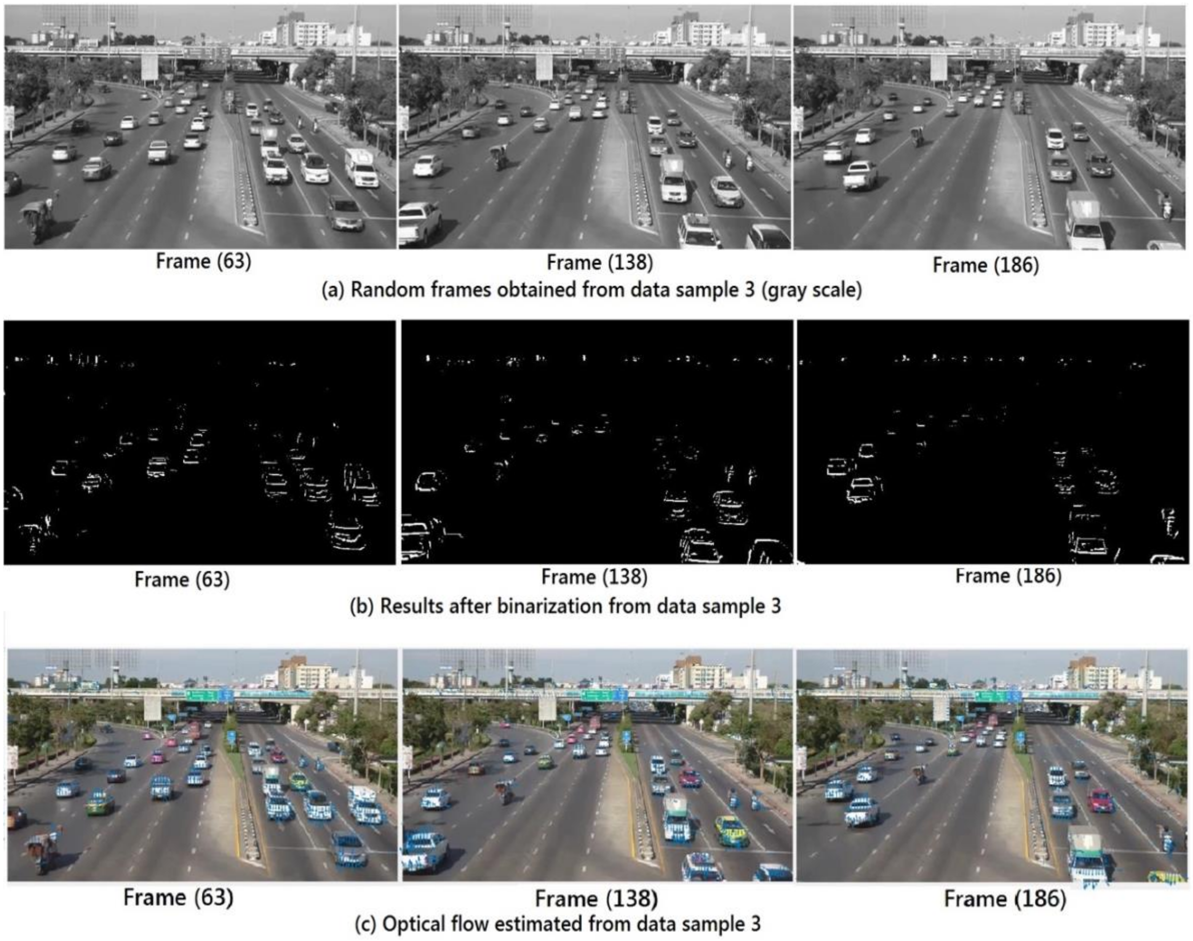


Figure 3.6 Activity maps and pixel sensitivity maps of video data sample 2 by different approaches; [A] by Pan et al. in [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model.

<u>Video data sample-3 (Traffic Surveillance dataset)</u>:



(a) Random frames obtained from data sample 3 (gray scale)

(b) Results after binarization from data sample 3

(c) Optical flow estimated from data sample 3

Figure 3.7 Simulation results from video data sample 3 using SAM framework.

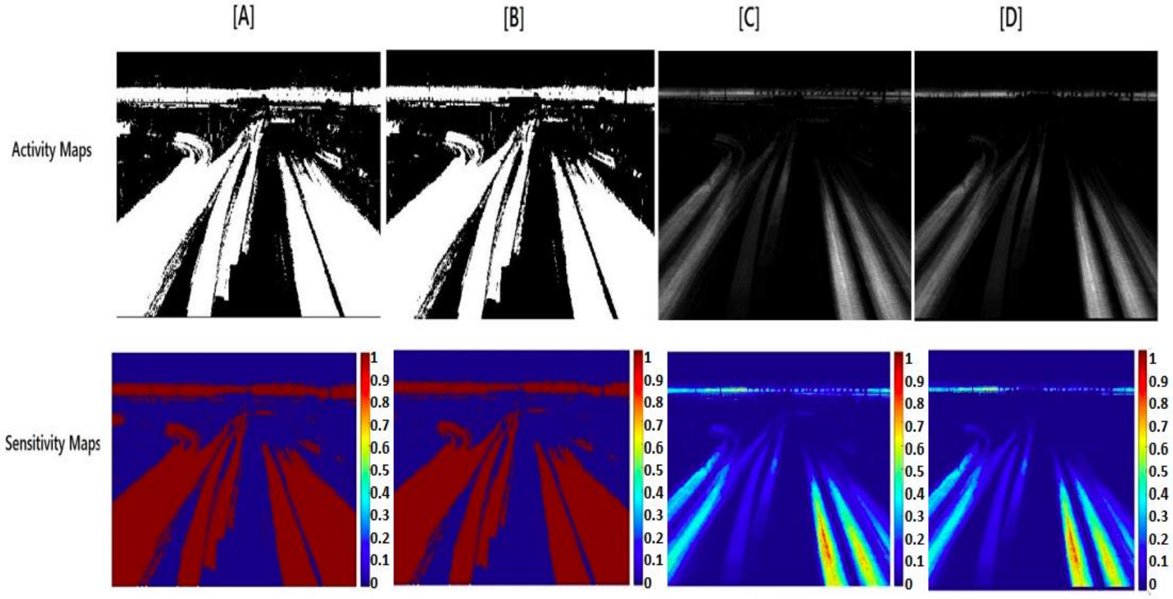
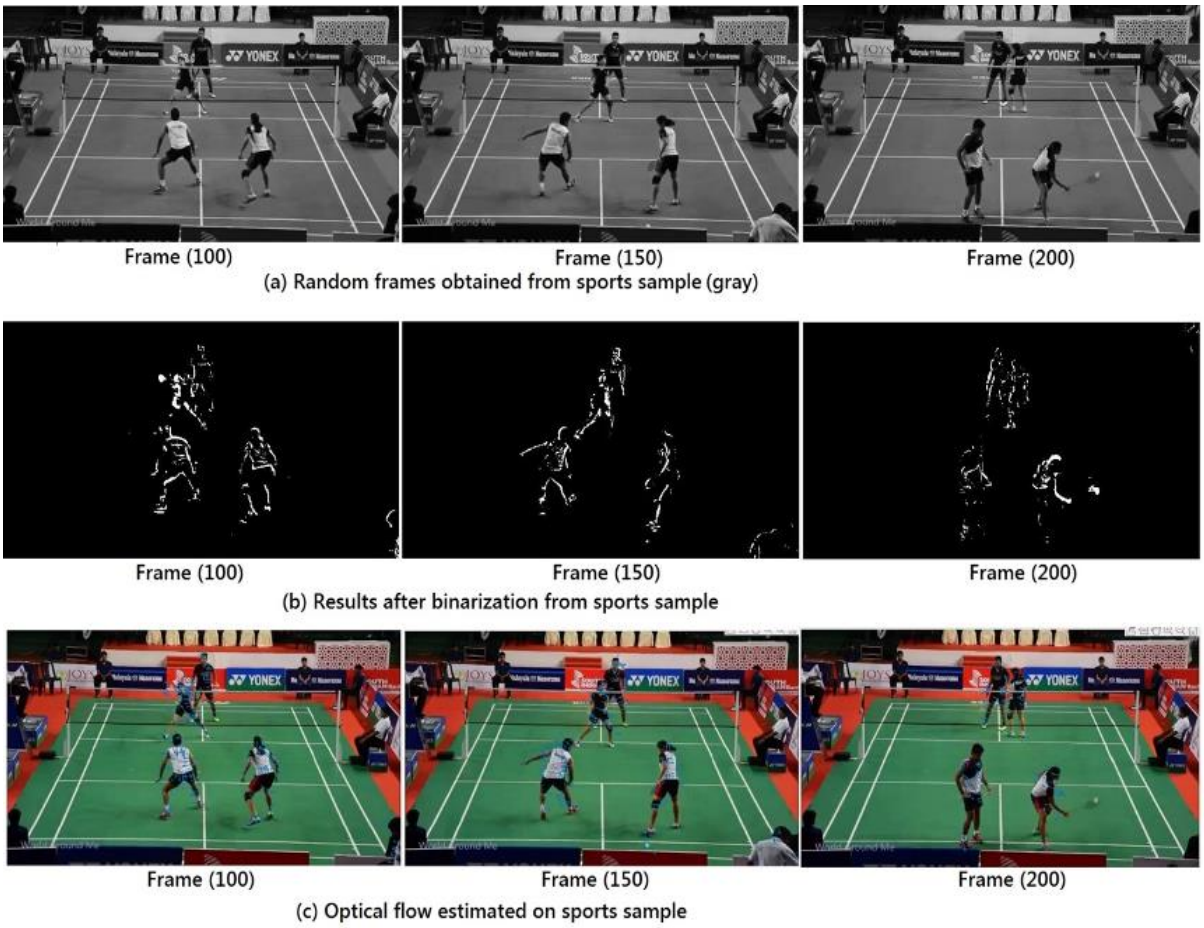


Figure 3.8 Activity maps and pixel sensitivity maps of video data sample 3 by different approaches; [A] by Pan et al. in [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model.

Video data sample-4 (Sports dataset - Badminton):



(a) Random frames obtained from sports sample (gray)

(b) Results after binarization from sports sample

(c) Optical flow estimated on sports sample

Figure 3.9 Simulation results from video data sample 4 using SAM framework.

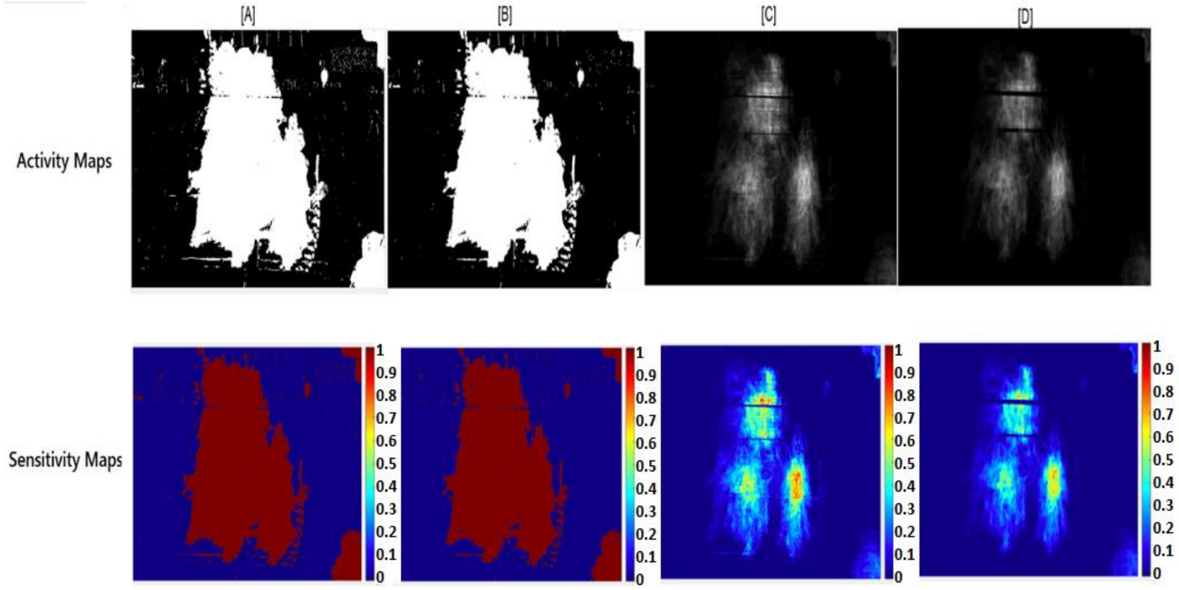


Figure 3.10 Activity maps and pixel sensitivity maps of video data sample 4 by different approaches; [A] by Pan et al. in [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model.

Video data sample-5 (Sports dataset - Sword fight):

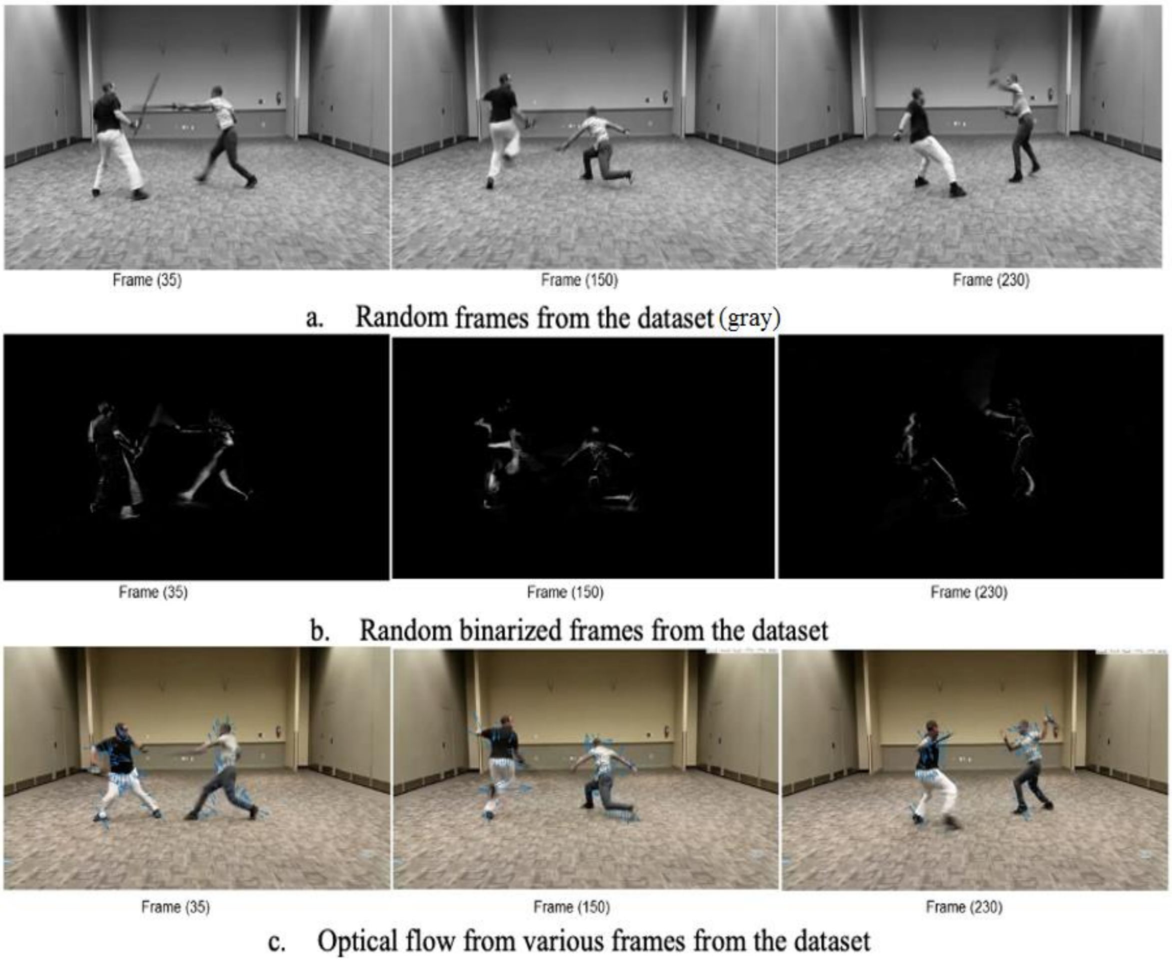

Frame (35)
Frame (150)
Frame (230)
a. Random frames from the dataset (gray)
Frame (35)
Frame (150)
Frame (230)
b. Random binarized frames from the dataset
Frame (35)
Frame (150)
Frame (230)
c. Optical flow from various frames from the dataset

Figure 3.11 Simulation results from video data sample 5 using SAM framework.

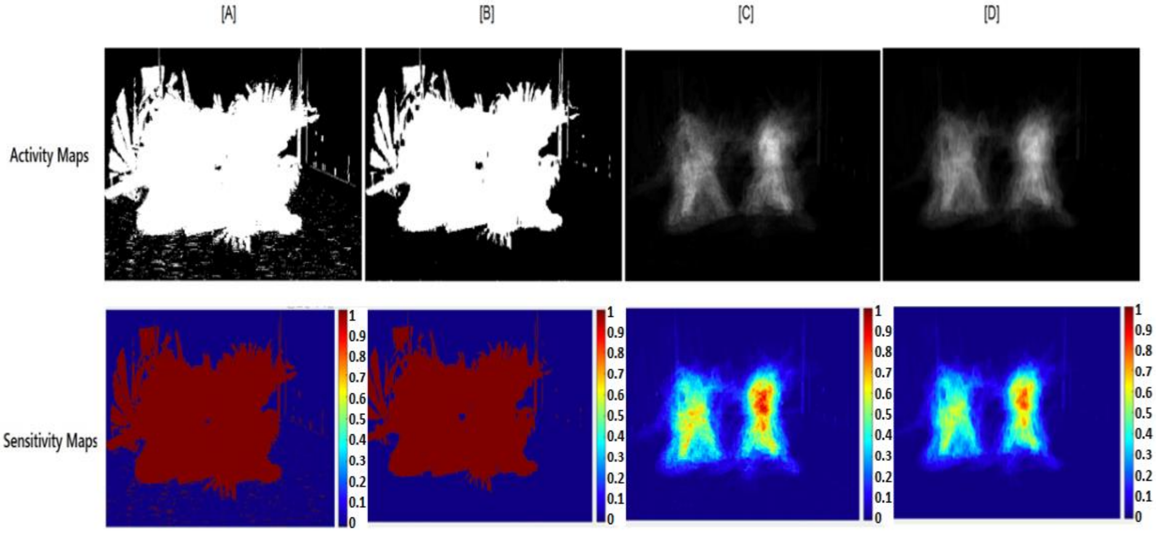
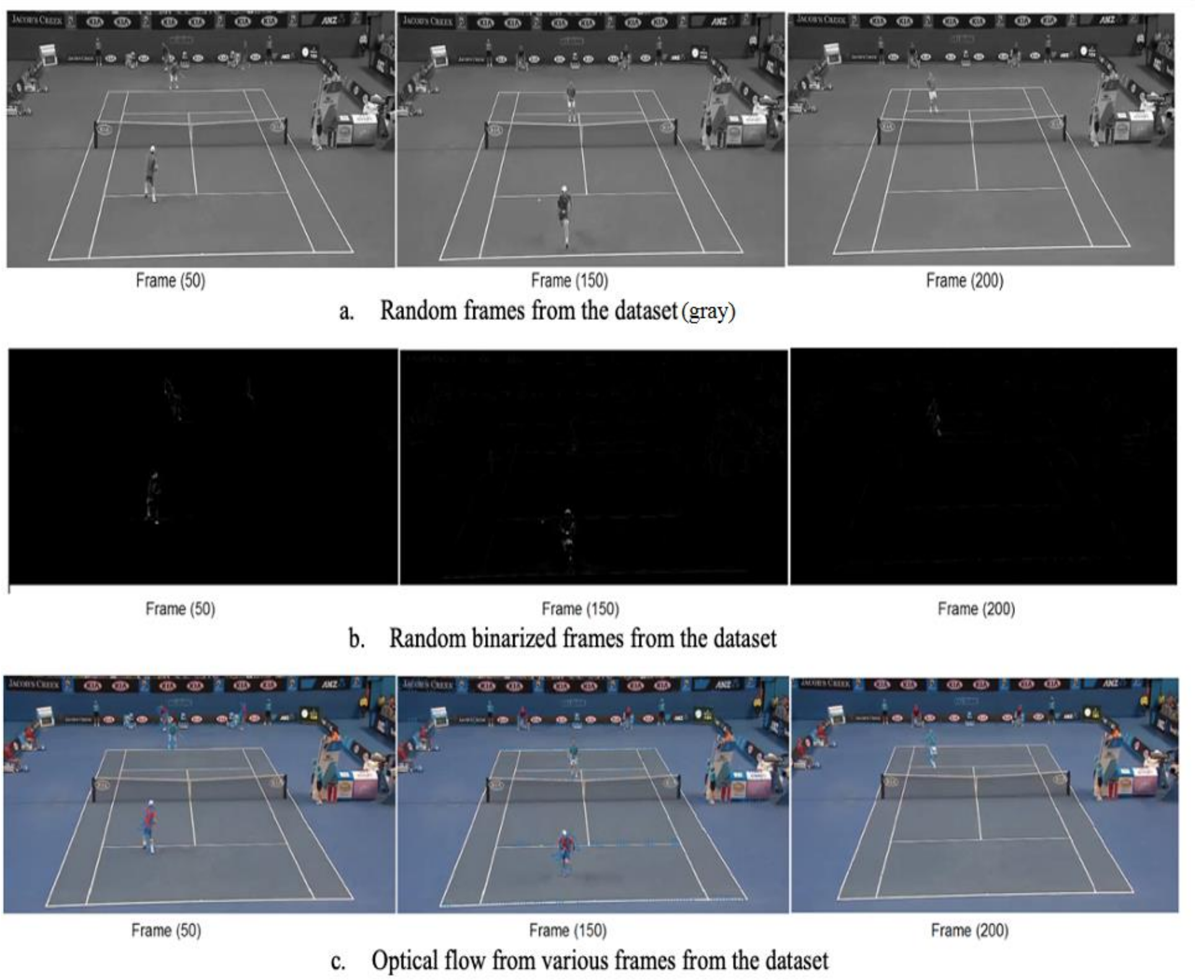


Figure 3.12 Activity maps and pixel sensitivity maps of video data sample 5 by different approaches; [A] by Pan et al. in [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model.

Video data sample-6 (Sports dataset - Tennis):



a. Random frames from the dataset (gray)

b. Random binarized frames from the dataset

c. Optical flow from various frames from the dataset

Figure 3.13 Simulation results from video data sample 6 using SAM framework

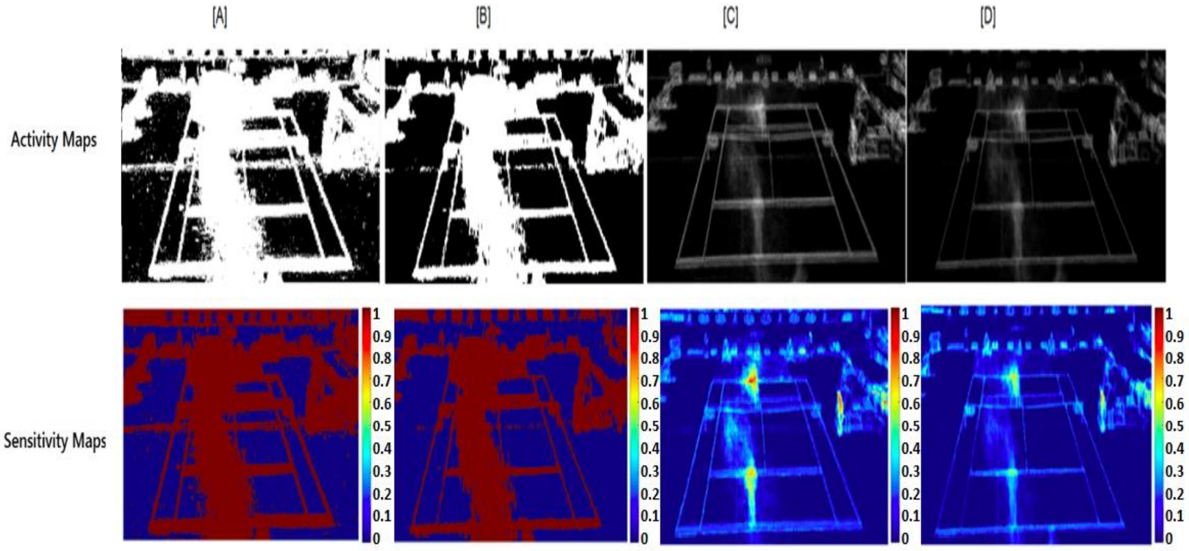


Figure 3.14 Activity maps and pixel sensitivity maps of video data sample 6 by different approaches; [A] by Pan et al. in [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model.

3.8 Results

A comparative performance analysis of different approaches proposed in [165], [166], [175], and the proposed SAM framework model in terms of performance parameters (*i.e.*, MOTA (%)) is presented in Table 3.1 hereinbelow.

Table 3.1 Comparison of performance parameters by different approaches tested on video data sample 1; [A] by [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model; and [E] is the true data obtained using markers.

Ref.	**TPC**	**TPR (%)**	**FPC**	**FPR (%)**	**TNC**	**TNR (%)**	**FNC**	**FNR (%)**	**MOTA (%)**
Video Data sample 1 (Traffic Surveillance):									
[A]	67,830	92	39,114	53.0	1,17,558	75.03	5898	3.76	43.15
[B]	66,245	89.85	32,761	44.4	1,23,911	79.09	7483	4.77	50.80
[C]	65,924	89.41	6,273	8.51	1,50,399	95.99	7804	4.98	86.51
[D]	65,138	88.34	1,071	0.14	1,55,591	99.31	8590	5.48	94.38
[E]	73,728	100	0	0	1,56,672	100	0	0	100

Table 3.2 Comparison of performance parameters by different approaches tested on video data sample 2; [A] by [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model; and [E] is the true data obtained using markers.

Ref.	**TPC**	**TPR (%)**	**FPC**	**FPR (%)**	**TNC**	**TNR (%)**	**FNC**	**FNR (%)**	**MOTA (%)**
Video Data sample 2 (Traffic Surveillance):									
[A]	83,262	94.50	43,149	48.97	99,146	69.67	4843	3.40	47.63
[B]	81,989	93.05	38,102	43.24	1,04,193	73.22	6116	4.29	52.46
[C]	80,594	91.47	8,122	9.21	1,34,173	94.29	7511	5.27	85.52
[D]	79,813	90.59	1622	0.18	1,40,673	98.86	8292	5.82	94.00
[E]	88,105	100	0	0	1,42,295	100	0	0	100

Table 3.3 Comparison of performance parameters by different approaches tested on video data sample 3; [A] by [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model; and [E] is the true data obtained using markers.

Ref.	**TPC**	**TPR (%)**	**FPC**	**FPR (%)**	**TNC**	**TNR (%)**	**FNC**	**FNR (%)**	**MOTA (%)**
Video Data sample 3 (Traffic Surveillance):									
[A]	1,06,274	96.15	41,827	37.84	78,051	65.11	4248	3.54	57.11
[B]	1,04,483	94.53	34,131	30.88	85,747	71.54	6039	5.03	64.09
[C]	1,02,173	92.44	9138	8.27	1,10,740	92.37	8349	6.96	84.77
[D]	1,00,628	91.04	1122	0.10	1,18,756	99.06	9894	8.25	91.65
[E]	1,10,522	100	0	0	1,19,878	100	0	0	100

Table 3.4 Comparison of performance parameters by different approaches tested on video data sample 4; [A] by [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model; and [E] is the true data obtained using markers.

Ref.	**TPC**	**TPR (%)**	**FPC**	**FPR (%)**	**TNC**	**TNR (%)**	**FNC**	**FNR (%)**	**MOTA (%)**
Video Data sample 4 (Traffic Surveillance):									
[A]	82107	95.35	26,187	30.41	1,18,105	81.85	4001	2.77	66.82
[B]	80926	93.98	19,223	22.32	1,25,069	86.68	5182	3.59	74.09
[C]	78446	91.10	5,982	6.94	1,38,310	95.8	7662	5.31	87.75
[D]	77102	89.54	2,321	2.69	1,41,971	98.39	9006	6.24	91.07
[E]	86108	100	0	0	1,44,292	100	0	0	100

Table 3.5 Comparison of performance parameters by different approaches tested on video data sample 5; [A] by [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model; and [E] is the true data obtained using markers.

Ref.	**TPC**	**TPR (%)**	**FPC**	**FPR (%)**	**TNC**	**TNR (%)**	**FNC**	**FNR (%)**	**MOTA (%)**
Video Data sample 5 (Sports dataset- Sword fight):									
[A]	66,121	91.87	23, 877	33.17	1,28,547	81.14	5,885	3.71	63.12
[B]	63,964	88.86	21, 232	29.49	1,37,192	86.59	8,012	5.05	65.46
[C]	58,372	81.10	12,121	16.84	146303	92.35	13,604	8.58	74.58
[D]	54, 232	75.34	8,962	12.45	1,49462	94.32	17744	11.2	76.35
[E]	71,976	100	0	0	158424	100	0	0	100

Table 3.6 Comparison of performance parameters by different approaches tested on video data sample 6; [A] by [165]; [B]. by Mehboob et al. in [166]; [C]. by Indu, S. in [175]; [D] is our proposed approach using SAM framework model; and [E] is the true data obtained using markers.

Ref.	**TPC**	**TPR (%)**	**FPC**	**FPR (%)**	**TNC**	**TNR (%)**	**FNC**	**FNR (%)**	**MOTA (%)**
Video Data sample 6 (Sports dataset- Tennis):									
[A]	46,185	94.34	20,863	42.61	1,60,602	88.50	2,768	1.52	55.87
[B]	43,266	88.38	18,286	37.35	1,63,179	89.92	5,687	3.13	59.52
[C]	38,128	77.89	12,112	24.74	1,69,353	93.32	10,825	5.97	69.29
[D]	37,109	75.81	8,934	18.25	1,72,531	95.08	11,844	6.52	75.23
[E]	48,953	100	0	0	1,81,465	100	0	0	100

The average performance of the proposed SAM framework is compared with systems proposed in [176] and [177] through video data sample 1 to video data sample 3 for traffic surveillance applications, and RGB sequence as presented in [178] with data sample 4 to data sample 6 for sports analytics application. The comparison of performance of the SAM framework model with [176], [177] and [178] is presented in Figure 3.15.

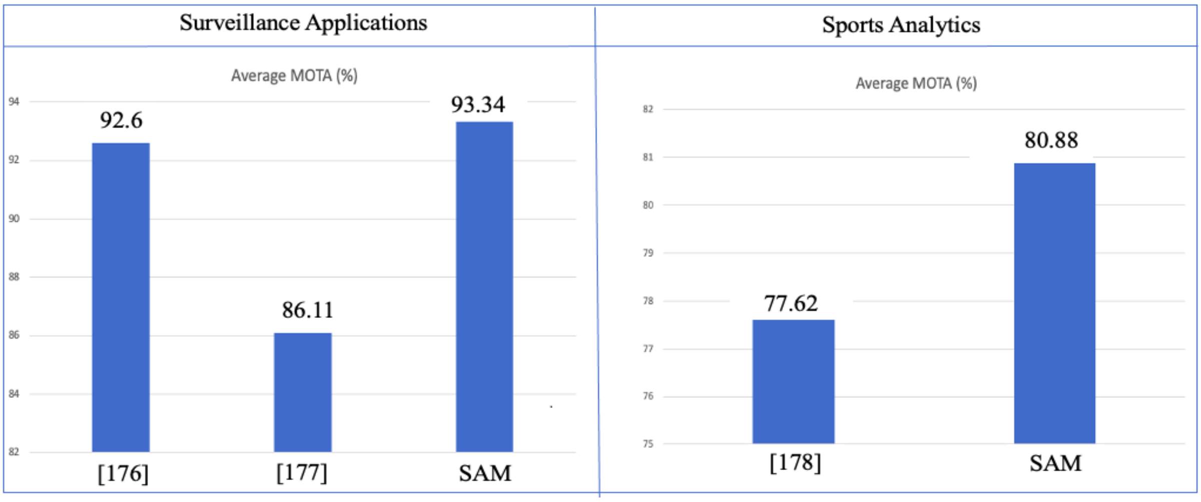


Figure 3.15 Comparison of SAM performance with [176], [177] and [178] (in average MOTA %).

3.9 Conclusion

Performance of a successful active vision system depends on the accuracy of images captured by the camera sensor, a spatiotemporal understanding of the scene, and data processing capabilities of the system. Advanced active vision systems typically use highly complex computation model to address the aforementioned issues, which may be problematic for systems with constrained resources. The SAM framework addresses the challenge of spatiotemporal activity mapping that can balance the performance of active vision system with limited resource availability.

The SAM framework model analyses the scene in both spatial and temporal perspectives, and generates adaptive activity maps for sensor reconfiguration so that the important region(s) can be captured near to the center of the camera sensor's field of view. The framework pre-processes the sensor data using simplistic image processing techniques like background subtraction, binarization, thresholding, and federated optical flow. The temporal relationship between the consecutive image frames is represented by a half-width Gaussian distribution. The proposed SAM framework's straightforward model yields highly accurate spatiotemporal activity mapping with low computation complexity and load, and thus requires low system resources. The performance is compared in terms of MOTA (%), where the SAM framework model outperforms contemporary systems presented in [165], [166], [175], [176], [177] and [178]. Specifically, the SAM framework model showcases 0.79% better average MOTA relative to [176] and 8.39% better average MOTA relative to [177], when tested on traffic surveillance datasets (*i.e.*, data samples 1, 2 and 3). The SAM framework model further showcases 4.21% better average MOTA as compared to [178], when tested on sports datasets (*i.e.*, data samples 4, 5 and 6). The development of SAM framework model resulted in two research publications cited as [175] and [179].

Chapter 4

ADAPTIVE SELF-RECONFIGURATION

Computer vision has seen tremendous advancement in technology and an exponential increase in the number of used case applications in the last decade. The growth in future leads to development and deployment of futuristic, advanced, and adaptive computer vision applications such as completely automated driverless vehicles, tele-immersion, advanced sports analysis, and the like. Most of the advanced computer vision applications require performing highly complex computations in real time. Presently, such capabilities are only confined to large datacentres which are not accessible to all, due to humongous expenses and unavailability for specific applications.

As most of the advanced computer vision technologies use artificial intelligence (in one form or the other), it is nearly impossible to train such systems in near real time for any unforeseen condition, even after extensive exploitation of datacenter resources. Thus, there is a need to develop an adaptive system that is capable of data and information sharing such that the training latency can be reduced. The contemporary art presents systems based on transfer learning to facilitate an adaptive trait to the system, however such systems are also far away to tackle unforeseen conditions without any prior training experience in near real time. To address the aforementioned problem, an Adaptive Self-Reconfiguration (AdapSR) framework is presented in this chapter.

4.1 Self-Reconfiguration

Self-Reconfiguration is a property of a system to understand a situation of operation and re-configure its performance parameters to optimize its performance. To achieve a state of self-reconfiguration, the system must be capable of understanding the objectives of operation, derive a relationship of reconfiguration parameters and their effects on the performance of the system. Specifically, a self-reconfigurable active vision system requires determining a relationship between calibration parameters of the camera sensors and re-configuration of the calibration parameters to capture better scene yielding to an improved understanding of activities from the scene. More particularly, if an active vision system employs a network of camera sensors (*i.e.*, a smart camera network), to possess self-reconfiguration the active vision system must be capable of understanding the effects of dynamic changes and unforeseen situations and/or activities to calibrate the configuration space of each camera sensor deployed in the network. The ability to adapt to such changes and situations, and reconfigure performance parameters for optimized overall performance of the system is known as self-adaptation [180].

Various approaches have been proposed in the past for self-adaptation of computer vision systems. Leong et al. in [181] proposed self-reconfiguration of parameters of unmanned aerial vehicles used for surveillance. Self-reconfiguration of distributed smart camera network for vehicle re-identification is proposed by Martinel et al. in [182] using deep learning models. Various models are compared by Nataranjan et al. in [183] for self-reconfiguration of computer vision systems for data extraction through active camera nodes of a computer vision system. However, for an efficient and accurate self-reconfiguration, the active vision system must be capable of identification of its state, its performance parameters, and most critically it must be capable of deriving a relationship between the performance parameters and the outcome of calibration of the parameters, which is found missing in the state-of-art. Further, an optimized self-reconfiguration of an active vision system demands self-adaptation. Figure 4.1 depicts a dynamic reconfiguration framework for SCNs proposed by Piciarelli et al. in [8].

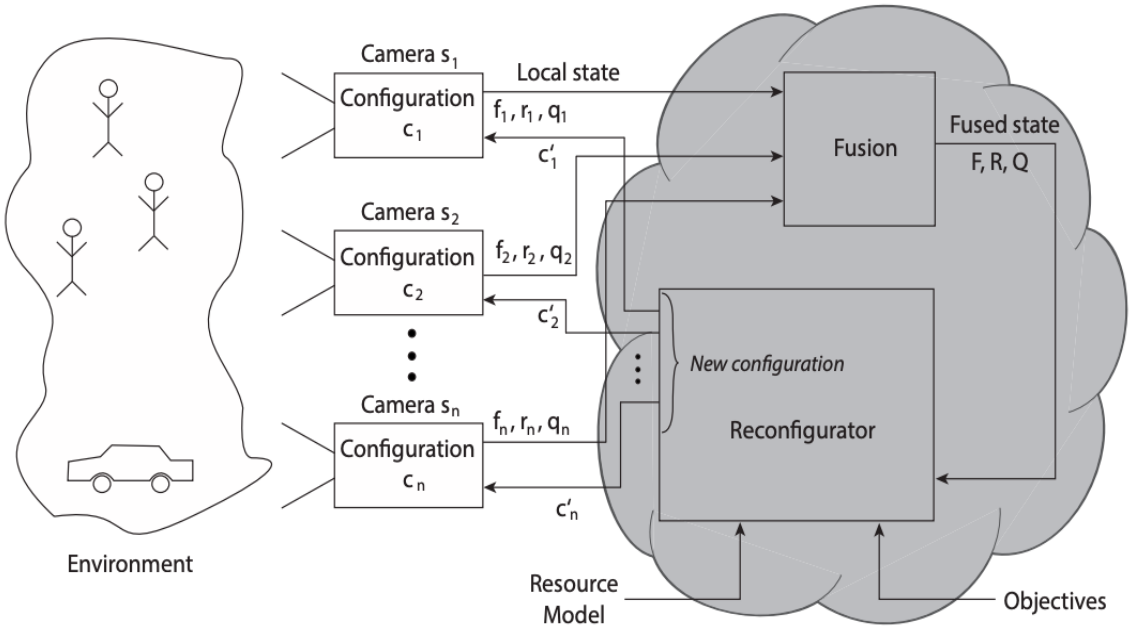


Figure 4.1 Dynamic Self-reconfiguration framework [8]

The dynamic reconfiguration framework of [8] used a Smart Camera Network to determine an overall state (F), overall quality (Q), and overall resource utilization (R) of the system from set of data corresponding to node's local states (f), local QoS of nodes (q) and resource utilization of the nodes (r), respectively. Piciarelli et al. in [8] through the dynamic reconfiguration framework further demonstrated the capabilities of a re-configurator to identify changes in parameters in accordance with a resource model and the system's goals. However, as the dynamic reconfiguration framework in [8] was developed for a centralized environment, the framework failed to tackle unforeseen conditions with utmost efficiency. To tackle an unforeseen condition that is not identified within the smart camera network, the framework had to completely determine the reconfiguration parameters from the scratch, which makes it unfavorable to tackle unforeseen conditions in near real-time, and thus limits the operational applications of the system. The dynamic reconfiguration framework proposed by Piciarelli et al. in [8] was later improved by Rudolf et al. in [190], Cai et al. in [191], and Suresh et al. in [192] for adaptive self-reconfiguration of smart camera networks, however, it also lacked adaptiveness.

4.2 Self-Adaptation

Self-adaptation is referred to as the ability of a system to learn and adapt to dynamic changes in its state of performance, and update its performance parameters accordingly through information and knowledge received from other systems. In contrast to self-reconfiguration, self-adaptation facilitates a system to enhance its performance through updating configuration space of nodes, active participants, model parameters, and algorithms shared by other systems in a network. Self-adaptation further enables a system to provide a best configuration space suitable to tackle a particular condition that may be unforeseen to another system, thus provides an established pretrained model for enhanced performance in terms of efficiency and timeliness of computation.

To achieve self-adaptation, a system must possess two fundamental properties: self-expression and self-awareness. Self-expression is an ability of a system to determine local states of each component deployed in the system towards the overall state in terms of the Quality of Service (QoS) delivered by the system (computed by way of the performance parameters of the system). Self-expression further enables the system to share the state parameters corresponding to its local and overall state, the objectives of operation and associated data and/or metadata to other systems associated with the system. Furthermore, self-expression enables the system to receive feedbacks from other systems based on the information shared by the system. Self-awareness is an ability of the system to question its local and overall operational states, and adjust its configuration space based on the feedback from other systems that is best suitable to tackle the unforeseen condition.

Rinner et al. in [181] proposed six major steps for self-adaptation based on a market-based approach to self-awareness: resource monitoring, object tracking, topology learning, object handover, strategy selection, and objective formation. In [182], Lewis et al. distinguished between explicit and implicit events and discussed the privacy, scope, and quality of self-adaptation. Lewis and Chandra in [183] discussed formal models for self-adaptation and their applications in Artificial Intelligence systems, conceptual systems, engineering, automotive systems, computing, and the like. Wang et al. in [184] discussed self-adaptation methods with online learning capabilities. Ali et al. in [185] proposed an auto-adaptive multi-stream architecture with multiple heterogeneous sensors and pipelined switches between processing states and ideal states to reduce power utilising a Field Programmable Gate Array (FPGA) implementation that demonstrated inter-frame adaptation capability with a relatively low overhead. In [186], Guettalfi et al. proposed an architecture for public and private self-awareness using actuators that incorporate QoS, resource estimation, a feedback mechanism, and state estimation. Zhu et al. in [187] and Lin et al. in [188] proposed unsupervised learning based self-adaptation for person re-identification. Wu et al. in [189] and Rudolph et al. in [190] proposed adaptive self-reconfiguration systems for computer vision applications based on sharing of information amongst nodes that are internally deployed in the smart camera network. However, both systems being designed for centralized systems limit the knowledge sharing to internal nodes of the camera network, and thus are confined to limited learning and knowledge sharing within the smart camera network. Further, the adaptive self-reconfiguration systems proposed in [189] and [190] were also prone to visual attacks if AI/ML models are utilized for operation. Rudolf et al. in [190], Cai et al. in [191], and Suresh et al. in [192] improved the dynamic reconfiguration framework proposed by Piciarelli et al. in [8] for adaptive self-reconfiguration of smart camera networks to enhance the detection accuracy, however, the operation in centralized network environment limits their adaptiveness and thus capabilities to tackle unforeseen conditions in near real-time.

4.3 Adaptive Self-Reconfiguration Framework

From the aforementioned, it is well established that the scene understanding and the reconfiguration of calibration parameters of camera sensors are interdependent, which makes it very challenging for a system to determine an accurate configuration space to derive optimized scene understanding for completely unknown environments in near real-time. With some preliminary understanding of an event, the latency and efficiency of complex computations used for deriving scene understanding can be highly improved.

To address the abovementioned challenge, The Adaptive Self-reconfiguration (AdapSR) framework is specifically designed to extend the scope of the dynamic re-configuration framework by Piciarelli et al. in [8] for a distributed network of systems. It is a well-known fact that sharing information is simpler and less exerting than deriving information, and thus requires less resources. The AdapSR framework mimics human learning and knowledge sharing based on past experiences to a distributed network of systems, and thus facilitates systems with a self-adaptive capability to tackle unforeseen situations more efficiently than those solutions operating in a centralized environment.

More particularly, the AdapSR framework utilizes data and information sharing among a distributed network of computer vision systems. In situations when an unforeseen activity is detected by a system, the AdapSR framework enables determining a similar activity that was analysed by any other system in the distributed network in the past and fine tuning its information and model to tackle the unforeseen conditions in near real-time without extensively exploiting the resources of datacenters. Thus, the AdapSR framework provides an efficient solution for efficient and adaptive active vision systems to tackle unforeseen conditions in near real time.

4.4 Model

As discussed earlier, the AdapSR framework is specifically designed to facilitate a plurality of smart camera networks with an adaptive self-reconfiguration capability to enable calibration of configuration space of each camera sensor network efficiently. The AdapSR model architecture is as shown in Figure 4.2 hereinbelow.

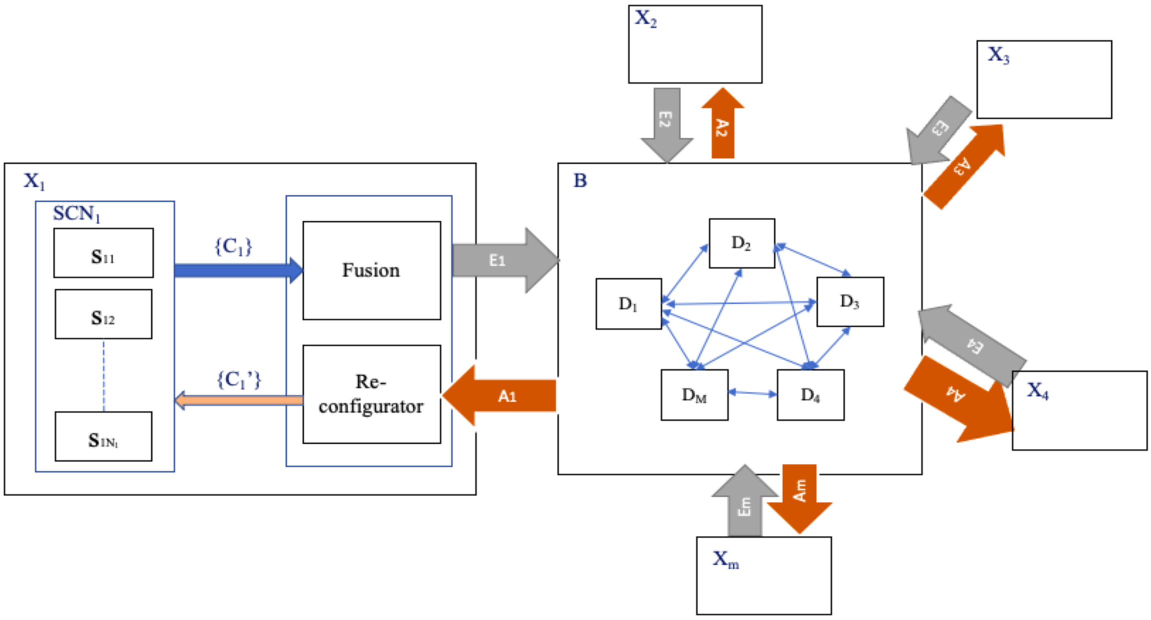


Figure 4.2 AdapSR architecture for active vision systems

Notations used:

X_i: i^{th} active vision system;

SCN_i: i^{th} smart camera network

s_{ij}: j^{th} camera sensor of the i^{th} smart camera network;

N_i: Number of camera sensors in the i^{th} smart camera network;

$\{C_i\}$: Set of input data for the i^{th} smart camera network;

$\{C_{i'}\}$: Set of output data for the i^{th} smart camera network;

E_i: Self-expression data for the ith active vision system;

A_i: Self-awareness data for the ith active vision system;

m: Number of active vision systems utilizing the AdapSR framework;

B: Distributed blockchain network comprising “M” number of datacenters; and

D_k: kth datacenter in the distributed blockchain network.

Specifically, the development of an operational model based on the AdapSR framework requires a plurality of active vision systems and a plurality of processing nodes co-operatively coupled to each other in a distributed network. More particularly, the plurality of processing nodes are datacenters coupled to each other in a distributed blockchain network. The functionality of an active vision system is defined by the objectives of the active vision system. Each active vision system of the plurality of active vision systems deployed in the AdapSR framework model includes a smart camera network that includes a plurality of camera sensors, an information fusion unit, and a reconfiguration unit. Each camera sensor of the plurality of camera sensors is configured to capture images of an environment to be analysed. Each smart camera network is configured to determine local state of each camera sensor in terms of local configuration space, values of the calibration parameters, and the amount of the resources utilized and left with each camera sensor. Each smart camera network is further configured to determine an overall state of the active vision system based on the local states of each camera sensor and the functionality of the active vision system. The information fusion unit associated with each active vision system is configured to attach a timestamp and a camera identifier to each image for identification. Further, the information fusion unit combines the timestamped images with camera identifiers to generate an image data for each active vision system. Furthermore, the information fusion unit is further configured to fuse the image data with the system’s state to generate a self-expression information corresponding to the active vision system.

The reconfiguration unit of each active vision system is configured to receive a self-awareness information from the plurality of processing nodes deployed in the distributed network of processing nodes. The reconfiguration unit is further configured to determine reconfiguration parameters from the self-awareness information and calibrate the configuration space of each camera sensor of the smart camera network based on the self-awareness information.

The plurality of processing nodes is configured to receive the self-expression information from each active vision system. The plurality of processing nodes is further configured to identify the functionality of each active vision system from the self-expression information, and segregate the active vision systems into a number of groups based on their respective functionalities.

Upon segregation of the active vision systems, the plurality of processing nodes is configured to determine a spatiotemporal sensitivity map for each active vision system based on the consecutive timestamped images from the self-expression information (as presented in Chapter 3). Furthermore, the plurality of processing nodes is configured to determine a QoS of each active vision system based on analysis of the spatiotemporal sensitivity map of each active vision system. The plurality of processing units further includes a distributed ledger to store models, sensitivity maps and configuration space associated with an activity identified by any active vision system of the plurality of activity systems corresponding to a particular functionality of the active vision system. The plurality of processing units is further configured match the spatiotemporal sensitivity map with the sensitivity maps derived from past experiences of the plurality of active vision systems in the distributed ledger based on the functionality of the active vision system to identify a best suitable sensitivity map for the active vision system and send the self-expression information to the active vision system including the configuration space of associated with the best suitable sensitivity map.

4.5 Process

The AdapSR framework provides calibration of configuration space of the camera sensors deployed in each active vision system of the plurality of active vision systems based on the self-expression information generated by the plurality of processing nodes. In operation, the AdapSR model, by way of each active vision system, further generates a self-expression information and sends the self-expression information to the plurality of processing nodes. Further, the AdapSR model, by way of the plurality of nodes segregates the plurality of active vision systems based on the functionality of each active vision system. Furthermore, the AdapSR model, by way of the plurality of processing nodes generates a spatiotemporal sensitivity map for each active vision system, and determine a QoS of each active vision system based on analysis of the corresponding spatiotemporal sensitivity map. Furthermore, the AdapSR model, by way of the plurality of processing nodes identifies a best suitable sensitivity map for the active vision system and send the self-expression information to the active vision system including the configuration space of associated with the best suitable sensitivity map.

Specifically, to evaluate the performance of the AdapSR framework, Region Proposal Network model (*i.e.*, a faster R-CNN model) presented by Ren et al. in [193] is used for segmentation and comparison of spatiotemporal sensitivity maps. It must be apparent to a person skilled in the art that the Region Proposal Network model is used just for illustration of the effectiveness of AdapSR framework as compared to the centralized reconfiguration systems, and the functionality of the AdapSR framework is not limited to it. The AdapSR framework can be utilized with any kind of segregation and comparison model without deviating from its scope. More particularly, the Region Proposal Network model (*i.e.*, a faster R-CNN model) presented by Ren et al. in [193] provides determination of a number of regions of interest (ROI) from the spatiotemporal sensitivity maps. Each region of interest is then classified by the Convolution Neural Network (CNN) classifier to identify a class corresponding to each Region of Interest detected of the number of Regions of Interest. The AdapSR model further generates a matching score by comparing each Region of Interest with the Regions of Interest of the predefined sensitivity maps stored in the distributed ledger to generate a matching score. Based on the matching score, one or more sensitivity map and their corresponding configuration parameters are determined matching the spatiotemporal sensitivity map of the active vision system that are used to generate self-expression information for the active vision system using CNN of the Proposal Network model.

The sensitivity map from the distributed ledger is selected for an active vision when the matching score is greater than a threshold value, that is determined using mean thresholding (as presented in Chapter 3) of all the activities determined by the plurality of processing nodes, and thus provides an adaptive improvement of the protocols over a period of deployment time. Further, the AdapSR model selects the sensitivity map with highest matching score above the mean threshold value to be most suitable for the active vision system. In a scenario, when none of the sensitivity maps has a matching score higher than the threshold value, the AdapSR framework proposes segregating the plurality of processing nodes into two categories: processor nodes and validator nodes. The processor nodes generate a set of protocols for determination of the configuration model for the active vision system using CNN and the validator nodes are configured to validate the configuration parameters based on update in the Quality of Service of the active vision system received in the form of the self-awareness information provided by the active vision system. Specifically, the AdapSR framework model identifies the center of Regions of Interest and determines its deviation from the centre of the spatiotemporal activity map. Further, based on the calibration parameters and the functionality of the active vision system from the self-expression information, the AdapSR framework model identifies an updated configuration space for the active vision system. For a seamless and unbiased operation of the plurality of processing nodes (*i.e.*, the processor nodes and the validator nodes), the plurality of processing nodes are categorized based on proof of active participation (POAP) consensus mechanism as presented in [194], and the validator nodes are rewarded based on proof of stake (POS) consensus mechanism as presented in [195]. Due to distribution of the task load amongst the processor and validator nodes, it is possible to tackle the unforeseen situation in near real-time, when encountered for the first time by the entire system of smart camera networks.

4.6 Performance Parameters

The performance of the proposed AdapSR framework is evaluated in terms of multi-object tracking accuracy (MOTA) *i.e.*, the accuracy of detection of Regions of Interest in the spatiotemporal sensitivity maps determined over specific periods of time (*e.g.*, epochs) that represents the rate of learning of the AdapSR framework with respect to other contemporary systems. Specifically, the MOTA (%) depends on true positive pixel count (TPC), true positive pixel detection rate (TPR), false positive pixel count (FPC), false positive pixel detection rate (FPR), true negative pixel count (TNC), true negative pixel detection rate (TNR), false negative pixel count (FNC), and false negative pixel detection rate (FNR) as presented in equation 4.1 hereinbelow.

$$MOTA\ (\%) = \{(Pt- Pf)/Pt\} * 100 \tag{4.1}$$

where, Pt represents the total pixel count in the spatiotemporal sensitivity map; and *'Pf'* represents the count of falsely detected or non-detected pixels in the spatiotemporal sensitivity map.

The total pixel count *'Pt'* in the spatiotemporal sensitivity map is represented by equation 4.2 as:

$$Pt = TPC + FPC + TNC +FNC \tag{4.2}$$

and the count of falsely detected pixels *'Pf'* in the spatiotemporal sensitivity map is represented by equation 4.3 as:

$$Pf = FPC + FNC \tag{4.3}$$

Based on the progressive MOTA (%) through training over epochs, the efficiency of the SAM framework is compared with other state of art systems.

4.7 Results

For illustration of the effectiveness of the proposed AdapSR framework, the model is tested using various video datasets, where each video dataset mimics the images captured by a single stationary camera sensor capturing consecutive images over a predefined period of time. Specifically, the video datasets include surveillance datasets of 30 frames per second (*i.e.*, data sample 1 and 2 as presented earlier in Chapter 3), resolution of 360x640 pixels per frame, and a duration of 10 seconds of each video dataset for multi-object detection and tracking application. To evaluate the performance of the AdapSR framework, Region Proposal Network model (*i.e.*, a faster R-CNN model) presented by Ren et al. in [193] is used for segmentation and comparison of spatiotemporal sensitivity maps. Due to the lack of resources required to develop a private blockchain network with datacenters, we used lower processing capabilities. The simulations and results are derived using MATLAB Image Processing Toolbox on a work station (GPU) with 128 GB of Random-access memory and Intel(R) Xeon(R) Silver 4214 CPU @ 2.19-2.20 GHz. The results are represented with respect to training cycles or epochs (Ti) (each of 15 minutes duration). It must be apparent to a person skilled in the art that the processing capabilities of a datacenter are much higher and can be used to achieve results in near real-time.

The performance of the proposed AdapSR framework is compared with the dynamic reconfiguration model by Piciarelli et al. in [8], in terms of MOTA when trained progressively over epochs and is presented in Table 4.1 and Figure 4.3 for data sample 1, and Table 4.2 and Figure 4.4 for data sample 2, respectively.

Data Sample-1:

Table 4.1 Comparison of performance of AdapSR with [8] tested on video data sample 1 in terms of MOTA%, trained over epochs (T) of 15 minutes each.

Pixels: 640 × 360		**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
[8]	TPC	30,541	41,556	52,956	56,871	60,279	74,638	76,267
	TNC	18,719	20,268	28,514	33,400	39,277	35,098	41,905
	FPC	80,271	74,352	69,183	66,418	63,116	60,947	59,104
	FNC	1,00,869	94,024	79,747	73,711	67,728	59,717	53,124
	MOTA (%)	**21.38**	**26.92**	**35.36**	**39.18**	**43.21**	**47.63**	**51.29**
	Training cycles to obtain above 80% MOTA: **18**							
AdapSR	TPC	63,018	69,217	72,141	76,238	78,908	79,519	86,211
	TNC	33,128	37,320	41,169	46,473	48,042	52,793	51,775
	FPC	47,982	43,755	40,073	36,117	36,431	31,824	29,273
	FNC	86,272	80,108	77,017	71,512	67,019	66,264	63,141
	MOTA (%)	**41.73**	**46.24**	**49.18**	**53.26**	**55.10**	**57.34**	**59.89**
	Training cycles to obtain above 80% MOTA: **12**							

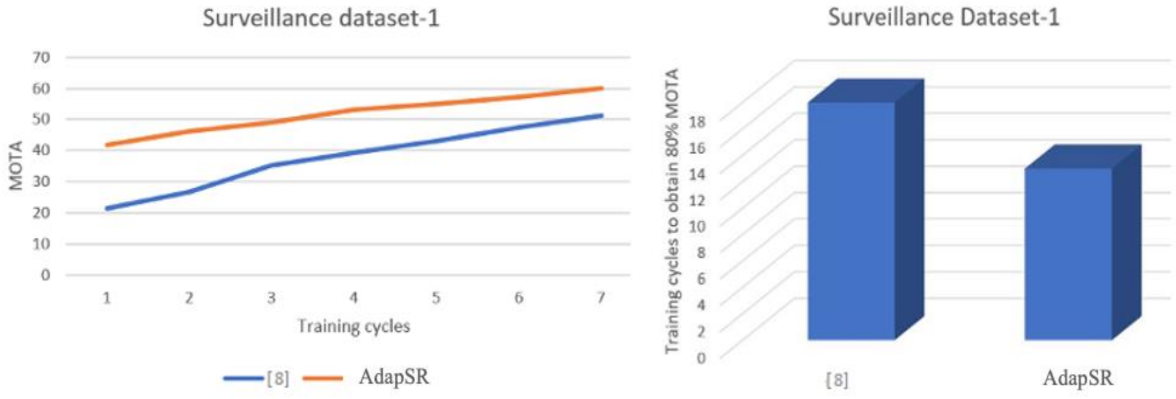


Figure 4.3 Comparison of the performances of the system in [8] and AdapSR for data sample 1.

Data Sample-2:

Table 4.2 Comparison of performance of ADAPSR with [8] tested on video data sample 2 in terms of MOTA%, trained over epochs (T) of 15 minutes each.

Pixels: 640 × 360		**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
[8]	TPC	23,211	27,324	29,841	33,266	36,421	39,972	41,101
	TNC	50,080	58,477	66,581	69,515	76,451	77,601	83,591
	FPC	89,233	85,161	82,686	79,957	75,277	72,098	69,035
	FNC	67,876	59,438	51,292	47,662	42,251	40,729	36,673
	MOTA (%)	**31.81**	**37.24**	**41.85**	**44.61**	**48.99**	**51.03**	**54.12**
	Training cycles to obtain above 80% MOTA: **15**							
ADAPSR	TPC	38,211	40,128	41,007	42,091	43,108	43,236	43,901
	TNC	70,860	77,652	83,017	84,952	90,317	92,884	96,666

	FPC	77,102	71,928	69,982	67,041	66,101	65,384	63,687
	FNC	44,227	40,692	36,394	36,316	30,874	28,896	26,146
	MOTA (%)	**47.34**	**51.12**	**53.83**	**55.14**	**57.91**	**59.08**	**61.01**
	Training cycles to obtain above 80% MOTA: **11**							

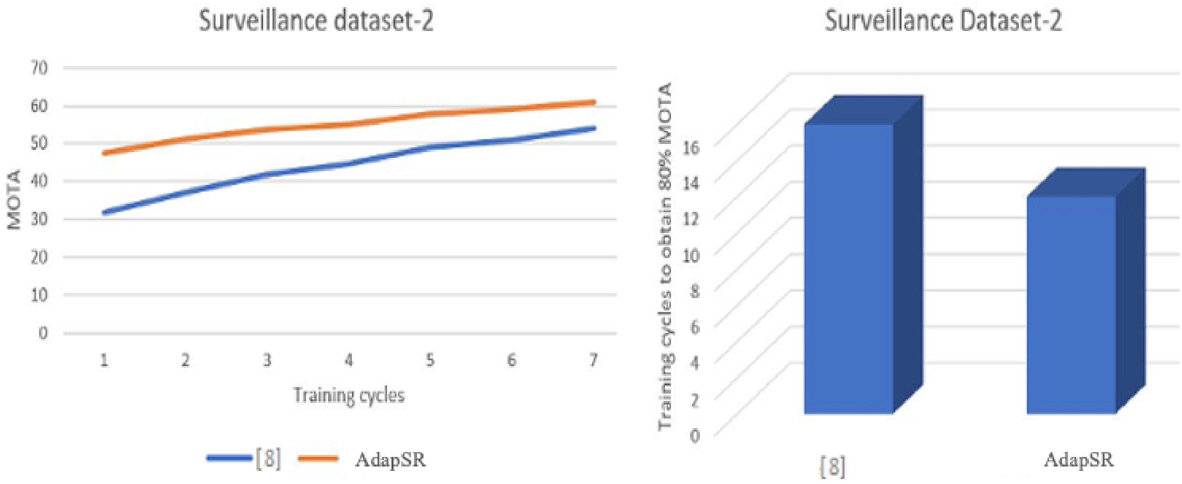


Figure 4.4 Comparison of the performances of the system in [8] and AdapSR for data sample 2

4.8 Conclusion and Scope

The functionality of active vision systems and the reconfiguration of the sensors that feed those systems with data are interdependent. However, it can be difficult to reconfigure the calibration space of an active vision system using a network of sensors. The contemporary computer vision systems struggle miserably to deal with unforeseen conditions, as it takes ample amount of time to develop understanding of the unforeseen condition. Thus, it is almost impossible to reconfigure such a system in real-time. Further, to process sensor data and derive an understanding of the scene, majority of contemporary active vision systems rely on Artificial Intelligence (AI) based models that are vulnerable to visual attacks (such as adversarial attack). The proposed AdapSR framework model provides fast and efficient reconfiguration of a smart camera networks to tackle unforeseen conditions by deriving an understanding of the condition based on past events experienced by other smart camera networks coupled to a blockchain network. The blockchain network of the AdapSR framework acts as a system of systems, connecting a number of smart camera networks together in a distributed environment. The performance of AdapSR framework surpasses the state of art dynamic reconfiguration presented in [8] that is the base of various adaptive reconfiguration systems in terms of multi-object tracking accuracy (MOTA) and processing latency. With some limitations and presumptions, the proposed AdapSR framework is tested in a homogeneous sensor environment, however, the AdapSR framework aims to be developed for heterogeneous systems to broaden the scope of its applications in the future. The AdapSR framework model is presented in one of our research papers cited as [172].

Further, for storage of large data (corresponding to training models, datasets etc.) in the distributed ledger, the AdapSR framework proposes data compression using an autoencoder model which is presented in Chapter 5.

Chapter 5

AUTO-ENCODER FOR ADAPTIVE SELF-RECONFIGURATION

The Adaptive Self-Reconfiguration (AdapSR) framework presents an efficient solution for self-adaptation and self-reconfiguration of active vision systems. However, the accessibility of datacentres required by the AdapSR is limited and so is the data storage capability of each datacentre deployed in the blockchain network and performing active vision tasks for the AdapSR model. As the blockchain network determines, analyses and stores a number of activities associated with each active vision system utilizing its resources, with time the size and the data pertaining to activities and events analysed by the blockchain network increases drastically. As the storage capacity associated with each datacentre is limited, and the parametric data, datasets, protocols, and other data/metadata associated with the activities is distributed throughout the network of datacentres, it is a matter of concern for the blockchain network deploying datacenters to manage such a big data. Addition of new datacentres to the network is an expensive affair and leads to an everlasting increase in expenses of the network. Typically, the consecutive images captured by the smart camera network of each active vision systems used to determine the spatiotemporal activity maps and the training datasets used for training the models (such as the training dataset of fast R-CNN as proposed in Chapter 4) exploit the storage resources extensively.

Thus, storage of the training datasets and consecutive frames associated with the activities (hereinafter cumulatively referred to as “activity data”) in the distributed ledger needs immediate attention and demands a technical solution to optimize the storage size by appropriate and efficient compression of the activity data. There further remains a need to determine critical information associated with the activity data for efficient storage of the compressed form of the activity data such that the critical information associated with the data is not lost and the activity data can be retrieved in its original form whenever required by the blockchain network. To address this problem, we propose a simple yet effective auto-encoder model for compression of the activity data based on the properties of Gyrator Transform (GT).

5.1 Gyrator Transform

Gyrator transform (GT) is widely exploited in cryptography as presented in [196]-[198] due to its simplistic computations and resultant properties. The Gyrator Transform as presented in [199] is a linear canonical integral transform, that results in a twisted rotational effect in position-spatial frequency planes of phase space. Gyrator Transform is an extension to Fast Fourier Transform (FFT) that produces a similar effect to rotation of an input signal about the optical axis. The rotation angle (α) of transform is used as encryption parameter whereas negative of the rotation angle is utilized as the key to revert the effect at the decryption phase. This holds a strong security as the rotational angle can take any value from 0 to 2π. At π/2 the Gyrator Transform behaves as FFT. Gyrator Transform holds scalability, periodicity and additive property with respect to the rotation angle which makes it an efficient tool for cryptography.

Initially, the input data *'g(x,y)'* (preferably in the form of a matrix, however not limited to it) is multiplied with a first conversion factor to determine a first intermediate state *'$g_2(x,y)$'* as presented in equation 5.1 hereinbelow.

$$g_2(x,y) = e^{jxy(\Delta 1)(\Delta 1)\cot(\alpha)} \,.\, g(x,y) \qquad (5.1)$$

where, *'$e^{jxy(\Delta 1)(\Delta 1)\cot(\alpha)}$'* represents the first conversion factor, *x* and *y* represents the input coordinates, respectively, *'Δ1'* represents the differential value of input coordinates *x* and *y*, and α represents the rotation angle.

Further, a Discrete Fourier Transform (DFT) is applied to the first intermediate state *'$g_2(x,y)$'* to determine a second intermediate state *'$G_{\alpha,2}(p,q)$'*, which upon transpose generates a third intermediate state *'$G_{\alpha,3}(p,q)$'*. Furthermore, the third intermediate state *'$G_{\alpha,3}(p,q)$'* is multiplied by a second conversion factor as presented in equation 5.2 hereinbelow.

$$G_{\alpha,3}(p,q) = ((\Delta 2)^2.\ (|csc\ (\alpha)|/2\ \pi).\ e^{jpq(\Delta 2)(\Delta 2)\cot(\alpha)}).\ G_{\alpha,2}(p,q) \qquad (5.2)$$

where, *'$\Delta 2^2.\ (|csc\ (\alpha)|/2\ \pi).\ e^{jpq(\Delta 2)(\Delta 2)\cot(\alpha)}$'* represents the second conversion factor, *p* and *q* represents the output coordinates, respectively, *'Δ2'* represents the differential value of the output coordinates *p* and *q*, and *α* represents the rotation angle.

The transformation of data through the Gyrator Transform is shown in Figure 5.1 hereinbelow.

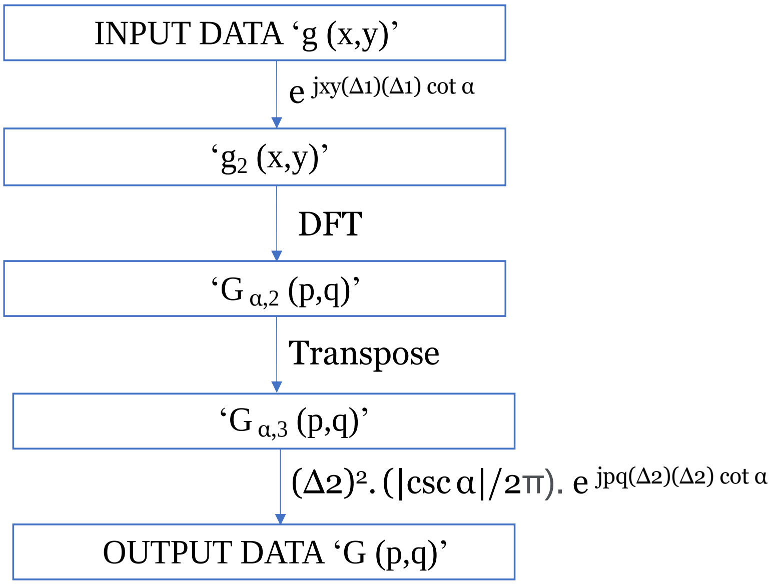

INPUT DATA 'g (x,y)'
$e^{jxy(\Delta 1)(\Delta 1)\cot\alpha}$
'g_2 (x,y)'
DFT
'$G_{\alpha,2}$ (p,q)'
Transpose
'$G_{\alpha,3}$ (p,q)'
$(\Delta 2)^2.\ (|csc\ \alpha|/2\pi).\ e^{jpq(\Delta 2)(\Delta 2)\cot\alpha}$
OUTPUT DATA 'G (p,q)'

Figure 5.1 Computation of Gyrator Transform for Autoencoder

5.2 Autoencoder Model

The compression properties of Gyrator transform as shown in Figure 5.1 are used by the proposed autoencoder for self-adaptive re-configurator to compress the activity data for storage in the distributed ledger of the blockchain network. From the above discussion on transformation of data by way of the Gyrator Transform (GT), it is well-established that the encryption-decryption of data typically depend on the dimension of the input and output data, the differential value of the input and output coordinates corresponding to the input and output data, respectively, and most importantly the rotation angle (α). Thus, the autoencoder model proposes selection of random values from 0 to 2π for selection of the rotation angle, a random value for output dimension that is less than the input dimension, and a random value for selection of the differential value of the output co-ordinates. Based on the selected values and the dimensions and differential value of input coordinates from the input images, the autoencoder model, by way of the plurality of processing nodes is configured to generate a feature vector for each image.

For compression of each image of the activity data, the autoencoder model is configured to perform pre-processing of each image by binarization and thinning of each image of the activity data. The autoencoder is further configured to encrypt each image of the activity data using the corresponding feature vector by way of Gyrator Transform (GT), which encrypts and compresses the data of each image of the activity data. Furthermore, the autoencoder model is configured to identify the non-zero pixels of each encrypted image. Upon identification of the non-zero pixels of each encrypted image, the autoencoder model is configured to generate an encrypted data for each image based on the metadata of the non-zero pixels of the encrypted image, and the feature vector, that is stored in the distributed ledger. The image can be retrieved using the Gyrator transform (GT) with the values in the feature vector using (-α) as the rotation angle.

5.3 Process

The autoencoder provides compression and encryption of the activity data by way of Gyrator Transform (GT) using random parametric values, that results in increased security and reduced size of the activity data for efficient storage of activity data in the distributed ledger. In operation, the autoencoder generates a random feature vector for each image of the activity data, and encrypts the images based on the corresponding feature vector using Gyrator Transform (GT). Upon encryption of the images, the autoencoder identifies non-zero pixel values of each encrypted image and generates an encrypted data for each image based on the metadata of the non-zero pixel values and the feature vector corresponding to the image. The encrypted data is then stored in the distributed ledger. The Gyrator transform is further used on the encrypted data with the encryption parameters in the feature vector, and a negative rotation angle (-α) to retrieve the original image. The autoencoder model is tested on a sample image in the next section.

5.4 Simulations and Results

For testing the efficiency of the proposed autoencoder, the autoencoder model is tested on a sample fingerprint image. The fingerprint image is selected to verify the performance of the autoencoder in retaining the important information associated with each fingerprint. It is a well-established fact that each fingerprint is different from the other based on its features known as minutiae. Minutiae carry the important information of each fingerprint image and thus are critical to each fingerprint. For illustration, the autoencoder model is tested with the basic three minutiae associated with any fingerprint that are (i) ridge bifurcation, (ii) ridge continuation, and (iii) ridge ending. To determine the abovementioned minutiae, the fingerprint image is pre-processed by binarization and thinning to one pixel width. Further, a (3x3) pixel mask is used on the fingerprint image to determine fingerprint minutiae shifting the mask to each pixel of the pre-processed image. The minutiae are determined by the values of the pixels surrounding the center of the mask. For example, when the center pixel of the (3x3) pixel mask is surrounded by two dark pixels on one side, such a situation corresponds to a ridge bifurcation, else when the center pixel is surrounded by only one dark pixel continuing along the ridge, such a situation corresponds to ridge continuation, else when the center pixel is not surrounded by any dark pixel on a side of the center pixel, such situation corresponds to ridge ending as shown in Figure 5.2.

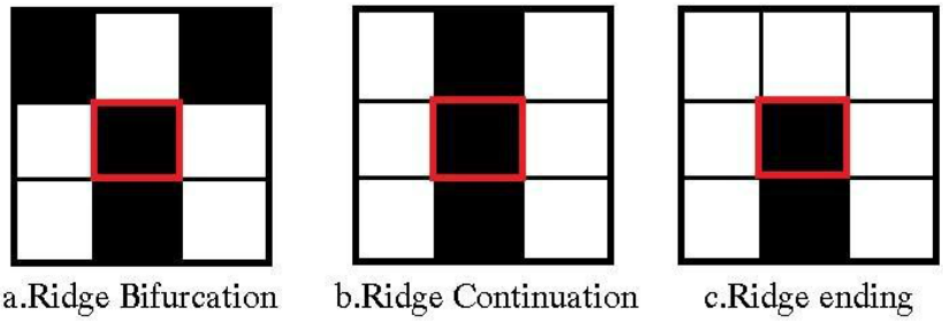

a.Ridge Bifurcation b.Ridge Continuation c.Ridge ending

Figure 5.2 Classification of fingerprint minutiae

For testing the efficiency of the autoencoder, we determined minutiae from original fingerprint image and then encrypted the original image using GT based autoencoder using rotation angle (α) equal to 0.8 π. The output pixel dimensions are kept equal to the input pixel dimensions, and the input and output differential values are kept equal to ‘1’ for simplicity. The results obtained by the autoencoder through the process are shown from Figure 5.3 to Figure 5.6 hereinbelow.

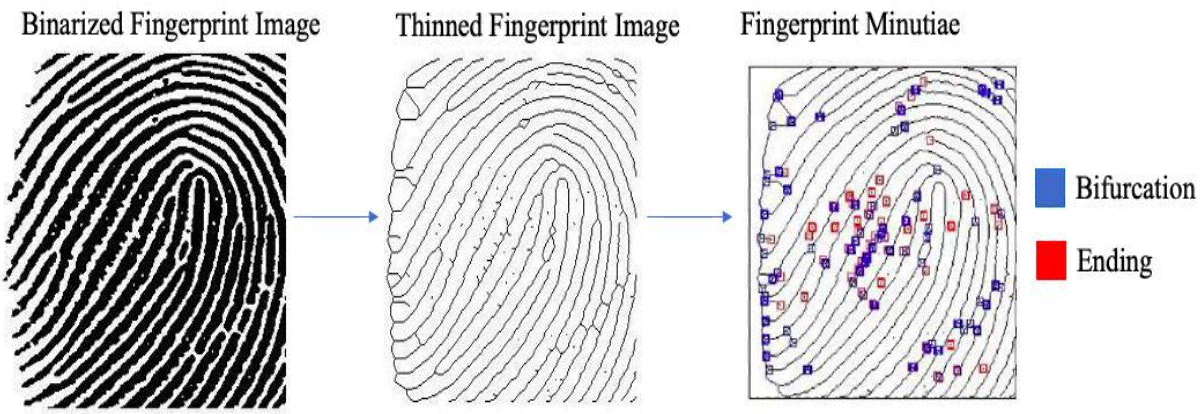


Figure 5.3 Minutiae determined from input image

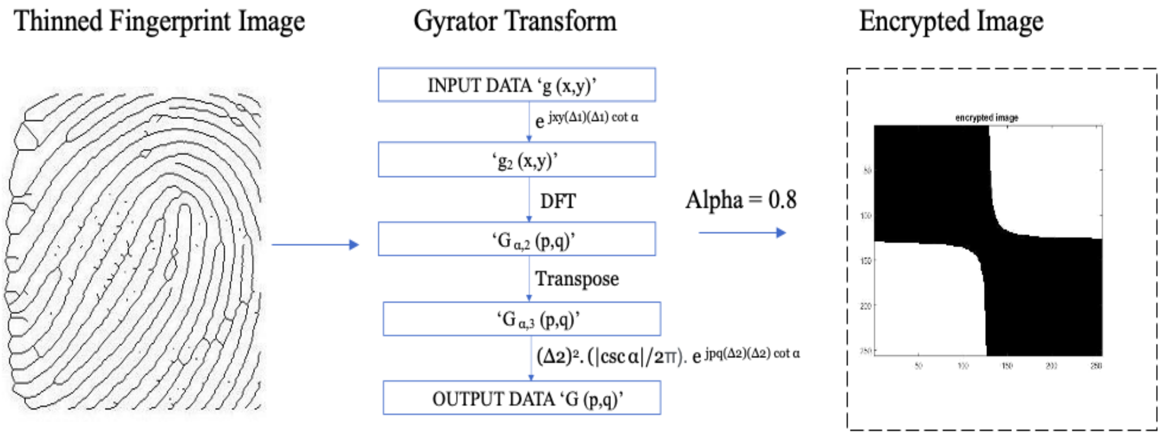


Figure 5.4 Encryption of pre-processed image using GT with α =0.8 π

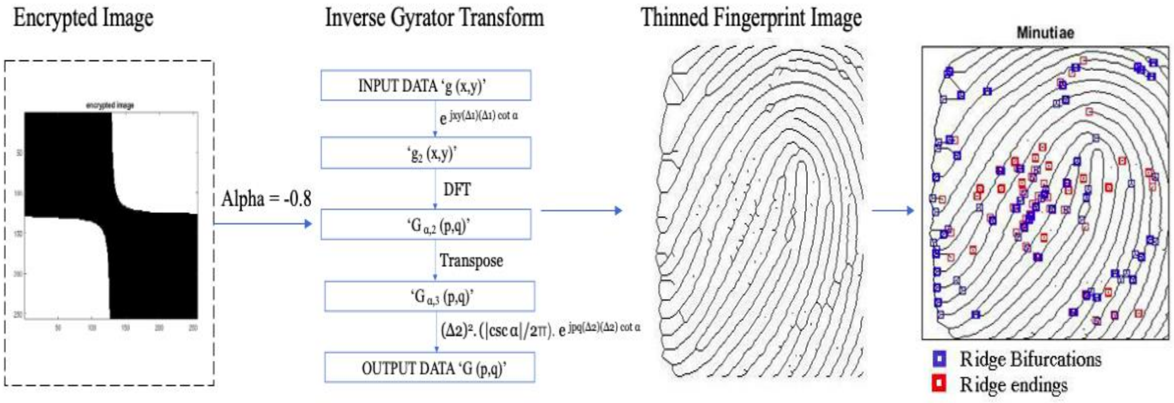


Figure 5.5 Minutiae determined from the encrypted image

The minutiae derived from the input pre-processed image and the minutiae derived by the decrypted thinned image (*i.e.*, received after decryption of the encrypted image) are compared, and it is observed that each minutia of the input image is preserved. Due to less information contained in the image encrypted by the autoencoder, the size of the image is reduced. Specifically, for the configuration mentioned hereinabove, a 33KB custom fingerprint image is pre-processed and encrypted resulting in an 8KB encrypted image with a feature vector of approximately 1KB associated with it.

5.5 Conclusion and Scope

By the comparison of the features of the input and output fingerprint images, it has been observed that the autoencoder encrypts the image without losing the critical information of the image. The autoencoder further provides an efficient reduction in the size of the image in terms of memory occupancy, which specific to the custom fingerprint image with a rotation angle value (α) equal to 0.8 π, output pixel dimensions equal to the input pixel dimensions, and the input and output differential values equal to ‘1’, resulted in 9KB data (including 1KB feature vector) compressed from 33KB (original size of the fingerprint image) such that the compressed image occupies only 0.28 times the memory size as compared to the original image. As the occupancy of the distributed ledger of the AdapSR framework model is highly affected by the humongous size of datasets and the consecutive images captured by each active vision system, the autoencoder provides a relief to the exploited utilization of the distributed ledger. It must be apparent to a person skilled in the art that the abovementioned ratio of the reduction in size is specific to the use of autoencoder in the specific configuration used on the custom fingerprint image and may vary with the type of information contained by the image, however the compression of the image is confirmed for any image without losing the critical features of the image.

The autoencoder is proposed to compress and encrypt activity data including the consecutive images and the image datasets, however the autoencoder model is yet to be modified and tested for encryption and compression of the parametric data, protocols, model data, and other data/metadata associated with the AdapSR model.

The use of Gyrator Transform of the autoencoder model for compression and encryption of image data is presented in one of our research papers cited as [200].

Chapter 6

DYNAMIC SPEED LIMIT ALLOCATION

Injuries, fatalities and deaths caused by road accidents are the major public health hazards affecting families mentally, physically, emotionally and financially. As presented in a road accident report of the year 2018-2019 by Delhi police [201], more than 90% of the vehicles on road are motorcycles and private cars with an average yearly growth of over 6% in India. Most of the causalities are faced by best productive age group of 25 to 50 years old which affects as an overall dip of 3% in GDP of a nation. New Delhi, the national capital of India records the second largest population density of the nation with a total population of over 32 million people. As reported in the report [201], over 5000 road accidents, and over 1400 deaths due to road accidents is recorded every year in the capital. An average of 24.63% are reported severe accidents every year out of which over 85% of the victims with fatal injuries and deaths are pedestrians, car occupants and two-wheeler riders. Various factors like over speeding, red light jumping, driving without helmet, drink and drive, using mobile phones while driving, wrong lane driving and poor maintenance of road and vehicles enhance the possibility of road accidents, over speeding being one of the major reasons of fatal injuries and death. The main reason of such accidents is impatience of people due to traffic jams, but what causes traffic jams? The answer is simple, traffic density is uneven throughout the day, however, traffic speed limits are fixed. This fixed speed limit causes chaos and traffic jams resulting in higher number of accidents, causalities and even death. Thus, there remains a need of an automated dynamic speed allocation system that can predict and allocate speed limit of an area based on a number of parameters such as the traffic analysis, conditions of the road and probability of accidents at a specific speed limit particular to that area. This chapter presents a Dynamic Speed Allocation (DSA) framework for prediction of traffic speed limit for different areas.

6.1 Dynamic Speed Allocation Framework

The Dynamic Speed Allocation (DSA) framework presents dynamic speed allocation, and is inspired by the AdapSR framework as presented in Chapter 4 for adaptive prediction of suitable speed limits in different areas. The DSA framework, based on input data corresponding to different areas in the form of a plurality of parameters such as traffic density, accident count, static speed limit etc., predicts a most suitable speed limit for each area.

6.2 Model

The operational model based on the DSA framework requires a plurality of smart sensor networks deployed in different areas and a plurality of processing nodes co-operatively coupled to each other in a distributed network. More particularly, the plurality of processing nodes can be datacenters coupled to each other in a distributed blockchain network (when deployed for real time dynamic traffic speed limit allocation based on real-time analysis of video feed of each area). Each smart sensor network is configured to obtain in real time, a number of parameters that correspond to determination of speed limit of an area. Specifically, the speed limit of each area depends on two major factors: Traffic density, and probability of accidents. The objective of the DSA framework is to determine an optimum speed limit for each area for minimizing the traffic density and probability of accidents. The DSA model architecture is as shown in Figure 6.1 hereinbelow.

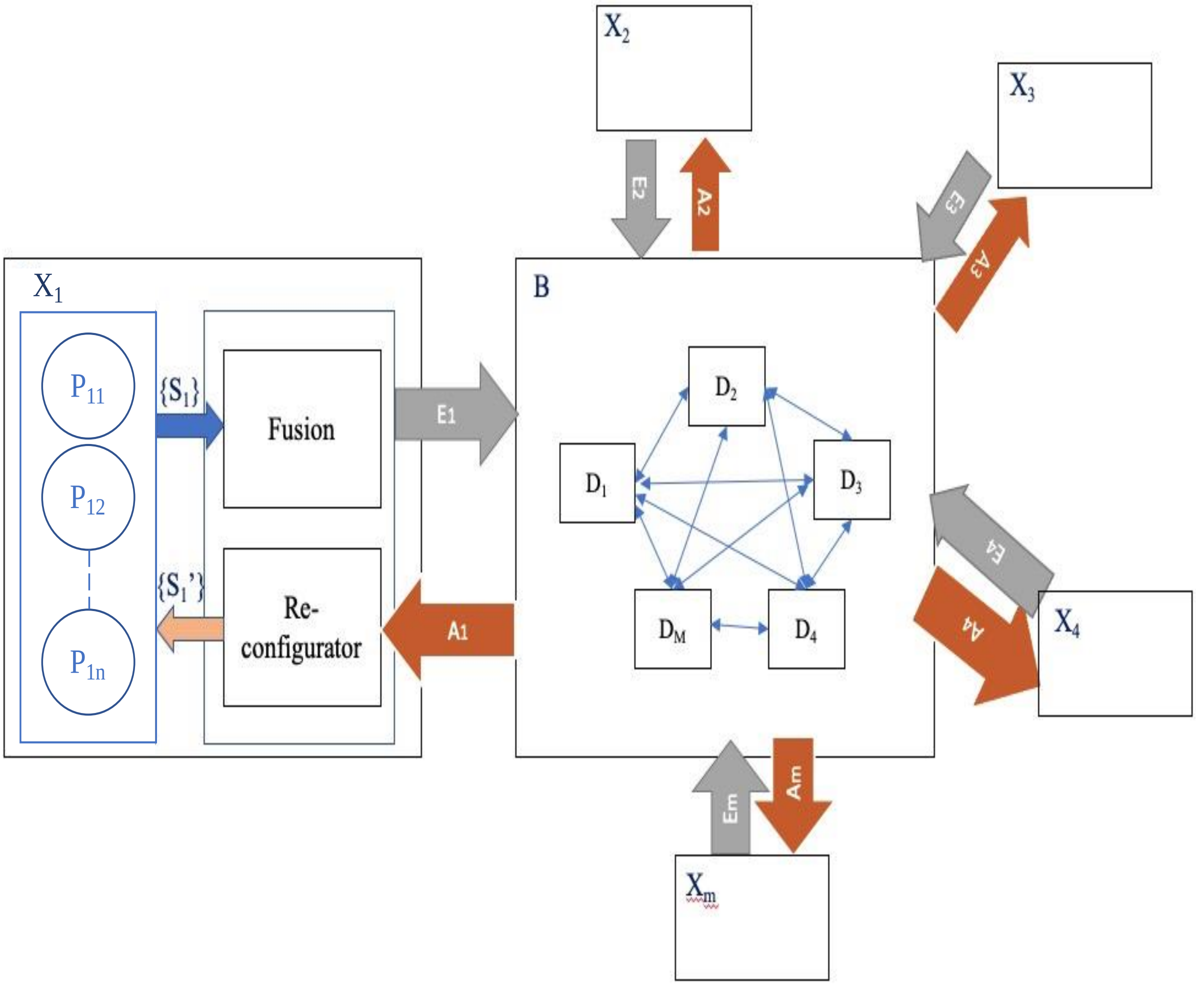


Figure 6.1 Framework for dynamic speed limit allocation.

Notations used:

X_i: ith area under observation;

P_{ij}: jth parameter of the ith area;

n_i: Number of parameters in the ith area;

$\{S_i\}$: Input parametric data of ith area;

$\{S_{i'}\}$: Revised parametric data for the ith area;

E_i: Self-expression data (*i.e.*, cost function) for the ith area;

A_i: Self-awareness data (*i.e.,* optimized cost function) for the ith area;

m: Count of areas utilizing the DSA framework;

B: Distributed network comprising “M” number of datacenters/processors; and

D_k: kth datacenter in the distributed network.

Each smart sensor network includes a number of smart sensors for determination of the plurality of parameters associated with the speed limit of that particular area. The area further has a fusion unit to generate an initial cost function comprising weighted sum of the plurality of parameters, that is transferred to the plurality of processing nodes of the distributed network. Each area further has a re-configurator unit that is configured to receive an optimized cost function from the plurality of processing nodes and derive the revised values of the plurality of parameters based on the optimized cost function. The plurality of processing nodes is configured to receive the initial cost function from each area, and derive a relationship between the plurality of parameters for optimizing the cost function. The plurality of processing nodes is further configured to compare the initial cost function of each area to determine a best area with the highest initial cost function. Furthermore, the plurality of processing nodes is configured to determine a relative efficiency of speed limit (in terms of a normalized efficiency score) of each area in reference to the best area. Based on the normalized efficiency score of each area, the plurality of processing nodes is configured to generate the optimized cost function of each camera with optimized values of each parameter of the plurality of parameters and a relationship between the plurality of parameters, that is specific to each area. Specifically, as the plurality of parameters for DSA framework include traffic density (T), static speed limit (V), and probability of accident *i.e.,* measured in terms of count of accidents per unit area (A), such that the speed limit (V) is inversely related to the count of accidents per unit area (A) and the traffic density (T). Thus, the initial cost function (E) (*i.e.,* the self-expression) of each area is a weighted sum of the plurality of parameters as presented in equation 6.1 hereinbelow.

$$E = \sum(W_1.V) + (W_2.\ (\frac{1}{T})) + \ (W_3.\ (\frac{1}{A})) \tag{6.1}$$

The plurality of processing nodes utilizes the parametric data from each area to train a neural network for determination of the relationship between the parameters. Specifically, the neural network is configured to determine based on the count of accidents per unit area (A) and the traffic density (T) of each area, a best area amongst all the areas. Further, the plurality of processing nodes, by way of the neural network is configured to determine the relative score for each area, and update the speed limit (V), and predict the count of accidents per unit area and the traffic density based on the updated speed limit. The plurality of processing nodes is further configured to send the updated parametric value to each area, where based on the updated speed limit, the static speed limit is adjusted to compare the effect of the updated speed limit on the traffic density and the accidents per unit area. Each smart sensor network is further configured to generate an updated set of parametric data based on the updated speed limit which is compared with the predicted values by the plurality of processing nodes to determine an error function (or the loss function). After iterative updating of the parametric values, the DSA framework model achieves a steady relationship between the plurality of parameters in terms of individual weights corresponding to each area. The steady relationship in terms of weights is further used to dynamically allocate the speed limit to each area.

6.3 Process

The DSA model provides iterative updating of the plurality of parameters of each area associated with the speed limit to derive a relationship between each parameter specific to the area, and thus determine a dynamic speed limit for each area. The DSA model further predicts the traffic density and the effect of the updated speed limit on the total count of accidents in each area. In operation, the smart sensor networks associated with each area provide the initial cost function in terms of weighted sum of the plurality of parameters. The plurality of processing nodes in the distributed network, by way of the neural network, using a combined parametric data received from the different areas, determines an efficiency score for each area and further determines an updated parametric data for each area. The updated parametric area includes the updated speed limit, the predicted accident count (based on the updated speed limit), and the predicted traffic density (based on the updated speed limit) for each area, which is sent to the respective area. The updated speed limit is imposed to the respective area, and a new parametric data is obtained. The new parametric data is further sent to the plurality of processing nodes to determine a between the predicted traffic density, the predicted accident count and the actual effect of the updated speed limit on the traffic density and the accident count to determine the loss function. The loss function is minimized by iteratively updating the weights (*i.e.*, training the neural network) to determine relationship between the plurality of parameters (in terms of the weights associated with the plurality of parameters in the cost function) to determine the optimized cost function for each area. The optimized cost function is further used by each area to derive the dynamic speed limit based on the dynamic value of plurality of parameters for each area.

6.4 Results

The DSA model is tested on the dataset presented in [201] for traffic density of 2-wheelers and cars, number of accidents of 2-wheelers and cars and the static speed limit of different areas of New Delhi, India, to determine the dynamic speed limit for each area. Table 6.1 and Table 6.2 present the results generated by the DSA model for dynamic speed limit and prediction of the accident count based on the dynamic speed limit of the different areas of New Delhi, for 2-wheelers, and 4-wheelers, respectively.

Table 6.1 Dynamic speed limit allocation and prediction of the accident count based on the dynamic speed limit of the different areas of New Delhi, for 2-wheelers. The data is a monthly data derived from a yearly report [201], thus is scaled accordingly.

S.No	Location	Two wheeler Count	Area (km)	Traffic density	Static speed limit (km/h)	Two Wheeler Accident Count	Efficiency Score 2-wheelers	Revised speed limit (km/h)	Predicted Decrease In Accident 2-w (%)
1	August Kranti Marg	4841	9	537.9	60	1.16	1	60	0
2	Mayapuri Road	10289	10	1028.9	30	2.25	0.986027	29.6	2.79
3	Auchandi Bawana	2228	7	318.3	50	1.33	0.9823334	49.1	3.53
4	Africa Avenue	14875	13	1144.2	60	3.018	0.965242	57.9	6.95
5	Rani Jhansi Road	11816	7	1688.0	50	5.16	0.958388	47.9	8.32
6	Sardar Patel Marg	3666	10	366.6	50	1.25	0.954599	47.7	9.08
7	Bahadur Shah Zafar Marg	2417	12	201.4	50	2.66	0.921318	46.1	15.74
8	Prithviraj Road	1765	8	220.6	60	3.018	0.917445	55	16.51
9	Guru Ravidas Marg	4231	7	604.4	60	8.16	0.915815	54.9	16.83
10	Pankha Road	7774	10	777.4	65	11	0.914754	59.5	17.05
11	Maa Anandmayee Marg	13675	11	1243.2	70	19.33	0.913978	64.0	17.20
12	Aurobindo Marg	8029	10	802.9	40	13.16	0.912808	36.5	17.44
13	Nangloi Najafgarh Road	4219	8	527.4	50	10	0.911239	45.6	17.75
14	Lodhi Road	1664	9	184.9	50	4.5	0.90926	45.5	18.15
15	Mehrauli Badarpur Road	16389	13	1260.7	50	31	0.908164	45.4	18.37
16	Bawana Road	5644	15	376.3	50	10.25	0.907263	45.4	18.55
17	Wazirabad Road	14390	12	1199.2	70	34.018	0.907062	63.5	18.59
18	Vikas Marg	7609	12	634.1	50	18.25	0.906861	45.3	18.62
19	Najafgarh Road	12351	10	1235.1	60	47.66	0.904943	54.3	19.01
20	Mathura Road	18370	37	496.5	50	37	0.901778	45.1	19.64
21	Grand Trunk Road	6566	12	547.2	100	42.82	0.901772	90.2	19.644
22	Rohtak Road	25102	27	929.7	60	75.16	0.901470	54.1	19.706
23	GT Karnal Road	25333	46	550.7	100	61.25	0.900749	90.1	19.85
24	Ring Road	16965	15	1131.0	50	144.166	0.900439	45	19.91
25	Outer Ring Road	72143	101	714.3	70	119.66	0.9	63	20

The results are presented in Tables 6.1 and 6.2 for a relative decrease of maximum 10 % in the speed limits of two-wheelers in one iteration. The difference in speed limits based on the efficiency score through the DSA model is presented in Figure 6.2, and the effect of the updated speed in terms of predicted 2-wheeler accidents is presented in Figure 6.3 hereinbelow.

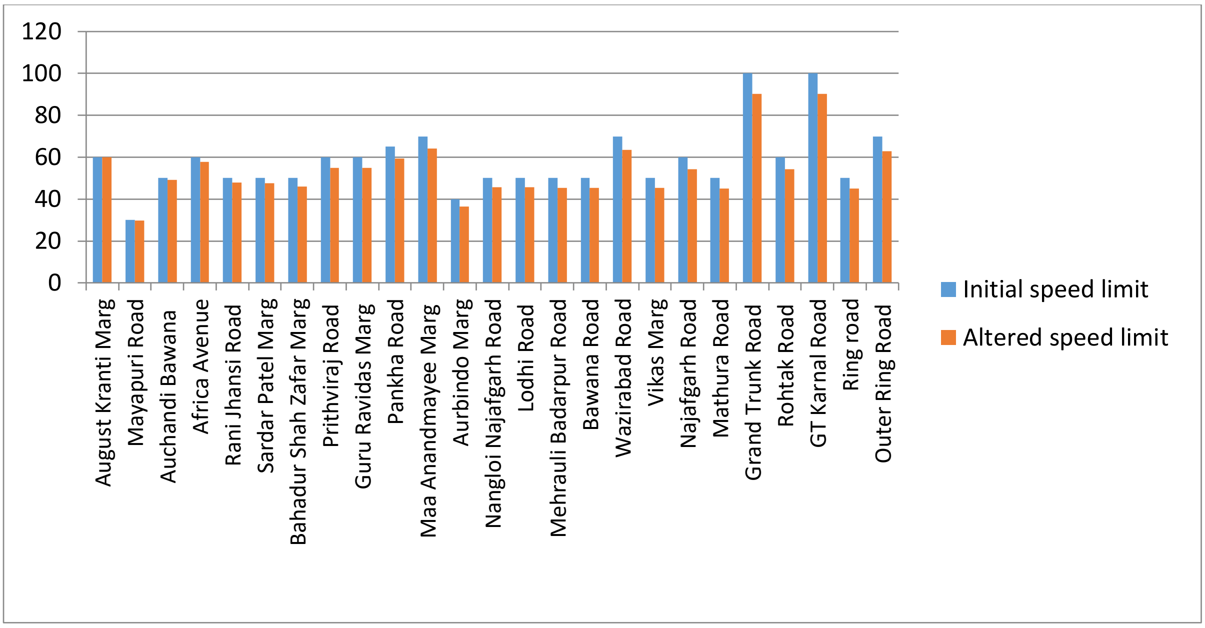


Figure 6.2 Comparison of speed limits of 2-wheelers before and after DSA.

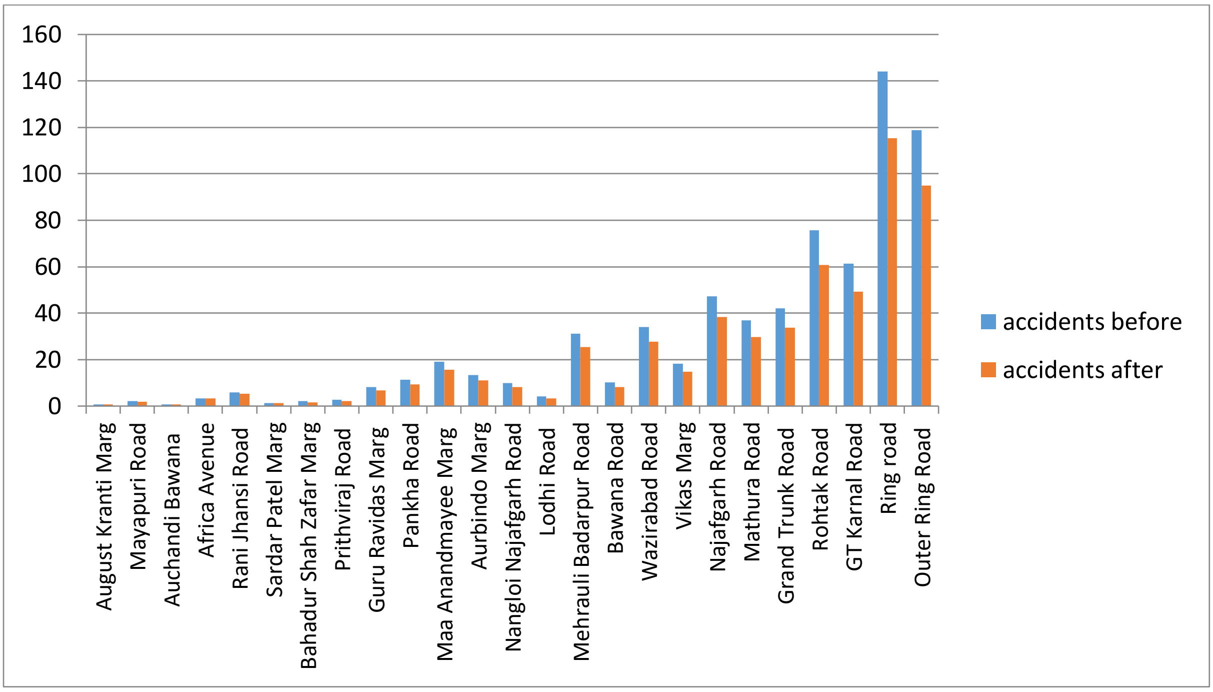


Figure 6.3 Comparison of accidents of 2-wheelers before and after DSA (predicted)

Table 6.2 Dynamic speed limit allocation and prediction of the accident count based on the dynamic speed limit of the different areas of New Delhi, for 4-wheelers. The data is a monthly data determined from a yearly report [201], thus is scaled accordingly.

S.No	Location	Four wheeler Count	Area (km)	Traffic density	Static speed limit (km/h)	Four Wheeler Accident Count	Efficiency Score 4-wheelers	Revised speed limit (km/h)	Predicted Decrease In Accident 4-w (%)
1	August Kranti Marg	8468	9	940.9	60	0.083	1	60	0
2	Mayapuri Road	5283	10	528.3	30	0.166	0.93528	28.1	12.94
3	Auchandi Bawana	1770	7	252.9	50	0.083	0.94482	47.2	11.036
4	Africa Avenue	22686	13	1745.1	60	0.22	0.9589	57.5	8.22
5	Rani Jhansi Road	4080	7	582.9	50	0.44	0.91265	45.6	17.47
6	Sardar Patel Marg	6419	10	641.9	50	0.083	0.95498	45.7	9.004
7	Bahadur Shah Zafar Marg	2706	12	225.5	50	0.166	0.91392	46.1	17.216
8	Prithviraj Road	3815	8	476.9	60	0.166	0.92211	55.3	15.578
9	Guru Ravidas Marg	1829	7	261.3	60	0.664	0.90335	54.2	19.29
10	Pankha Road	5741	10	574.1	65	0.833	0.90609	58.9	18.78
11	Maa Anandmayee Marg	5455	11	495.9	70	1.33	0.90273	63.2	19.454
12	Aurobindo Marg	11632	10	1163.2	40	0.916	0.91098	36.4	17.804
13	Nangloi Najafgarh Road	2407	8	300.9	50	0.75	0.90335	45.2	19.33
14	Lodhi Road	5438	9	604.2	50	0.25	0.91873	45.9	16.254
15	Mehrauli Badarpur Road	10200	13	784.6	50	2.166	0.90253	45.1	19.494
16	Bawana Road	4183	15	278.9	50	0.75	0.90283	45.1	19.43
17	Wazirabad Road	4429	12	369.1	70	2.25	0.9005	63	19.9
18	Vikas Marg	6671	12	555.9	50	1.33	0.90324	45.2	19.352
19	Najafgarh Road	6331	10	633.1	60	3.33	0.90098	54.1	19.804
20	Mathura Road	23597	37	637.8	50	2.66	0.9015	45.1	19.7
21	Grand Trunk Road	6832	12	569.3	100	3	0.90098	90.1	19.804
22	Rohtak Road	14858	27	550.3	60	5.5	0.90001	54	20
23	GT Karnal Road	21861	46	475.2	100	4.83	0.9	90	20
24	Ring Road	22997	15	1533.1	50	10.33	0.9005	45	19.9
25	Outer Ring Road	122085	101	1208.8	70	8.5	0.9004	63	19.98

The difference in speed limits based on the efficiency score through the DSA model is presented in Figure 6.4, and the effect of the updated speed in terms of predicted 2-wheeler accidents is presented in Figure 6.5 hereinbelow.

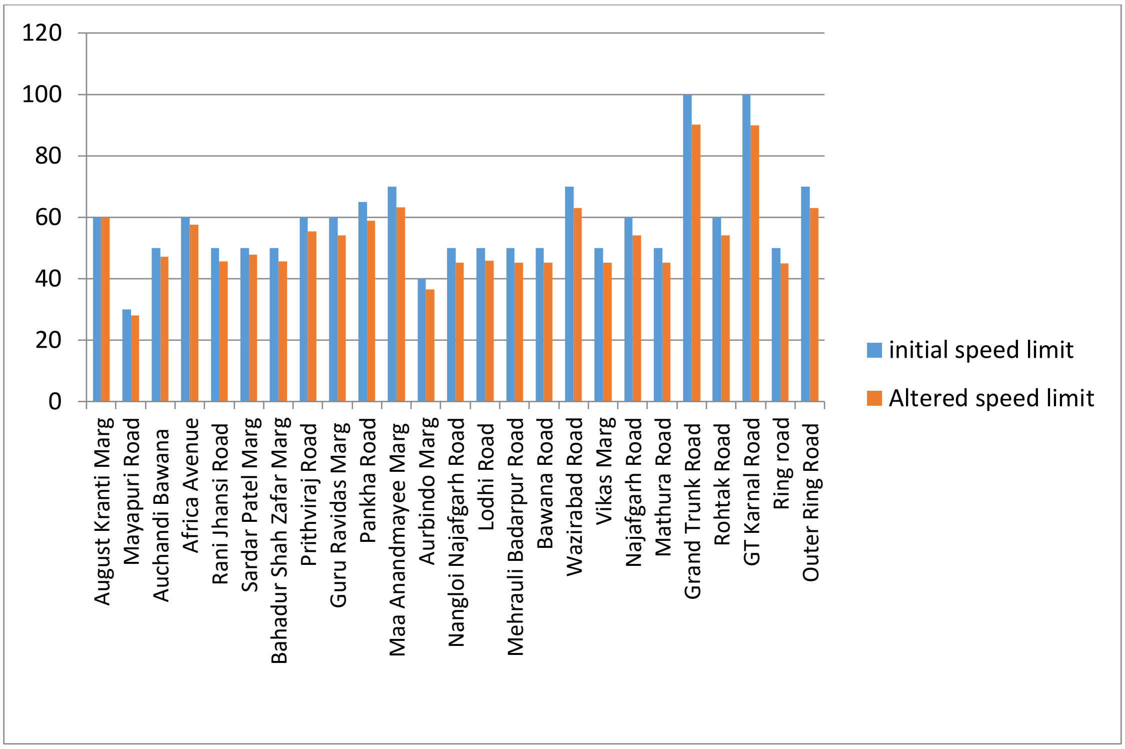


Figure 6.4 Comparison of speed limits of 4-wheelers before and after DSA

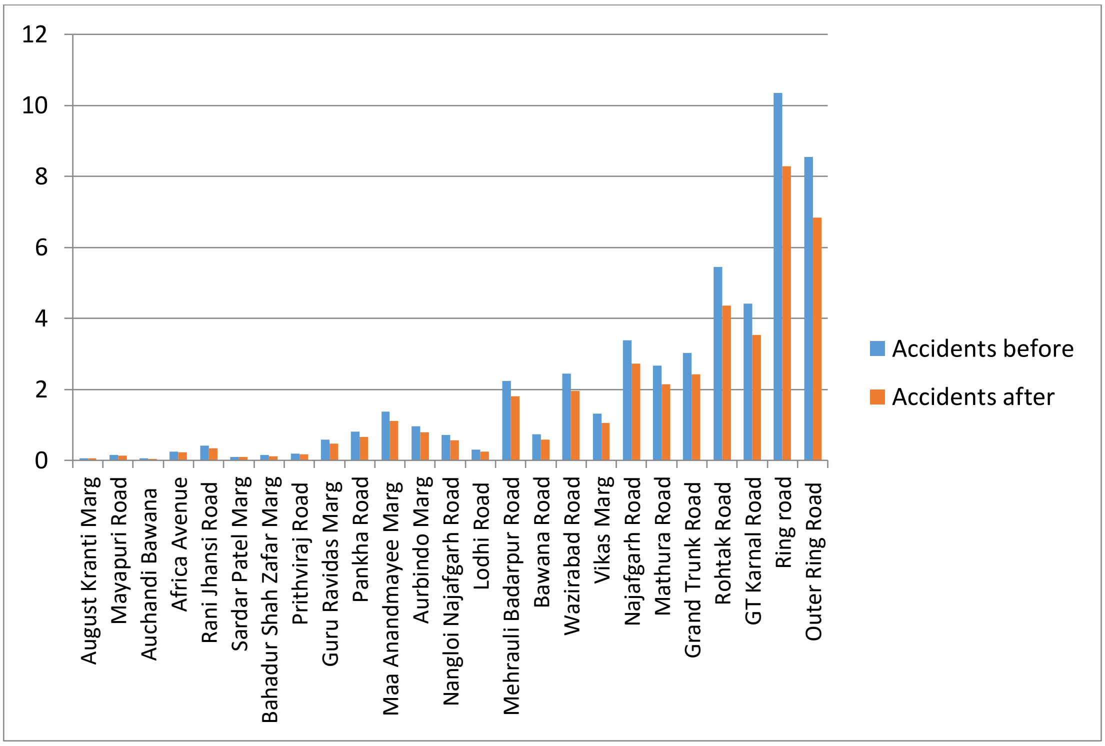


Figure 6.5 Comparison of accidents of 2-wheelers before and after DSA (predicted)

6.5 Conclusion and Scope

The DSA framework is based on the architecture of SAM framework presented in Chapter 4, and provides an efficient adaptive dynamic speed limit allocation based on the plurality of parameters derived from different areas under observation. The effect of the change speed limit on the accident count for 2-wheelers and 4-wheelers is shown in Figure 6.6 and Figure 6.7, respectively. The effect of the change in speed limit on the change in accidents is found to be in accordance with the relationship presented by Cameron et al. in [202], that validates the effectiveness of the proposed DSA framework model.

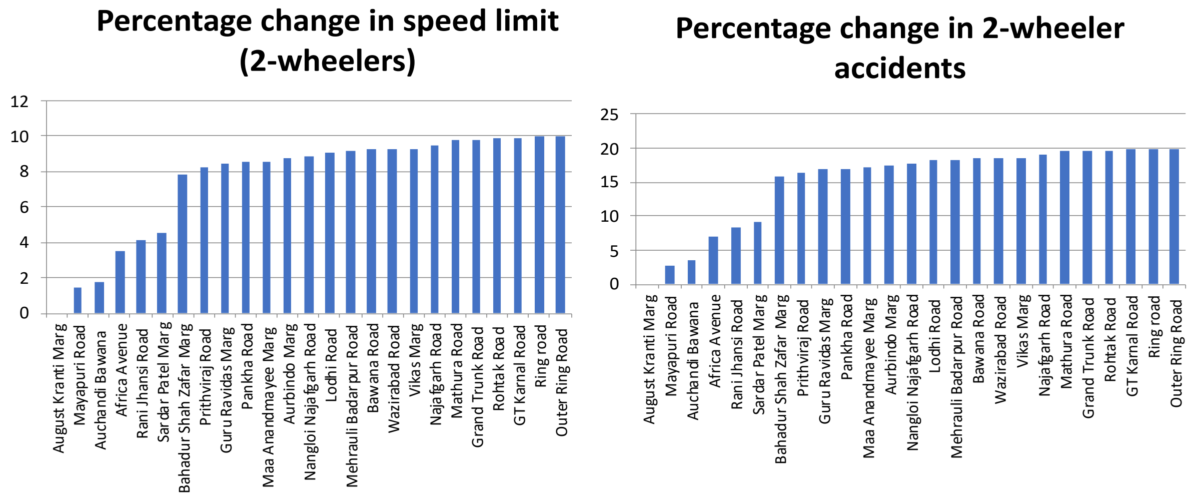


Figure 6.6 Effect of speed limit change on 2-wheeler accidents (predicted)

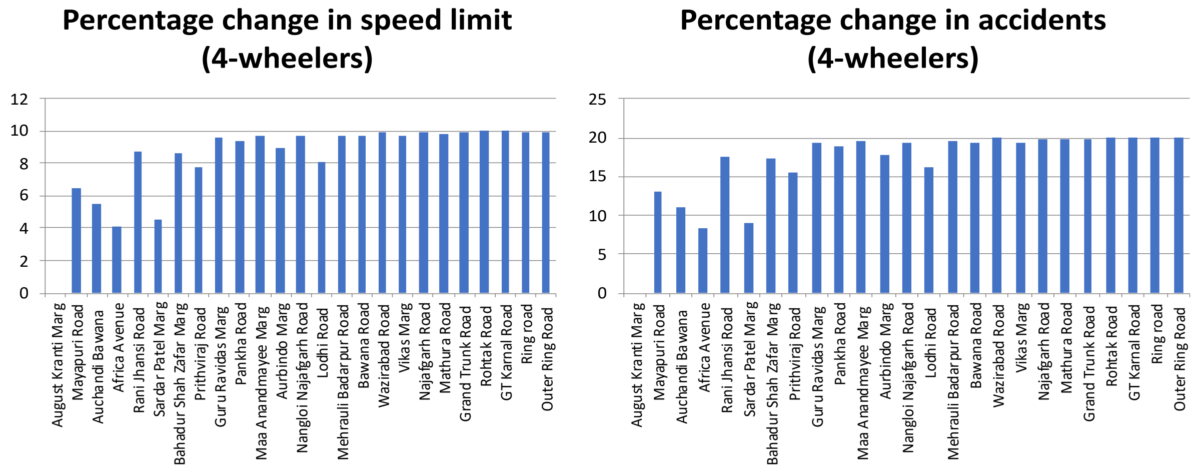


Figure 6.7 Effect of speed limit change on 4-wheeler accidents (prediction)

The DSA framework is tested on a steady dataset [201], and not on a real-time analysis of video feeds from different areas. Thus the traffic density is assumed to be constant, however for real-time applications, the traffic density can be determined by the total count of two-wheelers and four-wheelers per unit area dynamically captured by a computer vision system. Further, the DSA framework can be used to regularize the traffic speed based on the actual traffic density of each area, which requires development and deployment of the DSA model in cooperation with a network of computer vision system, and thus requires future advancements.

The DSA framework is presented in a research paper submitted for publication, that is under review.

Chapter 7

CONCLUSION AND FUTURE SCOPE

This chapter will summarize the major contributions and achievements that come out of the present work. The summary of the major contributions follows in the following Section 7.1. Despite the significant contributions, no research is said to be complete unless it directs to a few topics for future research. Hence, the potential work that can be explored further is briefly discussed as directions to future work in the Section. 7.2.

7.1 Summary of Major Contributions

The essence of this thesis work is to design and develop efficient techniques for adaptive reconfiguration of calibration parameters of active vision systems to capture objects of interest with enhanced resolution, that yields to an improved understanding of the scene. In order to address the limitations of various aspects in this field, several innovative frameworks and methods have been suggested under current work which are summarized as follows:

- Smart camera networks deployed for a variety of applications require different resource utilization models based on the objectives and resource availability. Thus, there is a requirement to understand the requirements and objectives prior to designing an active vision system prior to its deployment in computer vision applications. This thesis segregates the computer vision applications into two categories based on the requirements, objectives and resource availability, and proposes framework models specifically designed to fulfil the requirements of each category. Particularly, both the framework models are designed for an directed objective of adaptive reconfiguration of the smart sensor networks deployed for active vision applications for improved object detection and enhanced understanding of the scene.

- Advanced and complex systems are the possible state-of-art options for re-configuration of sensor's parameters. However, such approaches are not feasible to be used for mobile systems with limited computational, power, and storage resources. To address the aforementioned challenge, this thesis presents the SAM framework for spatiotemporal activity mapping that can balance the performance of active vision system with limited resource availability, and is used for reconfiguration of the active vision system.

  The SAM framework model analyses the scene in both spatial and temporal perspectives, and generates adaptive activity maps for sensor reconfiguration so that the important region(s) can be captured in the camera sensor's field of view. The framework pre-processes the sensor data using simplistic image processing techniques like background subtraction, binarization, thresholding, and federated optical flow. The temporal relationship between the consecutive image frames is represented by a half-width Gaussian distribution. The proposed SAM framework's straightforward model yields highly accurate spatiotemporal activity mapping with low computation complexity, and thus employ system low resources. The performance is compared in terms of MOTA, where the SAM framework model outperforms contemporary systems presented in [165], [166], [176], [177] and [178]. Specifically, the SAM framework model showcases 0.79% better average MOTA relative to [176] and 8.39 % better average MOTA as relative to [177], when tested on traffic surveillance datasets (*i.e.*, data samples 1, 2 and 3). The SAM framework model further showcases 4.21% better average MOTA as compare to [178], when tested on sports datasets (*i.e.*, data samples 4, 5 and 6).

- The contemporary computer vision systems struggle miserably to deal with unforeseen conditions, as it takes ample amount of time to develop understanding of the unforeseen condition. Thus, it is almost impossible to reconfigure such a system in real-time. Further, to process sensor data and derive an understanding of the scene, majority of contemporary active vision systems rely on Artificial Intelligence (AI) based models that are vulnerable to visual attacks (such as adversarial attack). The proposed AdapSR framework model provides fast and efficient reconfiguration of a smart camera networks to tackle unforeseen conditions by deriving an understanding of the condition based on past events experienced by other smart camera networks coupled to a blockchain network. The blockchain network of the AdapSR framework acts as a system of systems, connecting a number of smart camera networks together in a distributed environment. The performance of AdapSR framework surpasses the state of art dynamic reconfiguration presented in [8] that is the base of various adaptive reconfiguration systems in terms of multi-object tracking accuracy (MOTA) and processing latency.
- The AdapSR framework model relies on storage of activity data, models, parametric data and large image datasets and consecutive images to be saved in the distributed ledger of the distributed network utilized by the AdapSR model. However, the accessibility of datacentres required by the AdapSR is limited and so is the data storage capability of each datacentre deployed in the blockchain network and performing active

vision tasks for the AdapSR model. Thus, the AdapSR model requires compression of large datasets for optimum utilization of storage resources. To achieve the same, the autoencoder presented in Chapter 5 presents a model for compression of the large image data by utilizing the properties of Gyrator Transform for image encryption without hampering the features of the image. Further the autoencoder presents generating a feature vector for each compressed image that includes the metadata of each compressed image, and is associated with each image before storage in the distributed ledger. The autoencoder showcases generation of the compressed image that occupies only 0.28 times the memory size as compared to the original image.

- The application of the proposed AdapSR framework is not limited to self-reconfiguration of calibration parameters of the smart sensor network. Rather, the adaptive reconfiguration properties of the AdapSR framework can be used for reconfiguration of any set of parameters, and is not specific only to computer vision applications. To demonstrate the same, the DSA model is realized based on the architecture of AdapSR framework for dynamic allocation of speed limits to various locations (areas), and is used to establish a relationship between speed limit variation and possibility of accident avoidance.

7.2 Future Directions

In this thesis, numerous reconfiguration frameworks for active vision systems are investigated and explored in detail to provide novel contribution in this area. But there are some research dimensions that arise out of the current work which demand future study. These dimensions are summarized as directions to future work and are enlisted as follows:

- The models for the SAM framework as presented in Chapter 3 and the AdapSR framework as presented in Chapter 4 are simulated and tested on video datasets with a few limitations of the specifications of the video dataset. The efficiency of both the models is yet to be tested on real-time video feed.

- Deployment of the AdapSR framework (as presented in Chapter 4) is expensive due to the utilization of distributed blockchain network, and thus the AdapSR framework is presented and tested through simulations and model ideally. However, a real-life deployment of the AdapSR framework is yet to be achieved for real-time adaptive self-reconfiguration of a number of active vision systems.
- The AdapSR framework is presented and tested in Chapter 4 for a homogeneous sensor environment, however, improvements in the AdapSR framework is required to be developed for heterogeneous systems to broaden the scope of its applications in the future.
- The autoencoder presented in Chapter 5 is tested and presented for compression of the image datasets, however, the autoencoder model is yet to be modified and tested for encryption and compression of the parametric data, protocols, model data, and other data/metadata associated with the AdapSR model.
- The DSA framework as presented in Chapter 6 can be used to regularize the traffic speed based on the actual traffic density of each area, which requires development and deployment of the DSA model in cooperation with a network of computer vision system, and thus requires future advancements.

References

[1] M. Reisslein, B. Rinner, B. R. Chowdhury, “A Smart Camera Networks,” In Computer, vol. 47, no. 5, pp. 23–25, May 2014, doi: 10.1109/MC.2014.134.

[2] T. Zhang, W. Aftab, L. Mihaylova, C. L. Wheeler, S. Rigby, D. Fletcher, S. Maddock, G. Bosworth, “Recent Advances in Video Analytics for Rail Network Surveillance for Security, Trespass and Suicide Prevention - A Survey,” In Sensors, vol. 22, no.12, 4324, June 2022, doi: 10.3390/s22124324.

[3] R. Theagarajan, F. Pala, X. Zhang, B. Bhanu, “Soccer: Who has the ball? Generating visual analytics and player statistics,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, Utah, USA, pp. 1749–1757, June 2018, doi: 10.1109/CVPRW.2018.00227.

[4] C. Wu, A.H. Khalili, H. Aghajan, “Multiview activity recognition in smart homes with spatio-temporal features,” In Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras, New York, USA, pp. 142–149, September 2010, doi: 10.1145/1865987.1866010.

[5] S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, G. Wang, “Real-time obstacle detection and tracking for sense-and-avoid mechanism in UAVs,” In IEEE Transactions on Intelligent Vehicles, vol 3, no. 2, pp. 185–197, June 2018, doi: 10.1109/TIV.2018.2804166.

[6] M. Agarwal, P. Parashar, A. Mathur, K. Utkarsh, A. Sinha, “Suspicious Activity Detection in Surveillance Applications Using Slow-Fast Convolutional Neural Network,” In Proceedings of Advances in Data Computing, Communication and Security, Springer, Berlin/Heidelberg, Germany, vol. 106, pp. 647–658, March 2022, doi: 10.1007/978-981-16-8403-6_59.

[7] A. Hanson, “Introduction to Computer Vision Systems,” Book Chapter in Computer Vision Systems, 1st Edition, Elsevier, Amsterdam, The Netherlands, pp. 127-156, January 1978.

[8] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, G. L. Foresti, “Dynamic reconfiguration in camera networks: A short survey,” In IEEE Transactions on Circuits and Systems Video Technolgy, vol. 26, No. 52015, pp. 965–977, May 2016, doi: 10.1109/TCSVT.2015.2426575.

[9] T. C. Jesus, D. G. Costa, P. Portugal, F. Vasques, “A Survey on Monitoring Quality Assessment for Wireless Visual Sensor Networks,” In Future Internet, vol. 14(7), no. 213, pp. 1-26, July 2022, doi: 10.3390/fi14070213.

[10] S. Indu, S. Chaudhury, N.R. Mittal, A. Bhattacharyya, "Optimal sensor placement for surveillance of large spaces," In Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Como, Italy, pp. 1–8, October 2009, doi: 10.1109/ICDSC.2009.5289398.

[11] G. Zhang, B. Dong, J. Zheng, "Visual Sensor Placement and Orientation Optimization for Surveillance Systems," In Proceedings of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Krakow, Poland, pp. 1–5, November 2015, doi: 10.1109/BWCCA.2015.19.

[12] L. C. Da'Silva, R. M. Bernardo, H. A. De'Oliveira, P. F. Rosa, "Multi-UAV agent-based coordination for persistent surveillance with dynamic priorities," In Proceedings of the International Conference on Military Technologies (ICMT), Brno, Czech Republic, pp. 765–771, May 2017, doi: 10.1109/MILTECHS.2017.7988859.

[13] M. A. Jamshed, M. F. Khan, K. Rafique, M. I. Khan, K. Faheem, S. M. Shah, A. Rahim, "An energy efficient priority based wireless multimedia sensor node dynamic scheduler," In Proceedings of the 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET), Islamabad, Pakistan, pp. 1–4, December 2015, doi: 10.1109/HONET.2015.7395435.

[14] A. Vejdanparast, "Improving the Fidelity of Abstract Camera Network Simulations," Ph.D. Thesis, Aston University: Birmingham, UK, 2020.

[15] X. Wang, H. Zhang, H. Gu, "Solving Optimal Camera Placement Problems in IOT Using LH-RPSO," In IEEE Access, vol. 8, pp. 40881–40891, September 2019, doi: 10.1109/ACCESS.2019.2941069.

[16] N. J. Redding, J. F. Ohmer, J. Kelly, T. Cooke, "Cross-matching via feature matching for camera handover with non-overlapping fields of view," In Proceedings of the 2008 Digital Image Computing: Techniques and Applications, Canberra, ACT, Australia, pp. 343–350, January 2008, doi: 10.1109/DICTA.2008.38.

[17] L. Esterle, P. R. Lewis, M. Bogdanski, B. Rinner, X. Yao, "A socio-economic approach to online vision graph generation and handover in distributed Smart Camera Networks," In Proceedings of the 5th ACM/IEEE International Conference on Distributed Smart Cameras, Ghent, Belgium, pp. 1–6, August 2011, doi: 10.1109/ICDSC.2011.6042902.

[18] J. L. Lin, K. S. Hwang, C. Y. Huang, “Active and Seamless Handover Control of Multi-Camera Systems With 1-DoF Platforms,” In IEEE Systems Journal, vol. 8, no. 3, pp. 769–777, 2012, doi: 10.1109/JSYST.2012.2224611.

[19] E. L. Hall, J. B. Tio, C. A. Mc’Pherson, F. A. Sadjadi, “Measuring curved surfaces for robot vision,” In Computer, vol. 15, pp. 42–54, 1982, doi: 10.1109/MC.1982.1653915.

[20] R. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” In IEEE Journal of Robotic Automation, vol. 3, no. 4, pp. 323–344, 1987, doi: 10.1109/JRA.1987.1087109.

[21] O. D. Faugeras, “The Calibration Problem for Stereo,” In Proceedings of the Computer Vision and Pattern Recognition, Miami, FL, USA, vol. 52, pp. 15–20, 1986, doi: https://doi.org/10.1007/978-3-642-74567-6_15.

[22] J. Weng, P. Cohen, M. Herniou, “Camera calibration with distortion models and accuracy evaluation,” In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 10, pp 965–980, 1992, doi: 10.1109/34.159901.

[23] H. D. Whyte, T. Bailey, “Simultaneous localization and mapping: Part I,” In IEEE Robotics and Automation Management, vol 13, no. 2, pp. 99–110, 2006, doi: 10.1109/MRA.2006.1638022.

[24] H. D. Whyte, T. Bailey, "Simultaneous localization and mapping (SLAM): Part II,” In IEEE Robot. Autom. Mag., Vol 13, no. 3, pp. 108–117, 2006, doi: 10.1109/MRA.2006.1678144.

[25] O. Özyeşil, V. Voroninski, R. BAdapSRi, A. Singer, “A survey of structure from motion,” In Cambridge University Press Acta Numerica, vol. 26, pp. 305–364, May 2017, doi: 10.1017/S096249291700006X.

[26] D. Fox, W. Burgard, F. Dellaert, S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA, pp. 343-349, 1999, doi: http://robots.stanford.edu/papers/fox.aaai99.pdf.

[27] W. E. Mantzel, C. Hyeokho, G. B. Richard, “Distributed camera network localization,” In Proceedings of the Conference Record of the Thirty-Eighth Asilomar Conference on Signals,

Systems and Computers, Pacific Grove, CA, USA, Vol 2, pp. 1381–1386, November 2004, doi: 10.1109/ACSSC.2004.1399380.

[28] E. Brachmann, C. Rother, “Learning less is more-6d camera localization via 3d surface regression,” In Proceedings of the Computer Vision and Pattern Recognition, Salt lake city, Utah, USA, pp. 4654–4662, June 2018, doi: 10.48550/arXiv.1711.10228.

[29] Z. Tang, Y. S. Lin, K. H. Lee, J. N. Hwang, J. H. Chuang, Z. Fang, “Camera self-calibration from tracking of moving persons,” In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Maxico, pp. 265–270, December 2016, doi: 10.1109/ICPR.2016.7899644.

[30] C. Zheng, H. Qiu, C. Liu, X. Zheng, C. Zhou, Z. Liu, Z. J. Yang, “A Fast Method to Extract Focal Length of Camera Based on Parallel Particle Swarm Optimization,” In Proceedings of the 37th Chinese Control Conference (CCC), Wuhan, China, pp. 9550–9555, July 2018, doi: 10.23919/ChiCC.2018.8483981.

[31] G. Führ, C. R. Jung, “Camera self-calibration based on nonlinear optimization and applications in surveillance systems,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 5, pp. 1132–1142, 2015, doi: 10.1109/TCSVT.2015.2511812.

[32] Q. Yao, H. Sankoh, K. Nonaka, S. Naito, “Automatic camera self-calibration for immersive navigation of free viewpoint sports video,” In Proceedings of the 18th International Workshop on Multimedia Signal Processing (MMSP), Montreal, Canada, pp. 1–6, September 2016, doi: 10.1109/MMSP.2016.7813399.

[33] F. Li, H. Sekkati, J. Deglint, C. Scharfenberger, M. Lamm, D Clausi, J. Zelek, A. Wong, “Simultaneous projector-camera self-calibration for three-dimensional reconstruction and projection mapping,” IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 74–83, 2017, doi: 10.1109/TCI.2017.2652844.

[34] J. Heikkila, “Using sparse elimination for solving minimal problems in computer vision,” In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, pp. 76–84, October 2017, doi: 10.1109/ICCV.2017.18.

[35] Z. Tang, Y. S. Lin, K. H. Lee, J. N. Hwang, J. H. Chuang, “ESTHER :Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans,” In IEEE Access, vol. 7, pp. 10754–10766, 2019, doi: 10.1109/ACCESS.2019.2891224.

[36] D. Marinakis, G. Dudek, “Topology inference for a vision-based sensor network,” In Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05), Victoria, British Columbia, Canada, pp. 121–128, May 2005, doi : 10.1109/CRV.2005.81.

[37] A. Van Den Hengel, A. Dick, R. Hill, “Activity topology estimation for large networks of cameras,” In Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, Sydney, Australia, pp. 44, November 2006, doi: 10.1109/AVSS.2006.17.

[38] H. Detmold, A. V. D. Hengel, A. Dick, A. Cichowski, R. Hill, E. Kocadag, K. Falkner, D. S. Munro, “Topology estimation for thousand-camera surveillance networks,” In Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras, Vienna, Austria, pp. 195–202, September 2007, doi: 10.1109/ICDSC.2007.4357524.

[39] P. Clarot, E. B. Ermis, P. M. Jodoin, V. Saligrama, “Unsupervised camera network structure estimation based on activity,” In Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Como, Italy, pp. 1–8, 2009, doi: 10.1109/ICDSC.2009.5289362.

[40] X. Zou, B. Bhanu, B. Song, A. K. Roy-Chowdhury, “Determining topology in a distributed camera network,” In Proceedings of the IEEE International Conference on Image Processing, San Antonio, Texas, U.S.A, Volume 5, pp. V-133, 2007, doi: 10.1109/ICIP.2007.4379783.

[41] R. Farrell, L. S. Davis, “Decentralized discovery of camera network topology,” In Proceedings of the 2nd ACM/IEEE International Conference on Distributed Smart Cameras, Palo Alto, CA, USA, pp. 1–10, 2008, doi : 10.1109/ICIP.2007.4379783.

[42] M. Zhu, A. Dick, A. V. D. Hengel, “Camera network topology estimation by lighting variation,” In Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA), Adelaide, Australia, pp. 1–6, 2015, doi: 10.1109/DICTA.2015.7371245.

[43] G. Mali, M. Sudip, “TRAST: Trust-based distributed topology management for wireless multimedia sensor networks,” In IEEE Transactions on Computers, vol. 65, no. 6, pp. 1978–1991, 2016, doi: 10.1109/TC.2015.2456026.

[44] T. Feigang, Z. Xiaoju, L. Quanmi, L. Jianyi, “A Camera Network Topology Estimation Based on Blind Distance,” In Proceedings of the 11th International Conference on Intelligent

Computation Technology and Automation (ICICTA), Changsha, China, pp. 138–140, 2018, doi: 10.1109/ICICTA.2018.00039.

[45] Z. Li, J. Wang, J. Chen, “Estimating Path in camera network with non-overlapping FOVs,” In Proceedings of the 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, pp. 604–609, 2018, doi: 10.1109/ICSAI.2018.8599452.

[46] A. Kansal, M. B. Srivastava, “An environmental energy harvesting framework for sensor networks” In Proceedings of the 2003 International Symposium on Low Power Electronics and Design, Seoul, Korea, pp. 481–486, 2003, doi: 10.1145/871506.871624.

[47] M. Bramberger, M. Quaritsch, T Winkler, B. Rinner, H. Schwabach, “Integrating multi-camera tracking into a dynamic task allocation system for smart cameras,” In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Como, Italy, pp. 474-479, 2005, doi: 10.1109/AVSS.2005.1577315.

[48] M. Bramberger, B. Rinner, H. Schwabach, “A method for dynamic allocation of tasks in clusters of embedded smart cameras,” In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Waikolova, USA, vol. 3, pp. 2595–2600, 2005, doi: 10.1109/ICSMC.2005.1571540.

[49] D. R. Karuppiah, R. A. Grupen, Z. Zhu, A. R. Hanson, “Automatic resource allocation in a distributed camera network,” In Machine Vision Applications, vol. 21, pp. 517–528, 2010, doi: 10.1007/s00138-008-0182-7.

[50] B. Dieber, C. Micheloni, B. Rinner, “Resource-aware coverage and task assignment in visual sensor networks,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 10, pp. 1424–1437, 2011, 10.1109/TCSVT.2011.2162770.

[51] B. Dieber, L. Esterle, B. Rinner, “Distributed resource-aware task assignment for complex monitoring scenarios in visual sensor networks,” In Proceedings of the 6th International Conference on Distributed Smart Cameras (ICDSC), Hong Kong, China, pp. 1–6, 2012.

[52] C. Kyrkou, C. Laoudias, T. Theocharides, C. G. Panayiotou, M. Polycarpou, “Adaptive energy-oriented multitask allocation in smart camera networks,” In IEEE Embedded Systems Letter, vol. 8, no. 2, pp. 37–40, 2016, doi: 10.1109/LES.2016.2526071.

[53] Y. Wang, J. Zhang, Z. Liu, Q. Wu, P. A. Chou, Z. Zhang, Y. Jia, “Handling occlusion and large displacement through improved RGB-D scene flow estimation,” IEEE Transactions on

Circuits and Systems for Video Technology, vol. 26, no. 7, pp. 1265–1278, 2015, doi: 10.1109/TCSVT.2015.2462011.

[54] W. Ouyang, X. Zeng, X. Wang, “Partial occlusion handling in pedestrian detection with a deep model,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 11, pp. 2123–2137, 2015, doi: 10.1109/TCSVT.2015.2501940.

[55] M. I. Shehzad, Y. A. Shah, Z. Mehmood, A. W. Malik, S. Azmat, “K-means based multiple objects tracking with long-term occlusion handling,” In IET Computer Vision, vol. 11, pp. 68–77, 2016, doi: 10.1049/iet-cvi.2016.0156.

[56] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova, J. A. Chambers, “Multi-target tracking and occlusion handling with learned variational Bayesian clusters and a social force model.” In IEEE Transactions on Signal Processing, vol. 64, pp. 1320–1335, 2015, doi: 10.1109/TSP.2015.2504340.

[57] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, “Vision-based occlusion handling and vehicle classification for traffic surveillance systems,” In IEEE Intelligent Transportation System Management, vol. 10, no. 2, pp. 80–92, 2018, doi: 10.1109/MITS.2018.2806619.

[58] S. Zhao, S. Zhang, L. Zhang, “Towards occlusion handling: Object tracking with background estimation,” In IEEE Transactions on Cybernetics, vol. 48, no. 7, pp. 2086–2100, 2017, doi: 10.1109/TCYB.2017.2727138.

[59] Y. Liu, X. Y. Jing, J. Nie, H. Gao, J. Liu, G. P. Jiang, “Context-Aware Three-Dimensional Mean-Shift with Occlusion Handling for Robust Object Tracking in RGB-D Videos,” In IEEE Transactions on Multimedia, vol. 21, no. 3, pp. 664–677, 2018, doi: 10.1109/TMM.2018.2863604.

[60] X. Feng, Y. Jiang, X. Yang, M. Du, X. Li, “Computer vision algorithms and hardware implementations: A survey,” In Integration, vol. 69, pp. 309–320, 2019, doi: 10.1016/j.vlsi.2019.07.005.

[61] S. A. Hørup, S. A. Juul, H. H. Larsen, “The Art of General-Purpose Computations on Graphics Processing units”, Technical Project Report, Aalborg University, Aalborg, Denmark, 2011,https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f1879480362c92e8dfc720005fcbe2c7ffeeac5b.

[62] Y. Guo, J. Liu, G. Li, L. Mai, H. Dong, “Fast and Flexible Human Pose Estimation with HyperPose,” In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, pp. 3763–3766, 2021, doi: 10.1145/3474085.3478325.

[63] S. Tan, B. Knott, Y. Tian, D. J. Wu, “CryptGPU: Fast privacy-preserving machine learning on the GPU,” In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp. 1021–1038, 2021, doi: 10.1109/SP40001.2021.00098

[64] H. Irmak, D. Ziener, N. Alachiotis, “Increasing Flexibility of FPGA-based CNN Accelerators with Dynamic Partial Reconfiguration,” In Proceedings of the 31st International Conference on Field-Programmable Logic and Applications (FPL), Dredsen, Germany, pp. 306–311, 2021, doi: 10.1109/FPL53798.2021.00061.

[65] A. Costa, N. Corna, F. Garzetti, N. Lusardi, E. Ronconi, A. Geraci, “High-Performance Computing of Real-Time and Multichannel Histograms: A Full FPGA Approach,” In IEEE Access, vol. 10, pp. 47524–47540, 2022, doi: 10.1109/ACCESS.2022.3169760.

[66] M. A. Carbajal, R. P. Villa, D. E. Palazuelos, G. J. Astorga, Rubio Astorga, “Reconfigurable Digital FPGA Based Architecture for 2-Dimensional Linear Convolution Applications,” In Identitad Energetica, Madrid, Spain, 2021, doi: http://www.cinergiaug.org/Revista/RIE_V4_N1_Dic2021.pdf.

[67] H. Xiong, K. Sun, B. Zhang, J. Yang, H. Xu, “Deep-Sea: A Reconfigurable Accelerator for Classic CNN,” Wirel. Commun. Mob. Comput., vol. 2022, no. 4726652, 2022, doi: 10.1155/2022/4726652.

[68] L. Wei, L. Peng, “An Efficient OpenCL-Based FPGA Accelerator for MobileNet,” In Journal of Physics: Conference Series, vol. 1883, no. 012086, 2021, doi: 10.1088/1742-6596/1883/1/012086.

[69] R. Szeliski, “Scene Reconstruction from multiple cameras,” In Proceedings of the International Conference on Image Processing (ICISP), Vancouver, BC, Canada, vol. 1, pp. 13–16, 2000, doi: 10.1109/ICIP.2000.900880.

[70] B. Micušık, D. Martinec, T. Pajdla, “3D metric reconstruction from uncalibrated omnidirectional images,” In Proceedings of the Asian Conf. on Comp. Vision (ACCV’04), Jeju Island, Korea, 2014, doi: ftp://cmp.felk.cvut.cz/pub/cmp/articles/micusik/Micusik-ACCV2004.pdf.

[71] L. Peng, Y. Zhang, H. Zhou, T. Lu, “A robust method for estimating image geometry with local structure constraint,” In IEEE Access, vol. 6, pp. 20734–20747, 2018, doi: 10.1109/ACCESS.2018.2803152.

[72] D. N. Brito, C. F. Nunes, F. L. Padua, A. Lacerda, “Evaluation of interest point matching methods for projective reconstruction of 3D scenes,” In IEEE Latin American Transactions, vol. 14, no. 3, pp. 1393–1400, 2016, doi: 10.1109/TLA.2016.7459626.

[73] S. Milani, “Three-dimensional reconstruction from heterogeneous video devices with camera-in-view information,” In Proceedings of the IEEE International Conference on Image Processing (ICIP), Quebec, Canada; pp. 2050–2054, 2015, doi: 10.1109/ICIP.2015.7351161.

[74] H. Aliakbarpour, V. S. Prasath, K. Palaniappan, G. Seetharaman, J. Dias, “Heterogeneous multi-view information fusion: Review of 3-D reconstruction methods and a new registration with uncertainty modeling,” In IEEE Access, vol. 4, pp. 8264–8285, 2016, doi: 10.1109/ACCESS.2016.2629987.

[75] C. Wang, X. Guo, “Plane-Based Optimization of Geometry and Texture for RGB-D Reconstruction of Indoor Scenes,” In Proceedings of the International Conference on 3D Vision (3DV), Verona, Italy; pp. 533–541, 2018, doi: 10.1109/3DV.2018.00067.

[76] D. Ma, G. Li, L. Wang, “Rapid Reconstruction of a Three-Dimensional Mesh Model Based on Oblique Images in the Internet of Things,” In IEEE Access, vol. 6, pp. 61686–61699, 2018, doi: 10.1109/ACCESS.2018.2876508.

[77] K. Ichimaru, R. Furukawa, H. Kawasaki, “CNN based dense underwater 3D scene reconstruction by transfer learning using bubble database,” In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikolova Village, Hawaii, USA, pp. 1543–1552, 2019, doi: 10.1109/WACV.2019.00169.

[78] P. Viola, M. Jones, “Robust real-time object detection,” In International Journal of Computer Vision, vol. 4, pp. 34–47, 2001, doi: 10.1023/B:VISI.0000013087.49260.fb.

[79] P. Piccinini, A. Prati, R. Cucchiara, “Real-time object detection and localization with SIFT-based clustering,” Image Vis. Comput., vol. 30, no. 8, pp. 573–587, 2012, doi: 10.1016/j.imavis.2012.06.004.

[80] N. Dalal, B. Triggs, “Histograms of oriented gradients for human detection,” In Proceedings of the IEEE computer society conference on computer vision and pattern

recognition (CVPR'05), San Diago, California, USA, vol 1, pp. 886–893, 2005, doi: 10.1109/CVPR.2005.177.

[81] S. Aslani, H. Mahdavi-Nasab, “Optical flow based moving object detection and tracking for traffic surveillance,” In International Journal of Electrical, Compututer, Energetic Electronics and Communication Engineering, vol. 7(9), pp. 1252–1256, 2013, doi.org/10.5281/zenodo.1088502.

[82] J. Huang, W. Zou, J. Zhu, Z. Zhu, “Optical flow based real-time moving object detection in unconstrained scenes,” arXiv 2018, arXiv:1807.04890.

[83] S. Tougaard, “Practical algorithm for background subtraction,” In Surface Science, Elsevier, vol. 216, no. 3, pp. 343–360, 1989, doi: 10.1016/0039-6028(89)90380-4.

[84] J. Rieke, “Object detection with neural networks-a simple tutorial using keras,” In Towards Data Science, Vol. 6(12), 2017, https://towardsdatascience.com/object-detection-with-neural-networks-a4e2c46b4491.

[85] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You only look once: Unified, real-time object detection,” In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 779–788, 2016, doi: 10.48550/arXiv.1506.02640.

[86] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In Proceedings of the Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[87] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, “Single-shot refinement neural network for object detection,” In Proceedings of the Computer Vision and Pattern Recognition, Salt lake city, USA, pp. 4203–4212, 2018, doi: 10.48550/arXiv.1711.06897.

[88] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, “Libra R-CNN: Towards balanced learning for object detection,” In Proceedings of the Computer Vision and Pattern Recognition, Long beach, CA, USA, pp. 821–830, 2019, dio: 10.48550/arXiv.1904.02701.

[89] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, “SSD: Single shot multibox detector,” In Proceedings of the European Conference on Computer Vision, October 11-14, 2016, Amsterdam, The Netherlands; pp. 21–37, doi: 10.48550/arXiv.1512.02325.

[90] S. M. Roy, A. Ghosh, “Real-time adaptive Histogram Min-Max Bucket (HMMB) model for background subtraction,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 7, pp. 1513–1525, 2017, doi: 10.1109/TCSVT.2017.2669362.

[91] W. Min, M. Fan, X. Guo, Q. Han, “A new approach to track multiple vehicles with the combination of robust detection and two classifiers,” In IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 174–186, 2017, doi: 10.1109/TITS.2017.2756989.

[92] Y. Wu, X. He, T.Q. Nguyen, “Moving object detection with a freely moving camera via background motion subtraction,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, pp. 236–248], 2015, doi: 10.1109/TCSVT.2015.2493499.

[93] W. Hu, Y. Yang, W. Zhang, Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition,” In IEEE Transactions on Image Processing, vol. 26, no. 2, pp. 724–737, 2016, doi: 10.1109/TIP.2016.2627803.

[94] H.S. Parekh, D.G. Thakore, U.K. Jaliya, “A survey on object detection and tracking methods,” In International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, pp. 2970–2979, 2014, https://www.rroij.com/open-access/a-survey-on-object-detection-and-tracking-methods.pdf.

[95] A. Yilmaz, O. Javed, M. Shah, “Object tracking: A survey,” In ACM Computer Survey, vol. 38, no. 4, pp. 13, 2006, doi: 10.1145/1177352.1177355.

[96] C. J. Du, H. J. He, D. W. Sun, “Object Classification Methods,” In Computer Vision Technology for Food Quality Evaluation; Academic Press: Cambridge, MA, USA, 2016, pp. 87–110, doi: 10.1016/B978-0-12-802232-0.00004-9.

[97] M. Ankerst, C. Elsen, M. Ester, H.P. Kriegel, “Visual classification: An interactive approach to decision tree construction,” In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diago, California, USA, pp. 392–396, 1999, doi: 10.1145/312129.312298.

[98] S. Anurag, Eu. Han, V. Kumar, V. Singh, “Parallel formulations of decision-tree classification algorithms,” In High Performance Data Mining; Springer, Boston, MA, pp. 237–261, 1998, doi: 10.1109/ICPP.1998.708491.

[99] F. Schroff, A. Criminisi, A. Zisserman, “Object Class Segmentation using Random Forests,” In Proceedings of the British Machine Vision Conference, University of Leeds, England, pp. 1–10, 2008, doi: 10.5244/C.22.54.

[100] T. Bayes, “LII - An essay towards solving a problem in the doctrine of chances,” technical Letter by the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S,” Philos. Trans. R. Soc. Lond., vol 53, pp. 370–418, 1763, doi: 10.1098/rstl.1763.0053.

[101] K. M. Leung, “Naive Bayesian Classifier,” Thesis Report, Polytechnic University Department of Computer Science/Finance and Risk Engineering, New York, USA, pp. 123–156, 2007, doi: 10.1128/AEM.00062-07.

[102] I. Kononenko, “Semi-Naive Bayesian Classifier,” In European Working Session on Learning, Springer, Berlin, Heidelberg, pp. 206–219, 1991, doi: 10.1007/BFb0017015.

[103] W.R. Klecka, R.I. Gudmund, W.R. Klecka, “Discriminant Analysis,” Book, Sage, New York, NY, USA, 1980; Volume 19.

[104] S. Menard, “Interpretting the canonical discriminant functions,” Book Chapter in Applied Logistic Regression Analysis, vol. 19, New York, USA, 2002, pp-23-40.

[105] T. Hastie, R. Tibshirani, “Discriminant adaptive nearest neighbor classification,” In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 6, pp. 607–616, 1996, doi: 10.1109/34.506411.

[106] K. S. Durgesh, B. Lekha, “Data classification using support vector machine,” Journal of Theoritical Applied Information Technology, vol. 12, pp. 1–7, 2010, https://www.researchgate.net/publication/285663733_Data_classification_using_support_vector_machine.

[107] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back, “Face recognition: A convolutional neural-network approach,” In IEEE Transactions on Neural Networks, vol. 8, pp. 98–113, 1997, doi: 10.1109/72.554195.

[108] F. Murtagh, “Multilayer perceptrons for classification and regression,” In Neurocomputing, vol. 2, pp. 183–197, 1991, doi: 10.1016/0925-2312(91)90023-5.

[109] N. Jmour, S. Zayen, A. Abdelkrim, “Convolutional neural networks for image classification,” In Proceedings of the International Conference on Advanced Systems and

Electric Technologies (IC_ASET), Hammamet, Tinisia, pp. 397–402, 2018, doi: 10.1109/ASET.2018.8379889.

[110] J. P. De'Villiers, F. W. Leuschner, R. Geldenhuys, "Centi-pixel accurate real-time inverse distortion correction," In Proceedings of the International Symposium on Optomechatronic Technologies, West Harbor, San Diago, CA, USA, vol 7266, pp. 726611, 2008, doi: 10.1117/12.804771.

[111] B. Caprile, V. Torre, "Using vanishing points for camera calibration," In International Journal of Computer Vision, vol. 4, pp. 127–139, 1990, doi: 10.1007/BF00127813.

[112] A. Wang, T. Qiu, L. Shao, "A simple method of radial distortion correction with centre of distortion estimation," In Journal of Mathematical Imaging and Vision, vol. 35, pp. 165–172, 2009, doi: 10.1007/s10851-009-0162-1.

[113] R. Hartley, S.B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 8, pp. 1309–1321, 2007, doi: 10.1109/TPAMI.2007.1147.

[114] K. Huang, S. Ziauddin, M. Zand, M. Greenspan, "One Shot Radial Distortion Correction by Direct Linear Transformation," In Proceedings of the IEEE International Conference on Image Processing (ICIP), Abu Dhabi, Dubai, pp. 473–477, 2020, doi: 10.1109/ICIP40778.2020.9190749.

[115] H. Zhao, Y. Shi, X. Tong, X. Ying, H. Zha, "A Simple Yet Effective Pipeline For Radial Distortion Correction," In Proceedings of the IEEE International Conference on Image Processing (ICIP), Abu Dhabi, Dubai, pp. 878–882, 2020, doi: 10.1109/ICIP40778.2020.9191107.

[116] Y. M. Wang, Y. Li, J. B. Zheng, "A camera calibration technique based on OpenCV," In Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences, Chengdu, China, pp. 403–406, 2010, doi: 10.1109/ICICIS.2010.5534797.

[117] S. Lee, H. Hong, "A robust camera-based method for optical distortion calibration of head-mounted displays," In IEEE Virtual Reality (VR), Lake Buena Vista, FL, USA, 2013, pp. 27-30, doi: 10.1109/VR.2013.6549353.

[118] Z. Wang, M. Liu, S. Yang, S. Huang, X. Bai, X. Liu, J. Zhu, X. Liu, Z. Zhang, “Precise full-field distortion rectification and evaluation method for a digital projector,” In Optical Review, vol. 23, pp. 746–752, 2016, doi: 10.1007/s10043-016-0255-1.

[119] S. Yang, M. Srikanth, D. Lelescu, K. Venkataraman, “Systems and Methods for Depth-Assisted Perspective Distortion Correction,” In U.S. Patent application no. 9,898,856, 2018.

[120] G. Finlayson, H. Gong, R.B. Fisher, “Color homography: Theory and applications,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 1, pp. 20–33, 2017, doi: 10.1109/TPAMI.2017.2760833.

[121] H. Wang, J. Yang, B. Xue, X. Yan, J. Tao, “A novel color calibration method of multi-spectral camera based on normalized RGB color model,” In Results in Phyics, vol. 19, pp. 103498, 2020, doi: 10.1016/j.rinp.2020.103498.

[122] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, X. Pan, “Mat: Motion-aware multi-object tracking,” In Neurocomputing, vol. 476, pp. 75–86, 2022, doi: 10.1016/j.neucom.2021.12.104.

[123] T. Meinhardt, A. Kirillov, L. Leal-Taixe, C. Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, pp. 8844–8854, 2022, doi: 10.48550/arXiv.2101.02702.

[124] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pp. 1337–1342, 2003, doi: 10.1109/TPAMI.2003.1233909.

[125] W. Zhang, B. Ma, K. Liu, R. Huang, “Video-based pedestrian re-identification by adaptive spatio-temporal appearance model,” In IEEE Transactions on Image Processing, vol 26, pp. 2042–2054, 2017, doi: 10.1109/TIP.2017.2672440.

[126] X. Yang, M. Wang, D. Tao, “Person re-identification with metric learning using privileged information,” In IEEE Transactions on Image Processing, vol. 27, pp. 791–805, 2017, doi: 10.1109/TIP.2017.2765836.

[127] S. Geng, M. Yu, Y. Guo, Y. Yu, “A Weighted Center Graph Fusion Method for Person Re-Identification,” In IEEE Access, vol. 7, pp. 23329–23342, 2019, doi: 10.1109/ACCESS.2019.2898729.

[128] X. Yang, Y. Tang, N. Wang, B. Song, X. Gao, “An End-to-End Noise-Weakened Person Re-Identification and Tracking with Adaptive Partial Information,” In IEEE Access, vol. 7, pp. 20984–20995, 2019, doi: 10.1109/ACCESS.2019.2899032.

[129] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, J. Luo, “Anatomy-aware 3d human pose estimation with bone-based pose decomposition,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, pp. 198–209, January 2021, doi: https://doi.org/10.48550/arXiv.2002.10322 .

[130] M. Straka, S. Hauswiesner, M. Rüther, H. Bischof, “Skeletal Graph Based Human Pose Estimation in Real-Time,” In Proceedings of the BMVC, Dundee, UK, pp. 1–12, 2011, doi: 10.5244/C.25.69.

[131] L. W. Campbell, A. F. Bobick, “Using phase space constraints to represent human body motion”, In Proceedings of the International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 338–343, 1995, doi: 10.1109/ICCV.1995.466880.

[132] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio, “Patch-Based Experiments with Object Classification in Video Surveillance”, In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Puerto Rico, vol. 97, pp. 193–199, 1997, doi.org/10.1007/978-3-540-74607-2_26.

[133] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, “Image captioning with semantic attention,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 4651–4659, 2016, doi: 10.48550/arXiv.1603.03925.

[134] P. Wcg, “Role of the manuscript reviewer,” In Medical Journal, Singapore, vol. 50, pp. 931–934, 2009.

[135] J. F. Polak, “The role of the manuscript reviewer in the peer review process,” Am. J. Roentgenol., vol. 165, pp. 685–688, 1995.

[136] H. Nguyen, B. Bhanu, A. Patel, R. Diaz, “VideoWeb: Design of a wireless camera network for real-time monitoring of activities,” In Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Como, Italy, pp. 1–8, 2009, doi: 10.1109/ICDSC.2009.5289418.

[137] O.W. Ibraheem, A. Irwansyah, J. Hagemeyer, M. Porrmann, U. Rueckert, “Reconfigurable vision processing system for player tracking in indoor sports,” In Proceedings

of the Conference on Design and Architectures for Signal and Image Processing (DASIP), Dresden, Germany; pp. 1–6, 2017, doi: 10.1109/DASIP.2017.8122114.

[138] Y. Xiang, A. Alahi, S. Savarese, "Learning to Track: Online multi-object tracking by decision making," In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, pp. 4705–4713, 2015, doi: 10.1109/ICCV.2015.534.

[139] I. Laptev, B. Caputo, "Recognizing human actions: A local SVM approach," In Proceedings of the 17th International Conference on Pattern Recognition, Washington DC, USA, pp. 32–36, 2004, doi: 10.1109/ICPR.2004.1334462.

[140] A.N. Duy, M. Yoo, "Calibration-Net: LiDAR and Camera Auto-Calibration using Cost Volume and Convolutional Neural Network," In Proceedings of the 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea, pp. 141–144, 2022, doi: 10.1109/ICAIIC54071.2022.9722671.

[141] Y. Cao, H. Wang, H. Zhao, X. Yang, "Neural-Network-Based Model-Free Calibration Method for Stereo Fisheye Camera," In Front, Bioeng. Biotechnol., vol. 10, 955233, 2022, doi: 10.3389/fbioe.2022.955233.

[142] H. Chen, S. Munir, S. Lin, "RFCam: Uncertainty-aware Fusion of Camera and Wi-Fi for Real-time Human Identification with Mobile Devices," In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol 6, No. 2, pp. 1–29, July 2022, doi : 10.1145/3534588.

[143] T. T. Dufera, Y. C. Seboka, C. F. Portillo, "Parameter Estimation for Dynamical Systems Using a Deep Neural Network," In Applications of Computer Intelligence and Soft Computing, vol. 2022, no. 2014510, 2022, doi: 10.1155/2022/2014510.

[144] A. Doula, A. Sanchez Guinea, M. Mühlhäuser, "VR-Surv: A VR-Based Privacy Preserving Surveillance System," In Proceedings of the CHI Conference on Human Factors in Computing Systems Extended Abstracts, New Orleans, NY, USA, pp. 1–7, 2022, doi: 10.1145/3491101.3519645.

[145] C. Pooja, K. Jaisharma, "Novel Framework for the Improvement of Object Detection Accuracy of Smart Surveillance Camera Visuals Using Modified Convolutional Neural Network Technique Compared with Global Color Histogram," In ECS Transactions, vol. 107, no. 18823, 2022, doi: 10.1149/10701.18823ecst.

[146] T. Jiang, Q. Zhang, J. Yuan, C. Wang, C. Li, “Multi-Type Object Tracking Based on Residual Neural Network Model,” In Symmetry, vol. 14, no. 8, 2022, doi: 10.3390/sym14081689.

[147] T. Jaganathan, A. Panneerselvam, S.K. Kumaraswamy, “Object detection and multi-object tracking based on optimized deep convolutional neural network and unscented Kalman filtering,” In Concurrncy and Computation: Practice and Experience, vol. 34, no.25, 2022, doi: 10.1002/cpe.7245.

[148] T. R. Deshpande, S. U. Sapkal, “Development of Object Tracking System Utilizing Camera Movement and Deep Neural Network,” In Proceedings of the 2022 IEEE Region 10 Symposium (TENSYMP), Mumbai, India, pp. 1–6, 2022, doi: 10.1109/TENSYMP54529.2022.9864420.

[149] S. M. Praveenkumar, P. Patil, P. S. Hiremath, “Real-Time Multi-Object Tracking of Pedestrians in a Video Using Convolution Neural Network and Deep SORT,” In Proceedings of the ICT Systems and Sustainability, (ICT4SD), Goa, India, pp. 725–736, 2022, doi: 10.1007/978-981-16-5987-4_73.

[150] M. Jhansi, S. Bachu, N. U. Kumar, M. A. Kumar, “IODTDLCNN: Implementation of Object Detection and Tracking by using Deep Learning based Convolutional Neural Network,” In Proceedings of the 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichy, India, pp. 1–6, 2022, doi: 10.1109/ICEEICT53079.2022.9768632.

[151] J. Barazande, N. Farzaneh, “WSAMLP: Water Strider Algorithm and Artificial Neural Network-based Activity Detection Method in Smart Homes,” J. AI Data Min., vol. 10, pp. 1–13, 2022, doi: 10.22044/jadm.2021.10781.2215.

[152] P. K. Y. Wong, H. Luo, M. Wang, J.C. Cheng, “Enriched and discriminative convolutional neural network features for pedestrian re-identification and trajectory modelling,” In Computer Aided Civil Infrastructural Engineering, vol. 37, pp. 573–592, 2022, doi: 10.1111/mice.12750.

[153] Y. Yao, X. Jiang, H. Fujita, Z. Fang, “A sparse graph wavelet convolution neural network for video-based person re-identification,” In Pattern Recognition, vol. 129, 108708, 2022, doi: 10.1016/j.patcog.2022.108708.

[154] M. Mohana, S. Alelyani, M. S. Alsaqer, “Fused Deep Neural Network based Transfer Learning in Occluded Face Classification and Person re-Identification,” In Online article arXiv:2205.07203, 2022, doi: 10.48550/arXiv.2205.07203.

[155] C. You, H. Zheng, Z. Guo, T. Wang, T. Wu, “Tampering detection and localization base on sample guidance and individual camera device convolutional neural network features,” In Expert Systems, vol. 40, no. 1, August 2022, doi: 10.1111/exsy.13102.

[156] S. Karamchandani, S. Bhattacharjee, D. Issrani, R. Dhar, “SLAM Using Neural Network-Based Depth Estimation for Auto Vehicle Parking,” In IOT with Smart Systems, Springer, Berlin/Heidelberg, Germany, pp. 37–44, 2022, doi: 10.1007/978-981-16-3945-6_5.

[157] Expert Market Search, “AI In Computer Vision Market based on Component (Hardware, Software), Vertical (healthcare, security, automotive, agriculture, sports & entertainment, and others), and Region–Global Forecast to 2027,” Research Report by Expert Market Search, 2023, https://www.expertmarketresearch.com/reports/ai-in-computer-vision-market.

[158] N. Andriyanov, “Methods for preventing visual attacks in convolutional neural networks based on data discard and dimensionality reduction,” In Applied Sciences, vol. 11, no. 11, 5235, 2021, doi: 10.3390/app11115235.

[159] B. Wang, M. Zhao, W. Wang, X. Dai, Y. Li, Y. Guo, “Adversarial Analysis for Source Camera Identification,” In IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 11 pp. 4174–4186, 2021, doi: 10.1109/TCSVT.2020.3047084.

[160] C. Zhang, P. Benz, C. Lin, A. Karjauv, J. Wu, I.S. Kweon, “A survey on universal adversarial attack,” In 30th International Joint Conference on Artificial Intelligence, Montreal, pp. 4687- 4694, August 2021, doi: 10.48550/arXiv.2103.01498.

[161] D. M. Edwards, E. D. Rawat, “Study of Adversarial Machine Learning with Infrared Examples for Surveillance Applications,” In Electronics, vol. 9, no. 8, 1284, August 2020, doi: 10.3390/electronics9081284.

[162] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, “A survey on adversarial attacks and defences,” CAAI Trans. Intell. Technol., vol. 6, pp. 25–45, 2021, doi: 10.1049/cit2.12028.

[163] N. Akhtar, A. Mian, N. Kardan, M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," IEEE Access, vol. 9, pp. 155161–155196, 2021, doi: 1 0.1109/ACCESS.2021.3127960.

[164] M. A. R. Ahad, "Motion History Images for Action Recognition and Understanding", Book Chapter in Action Representations, 2013, pp. 19-29, doi.org/10.1007/978-1-4471-4730-5.

[165] X. Pan, Y. Guo, A. Men, "Traffic Surveillance System for Vehicle Flow Detection," In Proceedings of Second International Conference on Computer Modeling and Simulation, Sanya, China, pp. 314-318, 2010, doi: 10.1109/ICCMS.2010.75.

[166] F. Mehboob, M. Abbas, R. Almotaeryi, R. Jiang, S.A. Maadeed, A. Bouridane, "Traffic Flow Estimation from Road Surveillance," IEEE International Symposium on Multimedia (ISM), Miami, FL, USA, pp. 605-608, 2015, doi: 10.1109/ISM.2015.14.

[167] M. Stuede, M. Schappler, "Non-Parametric Modeling of Spatio-Temporal Human Activity Based on Mobile Robot Observations," In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, doi: 10.1109/IROS47612.2022.9982067.

[168] S. Sattar, Y. Sattar, M. Shahzad, M. M. Fraz, "Group Activity Recognition in Visual Data: A Retrospective Analysis of Recent Advancements," In Proceedings of International Conference on Digital Futures and Transformative Technologies (ICoDT2), IEEE, Islamabad, Pakistan, pp. 1-8, 2021, doi: 10.1109/ICoDT252288.2021.9441478.

[169] L. Zhao, Y. Gao, J. Ye, F. Chen, Y. Ye, C.T. Lu, N. Ramakrishnan, "Online dynamic multi-source feature learning and its application to spatio-temporal event forecasting," In ACM transactions on knowledge discovery from data, vol. 1, no. 1, 2021, https://cs.emory.edu/~lzhao41/materials/papers/TKDD2020_preprinted.pdf.

[170] L. Yuanqiang, H. Jing, "A Sports Video Behavior Recognition Using Local Spatiotemporal Patterns," In Mobile Information Systems, vol. 2022, 2022. doi: 10.1155/2022/4805993.

[171] R. Yan, X. Shu, C. Yuan, Q.Tian, J. Tang, "Position-aware participation-contributed temporal dynamic model for group activity recognition," In IEEE Transactions on Neural

Networks and Learning Systems, Early Access, vol. 3, no. 12, 2021, doi: 10.1109/TNNLS.2021.3085567.

[172] Shashank, I. Sreedevi, "Distributed Network of Adaptive and Self-Reconfigurable Active Vision Systems," In Symmetry, vol. 14, no. 11, p-2281, 2022, doi: https://doi.org/10.3390/sym14112281.

[173] A.T. Ali, E. L. Dagless, "Computer vision for security surveillance and movement control," In IEE Colloquium on Electronic Images and Image Processing in Security and Forensic Science, pp. 6-1, 1990, https://ieeexplore.ieee.org/document/190222.

[174] F.Y. Shih, O.R. Mitchell, "Automated fast recognition and location of arbitrarily shaped objects by image morphology," In Proceedings CVPR'88: The Computer Society Conference on Computer Vision and Pattern Recognition, pp. 774-775, 1988, doi: 10.1109/CVPR.1988.196322.

[175] Shashank, S. Indu, "Sensitivity-Based Adaptive Activity Mapping for Optimal Camera Calibration," In Proceedings of International Conference on Intelligent Computing and Smart Communication, Springer, 2019, Tehri, India, pp. 1211-1218, doi.org/10.1007/978-981-15-0633-8_118.

[176] J.I. Isaac, J. Martin, R. Barco, "A low-complexity vision-based system for real-time traffic monitoring," In IEEE Transactions on Intelligent Transportation Systems, vol. 18, no.5, pp. 1279-1288, 2016, doi: 10.1109/TITS.2016.2603069.

[177] K. Achim, M. Sefati, S. Arya, A. Rachman, K. Kreisköther, P. Campoy, "Towards Multi-Object Detection and Tracking in Urban Scenario under Uncertainties," In Proceedings of 4th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS, Funchal, Madeira, Portugal, pp. 156-167, 2018, doi: 10.5220/0006706101560167.

[178] G. Rikke, T.B. Moeslund, "Constrained multi-target tracking for team sports activities," In IPSJ Transactions on Computer Vision and Applications, vol. 10, no. 1, pp. 1-11, 2018, doi: 10.1186/s41074-017-0038-z.

[179] Shashank, Indu Sreedevi, "Spatiotemporal activity mapping for enhanced multi-object detection with reduced resource utilization," In Electronics, vol. 12, no. 1, p-37, 2022, https://doi.org/10.3390/electronics12010037.

[180] J. C. SanMiguel, C. Micheloni, K. Shoop, G.L. Foresti, A. Cavallaro, “Self-reconfigurable smart camera networks,” Computer, vol. 47, no. 5, pp. 67–73, 2014, doi: 10.1109/MC.2014.133.

[181] B. Rinner, L. Esterle, J. Simonjan, G. Nebehay, R. Pflugfelder, G.F. Dominguez, P.R. Lewis, “Self-aware and self-expressive camera networks,” Computer, vol. 48, no. 7, pp. 21–28, 2015, doi: 10.1109/MC.2015.209.

[182] P. R. Lewis, A. Chandra, K. Glette, “Self-awareness and Self-expression: Inspiration from Psychology,” In Self-Aware Computing Systems, Springer, Berlin/Heidelberg, Germany, pp. 9–21, 2016, doi: 10.1007/978-3-319-39675-0_2.

[183] K. Glette, P.R. Lewis, A. Chandra, “Relationships to Other Concepts,” In Self-aware Computing Systems, Springer, Berlin/Heidelberg, Germany, pp. 23–35, 2016, doi: 10.1007/978-3-319-39675-0_3.

[184] S. Wang, G. Nebehay, L. Esterle, K. Nymoen, L.L. Minku, “Common Techniques for Self-awareness and Self-expression,” In Self-Aware Computing Systems. Natural Computing Series, Springer, Berlin/Heidelberg, Germany, pp. 113–142, doi: 10.1007/978-3-319-39675-0_7.

[185] A. Isavudeen, N. Ngan, E. Dokladalova, M. Akil, “Auto-adaptive multi-sensor architecture,” In Proceedings of the International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, pp. 2198–2201, 2016, doi: 10.1109/ISCAS.2016.7539018.

[186] Z. Guettatfi, P. Hübner, M. Platzner, B. Rinner, “Computational self-awareness as design approach for visual sensor nodes,” In Proceedings of the 12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), Madrid, Spain, pp. 1–8, 2017, doi: 10.1109/ReCoSoC.2017.8016147.

[187] Z. Zhu, Y. Luo, S. Chen, G. Qi, N. Mazur, C. Zhong, Q. Li, “Camera style transformation with preserved self-similarity and domain-dissimilarity in unsupervised person re-identification,” In Journal of Visual Communication and Image Representation, vol. 80, 2021, doi:10.1016/j.jvcir.2021.103303.

[188] S. Lin, J. Lv, Z. Yang, Q. Li, W.S. Zheng, “Heterogeneous graph driven unsupervised domain adaptation of person re-identification,” In Neurocomputing, vol. 471, pp. 1–11, 2022, doi: 10.1016/j.neucom.2021.11.009.

[189] M. Wu, C. Li, Z. Yao, “Deep Active Learning for Computer Vision Tasks: Methodologies, Applications, and Challenges,” In Applied Science, vol. 12, no. 16 , 2022, doi: 10.3390/app12168103.

[190] S. Rudolph, S. Tomforde, J. Hähner, “On the Detection of Mutual Influences and Their Consideration in Reinforcement Learning Processes,” 2019, doi: 10.48550/arXiv.1905.04205.

[191] L. Cai, H. Ma, Z. Liu, Z. Li, Z. Zhou, “Coverage Control for PTZ Camera Networks Using Scene Potential Map,” In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, pp. 1–6, 2022, doi: 10.1109/ICME52920.2022.9859676.

[192] S. Suresh, V. Menon, “An Efficient Graph Based Approach for Reducing Coverage Loss From Failed Cameras of a Surveillance Network,” In IEEE Sensors Journal, vol. 22, no. 8, pp. 8155–8163, 2022, doi: 10.1109/JSEN.2022.3157819.

[193] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 39, no. 06, pp. 1137-1149, 2017, doi: 10.48550/arXiv.1506.01497,doi: 10.1109/TPAMI.2016.2577031.

[194] K. Liang, “Fission: A Provably Fast, Scalable, and Secure Permissionless Blockchain,”, Technical Article in Cryptography and Security, Cornell University, december 2018, doi: 10.48550/arXiv.1812.05032.

[195] W. Zhao, “On Nxt Proof of Stake Algorithm: A Simulation Study,” In IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 1–12, 2022, doi: 10.1109/TDSC.2022.3193092.

[196] M.S. Milan, P. C. Elisabet , "Secure image encryption and authentication using the photon counting technique in the Gyrator domain," In Proceedings of IEEE 20th Symposium on Signal Processing, Images and Computer Vision (STSIVA), pp. 1-6, 2015, doi: 10.1109/STSIVA.2015.7330460.

[197] J.A.Rodrigo, T. Alieva, M.L.Colvo, “Applications of gyrator transform for image processing,” In Optics Communications, vol. 278, no.2, pp. 279-284, 2007, doi: 10.1016/j.optcom.2007.06.023.

[198] S. P. Chang, J. J. Dian, “Properties, Digital Implementation, applications and self image phenomenon of Gyrator Transform,” In Proceedings of 17th European Signal Processing Conference(EUSIPCO), pp. 441-445, 2009, https://ieeexplore.ieee.org/document/7077823.

[199] S. Yeom, B. Javidi, E. Watson, “Photon counting passive 3D image sensing for automatic target recognition,” Optics express, vol. 13, no. 23, pp.9310-9330, 2005, doi: 10.1364/OE.15.001513.

[200] Shashank, Indu Sreedevi, “Cryptography for Biometric Fingerprint Information Using Gyrator Transform” In Proceedings of International Conference on Signal Processing VLSI and Communication Engineering, IEEE, 2019, https://10.1109/ICSPVCE46182.2019.

[201] Delhi Traffic police, “Road accidents in Delhi 2019” https://delhitrafficpolice.nic.in/sites/default/files/uploads/2020/Road-accident-in-delhi2019.pdf.

[202] Cameron, H. Max , E. Rune, "Nilsson's Power Model connecting speed and road trauma: Applicability by road type and alternative models for urban roads,” Accident Analysis & Prevention, vol. 42, no. 6, pp. 1908-1915, 2010, doi: 10.1016/j.aap.2010.05.012.

Appendix-A

List of Publications

1. **Shashank**, Indu Sreedevi, "**Spatiotemporal activity mapping for enhanced multi-object detection with reduced resource utilization**," In Electronics, **SCIE Journal (IF: 2.94)**, vol.12, no. 1, p-37, 2022. https://doi.org/10.3390/electronics12010037.

2. **Shashank**, Indu Sreedevi, "**Distributed Network of Adaptive and Self-Reconfigurable Active Vision Systems**," In Symmetry, **SCIE Journal (IF:. 2.69)**, vol. 14, no. 11, p-2281, 2022. https://doi.org/10.3390/sym14112281.

3. **Shashank,** Indu Sreedevi, **"Sensitivity-Based Adaptive Activity Mapping for Optimal Camera Calibration."** in **International Conference** on Intelligent Computing and Smart Communication 2019, pp. 1211-1218. Springer, Singapore, 2020**.** https://doi.org/10.1007/978-981-15-0633-8_118

4. **Shashank,** Indu Sreedevi**, "Cryptography for Biometric Fingerprint Information Using Gyrator Transform" International Conference** on Signal Processing VLSI and Communication Engineering**,** IEEE, 2019**. (Accepted and Presented).** https:// 10.1109/ICSPVCE46182.2019

Appendix-B

List of Patents Filed

1. **Shashank**, Professor Indu Sreedevi, Delhi Technological University, Indian patent Application No. IN**202311030910** titled "**System and method for reconfiguration of a sensing unit**" filed at Indian Patent Office (IPO).

2. **Shashank**, Professor Indu Sreedevi, Delhi Technological University, Indian patent Application No. IN**202311030913** titled "**System and method for adaptive reconfiguration of sensor networks**" filed at Indian Patent Office (IPO).

Appendix-C

Biodata

Shashank completed his **B.Tech** degree in Electronics and Communication Engineering (with CGPA 8.38/10) from **Jamia Millia Islamia university, New Delhi** in 2014, and **M.Tech** degree (with CGPA 8.08/10) specializing in Microwave and Optical Communication Engineering from **Delhi Technological university**, **New Delhi** in 2016. Shashank joined **Delhi Technological University, New Delhi** as a full time Ph.D. Scholar in Electronics and Communication department under the supervision of **Professor S. Indu** in 2017. His areas of research interest are Computer vision, Artificial Intelligence, Smart Sensor Networks, and Blockchain Technology. During the period of his Ph.D. research, Shashank has published two papers in SCIE indexed Journals, and has presented two papers in International Conferences. He has also filed two Indian Patent Applications to protect his research models developed through his Ph.D. research. More particularly, Shashank has extensively researched on developing frameworks for adaptive self-reconfiguration of active vision systems. He has proposed various robust and efficient frameworks and models in this research area.