A NOVEL DATA MINING FRAMEWORK FOR OBJECT DETECTION IN VIDEO SURVEILLANCE SYSTEM USING CELLULAR LOGIC ARRAY

By

SURENDER SINGH (SAP ID: 500033583)

(SCHOOL OF COMPUTER SCIENCE AND ENGINEERING)

Submitted

IN PARTIAL FULFILLMENT OF THE REQUIREMENT OF THE DEGREE OF DOCTOR OF PHILOSOPHY

(ENGG. - COMPUTER SCIENCE)

TO

UPES

UNIVERSITY OF PETROLEUM & ENERGY STUDIES, DEHRADUN

March 2018

Under the Guidance of:

Dr. Ajay Prasad

Associate Professor,

SOCSE, UPES, Dehradun, Uttarakhand

Dr. Kingshuk Srivastava

Assistant Professor

SOCSE, UPES, Dehradun, Uttarakhand

Dr. Suman Bhattacharya

Project Head, IPR Management Services

TCS, Bhubaneswar, Orissa

ACKNOWLEDGMENT

First and foremost, my all devotions are to my Satguru, Radhaswami Dayal, Paramsant, Kanwar Sahib Ji Maharaj for his countless blessings. My unrestrained appreciation goes to my advisors; Dr. Ajay Prasad, Dr. Kingshuk Srivastava, Dr. Suman Bhattacharya for all the help and support they have extended to me throughout the course of this work. I simply cannot imagine how things would have proceeded without their help, support, patience, and contributions. I also extend my special acknowledgment to Dr. Manish Prateek, Director, SoCSE, UPES, India, Dr. Manas Ranjan Pradhan, Dean, INTI, Malaysia for their valuable time in consultations and discussions. I am also thankful to Dr. E. G. Rajan, Founder Director, Pentagram Research Centre (P) Limited, Hyderabad, for providing insight and expertise on the cellular logic array that greatly assisted the research.

I also acknowledge my many colleagues and friends as I had a pleasant, enjoyable and fruitful company with them. I am also extremely grateful to the celebrated authors whose noteworthy works have been consulted and referred in my research work. Finally, I wish to express my gratitude to my family members especially my wife for being patient with me and offering words of encouragement to spur my spirit at moments of depression.

Surender Singh

DECLARATION BY SCHOLAR

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or another institute of higher learning, except where due acknowledgment has been made in the text.

SURENDER SINGH
SAP ID: 500033583
Date: May, 2018
Place: Dehradun

UPES
UNIVERSITY WITH A PURPOSE

This is to certify that the thesis entitled "**A NOVEL DATA MINING FRAMEWORK FOR OBJECT DETECTION IN VIDEO SURVEILLANCE SYSTEM USING CELLULAR LOGIC ARRAY**" is being submitted by **Surender Singh (500033583)** in fulfillment for the Award of DOCTOR OF PHILOSOPHY in (Computer Science & Engineering) to the University of Petroleum and Energy Studies. Thesis has been corrected as per the evaluation reports dated 08/01/2019 and all the necessary changes / modifications have been inserted/incorporated in the thesis.

Guide:
Dr. Ajay Prasad
Associate Professor
SCS, UPES, Dehradun

Co-Guide:
Dr. Kingshuk Srivastava
Assistant Professor
SCS, UPES, Dehradun

Dean, School of Computer Science

Date: 7/2/2019

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

UPES

www.upes.ac.in

THE NATION BUILDERS UNIVERSITY

THESIS COMPLETION CERTIFICATE

This is to certify that the thesis on **"A Novel Data Mining Framework for Object Detection in Video Surveillance System using Cellular Logic Array"** submitted by **Surender Singh** to School of Computer Science and Engineering, UPES, Dehradun in Partial completion of the requirements for the award of the Degree of Doctor of Philosophy (Engineering- Computer Science) is an original work carried out by him under our joint supervision and guidance.

It is further certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.

Internal Guide

Dr. Ajay Prasad
Associate Prof. SoCSE,
UPES, Dehradun

External Guide

Dr. Suman Bhattacharya
Project Head, IPR Management Services
TCS, Bhubaneswar, Orissa

Internal Co-Guide

Dr. Kingshuk Srivastava
Assistant Prof. SoCSE,
UPES, Dehradun

DECLARATION BY SCHOLAR

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or another institute of higher learning, except where due acknowledgment has been made in the text.

SURENDER SINGH

Date: April 17, 2018

Place: Dehradun

THESIS COMPLETION CERTIFICATE

This is to certify that the thesis on **"A Novel Data Mining Framework for Object Detection in Video Surveillance System using Cellular Logic Array"** submitted by **Surender Singh** to School of Computer Science and Engineering, UPES, Dehradun in Partial completion of the requirements for the award of the Degree of Doctor of Philosophy (Engineering- Computer Science) is an original work carried out by him under our joint supervision and guidance.

It is further certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.

Dr. Ajay Prasad	**Dr. Suman Bhattacharya**
(Guide)	(External Guide)

Dr. Kingshuk Srivastava

(Co-guide)

TABLE OF CONTENT

values of neighboring cells. The cellular logic array-based framework uses pattern-directed search and replace techniques and by virtue of its working, it is inherently parallel and guarantees speed and precision. Digital images by virtue of its cellular automaton type configuration easily fit into cellular logic array processing (CLAP). This was the motivation behind using CLAP for obtaining effective and efficient results in object detection in video surveillance. The main objective of this research work is to design and implement a CLAP based data mining framework for fast, effective and accurate object detection in video surveillance system.

Open Source Computer Vision (OpenCV) and CodeBlocks are used for implementing the existing algorithms from scratch. OpenCV is a widely accepted cross-platform and open source libraries of C++ for manipulating digital images and real-time computer vision. At a later stage, MATLAB is also used to implement the existing system and the proposed algorithm as this is easy to use and flexible enough for experimenting with various options or modifications in algorithms. The CDnet2012 dataset has been used for evaluation and validation of the existing and the proposed algorithms. The data set consists of six categories namely baseline, camera jitter, dynamic background, intermittent object motion, shadow and thermal imagery scenes captured from different types of cameras in different lighting conditions, level of noise and compression techniques to throw a challenging test environment for evaluation purpose.

Effectivity and efficiency are two criteria used for measuring the ability of object detection algorithms. Effectivity is measured using Recall, Precision, F1 score or (F-measure which is harmonic mean of precision and recall) and PSNR metrics which are frequently used for the evaluation of effectivity or quality of detection in literature. Recall and Precision are two most used metrics for binary classifier but often requires a trade-off between themselves because Recall favors methods with a low false negative rate and Precision favors methods with a low false positive rate. The overall effect of both is represented by an F1 score. All three metrics along with Peak-Signal-to-Noise Ratio (PSNR) and Precision-Recall (PR) trade-off curves are used to measure the quality of detection. The best trade-off threshold is

EXECUTIVE SUMMARY

There are scores of applications such as hot pursuit of a criminal by law enforcement agencies, video record of accidents, automatic parking systems, object detection, complying with safety standards in production centres, objects tracking, crowd management, real-time monitoring of water logging to manage sewage and drainage systems require real-time event management in which an effective surveillance technique can achieve better results. The real-time management necessitates video surveillance system which detects situations in video flow that represent a security threat and trigger an alarm accordingly. Human monitoring and analysis of surveillance video are complex and multi-technology processes. Especially in the multi-camera environment, it is a very labor-intensive, error-prone and costly affair. Widespread use of security cameras and the huge size of data call for new frameworks for analyzing video data and image automatically to reduce the cost of reviewing and analysis in real time. The current techniques of video data mining are slow and inaccurate making these less useful for real-time video analytics. Existing video surveillance systems take care of video capture, store, and transmission of video to remote places but devoid of efficient threat detection and analysis leaving these functions exclusively to human operators for manual analysis. Therefore, there is an urgent need for a surveillance system which is fast, efficient and accurate.

Several methods have been proposed for object identification and tracking in video data mining literature, but nearly all of these process an image or video sequentially either in spatial or frequency domain or both. Cellular logic array-based representation of images and processing can inherently parallelize image processing techniques. Various image processing techniques have been devised using cellular automata but not tested on data mining processes for video surveillance purpose. A cellular logic array is a collection of homogeneous cells whose values evolve iteratively through a number of discrete steps. The cell values are simultaneously updated by a logic or formula involving calculation from the

values of neighboring cells. The cellular logic array-based framework uses pattern-directed search and replace techniques and by virtue of its working, it is inherently parallel and guarantees speed and precision. Digital images by virtue of its cellular automaton type configuration easily fit into cellular logic array processing (CLAP). This was the motivation behind using CLAP for obtaining effective and efficient results in object detection in video surveillance. The main objective of this research work is to design and implement a CLAP based data mining framework for fast, effective and accurate object detection in video surveillance system.

Open Source Computer Vision (OpenCV) and CodeBlocks are used for implementing the existing algorithms from scratch. OpenCV is a widely accepted cross-platform and open source libraries of C++ for manipulating digital images and real-time computer vision. At a later stage, MATLAB is also used to implement the existing system and the proposed algorithm as this is easy to use and flexible enough for experimenting with various options or modifications in algorithms. The CDnet2012 dataset has been used for evaluation and validation of the existing and the proposed algorithms. The data set consists of six categories namely baseline, camera jitter, dynamic background, intermittent object motion, shadow and thermal imagery scenes captured from different types of cameras in different lighting conditions, level of noise and compression techniques to throw a challenging test environment for evaluation purpose.

Effectivity and efficiency are two criteria used for measuring the ability of object detection algorithms. Effectivity is measured using Recall, Precision, F1 score or (F-measure which is harmonic mean of precision and recall) and PSNR metrics which are frequently used for the evaluation of effectivity or quality of detection in literature. Recall and Precision are two most used metrics for binary classifier but often requires a trade-off between themselves because Recall favors methods with a low false negative rate and Precision favors methods with a low false positive rate. The overall effect of both is represented by an F1 score. All three metrics along with Peak-Signal-to-Noise Ratio (PSNR) and Precision-Recall (PR) trade-off curves are used to measure the quality of detection. The best trade-off threshold is selected for each method in each scenario for comparison purpose but algorithms are ranked on the basis of visual analysis of P-R curves. PSNR helps in finding the level of dissimilarity between extracted motion and corresponding ground truth. Its correlation with the F1 score is an additional verification of the outcome. Finally, the efficiency of the algorithm is measured by a crude form of absolute time measuring i.e. Execution Time. It is measured in seconds or milliseconds on an i3 3rd generation Pentium 2.4 GHz processor and taken as the average time taken by the algorithm in the processing of a single frame. It is averaged over the whole range of video sequence except training frames.

In the primary phase of research, major problems related to object detection in the current video surveillance system are identified. A thorough literature survey has been done to identify the requirements of video surveillance systems and to evaluate the existing techniques for the identification of various factors that influence the design of video surveillance framework. The empirical investigation on various existing object motion detection techniques of background subtraction and background modeling categorizes the present algorithms into two sets based on reliability and speed; 1) the highly reliable but complex & time-consuming such as Kernel Density Estimation and Histogram Detection and 2) the fast but average on reliability algorithms such as Adaptive Mean Background Subtraction (AM BGS) and Adaptive Median Background Subtraction (AMD BGS). As fast algorithms are only suitable for real-time video surveillance, AM and AMD algorithms are selected for making improvement by the application of the CLAP in further research.

In the second step, the key algorithm of object detection is formulated to improve AM and AMD by extracting edges using CLAP on the difference of the current edge frame and edge map background to mine objects in motion. Edges are the most visual parts of a video frame, which are perceived by the human visual system, these are extracted from the difference of the current frame and the background frame. Then, the edge difference is de-noised and filled up to register the motion part in a frame. Most of edge detection programs work on pixel level by identifying the convex region of same gray level intensity and extracting edges by isolating pixel having a differing level of intensity by thresholding. Two different images may be shot in quite a different setup and they can differ in background intensities greatly due to variation in the reflectance, illumination, orientation, and depth of scene surfaces necessitating different values of threshold in edge detection process. This gives us an idea that instead of fixing a single value for all types of images and scenarios or empirically identifying threshold values each time during edge detection process; a formula must be devised which is dependent on background intensities so that good quality edge detection can materialize every time. A mathematical model is presented for this.

Once the adaptive threshold is conceptualized then edge-based motion detection algorithm is implemented using CLAP edge using global as well as the local threshold of 16×16 blocks of the image. These algorithms are further quantitatively compared with existing edge algorithm such as Sobel and Canny edge methods for object detection. Although remarkable results are achieved as compared to other standard edge algorithms such as Canny and Sobel, only a little success is achieved in a dynamic environment in comparison to basic BGS methods. Adaptive thresholding methods (global as well as local) do not improve BGS much in comparison to fix thresholding method.

Subsequently, a new concept of '*Local Neighborhood Differencing'* (LND) is devised to be used in BGS algorithm using CLAP in which average difference between current frame and background is computed on 9-pixel neighborhood for determining foreground pixel. The CLAP is used to make the whole procedure of differencing fast enough for real-time processing. Instead of processing for every pixel, which is very slow, the whole frames of the current frame (I) and background (B) are shifted one step in all eight directions of a pixel to find the absolute local neighborhood difference at once for the whole frame. This processing is similar to cellular logic array processing in which every cell value is updating itself by a rule defined over the cell neighborhood. This processing is easily implemented by using CLAP processing and it helps greatly in suppressing the false positives (noisy pixel) and increasing recall rate. This results in the appreciable improvement of motion detection in all scenarios except camera jitter.

The new framework for data mining for object detection using cellular logic array has been designed by considering the specific requirement of video processing and the common flaws of the existing techniques. First, suitable algorithms which need improvement are identified through a literature survey and empirical evaluation. Subsequently, CLAP-EDGE and CLAP-LND methods are formulated and implemented. The proposed methods are validated by comparing with existing system quantitatively using benchmarked dataset by using popularly used metrics such as Recall and Precision. Integrated measures such as F1 score and PSNR are used for result validations. Results are also compared with the average time taken per frame for determining the suitability of the proposed algorithms in real-time surveillance scenario. So, the predefined research objective has been attained by a CLAP based novel data mining framework of object detection which improves the existing detection algorithms in video surveillance significantly in all scenarios except camera jitter. It requires a necessary multi-modal background modeling method to reduce noise which can be addressed in future research.

Keywords: Cellular Logic Array Processing (CLAP), Background Subtraction (BGS), Adaptive Mean (AM BGS), Adaptive Median (AMD BGS), Video Surveillance, Image Processing, Edge Detection, Local Neighborhood Differencing (LND)

LIST OF ABBREVIATIONS

AM	Adaptive Mean
AMD	Adaptive Median
ANN	Artificial Neural Networks
BCS	Basic Sequential Clustering
BG	Background
BGS	Background Subtraction
CB	Code Book
CF	Current Frame
CLAP	Cellular Logic Array Processing
CW	Code Words
DOG	Difference Of Gaussians
ET	Execution Time
FD	Frame Differencing
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
HD	Histogram Detection
KDE	Kernel Density Estimation
K-MEANS	K-means Clustering Based Detection
LND	Local Neighborhood Differencing

LOG	Laplacian Of Gaussians
LVPS	Logical Video Processing System
MOG	Mixture Of Gaussians
MPR	Mean Performance ratio
OpenCV	Open Source Computer Vision
PCA	Principal Component Analysis
PDF	Probability Distribution Function
PR	Precision-Recall
PSNR	Peak Signal To Noise Ratio
SG	Single Gaussian
SVM	Support Vector Machines
TN	True Negative
TP	True Positive

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

OBJECT DETECTION IN VIDEO SURVEILLANCE SYSTEM

1.1 INTRODUCTION

Manual monitoring of surveillance cameras is extremely difficult especially in multiple and moving cameras environment. It has been identified practically that a person can attend multiple screens for only 20 minutes without losing his/her focus on the screen (Green, Travis, & Downs, 2014). This shows that manual analysis of a huge number of surveillance cameras is quite difficult, costly and erroneous. Thus, it becomes necessary to identify methods which are robust, fast and precise in tracking objects of interest in a surveillance video. In an automated mode, a factual video scene might be clattered by many real-time factors such as lightning, shadows, clouds, occlusions and slow or abrupt changes in luminance. The problem is further compounded by variation in scale, shape, the position of the object of interest making detection and tracking a very difficult and erroneous process. Therefore, there is need of methods that can provide photometric and geometric invariant detection of objects by fulfilling real-time processing requirements of video surveillance systems.

These days, video surveillance systems are gaining popularity as a cost-effective security tool due to new development in technology and cheaper security cameras. Surveillance is used to provide security blanket across the target area and to monitor activities in real time and to sound the alarm automatically for suspicious activities (Dammalapati, Gera, & Gera, 2013). Recorded video is also required to investigate past events; locate or trace vehicles, objects or people. Various applications of video surveillance system like disaster information and management system, law and order management system, personnel training, criminals, vehicles, and object locating system etc. require real-time processing of video streams. Managing real-time events such as hot pursuit of a criminal by law enforcement agencies, video record of accidents, automatic parking systems, object detection, complying with safety standards in production centres, objects tracking, crowd management, real-time monitoring of water logging to manage sewage and drainage systems are some of the other applications where better results can be achieved with the help of an effective surveillance techniques.

In real time, video surveillance systems detect situations in video flow that represent a security threat and trigger an alarm accordingly. These systems can be classified into three types; Operator controlled automated video surveillance and intelligent video surveillance systems (Vishwakarma & Agrawal, 2013). In an operator controlled surveillance system, the video stream is analyzed manually; a person observes the video to determine if there is an on-going activity that requires an action. In the second approach, the automated video surveillance system uses motion detection techniques to determine response. An intelligent video surveillance system is that which extracts the relevant information from generic motion accurately and issue actions.

1.2 MOTIVATION AND NEED OF RESEARCH

Human monitoring and analysis of surveillance video is a complex and multi-technology process (Frejlichowski, Gościewska, Forczmański, Nowosielski, & Hofman, 2013). Especially in the multi-camera environment, it is a very labor-intensive, error-prone and costly task (Somhorst, 2012). Widespread use of security cameras has made it necessary to come up with a new framework for analyzing video data and image automatically to reduce the cost of reviewing and analysis in real time. At the same time, business managers have started to look into the mega size video data for useful inferences to make the optimum decision for business processes improvement which will lead to higher profitability and market capture in their business (Hakeem, et al., 2012). As per the annual reports of Seagate

Technology (Canfield, 2014), in the near future, more than two-thirds of the top companies are thinking to employ video surveillance security systems for operations improvement as well as security. HIS Markit@ Technology Inc. forecasted that by 2019, each day, video surveillance cameras installed globally, will produce 2.5 Exabytes of data (Cropley, 2015). This is a huge data and current techniques of video data mining are slow and inaccurate making these less useful for real-time video analytic (Venetianer & Deng, 2010). Existing video surveillance systems take care about video capture, store, and transmission of video to remote places but devoid of efficient threat detection and analysis system leaving these functions exclusively to human operators for manual analysis (Elliott, 2010). Therefore, there is an urgent need for a surveillance system which is fast, efficient and accurate.

Several methods have been proposed for object identification and tracking in video data mining literature (Vijayakumar & Nedunchezhian, 2012). But nearly all of these process an image or video sequentially either in spatial or frequency domain or both. Cellular logic array-based representation of images and processing can inherently parallelize image processing techniques (Rajan, 1993). Till now, image processing techniques have been implemented using cellular automata but not tested on data mining processes for video surveillance purpose. The problem of object identification is mapped to the cellular logic array by representing images or video frames as cellular automata and then rules are defined to modify these representations using various basic algorithms such as thinning, edge detection, registration, image erosion and dilation in order to process video streams (Rajan, 1993). Automatic video surveillance system has its own peculiar requirement of speed and robustness, especially for real-time analysis. Therefore, this research work strives to propose and implement a cellular logic array based data mining framework for fast, effective and accurate object detection in video surveillance system.

1.3 VIDEO SURVEILLANCE SYSTEM ARCHITECTURE

A Video surveillance system is comprised of scores of cameras to capture video, recording system to store video and a common channel for fast connectivity between different modules and to distribute the video feeds to central monitoring locations (Somhorst, 2012). It also requires high capacity servers to store the video streams and analyze these streams to raise alarms for suspicious activities and recognition of people and objects. A surveillance system also contains a mechanism to broadcast video-derived intelligence to managers. A good video system should have an ability to integrate video feeds (real video) with other sources of intelligence such as communication intercepts and primarily human intelligence (logical video) (United States Patent No. US8773532B2, 2014).

Video Surveillance which is used to monitor people, vehicles, equipment, and event of interest remotely, comprises several components from video capturing to video processing and analysis to presentation with the following sequenced stages; video capture module, video stream selection, video processing and measurement and finally visualization (Borges, Conci, & Cavallaro, 2013). In general, the processing framework of an automated video surveillance system includes the following stages:

1. Video Capture Module
2. Video Selection Module
3. Video Processing Module
4. Human Machine Interface Module.

Video capture and selection modules are responsible for streaming the desired frames of video for treating into processing module. Video selection module can convert multiple streams from multiple cameras into a single stream by employing a fuzzy-based selection methodology (Morioka, Kovacs, Joo-Ho, & Korondi, 2010). It can be decided on the basis of several parameters such as distance, illumination, accuracy etc. Video processing module is related to digital image processing techniques such as object identification and tracking methods (Honghai, Shengyong, & Naoyuki, 2013). This module processes video frames automatically and helps in detecting objects (peoples, equipment, vehicles) and event of interest for security purposes.

The proposed video surveillance system architecture is shown in **Fig. 1-1**. Thin line shows control structure while thick arrow signifies data streaming. Except for Video capture module, all other three modules are working under the framework of Logical Video Processing system. The algorithms employed in this system will be developed using cellular logic array framework that uses pattern-directed Search and Replace (SAR) techniques and by its working, it is inherently parallel, and therefore, it guarantees speed and precision.

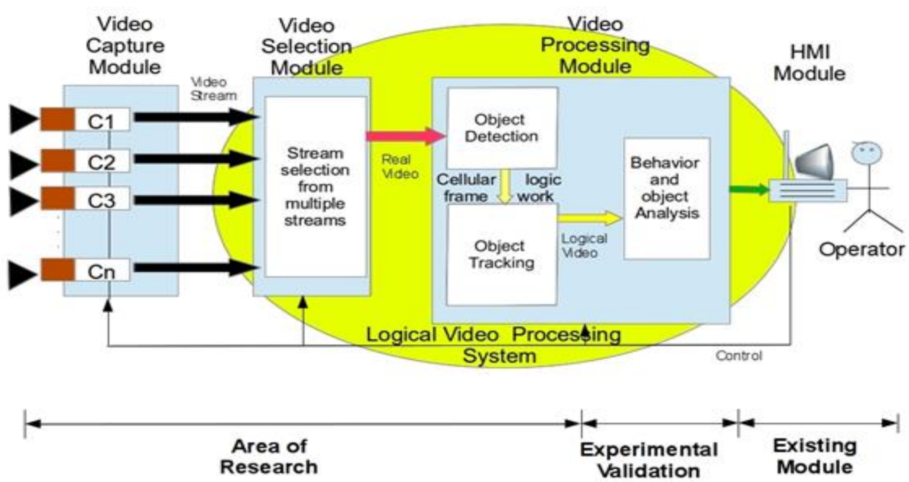


Fig. 1-1: The Proposed Architecture of CLAP based object detection

1.4 CELLULAR LOGIC ARRAY PROCESSING

The Cellular automaton was discovered by Neumann and Ulam in the 1950s but was made popular by Conway's "Game of Life" and Wolfram's paper, "Statistical mechanics of cellular automata". The cellular automaton is simple and straightforward. In the simplest form, it is an array of cells that update the homogeneous value in the cells simultaneously by following simple and local updating rules. Each cell value also called state of the cell is affected by the state of its neighborhood cells and greatly mimic natural phenomena. Although nature is continuous and so its event, the artificial cellular automaton which is implemented in discrete states in iterative mode can effectively simulate natural phenomenon and other problems which can be adapted in cellular automata configuration. A discrete state of the cellular array is called configuration. One of the greatest property of cellular automata is that they are self-organizing. Local neighborhood's effect is translated into a global phenomenon and can emulate the complex natural organizations. Different Components of Cellular Automaton include Lattice network of cells; number of cells; Shape of lattice (square, hexagonal etc.); dimension (single to multi-dimension); neighborhood depends on dimension and level of processing (popular neighborhoods are 3-neighborhood in single dimension, 5 and 9-neighborhood in two dimensions, 27-neighborhood in 3-dimensions).

Cellular Logic Array processing (CLAP) is based on cellular automata which are homogeneous structures or iterative cells of poly-dimensions. In cellular automata, each cell can have a finite state and a neighborhood defined by the number of cells it surrounds. Initially, at time t=0, a state is assigned randomly or seeded to each of cells and then a new generation of cells are evolved by following some rules which are homogeneous and exploits parallelism and local in nature. All cell states are updated by the same set of rules. Typically, the updating rule is same for each cell and does not change over time, and is applied to the whole grid simultaneously. Therefore, CLAP is essentially a computer algorithm that exhibits discreteness in space and time and operates on arrays of elements. The concept was applied to scores of applications in which time and space can be easily divided homogeneously. Image or video processing is one of such applications (Rajan, 1993) where the image is assumed as 2D-cellular automata and each pixel's value is updated using some updating rule. The type and extent of the neighborhood with updating rule define the application of image processing. Normally, the range of neighborhood is taken as $n \times n$ with processing pixel taken as a central element where n is usually an odd number for providing symmetry to operation. The working on cellular automata is shown in Algorithm 1-1.

Step 1: Seed initial configuration (Initialize cells by some initial values-specific or random depends on application)

Step 2: For each cell, calculate next state based on the current states of its neighbors.

Step 3: Simultaneously update all cells

Step 4: Go to step 2 until a stopping criterion is reached. (Stopping criterion may be some specific state or simply number of iterations)

Algorithm 1-1: Working of Cellular Automata

1.4.1 Applications of Cellular Automata

Cellular Automata models are easy to implement and fast, making them suitable for scores of application in physics, astronomy, biology and computer science. These are able to emulate many physical phenomena accurately. In computer science, besides image processing, these are being actively used for information security, encryption, simulating complex biological phenomenon such as honey bee work organization, genetic algorithm, neural network excitation which are further used to solve the complex problem of infinite search space.

1.4.2 Cellular Automata based Framework

The problem of object detection can be mapped to the cellular logic array by representing images or video frames as cellular automata and rules to modify these representations will be taken from various basic algorithms such as thinning, edge detection, registration, image erosion and dilation in order to process video streams. For better understanding the working of cellular automata or logic cellular array, a framework is given in **Fig. 1-2**: The proposed framework of the moving edge detection method to elucidate the process of moving edge detection from a sequence of video which is also used in our research work.

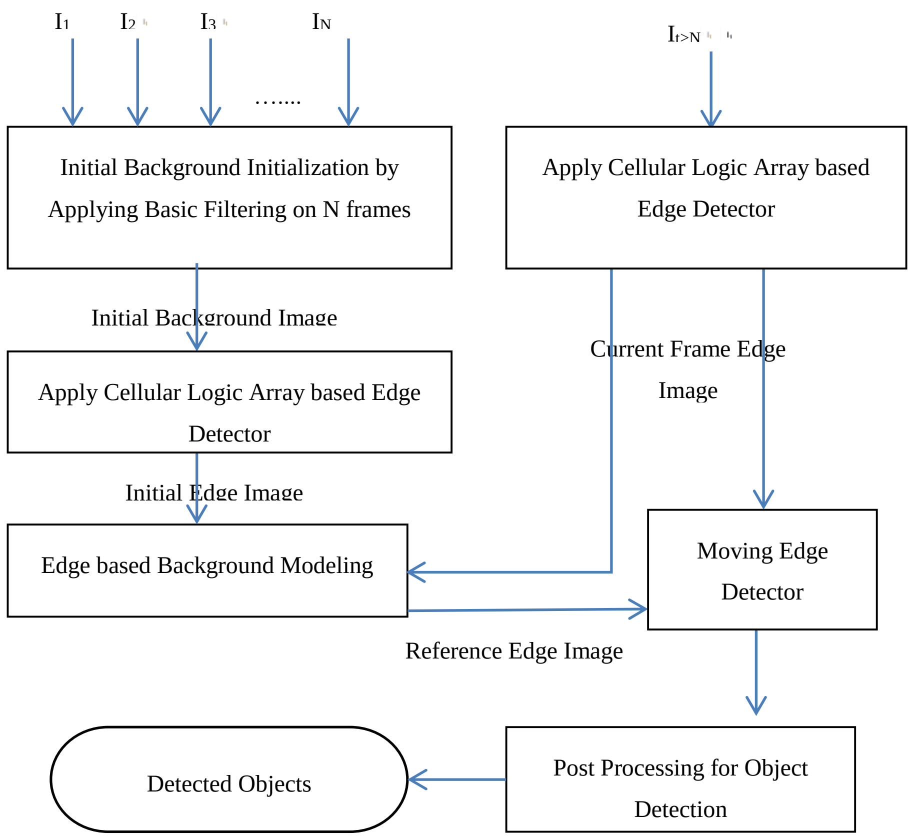


Fig. 1-2: The proposed framework of the moving edge detection method

Initially, N number of frames are used for preparing a background image. Cellular logic array based edge detection is used to find the initial edge background image which is later updated and maintained regularly by new edge image frames. Edge-based background modeling generates reference edge image which is used to detect moving edges. These moving edges are post-processed to find moving objects. A detailed method of object detection using edges' map is given in (Ramirez-Rivera, Murshed, & Chae, 2011). As this method uses a cellular logic array in edge detection and post-processing activities, it is faster and more robust than the existing methods.

1.5 VIDEO DATA MINING TECHNIQUES

Video data mining is a process which automatically extracts content and discovers patterns of the structure of the video, features of moving objects, spatial or temporal correlations of those features, objects activities, video events, etc. from vast amounts of video data. Video data mining can be classified in pattern detection, video clustering and classification and video association mining (Vijayakumar & Nedunchezhian, 2012). In case of images and frames of videos, the logic array of elements are image pixels which are processed by many digital image processing techniques such as motion/object detection, object classification, object tracking (Parekh, Thakore, & Jaliya, 2014). Video Data-mining means to unravel and explore knowledge from the video database. Consequently, these approaches can be divided into the following classifications:

1.5.1 Video Structure and Pattern Mining of Data

It is related to the syntactic level arrangement of information which is used to uncover the patterns in the video content. These patterns are further used to infer knowledge and decision making in related applications. It is an important step in machine learning and data analytics. Video structure mining is to ascertain the structure and patterns in the content for fast access to data from video streams while scene clustering and event mining are related to semantic knowledge which is used for video analytics.

1.5.2 Video Clustering and Classification

Clustering is an unsupervised activity which divides video data into different categories to assist in event identification, video compression, video indexing, video surveillance etc. In video surveillance systems, the identification of patterns and groups of moving objects support clustering analysis. This semantic representation leads to efficient indexing and browsing of video database. On the other side, classification techniques which organize video into already established categories. Rule-based classification techniques use the domain knowledge to semantic classification. Statistical approaches are also used to bridge the semantic gap for machine learning and creating inferences.

1.5.3 Video Association Mining

The two-steps process of establishing associations in the video is called Video association mining. In the first step, the video is segmented by using common features. The second step mine association among different segments to extracts knowledge which can be used for analytics and rule inference for further processing.

1.5.4 Video Motion Mining

A motion in a video represents temporal inconsistency of a pixel value in the video frames. Accurately identifying a motion is key to the success of object tracking and analysis. Motion can be of different types based on scene motion and camera motion (static/dynamic). These variations in setup give out a whole range of scenario. It is always difficult to suggest a single object motion detection method but we can suggest some optimum algorithms which work on nearly all type scenarios.

1.6 OBJECT MOTION DETECTION METHODS

In video streams, objects are identified either through features or using motion information. Temporal analysis of a pixel provides motion information over a period of time. A stable temporary history of a pixel suggests it to be a part of the background and frequent change in the value of a pixel hints toward some kind of motion and is suitable to be part of the motion and active object. A number of methods have been devised by researchers to detect objects from a video sequence. Literature broadly classifies object detection techniques into four approaches:

Background Subtraction, Temporal differencing, Statistical Approaches, Optical Flow. Background subtraction as the name suggests subtract the current image from a reference background and is the most commonly used scheme for object detection in static scenes (McIvor, 2000). The difference is thresholded to classify a pixel as foreground. The constructing of the reference background image is called as background modeling. The reference image is continuously updated to adapt the changes in dynamic background. In the temporal differencing scheme, the difference between consecutive frames is thresholded to categorize moving object detection in video streams. Although it is an adaptive method, it is highly susceptible to noise and highly dependent on frame rate. The third approach is statistical based object detection in which every pixel temporal history is utilized to predict the current value of the pixel to model the current background and finally, the motion is detected using subtraction with threshold method. The statistical methods provide better detection results but these are very slow in processing (Zhang, Tian, Yang, & Zhu, 2009). The fourth method is based on the optical flow of moving objects to detect an object in motion in video streams. Velocity and direction of every pixel are computed to predict the current value to detect object motion in the video even using a moving camera or moving background. Optical flow methods require very high computing and mostly not suitable for real-time video surveillance.

Object detection in the video is analogous to finding moving regions in the frames. In any frame every pixel is classified either as a foreground region, depicting objects in motion or as a background pixel which is immovable. Therefore, motion detection or moving object detection algorithm starts with segmentation of the moving part from the immobile part of frame i.e. Current Frame (*I*) is subtracted from Background Frame (*B*) of the scene. The I is taken when there is no movement in the scene or it is static. In the ideal case, this absolute difference when converted into binary value will give the whole movement in *I* but there is always the presence of noise in-camera picture. The jitter of the camera also introduces some noise in the scene. So, this difference is classified as the threshold with some value, to avoid noise and jitter in the difference or motion frame (M) (Benezeth, Jodoin, Emile, Laurent, & Rosenberger, 2008). The value of a pixel in motion frame at time t can be given as

$$M_t(x,y) = \begin{cases} 1 \;\; iff \;\; |I_t(x,y) - B(x,y)| \geq Th \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (1-1)}$$

Where t subscript denotes t^{th} frame, (x, y) is the position of a pixel in frames, I_t is the Current Frame and Th is a threshold value generally derived empirically.

The B, ideally, should be time invariant but due to introduction of several photometric or geometric variations in scene or objects in due course of time, the background image must be frequently adapted to the changed circumstances; otherwise the method will fail to find actual movements of objects in the scene. So an up-to-date copy of background must be maintained and adapted regularly to restrict the number of false positives in detection and equation can be rewritten as

$$M_t(x,y) = \begin{cases} 1 \;\; iff \;\; |I_t(x,y) - B_t(x,y)| \geq T \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (1-2)}$$

Where B_t is the current background which is to be used in finding M_t.

Above system is fast but its performance mainly depends on the selection of background frame which is also called reference frame. Modeling of this reference image holds the key of this algorithm. In summary, the above algorithm can be divided into following steps:

- Step1: Background initialization
- Step2: Background modeling or background maintenance
- Step3: Foreground detection

Background initialization may be the first frame or averaging of several initials frame. A good background frame results in the early convergence of background model (Colombari & Fusiello, 2010). Once background model is initialized properly then it is regularly updated and maintained to cope up with the problem of slow or abrupt light variation, shadows, the velocity of object motion, occlusions, ghost objects in scene etc. (Lee & Hedley, 2002). Finally, pixels which vary more than a threshold value in the absolute difference of current frame and background model is classified as foreground pixel and is a part of the object in motion. In many variations of background subtraction method defined in the literature, step 1 and 3 described above remain same.

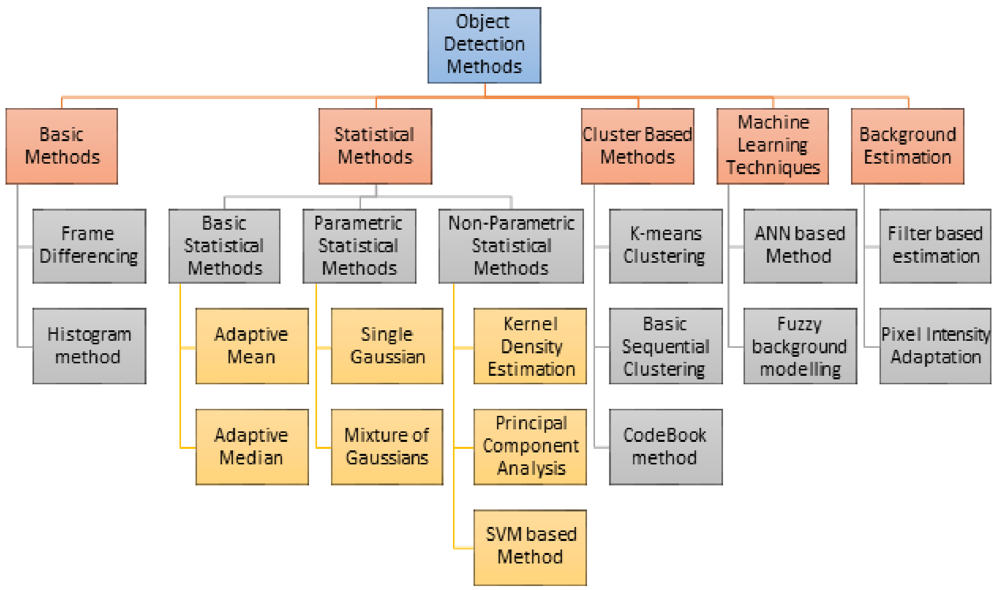


Fig. 1-3: Different Object Detection Methods

The author in the article (Bouwmans, 2014) classified different schemes of background modeling into five categories; i) Basic Background Modeling, ii) Statistical Background Modeling, iii) Background Modeling via Clustering, iv) Machine Learning Techniques and v) Background Estimation. These are depicted in **Fig. 1-3**. Some of these methods of object detection are explained below:

1.6.1 Basic Methods

- **Frame Differencing:**

Most logical and simplest method used for motion detection in a frame is "Frame Differencing" (FD) in which the current frame I_t is subtracted pixel by pixel from the previous frame I_{t-1} (Mingwu & Han, 2005). The difference is recognized as a threshold to avoid noise effect according to the following equation:

$$M_t(x,y) = \begin{cases} 1 \;\; iff \;\; |I_t(x,y) - I_{t-1}(x,y)| \geq Th \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (1-3)}$$

FD is also called "Temporal Difference", as the t^{th} frame is subtracted from $(t-1)^{\text{th}}$ frame. The method is simple but it may be possible that an image has been moving in previous frames before (t-1) time but not moving in (t-1) frame then it will not be detected. So, this method may not be able to acquire the entire target when applied directly. Further, it is highly sensitive to video frame rate, the speed of the objects in scene and threshold value. Higher motion may require larger threshold. Although frame differencing is useful in video compression where minimum change is required to construct a current frame from the previous frame requiring less memory space only to store subsequent frames, its utility in object detection is limited. To avoid these problems, usually, a special method of frame differencing is used that is called background subtraction method. Background modeling with Histogram:

Some authors have used temporal history of the pixel to decide about the most likely intensity value it can acquire in the next phase in order to generate a BF (Kuo, Chang, & Wang, 2009) (Jiang & Zhao, 2012) but huge memory consumption has restricted its use. In another method of histogram based detection, the temporal history of pixels is divided into several bins by identifying a probability of membership to every bin. A new pixel value is matched with corresponding bin and is classified as foreground pixel when the probability of the bin is less than a threshold.

1.6.2 Basic Statistical Background Modeling schemes:

- **Background Modeling with Average filter:**

In this scheme, background image is taken as average of N preceding frames (PF) and background BF can be represented as

$$B_t(x,y) = \frac{1}{N}\sum_{i=1}^{N} I_{t-i}(x,y) \qquad \text{Eqn. (1-4)}$$

Although it is very fast, it requires storing of 'N' frames at each execution and consumes lot of memory. The value of N depends on video frame rate and object motion speed. Large value of N increases accuracy but is a burden on memory; therefore, a compromise is made by adopting a running average method in which current background is modelled as a weighted sum of previous background and previous frame (Cheng, Huang, & Ruan, 2010). This scheme is also called Adaptive Mean Method (AM) and can be represented as follows:

$$B_{t+1}(x,y) = \alpha I_t(x,y) + (1-\alpha)B_t(x,y) \qquad \text{Eqn. (1-5)}$$

Where α is called the learning rate of background model and usually decided empirically.

A new approach of pixel selectivity is also used in the article (Cucchiara & Gualdi, Mobile video surveillance systems: An architectural overview, 2010), in which every pixel is marked either foreground or background to avoid the foreground pixels to become a part of background model (Cucchiara, Grana, Piccardi, & Prati, 2003) (Yan, Yu, Zhu, Lei, & Li, 2015). Once a pixel is marked foreground then it is not updated with **Eqn. 1-5**.

- **Background modeling with median filter:**

As the name suggests, instead of taking the average, the median of previous N frames is used to construct background image (McFarlane & Schofield, 1995).

$$B_t(x,y) = Med_{i=1}^{N}(I_{t-i}(x,y)) \qquad \text{Eqn. (1-6)}$$

Subsequently, the background image is updated in the following manner:

$$B_{t+1}(x,y) = \begin{cases} B_t(x,y) + 1 \ iff \ \ I_t(x,y) - B_t(x,y) > 0 \\ B_t(x,y) - 1 \ iff \ I_t(x,y) - B_t(x,y) < 0 \end{cases} \qquad \text{Eqn. (1-7)}$$

The median filter based method also called Adaptive Median Method (AMD) improves performance a bit as compared to mean filter in the form of less blurred intensities in foreground detection but still it can't handle the dynamicity of background properly. It also fails to detect properly slow moving objects. Further, sophisticated algorithms are proposed in the literature for this purpose (Ching, Cheung, & Chandrika, 2004).

1.6.3 Parametric Statistical methods for Background Modeling

As we have seen above, basic methods are fast but these are inefficient to handle peculiar requirement of the real world as they are unable to handle too slow and too fast motion of objects, abrupt changes in illumination, shadow, repetitive motions in the clutter and occlusions. Their global and constant thresholds also make them insufficient for challenging real-world problems. For proper modeling of background, the parametric statistical methods aim to understand the dynamicity of each pixel in the scene.

- **Background modeling with single Gaussian filter:**

In this method, intensity history of a pixel is assumed to vary with a Gaussian Probability Distribution Function (PDF) with mean μ and deviation σ (Wren, Azarbayejani, Darrell, & Pentland, 1997). Therefore, in a grey scale image sequence the probability density of the intensity value I_t of a pixel (x, y) caused by an object can be expressed as a Gaussian function with mean μ_t and standard deviation σ_t can be given as:

$$(I_t|\mu_t, \sigma_t) = \frac{1}{\sqrt{2\pi\sigma_t{}^2}} e^{-\frac{\left(I_t(x,y) - u_t(x,y)\right)^2}{2\sigma_t{}^2}} \qquad \text{Eqn. (1-8)}$$

Hence, Gaussian weights are used on temporal intensity values of a pixel while deriving a background instead of equal weights as in the case of the mean filter. The value of μ and σ are derived iteratively with the help of running averages to save memory by the following equations.

$$B_{t+1}(x,y) = u_{t+1}(x,y) = \alpha I_t(x,y) + (1-\alpha)u_t(x,y) \quad \text{Eqn. (1-9)}$$

$$\sigma_{t+1}^2(x,y) = \alpha\big(I_t(x,y) - u_t(x,y)\big)^2 + (1-\alpha)\sigma_t^2(x,y) \quad \text{Eqn. (1-10)}$$

Finally, foreground or motion frame is derived using equation.

$$M_t(x,y) = \begin{cases} 1 \;\; iff \;\; |I_t(x,y) - u_t(x,y)| > K\sigma \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \quad \text{Eqn. (1-11)}$$

Where K is a free threshold value and usually taken within a range of 1.5 to 2.5. A large K may accommodate more dynamic background while a small value of K may be required when there are subtle changes in the background.

Single Gaussian method can handle effectively background modeling from the scene where illumination is constant or gradual changing (in case of transition of day into night and vice versa), suggesting a single Gaussian PDF for a pixel value over time but in reality illumination may change abruptly (in case of switching a light on off in a room) which require a multimodal PDF. This necessitates background modeling with multiple Gaussian PDFs.

- **Background modeling with Mixture of Gaussian filter:**

Stauffer and Grimson (Stauffer & Grimson, 2000) argued that single Gaussian model fails to address the multi-modal history of a pixel and fails to extract foreground pixels in the highly dynamic background scene. The single Gaussian method also failed to adapt background model in case of an abrupt change in lighting condition of a scene. Therefore, they proposed to model a pixel history with a mixture of Gaussian (MOG) distribution. In this method values of a particular pixel (x, y) over time t frames can be modeled by a mixture of k Gaussian distributions (Value of k is usually from 3 to 5) by the following way:

$$P(I_t) = \sum_{i=1}^{k} \omega_i * \eta_i\left(I_t \middle| \mu_{i,t}\ , \sigma_{i,t}\right) \qquad \text{Eqn. (1-12)}$$

Where ω_i is the weight assigned to i^th^ Gaussian, η_i is i^th^ Gaussian distribution with mean $\mu_{i,t}$ and standard deviation $\sigma_{i,t}$ and is given as

$$\eta_i\left(I_t \middle| \mu_{i,t}, \sigma_{i,t}\right) = \frac{1}{\sqrt{(2\pi)\sigma_{i,t}^2}} e^{-\frac{\left(I_t - \mu_{i,t}\right)^2}{2\sigma_{i,t}^2}} \qquad \text{Eqn. (1-13)}$$

If pixel's new value I_t has a match with one of the distribution within K$\sigma_{i,t}$ then it is marked as background pixel:

$$B_t(x, y) = \begin{cases} 1 \ \ iff \ \ |I_t(x, y) - u_t(x, y)| \le K\sigma \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (1-14)}$$

where *K* is 2.5 and matching distribution mean and the standard deviation is updated as per the running average method:

$$\mu_{i,t+1} = (1 - \ \rho).\,\mu_{i,t} + \rho.\,I_{t+1} \qquad \text{Eqn. (1-15)}$$

and

$$\sigma_{i,t+1} = \sqrt{(1 - \ \rho).\,\sigma_{i,t}^2 + \rho.\left(I_{t+1} - \mu_{i,t+1}\right)^2} \qquad \text{Eqn. (1-16)}$$

where $\rho = \alpha.\,\eta_i\left(I_t \middle| \mu_{i,t}, \sigma_{i,t}\right)$ and α is learning rate

Weight ω_i assigned to i^th^ Gaussian is updated in the following manner:

$$\omega_{i,t+1} = (1 - \ \alpha).\,\omega_{i,t} + \alpha.\,M_{i,t+1} \qquad \text{Eqn. (1-17)}$$

Where $M_{i,t+1}$ is 1 for a matching Gaussian distribution and 0 for others. Weights are normalized each time after every iteration and their sum is made equal to 1.

If I_t is not matching to any of the k Gaussians following above criterion then pithe xel is marked as foreground pixel and a new distribution having mean as $u_{i,t+1}$=I_t and a prefixed high variance and very small weight is initialized and replaced with the least probably distribution having lowest ratio of ${}^{\omega_{i,t+1}}/_{\sigma_{i,t+1}}$ for the pixel. The MOG has better adaptability to the complex requirement of video surveillance, but due to its complex computation, it is slow and may not able to meet real-time needs.

1.6.4 Non-parametric Statistical Background modeling

In “non-parametric” modeling of background, a sample of temporal intensity values for each pixel in the image is used to estimate the density function of the pixel intensity. The probability of any newly observed intensity value of the pixel is used to classify it into background or foreground pixel.

- **Kernel Density Estimation:**

A very popular method of non-parametric density estimation is kernel density estimation (KDE) in which given a sample data set of size N, the probability of a pixel value to be x at time t is given by

$$p_{kde}(x_t) = \frac{1}{Nh^D} \sum_{i=1}^{N} K\left(\frac{x_t - x_i}{h}\right) \qquad \text{Eqn. (1-18)}$$

where K is called kernel function, N is the total number of recent sample points in D dimension which are updated recursively to update the model and *h* is called the bandwidth of kernel function which controls the smoothing of distribution (Elgammal, Duraiswami, Harwood, & Davis, 2002). A kernel function should fulfill the following properties:

$K(x) \geq 0,$

$$\int K(x).\,dx = 1$$

K(x) should be Symmetric

Many types of kernel functions considered in the past for object detection are uniform, Gaussian, quartic, triangular, Epnechnikov, cosine etc. (Soh, Hae, Mehmood, Hadi Ashraf, & Kim, 2013). A uniform kernel function is given by

$$K(u) = \begin{cases} 1 \; for \; |u_j| < \frac{1}{2} \; \forall \, j = 1 \ldots D \\ 0 \qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (1-19)}$$

This kernel is also known as Parzen window with a volume h^D in which equal weights are assumed for all the points considered in window function. This is called uniform distribution of weights in kernel function. The performance of KDE depends only on bandwidth h which if taken as a large value, will reduce the differences among the estimates of $p_{kde}(x_t)$ and p(x) for different data sets, but will increase the bias of $p_{kde}(x_t)$ with respect to the true density p(x). A small value of h will effect distribution in opposite manner. Its value is generally taken to minimize the integral squared error.

$$h_{MISE} = \arg\min \left\{ E \left| \int (p_{kde}(x) - p(x))^2 \ . dx \right| \right\} \qquad \text{Eqn. (1-20)}$$

Temporal histogram of a pixel is also density function with bandwidth h=1 but it has several drawbacks such as discontinuities of density, the curse of dimensionality etc. Elgammal *et. al.* (Elgammal, Duraiswami, Harwood, & Davis, 2002) used the median of the difference of consecutive frames over a sample size N as the bandwidth of their kernel to suppress the local-in-time movement of background. They also devised a method for suppression of false detection of objects by suggesting a displacement probability for a pixel and its connected region. These probabilities are recognized as the threshold for foreground identification. They have demonstrated that KDE is efficient in color modeling to avoid shadows and occlusions. The performance of KDE and MOG is stated to be almost similar in addressing the multimodal background and KDE has the advantage of not requiring any pre-knowledge about the number of mixtures in Gaussians but KDE consumes more memory than MOG.

- **Background modeling with Support vector machines:**

Support vector machines (SVM) are the supervised learning methods which are used in the linear or non-linear classification of data using advanced kernels for avoiding overfitting and underfitting of data. Several researchers have tried to use SVM for object detection by extracting features of pixels and then classifying them in background or foreground pixel. Junejo *et. al.* (Junejo, Bhutta, & Foroosh, 2011) used the optical flow of scene between two consecutive frames to regenerate unique features of the pixel by defining entropy, energy, and inertia related to the pixel. After that, they used SVM for classifying each pixel into background or foreground with the help of these features. The proposed method has been stated to present the excellent result on fountain sequence when compared with MOG but optical flow distribution is slow and largely not very reliable and accurate due to its sensitivity to noise, lighting, shadow, and occlusion. Miezianko and Pokrajac (Miezianko & Pokrajac, 2008) used SVM method to classify objects based on color, size, and shape.

- **Background modeling with Principal Component Analysis:**

Principal Component Analysis (PCA) is a statistical tool for orthogonal transformation of related data into linearly uncorrelated data. The transformation arranges principal components in a decreasing variance order and each component, being the eigenvector of the covariance matrix, is orthogonal to other components. Zhang et. al. (Zhang, Tian, Yang, & Zhu, 2009) analyzed the principal components of three consecutive frames extracted from the video stream for background and foreground classification. They found that the second component reveals most of the foreground or moving object information by suppressing most of the background. This component was used for generating and updating background by the procedure of 'top-hat' and 'bottom-hat' transformation. PCA is sensitive to the relative scaling of the original variables.

1.6.5 Background Modeling using clustering

- **K-means clustering based background modeling:**

Background modeling of a pixel can be achieved by classifying historical values of the pixel into k clusters (number of clusters, k, greatly varies with the type of scene) according to the behavior of its past history and similarity (Li, He, & Wang, 2008). The popular K-means algorithm has been used for this purpose. In this scheme, each cluster is parameterized with a weight w_i, mean u_i. Weight of a cluster is pea rcentage of membership it acquired in the past history. A cluster is classified as part of background when its weight crosses a certain threshold T_a. Once threshold weighted clusters define background about a pixel, then foreground segmentation can be done in the following manner.

The pixel value in the new incoming frame is matched with all the previously defined K clusters' mean values related to the pixel. If the minimum distance value is within a certain threshold limit T_b, then the current value is made a member of the shortest distanced cluster. The mean and weight of clusters are updated by the following equations:

$$u_{i,t+1} = (m_{i,t} \times u_{i,t} + I_t)/(m_{i,t} + 1) \qquad \text{Eqn. (1-21)}$$

$$w_{i,t+1} = \frac{(m_{i,t}+1)}{N} \qquad \text{Eqn. (1-22)}$$

Where $m_{i,t}$ is dega ree of membership of i^{th} cluster at time t, I_t is the current value of pixel and N is the total number of frames participated in clustering. Other clusters mean value will remain same but their weight will lessen as per their membership proportion which may rule out them from background modeling.

If the minimum distance exceeds the threshold limit T_b then a lowest weighted cluster is replaced with a new mean as the current value of the pixel and low weight (equivalent to a single member cluster) so that in case newly acquired foreground value remains static in future frames, it may gradually become a part of the background.

The procedure defined above is similar to MOG method, where the history of a particular pixel is assumed to imitate one of the Gaussians assumed in the initial phase. The K-means method is more robust because instead of assumptions it divides historical data into linearly separable segments. Although the standard K-means algorithm is a fast and easy technique of data clustering, it is a stochastic method and gets trapped in local minima. Final results of K-means method are not deterministic and greatly depends on the goodness of its initialization of centers in the first iteration which is done in a random fashion. Authors in (Kumar &

Sureshkumar, 2013) tried to find out a solution to this problem by manual seeding in K-means method. Here, instead of randomly choosing a hypothetical center for each cluster and then assigning memberships based on proximity, clusters are formed selectively by assigning membership manually in the first iteration based on the method of maximizing inter-cluster distance and minimizing intra-cluster distance. Once the initial membership of each cluster is defined then the standard K-means method is used for further clustering. Authors have reported better result by following the modified K-means method. Another problem with K-means clustering is with prefixing a number of clusters which model background scene but no one has paid attention to this problem.

- **Basic Sequential Clustering (BSC) based Background Modeling:**

Another clustering based method of background subtraction is BSC method in which instead of clustering at once from the collected data like in K-means clustering, online clustering is used where each pixel history is clustered right from the start of the first frame. Any new value of pixel is matched with centres of existing clusters by taking a minimum threshold distance (The number of allowed clusters is prefixed), if it matches, then the pixel's current value becomes a member of the matched cluster by increasing cluster's weight and modifying its mean and variance accordingly (Li, He, & Wang, 2008). Clusters that exceeds a weight threshold become the part of background modeling. If pixel's value does not match with any existing cluster within a range of threshold, the pixel is classified as a foreground pixel. Along with this, a new cluster is also initiated with the pixel's current value becoming its first member and assigning a very small weight and high variance to the new cluster. Xiao (Xiao, 2008) presented an improved BSC by adding the concept of merging of two or more similar clusters and reassignment scheme to avoid the problem of ordering syndrome with the general algorithm.

- **Background Subtraction Method Using Codebook (CB):**

A CB is a quantized representation of historical values of a pixel. For background modeling, every pixel creates a CB containing quantized values named as code words (CWs) with their intensity bounds, frequency, and access information. A pixel is classified as foreground or background based on color distance and intensity bound. If the present value is within a threshold distance with a code word and within intensity bound, then it is termed as a background pixel. CB of each pixel is used to find the latest usage and highest usage interval to determine the relevance of CW in background modeling. This information is also used to eliminate the redundant code words to obtain the refined initial codebook that can best represent the actual background (Kim, Chalidabhongse, Harwood, & Davis, Background Modeling and Subtraction by Codebook Construction, 2004). CW that occurs less than half of sampled frames is eliminated from codebooks for background modeling. Badal *et. al.* (Badal, Nain, Ahmed, & Sharma, 2015) proposed a modified codebook to address the problem of ghost regions in the background model. A ghost is a region in an image which is created when a temporary and stationary object moves in the background. In this method, CWs which used to be deleted due to non-usage for a longer period are retained for further use and this led to effective ghost elimination.

1.6.6 Machine learning based Background Modeling:

- **Background Modeling using Neural Networks:**

Machine learning based motion detection methods used SVM and artificial neural networks (ANN) to classify a motion pixel. The SVM and ANN parameters are learned during a training session to initialize the background model (Lin, Liu, & Chuang, 2002). Maddalena and Petrosino (Maddalena & Petrosino, 2008) model the background of a video with the weights of a neural network. In this approach, ANN is trained using frames of the video by extracting some feature such as texture, block or pixel. ANN is organized as a two-dimensional grid of neurons or nodes matching with each feature or block. Each of such neurons computes a weighted linear combination of incoming inputs for learning. Training makes clusters of incoming input data containing similar data. Once different regions of background model have classified more accurately in training, the current frame is converted into blocks to feed into the trained neural network. The block features of the foreground image will not match to any of the clusters formed at the training. The proposed approach can handle scenes containing moving backgrounds, gradual illumination variations but the ANN-based method has to train the background before detection by taking a few frames as training background making a delayed detection. These approaches take too much time but are interesting and may be useful in future. These methods also require huge memory and complex process of detection making these prohibitively slow for real-time detection.

- **Fuzzy Background Modeling:**

To address the uncertainty and imprecision in the classification and localization of moving pixel in the multi-modal environment, Baf et. al. (Baf, Bouwmans, & Vachon, 2008) proposed a fuzzy approach for background subtraction. In this work, the Choquet integral is used for the foreground detection and fuzzy operators for adaptive background maintenance. In comparison to crisp foreground detection, Fuzzy foreground detection is found to be more robust than the dynamic background and shadow environment (Bouwmans, 2012).

1.6.7 Background Estimation using Filters

Scott et. al. (Scott, Pusateri, & Cornish, 2009) presented an algorithm in which the background image related to a particular scene is estimated without having the prior knowledge of the scene. They assume that background pixels' intensities do not evolve quickly, while intensities of foreground pixels do. The evolution and their retention in the background image are controlled by adapting mean and variance of intensities by Kalman filter equations. High-intensity variances pixels are classified as foreground pixels and low variances pixels are included in the background model. The whole process of Kalman filter is to update mean and variance of the image as per the following equation iteratively.

Given the first t frames and calculation for μ_t and σ_t, The new weighted average μ_{t+1} is the estimated value of I'_{t+1}. The new actual intensity $I'_{t+1}(x,y)$ is compared with the measured $I_{t+1}(x,y)$. if it matches, it indicates a background else it is a foreground value. New estimation of the average and variance is calculated as

$$\mu_{t+1} = \mu_t + K(I_{t+1} - \mu_t) \qquad \text{Eqn. (1-23)}$$

$$\sigma'^2_{t+1} = \sigma^2_t + K(I_{t+1} - \mu_t)^2 \qquad \text{Eqn. (1-24)}$$

and the final variance is given as

$$\sigma^2_{t+1} = (1-K)\sigma'^2_t \qquad \text{Eqn. (1-25)}$$

where K is gain factor and is equal to $\frac{1}{n+1}$.

The background value is calculated as

$$B_{t+1}(x,y) = \begin{cases} 1 \;\; iff \;\; |I_{t+1}(x,y) - u_{t+1}(x,y)| \leq k\sigma_{t+1} \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (1-26)}$$

1.6.8 Background Subtraction using Pixel intensity adaptation:

Wang and Dudek (Wang & Dudek, 2014) proposed a background model which is not only based on each pixel's historical values but also based on the efficacies calculated using occurrence statistics. It removes the least useful background values from the model and selectively adapts to changes with different timescales. It also solves the problem of ghost generation. The high-frequency temporal noise is also controlled by individual decision threshold for each pixel. In this scheme, a small set of adaptive templates are created for each pixel by using the historical values of pixels. An occurrence metric related to each template helps to remove least used templates and replace them with new ones. The templates are ordered based on efficacy that assists the pixel adaptation to changes easily.

1.6.9 Optical flow based background Modeling:

In optical flow based method, pixels are classified as per their velocity and direction of movement (Lu & Manduchi, 2011). Pixels' positions that maintain optical flow in a time window are classified as moving objects and that show randomness are assigned to the background. Optical flow based methods fail in low-texture areas and produce halo effect around moving object boundaries.

As explained in above sections, there are a lot of algorithms proposed and evaluated in isolated ways by taking comfortable scenarios. There are lot of surveys which have compared different algorithms (Yilmaz, Javed, & Shah, 2006), (Benezeth, Jodoin, Emile, Laurent, & Rosenberger, 2008), (Bouwmans, 2011), (Athanesious & Suresh, 2012), (Parekh, Thakore, & Jaliya, 2014) but they lack objectivity in two ways; first, these surveys compare methods qualitatively not quantitatively and second, these methods have not been tested on rigorous benchmarks providing challenges in object detection. In the subsequent section, qualitative as well as quantitative comparisons of different popular object detection methods are presented using different scenarios in a benchmark video data set.

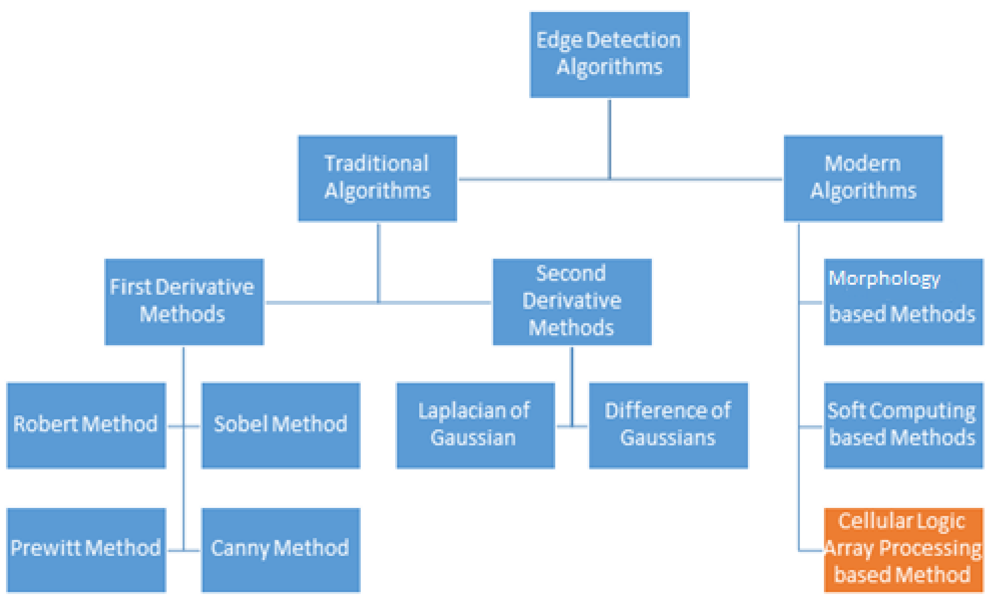


Fig. 1-4: Classification of Edge Detection Methods

1.7 EDGE DETECTION METHODS

Edge detection is one of the most basic procedures used in image processing application. Because of different variety of objects having different colors, texture, and shapes, there are numerous object detection methods researched and discussed in literature during past years (Basu, 2002), (Lakshmi & Ranarayanan, 2010), (Papari & Petkov, 2011), (Li, Xiong, Yin, & Liu, 2009). Many traditional edge detection algorithms devised for edge detection can be categorized into First Order Derivatives and Second order derivatives. These algorithms are depicted in **Fig. 1-4**.

1.7.1 First Order Derivative Based Edge Detection

As the name suggests, in first-order derivatives, the magnitude of the first gradient between adjacent pixels decides about the sharpness of edge and gradient vector judges the direction of maximum rate of change. In digital image processing, the discrete gradient is measured in terms of finite intensity differences between adjoining pixels. These are approximated by different masking filters such as Roberts (Roberts, 1963), Sobel (Sobel & Feldman, 1968) and Prewitt (Prewitt, 1970) operators which are given in **Fig. 1-5**. These filters separately find the vertical and horizontal gradient in one dimension(1D) which can be later combined to give a whole edge image. Roberts operator also called cross operators finds only oblique edges. Prewitt improved it by suggesting horizontal and vertical edges masks which are again improved by Sobel by proposing double weight for edge pixel.

The sharpening of the image through the first gradient unravel the finer details but it also enhances the noise in the image. These methods are not robust to noise and also infested with the problem of double edges. Later, Canny enhanced these methods by using non-maximum suppression, Hysteresis Thresholding and non-major edge points removal techniques to achieve stronger and finer edges in the image. Canny method, although, achieved very fine details of edges, it failed to control noise in the edge image. The Canny method also results in discontinued contours making object segments difficult in noisy environments.

R_x

0	1
-1	0

R_y

1	0
0	-1

Roberts' cross Operator

P_x

-1	0	1
-1	0	1
-1	0	1

P_y

-1	-1	-1
0	0	0
1	1	1

Prewitt Operator

S_x

-1	0	1
-2	0	2
-1	0	1

S_y

-1	-2	-1
0	0	0
1	2	1

Sobel Operator

LOG_2

-1	-1	-1
-1	8	-1
-1	-1	-1

LOG_1

0	-1	0
-1	4	-1
0	-1	0

Popular LOG Operators

Fig. 1-5: Edge Detection Masks

1.7.2 Second Order Derivative Based Edge Detection (Laplacian)

Second order derivatives such as Laplacian of Gaussians (LOG), Difference of Gaussians (DOGs) search for zero crossings in the second derivative of the image to find edges (Marr & Hildreth, 1980). These methods are very sensitive to noise and take much time for edge extraction. Two popular LOG operators are shown in **Fig. 1-5**. Other methods used in the past to extract edges are morphological edge detectors (Zhu S. , 2011) , soft computing based techniques (Mehrara, Zahedinejad, & Pourmohammad, 2009).

1.8 OBJECT DETECTION IN VIDEO SURVEILLANCE: PROBLEMS AND CHALLENGES

Real world video surveillance systems face complex and challenging environments owing to different contexts and applications for which these are installed (Brutzer, Hoferlin, & Heidemann, 2011). Bouwmans (Bouwmans, 2014) listed several challenges for which researchers need to devise background modeling techniques. Ideal object detection in the different setup of video surveillance requires fixed cameras, constant illumination and static background as preconditions which are never possible to maintain due to different peculiarities of indoor and outdoor scenes. Many problems identified by researchers in video surveillance are the poor quality of video having a lot of noise, camera jitter, slow and sudden change in illumination of a scene, shadows in a scene, the different requirement of indoor and outdoor scenes, clutter, camouflage, ghost, and occlusion problem in the scene. The problem is further compounded by variation in scale, shape, the position of the object of interest making detection and tracking a very difficult and erroneous process. All of these problems cannot be addressed by a single method due to different requirement. So, a major challenge in object detection is to propose, identify and implement a method which can address maximum problems or at least adapt itself to the changed requirements. These problems and challenges are summarized below:

1.8.1 Continuous or Sudden Changes in Light Intensity

Background modeling methods must be robust to detect object correctly in continuous or sharp light intensity changes. Surveillance systems are installed in both environments indoor or outdoor requiring them to adapt to light intensity changes. These changes may be sharp (in case of indoor) or gradual (in case of outdoor). Therefore, these methods must be adaptive to these changes. For example, a sudden switch of light or sunlight blocked by clouds strongly affects the appearance of the background model.

1.8.2 Dynamic Background

In environments, where continuous movement of objects are there backgrounds cannot be assumed constant and static. These undesired movements are normally iterative or periodic such as tree waving, water waves, flashing of traffic lights etc. and it should not be detected as foreground in object detection. A background modeling method must differentiate between the periodic or irregular movement of objects. To deal with such environment methods must model a pixel intensity using multimodal distributions. These distributions require large processing time. These methods must handle ghost objects in the background which may appear, hide and then reappear in the background.

1.8.3 Shadows

Shadow of objects in video streams are irrelevant and happens to be a major constraint in correctly mining foreground objects due to variation in illumination. Due to this variation, background pixels can be erroneously identified as foreground. Hence, a robust modeling algorithm is anticipated to remove or ignore shadows in object detection process.

1.8.4 Video noise

Noise can always creep into video signals during capturing and processing of images and naturally this may hamper the robust detection of objects. As the sources of noise are different there may be different forms of noise. It may occur due to camera jitter, aging of camera or sensor components such as lens etc. or information loss due to image compression. Background subtraction methods should mitigate or weaken the undesired effects of noise.

1.8.5 Other Challenges

There are several different types of requirement which are more application specific than general in nature. These are occlusion which may disturb the process of computing the stability of background frame; Camouflage when the object has poor contrast with its background making detection of foreground images difficult. Temporal differencing methods are very poor to handle this problem. Other than these, the occasional disappearance of objects and sporadic noise introduced in images or video streams due to the improper image capturing are real problems in object detection in video surveillance systems.

1.9 OUTLINE OF THE THESIS

The thesis is divided into the following eight chapters:

Chapter 1 introduces the basics of video surveillance with motivation to the proposed work. It is followed by the introduction to data mining concepts such as background subtractions, background modeling etc. with object detection methods. Edge detection methods are also familiarized because these are going to be used in our research methodology. Finally, problems and challenges of object detection are also summarized.

Chapter 2 is devoted to the literature survey containing works related to data mining and processing techniques in video surveillance. The chapter is divided into five sections covering each and every concept of data mining from video surveillance architecture to microscopic details in object detection such as various methods of object detection including edge-based detection using CLAP and adaptive thresholding

Chapter 3 describes the methodology of research including contribution, objectives, implementation environment, evaluation parameters, test data etc. It also sheds light on the use of cellular automaton in image processing and explains the design of cellular logic array based framework for data mining.

Chapter 4 presents a comparative evaluation of existing methods of object detection and discusses the experimental results with a view to deducing the final outcomes in a way that set the path of research direction of our work.

Chapter 5 describes the edge based object detection. After setting the main concept of edge detection using an adaptive threshold, a comparison is also made with existing methods and CLAP based edge detection methods with all the three types of thresholding.

Chapter 6 explains the main research contribution concept of local neighborhood differencing. It details how cellular logic arrays can be used to speed up the proposed method. A comparison with basic BGS methods on all the scenarios clearly established the supremacy of the proposed method in quality and efficiency both.

Chapter 7 sums up the results of existing and the proposed methods. It presents a comparison of results of the basic BGS method with the proposed methods and high quality and complex methods.

Chapter 8 concludes thesis with research directions and future scope in data mining in video surveillance techniques followed by bibliography and list of research publications in Appendix A.

CHAPTER 2

LITERATURE REVIEW

The literature review of the research work has focused on five areas. The first part of literature research is planned to establish the need for new fast and reliable object detection methods in real-time video surveillance. The second part of the literature review investigates the current techniques used for object detection. The third part of the literature survey is dedicated to exploring the use of CLAP in image processing especially edge detection methods with adaptive thresholding. The fourth section of this chapter is dedicated to getting an insight into edge based object detection. Finally, the emphasis of literature examination remains on the techniques employed for improving existing object detection methods using adaptive thresholding.

2.1 VIDEO SURVEILLANCE SYSTEMS

Recently many researchers have taken interest in video data mining techniques such as object identification and tracking due to multi-fold applications such as video surveillance, medical imaging, biometrics-based security system etc. A very good survey on video analysis of human dynamics was presented for biometric applications in (Wang & Singh, 2003). It provided a detailed survey on tracking of people and body parts such as the face, hands, fingers, legs, etc., and modeling of their behavior using motion analysis. Another article (Srinivasan, Porkumaran, & Sainarayanan, 2009) briefed about human body tracking in the surveillance area and discussed various kinds of background modeling methods such as background subtraction method, adaptive background subtraction method, adaptive Gaussian mixture method for 2-D and 3-D tracking etc. The authors in this used body positions and activities in video sequences for identification of the human body. The article in (Lefter, Rothkrantz, Bouchner, Burghouts, & Wiggers, 2010) described an audio/video based surveillance system for a car-driver identity recognition and car-driving behavior recognition system. They used an expert system to solve conflicts between audio and video for different scenarios. Another very good survey on visual surveillance of object motion and behavior is taken by Hu et. al. (Hu, Tan, Wang, & Maybank, 2004). They presented an overview of the developments in the field of video surveillance in dynamic scenes involving surveillance of people or vehicles. This survey identified five stages that are common for most surveillance systems. The first stage is described by pointing out the problems regarding multi-camera surveillance. The second stage was motion detection involving environment modeling, motion segmentation such as background subtraction, temporal difference and optical flow and shape based or motion based object classification. The third stage was related to object tracking involving region-based, contour-based, feature-based or model-based tracking. The second last stage of surveillance comprised of techniques for the analysis and recognition of motion patterns and the production and understanding of high-level descriptions of behaviors. The last stage used personal identification techniques for visual surveillance such as Human face and gait detection.

Another survey on contemporary remote surveillance systems for the public (Raty, 2010) focused on the evolution of video surveillance applicable to public safety. The work identified several future research directions such as real-time distributed architecture, intelligent cooperation between agents, addressing occlusion (objects become occluded by buildings, trees or other objects), the detection of ghosts, multi-sensor surveillance system etc. Another article (Hampapur, et al., 2005) proposed a new framework for a large-scale smart surveillance system to decrease the error rate. This paper gave an overview about the aspects for video surveillance systems. The article presented various challenges in the form of combining multiple sources of streams, automatic object detection, tracking, and classification. Authors also presented a technique *Face Cataloguer* for high-resolution face detection and a database storage technique for classifying video on the basis of objects.

Oh and Bandi (Oh & Bandi, 2002) proposed a framework for real-time data mining in video streams. The first stage of the framework grouped input frames followed by the second stage which extracts features from each segment. In the third stage, the segments are clustered into similar groups. This is followed by patterns discovery to detect objects, modeling and detect of events, video summarization, classification, and retrieval. This work also proposed a multilevel hierarchical clustering approach to group segments with similar categories using K-means algorithm and cluster validity method. The article (Yang, et al., 2009) described a high-level feature extraction framework in terms of low-level feature extraction; global, local and others, classifiers models such as Support Vector Machines (SVMs) ranking based on simple average fusion and linear weighted fusion and re-ranking. Here, textual information was extracted using automatic speech recognition and the information bottle principle. The global low-level features are based on color, textures, shape or a combination of those. The local low-level features were extracted using scale-invariant feature transform. The other feature Space-Time Interest Points computed locations and descriptors for space-time interest points in the video. Besides these, two additional features extraction of Region of Interest and the face feature. An article by Yokoi et. al. (Yokoi, Watanabe, & Ito, 2009) devised a mathematical model of event detection and implementation of different techniques such as change detection, human detection, human tracking and event detection.

The article by (Wani, Khan, & Patil, 2014) provided a processing framework for the behaviour analysis of the crowd using object detecting such as background subtraction, temporal differencing, and optical flow and using object tracking methods such as region based, active-contour based, feature-based and model-based to respond to accidents, crime, suspicious activities, terrorism to provide insights to improve evacuation planning and real-time situation awareness during public disturbances. It also discussed hidden Markov models for object analysis.

With a view to proposing a less computationally complexity and memory requirement system in real-time moving object detection and tracking algorithm on H.264 compressed video streams for IP video surveillance systems, a research work (Liu, Lu, & Zhang, 2007) proposed an algorithm for a real-life industrial perspective. The algorithm detects and segments regions having motion based on motion vectors embedded in the video stream without full decoding process and reconstruction of video frames using spatiotemporal filtering.

A detailed study of data mining techniques is presented in (Vijayakumar & Nedunchezhian, 2012). Authors described that video documents are generally unstructured in semantics and cannot be represented easily using relational data model. They cited that retrieval of data depends solely is based on the low-level feature extraction which is unpredictable due to lack of semantic relationship between high level and low-level features. They stressed that there is need of improved methods for the retrieval and mining process. They also categorized video data mining into video structuring, clustering, classifying, association, motion mining and pattern mining. There are many more works for video mining, but cellular logic array has not been used for moving object detection in video surveillance system. Only Pentagram Software (Pentagram, 2015) has used CLAP to develop a commercial product named as Logical Video Processing System (LVPS) for which very less information is available on company site. Therefore, CLAP based object mining is a fit case to implement and analysis.

2.2 EXISTING METHODS OF OBJECT DETECTION BASED ON BACKGROUND MODELING

As explained in previous chapter background subtraction methods are faster but need background modeling to address the object detection challenges. There are many background modeling methods. Some are efficient but consume too much time such as KDE. On the other side, few methods are fast but lack quality in detection. Therefore, we need to trade-off between fastness and quality while finding ways to improve quality in existing methods by proposing a new modification in background modeling, background registration, and background initialization.

There are several types of background initialization, ranging typically from selecting background as the first frame to averaging of several initials frame before invoking subtraction or detection process. A good initialization of background frame results in the early convergence of background model (Colombari & Fusiello, 2010). Some of the authors have used pixel's temporal histogram to decide about the most likely intensity value it can acquire in the next phase in order to generate a BF (Jiang & Zhao, 2012) but huge memory consumption has restricted its use. Further many sophisticated algorithms are proposed in the literature for this purpose (Ching, Cheung, & Chandrika, 2004). There are several background modeling as described in section 1.6. The different modifications and improvements taken in the past are described below:

2.2.1 Statistical methods for Background Modeling

Many types of kernel functions considered in the past for object detection are uniform, Gaussian, quartic, triangular, Epnechnikov, cosine etc. (Soh, Hae, Mehmood, Hadi Ashraf, & Kim, 2013). In one article (Elgammal, Duraiswami, Harwood, & Davis, 2002) authors used the median of the difference of consecutive frames over a sample size N as the bandwidth of their kernel to suppress the local-in-time movement of background. They also devised a method for suppression of false detection of objects by suggesting a displacement probability for a pixel and its connected region. These probabilities are recognized as the threshold for foreground identification. They have demonstrated that KDE is efficient in color modeling to avoid shadows and occlusions. The performance of KDE and MOG is almost similar in addressing the multimodal background and has the advantage of not requiring any pre-knowledge about the number of mixtures in Gaussians but KDE consumes more memory than MOG.

Junejo *et. al.* (Junejo, Bhutta, & Foroosh, 2011) used the optical flow of scene between two consecutive frames to regenerate unique features of the pixel by defining entropy, energy, and inertia related to the pixel. After that, they used SVM for classifying each pixel into background or foreground with the help of these features. The proposed method has been stated to present the excellent result on fountain sequence when compared with MOG but optical flow distribution is slow and largely not very reliable and accurate due to its sensitivity to noise, lighting, shadow, and occlusion. Authors in (Miezianko & Pokrajac, 2008) also used SVM method to classify objects based on color, size, and shape.

Zhang *et. al.* (Zhang, Tian, Yang, & Zhu, 2009) analyzed the principal components of three consecutive frames extracted from the video stream for background and foreground classification. They found that the second component reveals most of the foreground or moving object information by suppressing most of the background. This component was used for generating and updating background by the procedure of top-hat and bottom-hat transformations. This method was found to be very sensitive to the relative scaling of the original variables.

2.2.2 Background Modeling using clustering

Final results of K-means method are not deterministic and greatly depends on the goodness of its initialization of centers in the first iteration which is done in a random fashion. Kumar and Sureshkumar (Kumar & Sureshkumar, 2013) tried to find out a solution to this problem by manual seeding in K-means method. Here, instead of randomly choosing a hypothetical center for each cluster and then assigning memberships based on proximity, clusters were made selectively by assigning membership manually in the first iteration based on the method of maximizing inter-cluster distance and minimizing intra-cluster distance. Once the initial membership of each cluster was defined, then the standard K-means method was used for further clustering. Authors reported better result by following the proposed modification in K-means method. Another problem with K-means clustering is with prefixing a number of clusters which model background scene. No one has paid attention to this problem.

Xiao (Xiao, 2008) presented an improved BCS by adding the concept of merging of two or more similar clusters and reassignment scheme to avoid the problem of ordering syndrome with the general algorithm. They also proposed a modified codebook to address the problem of ghost regions in the background model. A ghost is a region in an image which is created when a temporary and stationary object moves in the background.

2.2.3 Machine learning based Background Modeling:

Authors in (Maddalena & Petrosino, 2008) modeled the background of a video with the weights of a neural network. In this approach, ANN was trained using frames of the video by extracting some feature such as texture, block or pixel. ANN was organized as a two-dimensional grid of neurons or nodes matching with each feature or block.

Baf *et. al.* (Baf, Bouwmans, & Vachon, 2008) proposed a fuzzy approach for background subtraction to address the uncertainty and imprecision in the classification and localization of moving pixel in a multi-modal environment. In this work, the Choquet integral was used for the foreground detection and fuzzy operators for adaptive background maintenance. In comparison to crisp foreground detection, Fuzzy foreground detection was found to be more robust in the dynamic background and shadow environments (Bouwmans, 2012).

2.2.4 Background Estimation using Filters

Wang and Dudek (Wang & Dudek, 2014) proposed a background model which was not only based on each pixel's historical values but also based on the efficacies calculated using occurrence statistics. It removed the least useful background values from the model and selectively adapted to changes with different timescales. It also solved the problem of ghost generation.

In another method based on optical flow, pixels were classified as per their velocity and direction of movement (Lu & Manduchi, 2011). Pixels' positions which maintained optical flow in a time window were classified as moving objects and those showed randomness, were assigned to the background. Optical flow based methods failed in low-texture areas and produced halo effect around moving object boundaries making it inefficient.

2.3 EDGE DETECTION AND CLAP BASED IMAGE PROCESSING FOR OBJECT DETECTION

Edge detection is one of the most basic procedures used in image processing application. Because of different variety of objects having different colors, texture, and shapes, there are numerous object detection methods researched and discussed in (Basu, 2002), (Lakshmi & Ranarayanan, 2010), (Papari & Petkov, 2011), (Lee & Hedley, 2002). Most popular category of edge detection method is first order derivative based edge detection in which edges are detected by computing the first derivative of the image. The magnitude of the gradient decides the sharpness of edge and the gradient vector gives the direction of maximum rate of change. Although the sharpening of the image through the first gradient unravel the finer details, it also enhances the noise in the image. Another problem associated with the first derivative is double edges. These edge detection algorithms can be enhanced by post-processing techniques such as non-maximum suppression, Hysteresis Thresholding, and non-major edge points removal. So what we finally get, is strong edges in the image. (Canny, 1986)

Second order derivative based edge detection (Laplacian) methods such as Laplacian of Gaussians (LOG), Difference of Gaussians (DOGs) which searches for zero crossings in the second derivative of the image to find edges. These methods are limited by their huge complexity of time and sensitivity to noise. (Marr & Hildreth, 1980)

Many attempts were made to extract edges with the help of morphological operations (NagaRaju, 2011), (RamaBai, Krishna, & SreeDevi, 2010) (Zhu S. , 2011). Soft computing based techniques were also devised for edge detection (Ezhoosh, 2010). Gao *et. al.* (Gao, Parslow, & Tan, 2001) proposed a hybrid method by combining Sobel edge detection operator with wavelet de-noising for edge detection in images containing white noise. Mohamed El-Sayed (El-Sayed, 2011) proposed an entropy-based high-quality edge detection method for decreasing computation time. Priyadarshini *et. al.* (Priyadarshini & Sahoo, 2010) proposed an automatic threshold-based edge detection method based on simple arithmetic and logic operations which claimed to perform better than Sobel's method and requiring less computation. Neha Mathur *et. al.* (Mathur, Mathur, & Mathur, 2016) developed a K-means segmentation method to obtain a local threshold value through histogram bins and cluster centers for Sobel operator. Although they did not estimate time, it seemed too heavy for real-time processing.

The use of cellular automata can be traced since the 1960s for many applications where parallelism can be exploited such as digital circuits and mathematical operations but first time, it was used for image processing only in 1974 when Duff et. al. (Duff, Watson, Fountain, & Shaw, 1973) developed a cellular logic array for based image processing. It described the hardware implementation of the cellular logic array for the use of image processing. They presented a tool namely CLIP3 for fast processing of image processing and pattern matching the application. Later, Prof. E G Rajan proposed a *cellular automata* based framework for image processing techniques for high-throughput data processing (Rajan, 1993). The central idea here was to assume digital image as *cellular array* and image processing algorithm as an *evolution* (updation rule) of the automaton. Various operations demonstrated in this paper are thinning, edge detection segmentation, erosion, and dilation. A Logical Video Processing system (LVPS) for image and video processing (Pentagram, 2015). In 2002, Authors in (Popovici & Popovici, 2002) also used cellular automata for image processing. In this paper, authors used two-dimensional cellular automata for removing noise from images. They also proposed cellular automata based model for edge detection and compared its working with SUSAN tool (Smith & Brady, 1997). Both of these works claimed that CLAP performance is better as compared to the traditional tools such as

SUSAN. Tapas Kumar *et. al.* (Kumar & Sahoo, 2010) compared cellular automata based edge detection with standard methods but without any quantitative evidence.

2.4 EDGE BASED OBJECT DETECTION

A lot of efforts have been made in the past to employ edge maps to extract object motion but due to lack of generalization and acceptability, there remains a lot of scope for improvements in this area. The following discussion presents a comprehensive study of work done in this field. Smith, in 2001, (Smith, Drummond, & Cipolla, 2000) proposed an edge-based segmentation method in a video sequence to detect single object detection and multiple object detection in frames using Bayesian's framework. The thesis demonstrated that edges contain sufficient motion information to determine motion labeling in a frame. The technique used an Expectation-Maximization algorithm to segment the frame into similar regions and then Bayesian probability was used to detect foreground segments from background segments. The method was applied to multiple video sequences and results claimed that the proposed approach provided accurate and efficient motion segmentation.

Changick & Jenq-Neng in (Changick & Jenq-Neng, 2002) used double-edge map obtained from the difference of successive frames which was used to get moving edges with current frame edges, previous frame edges, and background edge model. The proposed algorithm claimed to be fast for implementing in real-time surveillance system but it failed to update background model making it difficult to handle dynamic scenes. This work was further extended by (Sappa & Dornaika, 2006) proposing a three equidistant frames technique of motion detection. Two preliminary edge maps extracted from three frames were used for detecting moving edges by AND operation which was sufficient for high frame rate. For a lower frame rate scenario, it used an iterative scheme where equidistant frames were subtracted iteratively until no new edge information was obtained about the background. Once sufficient confidence was gained about the background, the algorithm switched from frame subtraction to background subtraction approach. The method was fast enough but failed to get a good result in camera jitter and random noise scenarios.

The article (Zhan, Duan, Xu, Song, & Luo, 2007) proposed an improved edge based object detection from contiguous frames and their difference by using Canny detector. This was followed by detection of moving area from difference image by counting a threshold of non-zero pixels over small blocks. Finally, block-connected component labeling was done to track the moving object. Experimental results asserted to overcome the limitations of the frame difference method by getting a high recognition rate and a high detection speed but the method needs to be tested over a large number of scenarios before implementing. Moreover, the method did not provide any solution to over and under sampling of frames.

In this work, (Li, Xiong, Yin, & Liu, 2009) Canny's edge image was used to build MoGs background with an objective to reduce the undesirable effect of sudden illumination on MoGs model. The presented results claimed to provide higher performance on real surveillance video but only two scenarios have been presented which does not effectively validate the result. Second, the paper did not consider the heavy computational cost of Canny's detector and MoG model making these techniques inapplicable for real-time surveillance. Wang et. al. (Wang, Zhang, Shi, & Zhong, 2013) proposed an edge based moving object segmentation algorithm which modeled background from image pixel values of the longest sequence to remove the problem of shadow and also post-processed the extracted image with a Gaussian filter to remove random noise. Although the method has claimed to remove the effect of shadow and noise, no conclusive evidence of applicability on different scenario except shadow problems has been put forward. The method was also limited by its huge memory requirement for background reconstruction.

The Gao's article (Gao, Parslow, & Tan, 2001) dealt with object detection based on perceptual vision. The consecutive frames of a video were processed to find the edge features based on generic curve segments and curve partition points. Then these frames were subtracted to find the average thresholded difference to detect moving object in the scene. Authors only analyzed result subjectively and the frame rate taken in this method was also high (5000 frames/sec) limiting the applicability of the proposed method in real-time scenarios. The article (Dong, Wang, & Jia, 2009) proposed a background subtraction technique for object detection based on RGB color space and edge ratio to identify shadow, object, and background by using. Separate threshold values were adapted for foreground and objects. Finally, area and edge ratio were used to rectify the misclassified object and shadow regions. Murshed et. al. (Murshed, Ramirez, & Chae, 2010), first, modeled a statistical background for each segment in the image and then used Canny edge based threshold method for motion detection. Background edge segments and moving edge segments were detected using statistical distribution maps and Chamfer distance maps respectively. The background was updated continuously to manage the dynamic scenes. In this method, only edge pixels were processed for faster execution.

Cui et. al. (Cui, Zeng, Cui, Fu, & Liu, 2011) used Canny edge detector to find edge map of contiguous frames and then edge pair difference is used to get moving objects. It got a better result than simple frame difference method but needs to be tested in different scenarios. Dhar and others (Dhar, Khan, Hasan, & Kim, 2011) proposed a gradient map based method in which gradient map difference of current frame and background were used to extract the moving objects with proper directional masking and thresholding. The proposed method was applied to different conditions such as indoor, outdoor, and foggy conditions and was claimed to be faster than traditional methods.

Authors in (Jabid, Mohammad, Ahsan, Abdullah-Al-Wadud, & Chae, 2011) mixed edge segmentation with a gradient map like feature called local directional pattern which provided the direction of an edge to detect moving objects. The watershed algorithm was also used to extract a regular boundary of the object as a post-processing operation. Priya and others (Priya, Mahesh, & Kuppusamy, 2014) presented an edge based video segmentation technique for finding the foreground objects in video streams. First, edges of the objects were detected using Canny edge detection method which was followed by a morphology motion filter and filling technique. The main benefits of edge detection based segmentation methods are fast processing and less requirement of storage. Warade *et. al.* (Warade, Kale, & Thakare, 2015) implemented a feature extraction technique on frame difference edge map. The extracted features such as color, texture, and shape determined the moving edges. Mukherjee and Kundu (Mukherjee & Kundu, 2013) employed Prewitt operator on background subtraction algorithm and compared it with Canny edge detection operator for extracting the objects in motion from video frames.

2.5 ADAPTIVE THRESHOLDING IN OBJECT DETECTION METHODS

In literature, many thresholding methods are used for object detection. Zidek and Hosovsky (Zidek & Hosovsky, 2014) classified threshold into static and dynamic thresholding. Static type includes binary, truncate and threshold to zero, band and multispectral (color) thresholding while dynamic thresholding contains Otsu or adaptive thresholding. They proposed a hybrid thresholding based on multispectral and Otsu method. Rosin *et. al.* (Rosin & Ioannidis, 2003) evaluated global thresholding techniques on object detection and classified methods into; i) Euler-number which specifies a single number for every stable block of image, ii) Poisson-noise modeling which expresses thresholding on the assumption that noise in image follows Poisson model for every pixel and iii) entropy-based method. The article (Chang & Aishy, 2006) proposed an adaptive threshold by taking an average of thresholds of every block derived using regions of change curve. Samanta and Sanyal (Samanta & Sanyal, 2012) presented a method of finding an adaptive threshold to get the edge map by using the mean and variance of 3 X 3 window of each pixel. Authors (Singh, Prasad, Srivastava, & Bhattacharya, 2017) have also devised an adaptive local threshold based on the neighborhood average to extract the edge map of an image.

Subudhi *et. al.* (Subudhi, Ghosh, & Nanda, 2016) proposed a spatial segmentation and temporal segmentation based object detection algorithm using an adaptive threshold based on entropy windowing approach. Nain *et. al.* (Nain, Jindal, A., & Jain, 2008) proposed a method to find the number of prominent peaks in the histogram of the image which represented the distinct regions in the image and a basis for thresholding. Firdousi and Parveen (Firdousi & Parveen, 2014) compared various local thresholding techniques such as Niblack's, Yanowitz and Bruckstein's, Bernsen's Techniques and concluded that the algorithm which followed local gray range method instead of local variance methods performed better than others. Hua et. al. (Hua & Ruichun, 2014) proposed an adaptive threshold for non-parametric Kernel Density Estimation to address the bimodal intensity distribution video sequences for object detection. Boufares et. al. (Boufares, Aloui, & Cherif, 2016) presented a discrete stationary wavelet transforms based adaptive threshold for motion detection in the adaptive background subtraction method. Isaac (Case, 2010) developed a method of finding adaptive threshold by calculating the second derivative of the cumulative sum of difference frame. He observed that adaptive threshold is that value of difference where the second derivative is approaching zero.

2.6 RESEARCH GAPS

As explained in above sections, huge numbers of algorithms are proposed for object detection, but most of these are evaluated in isolated ways by taking comfortable scenarios. There are lot of surveys which have compared different algorithms (Yilmaz, Javed, & Shah, 2006), (Benezeth, Jodoin, Emile, Laurent, & Rosenberger, 2008), (Bouwmans, 2011), (Athanesious & Suresh, 2012), (Parekh, Thakore, & Jaliya, 2014) but they lack objectivity in two ways; first, these surveys compare methods qualitatively not quantitatively and second, these methods have not been tested on a rigorous benchmark providing challenges in object detection. Most of the methods claimed to get good results and to overcome the limitations of the frame difference methods, but these need to be tested over a large number of scenarios before generalization and implementation.

Many methods also failed to provide any solution to over and under sampling of frames. These research works have not considered heavy computation cost of Canny’s detector and MoG model making technique ineffective for real-time surveillance. Some methods are also limited by their high memory requirements.

Time is the most important constraint in real-time video analysis and this factor requires attention while devising new techniques for object detection in real-time scenarios. Most of the researchers have not paid attention to this criterion while selecting and analyzing methods making these doubtful to be employed in real-time video surveillance.

Another area which remained open to research in this field is decidability of an optimum threshold automatically. Although many efforts have been made to model an adaptive threshold, it still lacks generalization. The factors which effect threshold value need to be identified and evaluated on different benchmarked dataset before making final conclusions.

So, keeping in mind the above-specified requirements, our research work strives to provide a data mining framework for object detection in video surveillance to address the poor quality of detection within timing constraints. Cellular automata based methods are assumed faster because these work on frame level rather than pixel level and will be implemented for achieving objectives.

CHAPTER 3

METHODOLOGY OF RESEARCH

3.1 CONTRIBUTION OF THESIS

This thesis proposes a cellular logic array based data mining framework for increasing efficiency in automatic video surveillance in terms of speed and precision. The proposed methods increase the quality of detection within reasonable time constraints and will help to reduce the error rate, delay and to increase the robustness of the object detection in video surveillance for real-time video analytics.

3.2 STATEMENT OF THE PROPOSAL

The existing methods of object detection in video surveillance give low-quality results besides being slow, costly and memory hungry. We need to find robust and fast methods with medium to low-end hardware configuration systems of objects detection for real-time video analytics. This research work strives for providing a novel data mining framework for object detection in video surveillance system using a cellular logic array for replacing contemporary, slow and unreliable techniques of video processing with faster and robust techniques, making surveillance system effective for real-time video analytics.

3.3 OBJECTIVE

To design and develop a novel data mining framework for object detection in video surveillance system using cellular logic array.

3.4 METHODOLOGY

As discussed in **section 1.3**, a complete video surveillance application is composed of three major steps; (i) video capturing, (ii) processing and (iii) analysis. Video frames are captured from single/multiple streams of static or dynamic cameras to process and analyze the scene further. Analysis can be targeted for general or specific objects. In video processing, there are series of operations such as objects representation, detection, identification and tracking for automatic monitoring of video scenes. Before starting the process of object identification, it must be defined in a unique way with its special features and descriptors such as centroid (Gemert, Veenman, Smeulders, & Geusebroek, 2010), set of points (Serby, Koller-Meier, & Gool, 2004), colors, edges, contours and silhouettes (Yilmaz, Li, & Shah, 2004), shapes (Comaniciu, Ramesh, & Meer, 2003), (Zhang, Collins, & Liu, 2004), skeleton (Li & Qi, 2010), probability densities functions such as Gaussian (Zhu & Yuille, 1996), a mixture of Gaussians (Paragios, Rousson, & Ramesh, 2002) and Parzen windows (Elgammal, Duraiswami, Harwood, & Davis, 2002), histograms (Comaniciu, Ramesh, & Meer, 2003), Templates (Fieguth & Terzopoulos, 1997) etc. This process is called object representation and is essential for further processing. The words 'detection', 'recognition' and 'identification' are used in literature interchangeably creating a lot of confusion making them indistinguishable at several times. We will stick to the formal definition of these words as defined by the Johnson Criteria (Johnson, 1958):

- Detection: "ability to distinguish an object from the background"
- Recognition: "ability to classify the object class (animal, human, vehicle, boat …)"
- Identification: "ability to describe the object in details (a man with a hat, a deer, a Jeep …)"

Taking a cue from above definitions, it can be clearly seen that object detection is the most primitive operation used in video processing application. Once the object is detected, then it is recognized or identified based on its representation. In both of the methods; Recognition and Identification, classification of different objects in a video scene ultimately lead to the discovery of the "tracking details" of the object of interest in the video. Finally, based on the tracking details, object analysis is done to make decisions.

Because of different variety of objects having different colors, texture, and shapes, there are numerous object detection methods researched and discussed in literature during past years. Bouwmans (Bouwmans, 2014) categorized different types of methods into traditional and recent approaches by listing nearly 15 years' comprehensive research in object detection along with resources, data sets, implementation codes. However, there is no up-to-date analytical review of these methods in literature available. This thesis aims to classify and analyze object detection methods with respect to areas of processing along with different implementation environment in order to give a clear picture of video processing specifically for the purpose of video surveillance applications. After analyzing the existing methods our next step is to formulate, design and implement a new cellular logic based data mining framework for object detection in video surveillance.

A key component to the design of a framework for video surveillance system is thorough knowledge and understanding of the factors responsible for affecting the performance of the system. Therefore, the following steps are taken to accomplish the stated objective:

1. A thorough literature survey is done to evaluate the required techniques for optimum result generation and the various factors that influence the design of video surveillance framework. The literature study also includes an investigation into the existing techniques of object detection in video surveillance.
2. In the second step, algorithm and mathematical model for object detection are conceptualized and formulated.

3. Next, a new framework using a cellular logic array is designed taking into consideration the specific requirement of video processing and the common flaws of the existing techniques.
4. Finally, the proposed framework is implemented and validated by comparing with the existing system.

3.5 IMPLEMENTATION ENVIRONMENT

Open Source Computer Vision (OpenCV) and CodeBlocks are used for implementing the existing algorithms from scratch. OpenCV is a widely accepted cross-platform and open source libraries of C++ for manipulating digital images and real-time computer vision (Pulli, Baksheev, Kornyakov, & Eruhimov, 2012). Although it contains several background methods such as MOG, no pre-built method is used in experiments. These are freshly built to process video frames at pixel level for effective comparison. CodeBlocks is a popular lightweight, cross-platform IDE used for providing an integrated development environment for several languages including C++ (CodeBlocks, 2017). GCC compiler is used for compiling C++ based methods and OpenCV functions and classes.

At a later stage, MATLAB is also used to implement the existing system and the proposed algorithm, as this is handy to use and flexible enough for experimenting with various options or modifications in algorithms. Once algorithms are stable then they can be implemented on real target language such as OpenCV etc.

3.6 PRE-PROCESSING AND POST PROCESSING METHODS

Although no pre-processing method is used for these experiments, various filters can be suggested to remove unwanted noise and effects on video (Sankari & Meena, 2011). In post-processing phase, the median blur filter is used in the 3×3 or 5×5 neighborhood because of its speed and effectiveness but other methods such as Gaussian blur and morphological functions can also be used. Morphological functions are more effective (Benezeth, Jodoin, Emile, Laurent, & Rosenberger,

2008) but these are computationally costly and may not be suitable for the analysis of video scene in real time scenarios. We have used 3×3 neighborhood median filter for comparative evaluation of existing methods but later on for evaluation of the proposed methods 5×5 neighborhood median filter used due to its better noise removal property.

Table 3-1: Different scenarios of video sequence selected for experimentation

Scenario1	**Scenario2**	**Scenario3**
Baseline Video Highway Data Set Total Frames=1700 Training Frames=469 Size=240X320	Camera Jitter Badminton Data Set Total Frames=1150 Training Frames=799 Size=480X720	Dynamic Background Fountain02 Data Set Total Frames=1499 Training Frames=499 Size=288X432
Scenario4	**Scenario5**	**Scenario6**
Intermittent objects Sofa Data Set Total Frames=2750 Training Frames=499 Size=240X320	Shadow Bus Station Data Set Total Frames=1250 Training Frames=299 Size=240X360	Thermal Imagery Park Data Set Total Frames=600 Training Frames=249 Frame Size=288X352

3.7 EVALUATION DATA SET

The CDnet2012 dataset (Goyette, Jodoin, Porikli, Konrad, & Ishwar, 2012) has been used for evaluation of algorithms. The data set consists of six categories namely baseline, camera jitter, dynamic background, intermittent object motion, shadow and thermal imagery scenes captured from different types of cameras in different lighting conditions, level of noise and compression techniques. The resolutions of the varying length videos in CDnet vary from 320 × 240 to 720 ×

576. One video sequence has been taken randomly from each of six categories for evaluation purpose. The first few hundreds of frames in each sequence are assumed as training frames for scene stabilization and the corresponding ground truths frames are labeled region of non-interest and are not used for evaluation purpose. Specification for different data set is given in Table 3-1.

3.8 EVALUATION PARAMETERS

Quality parameters/metrics in terms of accuracy and speed of automated video surveillance are listed in several research articles. Accuracy is measured in terms of a total number of falsely identified pixels. (Blair & Marion, 1985). Cielniak *et. al.* (Cielniak, Treptow, & Duckett, 2005) classified accuracy metrics in three classes; i) *Temporal Metrics* which measure the consistency of the system in detecting target object in adjoining frames, ii) *Detection Metrics* to count false positives and negatives of objects and iii) *Localization Metrics* that measure the extent of coverage of target objects in terms of pixels. Yin *et. al.* (Yin, Makris, & Velastin, 2007) further suggested metrics for performance evaluation of tracking algorithms. They suggested measuring speed in terms of time complexity of the algorithm used in the system. Memory requirements or Space complexity is another criterion which is also considered occasionally where memory is a constraint. However, measuring the effectiveness of a motion detection algorithm has no specific set of metrics.

Table 3-2: Binary Classifier Parameters

Parameters	**Formulas**	**Remarks**
Positive (P):	$P = SumPixel \; \forall \; FG \neq 0$	Foreground Pixels
Negative (N)	$N = SumPixel \; \forall \; FG = 0$	Background Pixels
True Positive (TP):	$TP = SumPixel \; \forall \; (FG \neq 0 \; \&\& \; \mathrm{GT} \neq 0)$	Correctly identified Pixels
False Positive (FP):	$FP = SumPixel \; \forall \; (FG \neq 0 \; \&\& \; \mathrm{GT} = 0)$	Incorrectly identified Pixels
True Negative (TN):	$TN = SumPixel \; \forall \; (FG = 0 \; \&\& \; \mathrm{GT} = 0)$	Correctly rejected Pixels
False Negative (FN):	$FN = SumPixel \; \forall \; (FG = 0 \; \&\& \; \mathrm{GT} \neq 0)$	Incorrectly rejected Pixels

Table 3-2 lists out four detection parameters viz. TP, FP, TN, FN, to measure the quality of object detection. In the presence of actual motion and desired motion in the form of ground truth, positive means pixel is discovered as part of motion (foreground- FG) and in case of negative, the pixel is judged as part of the background (BG). Hence, TP and TN can be defined as the sum of the numbers of rightly judged pixels as FG and BG respectively. FP and FN can be defined as the sum of the numbers of falsely rejected pixels as FG and BG respectively. With the help of these parameters, various statistical metrics such as Recall, Precision, Specificity, Accuracy, and F1 score are derived as shown in **Table 3-3**.

Table 3-3: Performance Metrics For Object Detection Methods

Metrics	**Formula**	**Remarks**
Sensitivity or Recall	$R = TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$ (True positive Rate- TPR)	The ability of correctly detecting foreground pixels
Specificity	$TNR = \frac{TN}{N} = \frac{TN}{TN + FP}$ (True Negative Rate − TNR)	The ability of the classifier to correctly detect background pixels
Precision (P):	$P = \frac{TP}{TP + FP}$	Quality of classification
Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$	System Accuracy
F1 score	$F1 = \frac{2TP}{2TP + FP + FN}$	Integrated Metrics of Recall and Precision.
Informed-ness (In)	$In = TPR + TNR - 1$	Integrated metrics
Peak Signal to Noise Ratio (PSNR)	$PSNR = 20. log_{10}. (I_{max}/\sqrt[2]{MSE})$ Where $MSE = \frac{1}{mn} \sum_{i=0}^{i=m-1} \sum_{j=0}^{j=n-1} ([FG_t(i,j) - GT_t(i,j)]^2)$ I_{max} is highest intensity i.e 255 for 8 bit pixel.	Only an approximation of human perception, do not consider the spatial relationship between pixels. (Winkler & Mohandas, 2008)
Similarity Measures	$S(A,B) = \frac{A \cap B}{A \cup B}$	Range is [0 1], 0 complete mismatch, 1 complete match
Speed	$S = \frac{Time}{N}$ Where N is the total number of frames.	The average processing time of a frame.

Precision is a fraction of retrieved relevant instances and recall is a fraction of relevant retrieved instances [McManus2008]. These are given in **Eqn. (3-1)** and

Eqn. (3-2):

$$Pr = \frac{\{Relevant\ Instances \wedge Retrieved\ Instances\}}{Retrieved\ Instances} \qquad \text{Eqn. (3-1)}$$

$$Re = \frac{\{Relevant\ Instances \wedge Retrieved\ Instances\}}{Relevant\ Instances} \qquad \text{Eqn. (3-2)}$$

Recall and Precision are two most used metrics for binary classifier but often require a trade-off between themselves because Recall favors methods with a low False Negative Rate (FNR) and Precision favors methods with a low False Positive Rate (FPR). The overall effect of both is represented by an F1 score.

		Ground Truth	
	Total Pixels (100)	**Actual FG Pixels (True)**	**Actual BG Pixels (False)**
Actual Outcome	**Retrieved FG Pixels (Positive)**	TP=28	FP=12
	Retrieved BG Pixels (Negative)	FN=14	TN=46

Precision= TP/(TP+FP) = 28/40=70%

Recall =TP/(TP+FN) =28/42= 68%

F-Measure= 2*P*R/(P+R)= 69%

Fig. 3-1: A Binary Classifier Example

Fig. 3-1 shows an example of a binary classifier and its associated parameters. Let us assume that we have 100 pixels' image and out of these, actual FG pixels are 42, and BG pixels are 58 but our detector is able to unearth only 28 FG pixels and 46

BG pixels then precision is 70% and recall is 68%. The combined measure of precision and recall is an F1 score which is harmonic mean of these two and is calculated 69%.

Effectivity and efficiency are two criteria used for measuring the ability of an object detection algorithm. Effectivity is measured using recall, precision, F1 score and PSNR metrics. These are well-studied in the literature (McManus, Renno, Makris, & Jones, 2008), (Kalirajan & Sudha, 2015) and also frequently used for the evaluation of effectivity or quality of detection. We have used all the three metrics along with peak-signal-to-noise ratio (PSNR) and precision-recall (PR) trade-off curves for all methods. PSNR helps in finding the level of dissimilarity between extracted motion and corresponding ground truth. Its correlation with the F1 score is an additional verification of the outcome. Finally, the efficiency of the algorithm is measured by a standard form of absolute time measuring i.e. Execution Time (ET). It is measured in seconds on an i3 Pentium 2.4 GHz processor and taken as the average time taken by the algorithm in the processing of a single frame. It is averaged over the whole range of video sequence except training frames. Although measuring absolute time is generally not an ideal criterion, an estimate can be provided and is suitable for comparison purpose.

For each method and each scenario, five to ten thresholds are defined based on the experiments and corresponding recall and precision for each threshold are calculated. Two extremes (recall, precision) points of (1,0) and (0,1) are assumed with these calculated pairs to draw curves. Threshold values greatly vary from algorithm to algorithm and are uniquely identified for each algorithm separately. The best trade-off threshold is selected for each method in each scenario for comparison purpose but algorithms are ranked on the basis of visual analysis of PR curves.

3.9 FEATURE SELECTION IN OBJECT DETECTION

A feature is a characteristic of an image which is relevant in the computation of algorithms. Various features are considered for object detection algorithms which vary in scales and types. Some features are based on pixel intensity in grey scale images, color, edge, texture. Parekh et. al. (Parekh, Thakore, & Jaliya, 2014) classified object detection algorithms in three categories based on the features researchers used in their algorithms. These are edge based, patch-based and hybrid in which multiple features are used. So, before deciding on an algorithm one must choose the feature, he/she will be choosing for object detection. In the forthcoming discussion, we are assuming grayscale images until it is specified otherwise. Most of the processing in object detection methods is done at the pixel level. But many research articles have also reported it on block level and frame level (Lin, Cao, & Zeng, 2014). Cellular logic array processes images at the frame level.

CHAPTER 4

COMPARATIVE EVALUATION OF EXISTING OBJECT DETECTION ALGORITHMS

Object detection in the video streams is analogous to find moving regions in the frames. In any frame every pixel is classified either as a foreground region, depicting objects in motion or as a background pixel which is immovable. Therefore, motion detection or moving object detection algorithm starts with segmentation of moving-part from immobile part of the frame i.e. Current Frame (I) is subtracted from Background Frame (B) of the scene. The BF is taken when there is no movement in the scene or it is static. In the ideal case, this absolute difference when converted into binary value will give the whole movement in I but there is always the presence of noise in the camera picture. The jitter of the camera also introduces some noise in the scene. So, this difference is thresholded with some value "Th" to avoid noise and jitter in the difference or motion frame (MF) (Benezeth, Jodoin, Emile, Laurent, & Rosenberger, 2008). The value of a pixel in motion frame at time t can be taken as **Eqn. (1-1).** This equation is reproduced here for readers' convenience.

$$M_t(x,y) = \begin{cases} 1 \;\; iff \;\; |I_t(x,y) - B(x,y)| \geq Th \\ 0 \qquad\qquad\qquad\qquad\quad otherwise \end{cases} \qquad \text{Eqn. (4-1)}$$

Where t subscript denotes tth frame, (x, y) is the position of a pixel in frames, B and I_t are background and the current frame respectively. Th is threshold value which is generally derived empirically.

The background ideally, should be time-invariant but due to the introduction of several photometric or geometric variations in the scene in due course of time, the background image must be frequently adapted to the changed circumstances; otherwise, the method will fail to find actual movements of objects in the scene. Therefore, an up-to-date copy of background must be maintained and adapted regularly to restrict the number of false positives in detection and equation can be rewritten as Eqn. (1-2). It is again reproduced below:

$$M_t(x,y) = \begin{cases} 1 \;\; iff \;\; |I_t(x,y) - B_t(x,y)| \geq T \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \qquad \text{Eqn. (4-2)}$$

Where B_t is the current background which is to be used in finding M_t.

Above system is fast but its performance mainly depends on the selection of background frame which is also called reference frame. Initialization and modeling of reference image or background hold the key to good object detection algorithm.

Background initialization may be taken simply as the first frame or averaging of several initials frame. A good background frame results in the early convergence of background model (Colombari & Fusiello, 2010). Once background model is initialized properly then it is regularly updated and maintained to cope up with the problem of slow or abrupt light variation, shadows, the velocity of object motion, occlusions, ghost objects in scene etc. (Lee & Hedley, 2002). Finally, pixels which vary more than a threshold value in the absolute difference of current frame and background model is classified as foreground pixel and is a part of the object in motion. A detailed study of the different types of background subtraction and background modeling technique is included in **section 1.6** of this thesis.

4.1 EVALUATION OF EXISTING ALGORITHMS

In the preliminary phase, a total of ten algorithms are implemented and evaluated in experiments using the baseline video sequence for both types of sequences; grey and color. In subsequent experiments, only grey sequences were considered because, in preliminary experiments, it was found that there was no substantial improvement in the quality of object detection and on the other side color based processing was taking too much time making it unsuitable for real-time video analysis.

Two grey methods; FD and K-means methods were also abandoned as FD is too simplistic and unsuitable for slow motion and other complex scenarios, while K-means was taking too much training time in clustering and performance, was also not superior to two other similar methods; BSC and CB. FD execution time is also similar to other adaptive methods which are far superior to it. Remaining eight methods consist of two statistical methods; AM and AMD which are relatively simple and fast as they rely on plain background subtraction. Other methods include two parametric methods; SG and MOG, two non-parametric probabilistic methods; HD and KDE. Finally, experimentations are also done using two clustering-based methods: BSC and CB methods. In each method, there are two or three parameters which affect the working of the method. The suitable range of each parameter are derived empirically and is described in **Table 4-1**.

Table 4-2 presents the best results obtained from experimentations corresponding the given threshold for each method. In the table, there are eight sections separated visibly from background grey colors. One is for the header of the table and two following sections are for grey and color baseline video sequence processing respectively. The same set of methods is used for both types of processing. Other five sections are related to grey video sequence processing for camera jitter, dynamic background, intermittent object motion, shadow and thermal imagery sequence. For each section, related to a corresponding threshold, five metrics are computed. These are execution time(ET) in seconds(s), Recall, Precision, F1 score, and PSNR. Low-value execution time and high values of recall, precision, F1 score, and PSNR are desirable. A median filter with the 3×3 neighborhood is used to remove any spurious noise from the foreground.

Table 4-1: Parameters for object detection methods

SN.	Method	Parameters	Range	Remarks
1.	AM	Threshold (td) Alpha (alpha)	10 to 70 (variable) 0.01 (Fixed)	Threshold Background Adaptation Rate
2.	AMD	Threshold (td) Alpha (alpha)	10 to 70 (variable) 0.01 (Fixed)	Threshold Background Adaptation Rate
3.	SG	Deviation Threshold (K) Alpha (alpha) Initial Threshold (sd_init)	0.3 to 3.0 (variable) 0.01 (Fixed) 6 (Fixed)	Deviation Threshold Background Adaptation Rate Initial Deviation
4.	MoG	Deviation Threshold (K) Alpha (alpha) Initial Threshold (sd_init)	0.3 to 3.0 (variable) 0.01 (Fixed) 6 (Fixed)	Deviation Threshold Background Adaptation Rate Initial Deviation
5.	HD	No of Bin (bin) Hist_bin_th_prob	16 to 112 (variable) 1.0/(2*bin) (variable)	Number of bins for pixel classification Foreground Detection Threshold
6.	KDE	Threshold (td_prob) Alpha (alpha) Initial Threshold (sd_init) Number of Kernels (nKernels)	0.01to 0.25(variable) 0.01 (Fixed) 6 (Fixed) nInitFrames/10 (Fixed)	Threshold Probability Background Adaptation Rate Initial Deviation Sampling frames called kernels
7.	BSC	Distance Th Number of Clusters (nClusters)	1 to 15 (variable) 4 (Fixed)	Threshold Number of clusters for pixel classification
8.	CB	Distance Th (dist_th) Number of code words (nCW)	1 to 30 (variable) 4 (Fixed)	Threshold Number of code words for pixel classification

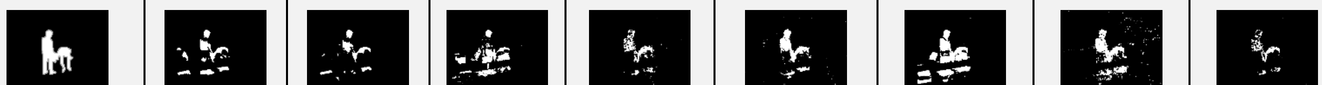
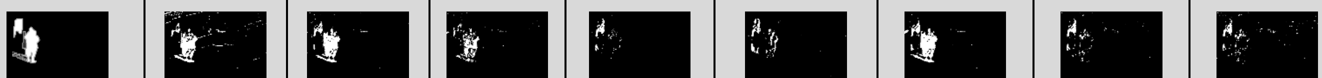
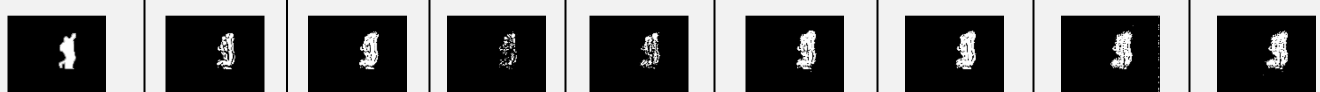
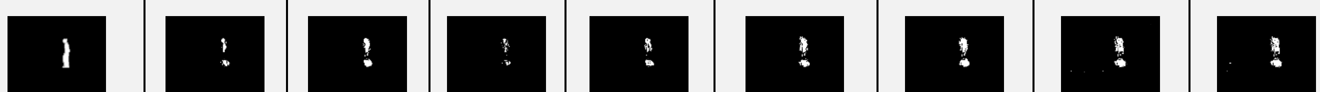
Table 4-2: Best results obtained by different methods for different scenarios

Scenarios	Methods /Metrics	Adaptive mean (AM)	Adaptive median (AMD)	Single Gaussian (SG)	Mixture of Gaussians (MOG)	Histogram Detection (HD)	Kernel Density Estimation (KDE)	Basic Sequential Clustering (BSC)	Codebook Detection (CB)
Baseline Video(Gray) Highway Data Set	Threshold	Th=30	Th=20	K=1.5	K=1.0	Bin=64	Td_prob=.05	Dist=4	Dist=4
	ET(s)	20	19	0.029	0.084	0.049	0.039	0.077	0.042
	Recall	0.925	0.9206	0.829	0.886	0.935	0.977	0.905	0.797
	Precision	0.7221	0.815	0.575	0.688	0.642	0.831	0.742	0.803
	F1 score	0.8111	0.8646	0.680	0.775	0.761	0.899	0.815	0.799
	PSNR	17.6084	18.7492	15.142	16.709	17.154	20.079	17.419	16.236
Baseline Video(Color) Highway Data Set	Threshold	Th=30	Th=20	K=1.5	K=1.3	Bin=48	Th_Pb=0.03	Th_Dist=4	Th_Dist=4
	ET(s)	41	37	0.055	0.0239	0.069	0.090	0.207	0.071
	Recall	0.9331	0.9264	0.9529	0.8992	0.8378	0.9193	0.9195	0.8046
	Precision	0.7137	0.8143	0.5644	0.7097	0.5734	0.8899	0.5875	0.7922
	F1 score	0.8088	0.8668	0.7089	0.7904	0.6808	0.9044	0.7169	0.7962
	PSNR	17.5943	18.8857	16.23	16.17	15.80	19.74	16.1016	16.03
Camera Jitter Badminton Data Set	Threshold	Th=45	Th=40	K=1.8	K=2.0	Bin=80	Th_Pb=0.03	Th_Dist=8	Th_Dist=8
	ET(s)	49	42	0.078	0.346	0.201	0.152	0.246	0.110
	Recall	0.6495	0.6404	0.7575	0.4926	0.8015	0.8291	0.453	0.324
	Precision	0.4868	0.5106	0.4556	0.3401	0.5985	0.6368	0.4104	0.4457
	F1 score	0.5565	0.5682	0.569	0.4024	0.6852	0.7203	0.4307	0.3752
	PSNR	15.5281	15.3036	16.3468	13.6792	17.1033	17.8159	13.403	11.8168
Dynamic Background Fountain02 Data Set	Threshold	Th=35	Th=30	K=1.8	K=1.8	Bin=64	Th_Pb=0.03	Th_Dist=6	Th_Dist=8
	ET(s)	20	20	0.035	0.126	0.075	0.040	0.087	0.055
	Recall	0.7133	0.8419	0.6687	0.7624	0.8209	0.8304	0.6682	0.5574
	Precision	0.539	0.5696	0.283	0.4644	0.7368	0.6892	0.5678	0.4932
	F1 score	0.614	0.6795	0.3977	0.5772	0.7766	0.7532	0.6139	0.5234
	PSNR	25.0757	27.3008	26.0259	26.1768	29.4276	27.3438	23.6062	21.7525

Intermittent objects Sofa Data Set	Threshold	Th=25	Th=20	K=1.5	K=0.5	Bin=80	Th_Pb=0.25	Th_Dist=1	Th_Dist=2
	ET(s)	21	19	0.026	0.052	0.074	0.057	0.071	0.046
	Recall	0.6786	0.5677	0.6279	0.6025	0.6273	0.685	0.7994	0.8024
	Precision	0.5131	0.4636	0.3285	0.2855	0.3207	0.6362	0.3965	0.4516
	F1 score	0.5844	0.5104	0.4313	0.3875	0.4244	0.6597	0.5301	0.5779
	PSNR	14.5222	13.5423	13.4149	14.0716	14.7415	14.4292	14.7656	14.9812
Shadow Bus Station Data Set	Threshold	Th=25	Th=20	K=1.3	K=0.8	Bin=48	Th_Pb=0.15	Th_Dist=2	Th_Dist=3
	ET(s)	18	21	0.035	0.102	0.051	0.040	0.068	0.041
	Recall	0.748	0.8849	0.7938	0.9048	0.827	0.8622	0.7401	0.6083
	Precision	0.6631	0.6336	0.5492	0.3371	0.4055	0.7472	0.5325	0.5414
	F1 score	0.703	0.7384	0.6492	0.4912	0.5442	0.8006	0.6194	0.5729
	PSNR	14.8201	16.4981	14.4556	Inf	16.4835	16.4781	15.1218	13.9586
Thermal Imagery Park Data Set	Threshold	Th=15	Th=10	K=1.2	K=1	Bin=32	Th_Pb=0.2	Th_Dist=2	Th_Dist=4
	ET(s)	27	24	0.035	0.107	0.047	0.051	0.085	0.051
	Recall	0.7313	0.7146	0.5262	0.6264	0.7696	0.7234	0.6292	0.7191
	Precision	0.6705	0.7863	0.386	0.5675	0.64	0.7393	0.7979	0.6873
	F1 score	0.6996	0.7488	0.4453	0.5955	0.6989	0.7312	0.7036	0.7028
	PSNR	21.263	21.1884	17.2174	17.7979	Inf	Inf	19.0281	20.978

Table 4-3: visual analysis of different methods for different scenarios

Scenarios	Ground Truth	Adaptive mean (AM)	Adaptive median (AMD)	Single Gaussian (SG)	Mixture of Gaussians (MOG)	Histogram Detection (HD)	Kernel Density Estimation (KDE)	Basic Sequential Clustering (BSC)	Codebook Detection (CB)
Baseline Video Highway Data Set Frame numbers Analyzed are 1020 and 1260	F1 score=1	0.7644	0.8466	0.6853	0.7891	0.7955	0.8897	0.7996	0.7825
	F1 score=1	0.7426	0.8115	0.4940	0.6678	0.4477	0.8572	0.7317	0.5096
Camera Jitter Badminton Data Frame numbers Analyzed are 920 and 1035	F1 score=1	0.5289	0.5238	0.6370	0.4431	0.7863	0.7328	0.4614	0.3572
	F1 score=1	0.6692	0.6804	0.6933	0.3219	0.6848	0.7807	0.3737	0.3608
Dynamic Background Fountain02 Data Set Frame	F1 score=1	0.7675	0.7960	0.4703	0.6257	0.9045	0.8678	0.6763	0.6110

numbers analyzed are 750 and 1250	F1 score=1	0.5076	0.6129	0.3157	0.5626	0.7000	0.6920	0.6124	0.4641
									
Intermittent objects Sofa Data Set Frame numbers Analyzed are 1650 and 2450	F1 score=1	0.6483	0.5834	0.4875	0.6084	0.6547	0.7441	0.6568	0.6194
									
	F1 score=1	0.5982	0.6058	0.5546	0.5695	0.6179	0.6533	0.6288	0.5850
									
Shadow Bus Station Data Set Frame numbers Analyzed are 825 and 1025	F1 score=1	0.6842	0.7221	0.6771	0.2360	0.3629	0.7967	0.4304	0.3027
									
	F1 score=1	0.6953	0.8127	0.6323	0.7978	0.8277	0.8352	0.8178	0.8204
									
Thermal Imagery Park Data Set Frame numbers Analyzed are 350 and 500	F1 score=1	0.7589	0.7955	0.5009	0.6573	0.7545	0.7799	0.7513	0.7596
									
	F1 score=1	0.5952	0.6951	0.3796	0.5762	0.6218	0.6631	0.6489	0.6460
									

4.2 RESULTS OF COMPARATIVE ANALYSIS OF EXISTING ALGORITHMS

Out of eight methods, AM, AMD, HD and KDE performance is above average. Simple but fast statistical methods such as AM and AMD perform better than more complex SG, MOG and cluster-based methods BSC and CB. Their performance is comparable to non-parametric methods HD and KDE methods. The best result for each method in a different video sequence is achieved for the different threshold. The threshold is called the curse of image processing and can’t be easily guessed. It means threshold has to be adaptive for a video sequence. Till now, no solution has been suggested in this regard. Precision is always on the higher side than Recall for all best threshold output. It means nearly all methods are biased for detecting false positive rather than identifying false negative. PSNR and F1 score correlate the best statistics across all categories of scenarios. Video sequences having small recurrent background motion such as water waves and tree movement and camera jitter are easily addressed by the performing algorithms. Thermal imagery is also addressed comfortably by all methods while intermittent motion and camera jitter scenarios present a challenge for all methods. PR curves for the different scenario are depicted in **Fig. 4-1** to **Fig. 4-6**.

If execution time is considered, for a small frame size of the order of 300X400, AM, AMD, HD methods are appropriate. They can be easily employed for real-time video surveillance. Although KDE performance tops in overall yet its execution rate is slow. For large frame size AMD is most appropriate followed by AM. Dataset wise performance comparison of these algorithms is given in **Table 4-2.**

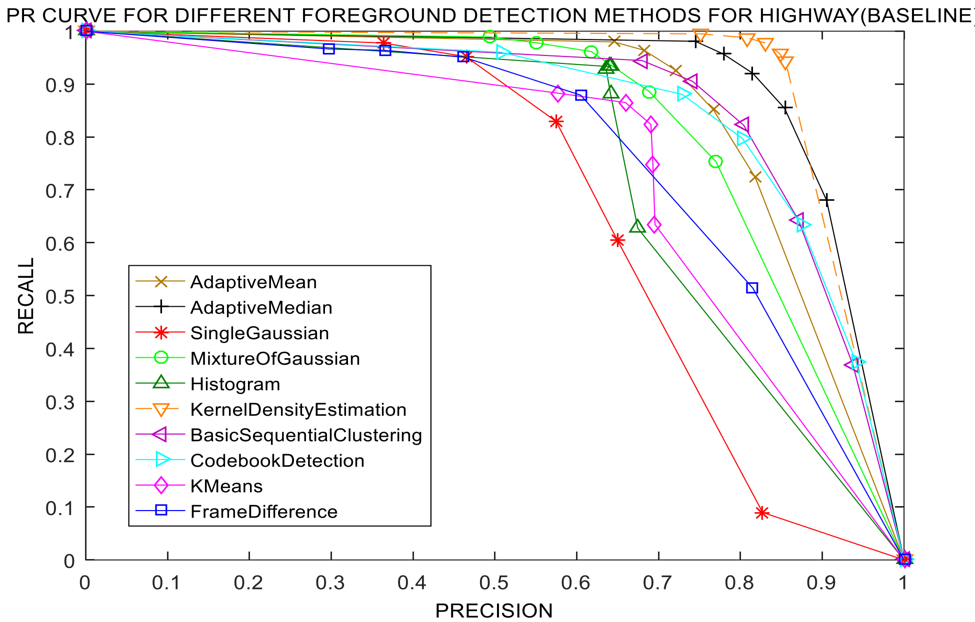


Fig. 4-1: Precision-Recall Curve for different object detection methods for the highway data set

For baseline highway data set as shown in **Fig. 4-1**, KDE and AMD fare better than others. The third distant method is BSC, AM, and CB. Surprisingly, SG and MOG have not performed very well and their ETs are also very high. The high area under Recall-Precision curve suggests the good performance of the methods.

Badminton data set is taken from a camera jitter video sequence. In this data set, top performing methods are KDE and HD while others are way behind as shown in **Fig. 4-2.** The worst performance is shown by cluster-based algorithms and MOG. In HD, optimal bin size is the key requirement.

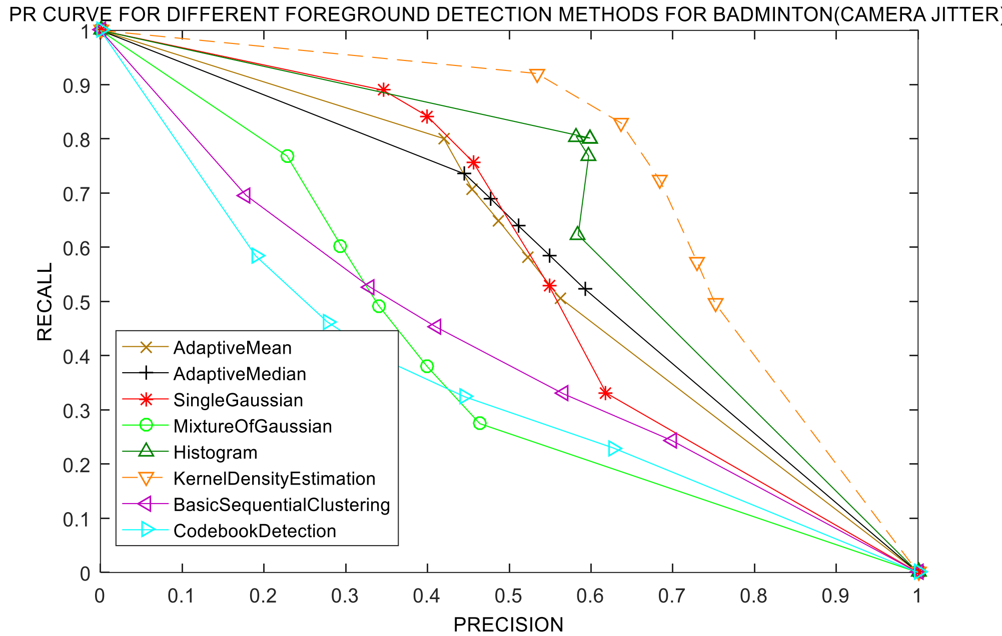


Fig. 4-2: Precision-Recall Curve for different object detection methods for the badminton data set

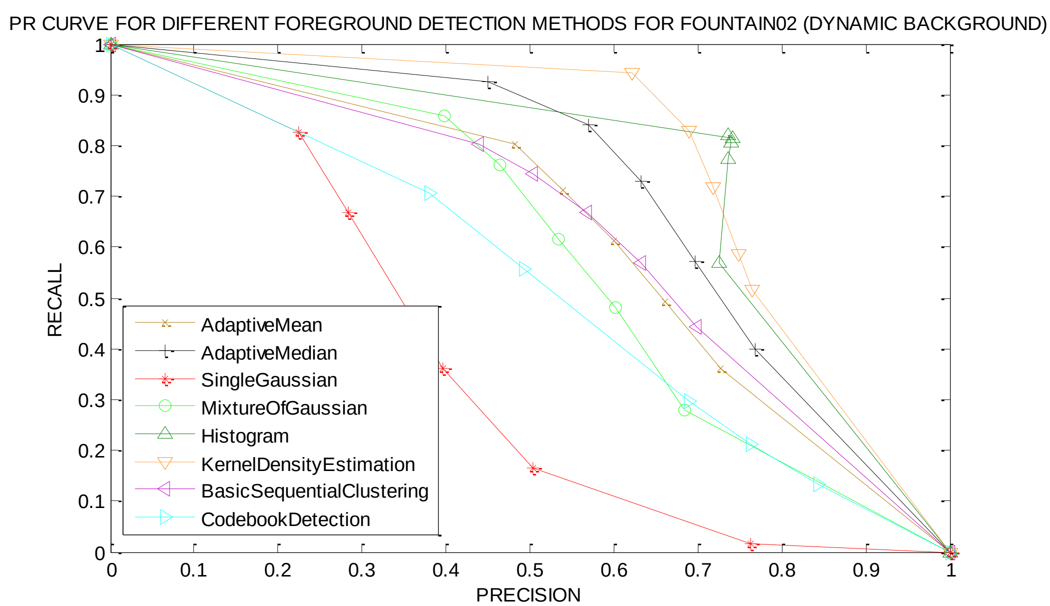


Fig. 4-3: Precision-Recall Curve for different object detection methods for the fountain02 data set

The dynamic background is pictured in the fountain02 data set for which PR curves are shown in **Fig. 4-3**, where again KDE and HD methods are winners. AMD is ranked third. MOG which is claimed to address multi-modal background has not performed well. A less area covered by P-R curve shows the inability of nearly all methods to perform up to the mark.

Sofa data set is a special challenge for object detection methods in which intermittent object movement is pictured. Experiments show that almost all methods fail to give good results in this dataset. This is the research area where more work needs to be done. The P-R curve shown in **Fig. 4-4** proves that again KDE is a better algorithm.

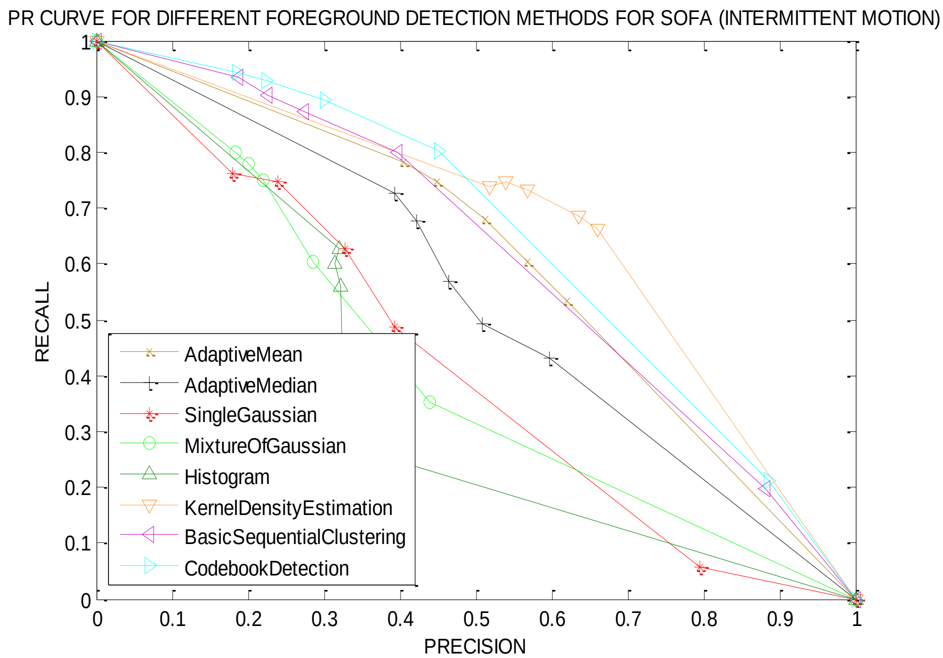


Fig. 4-4: Precision-Recall Curve for different object detection methods for the sofa data set

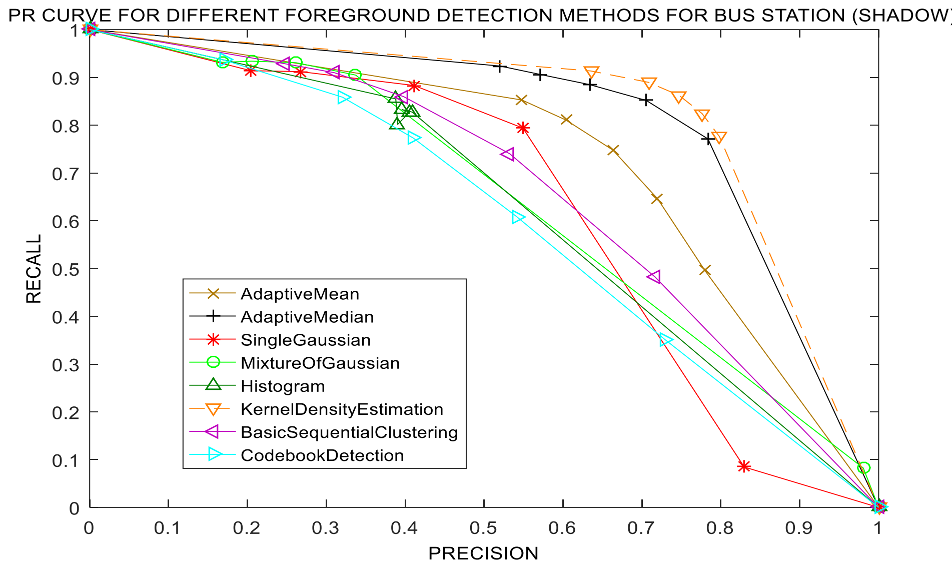


Fig. 4-5: Precision-Recall Curve for different object detection methods for the bus station data set

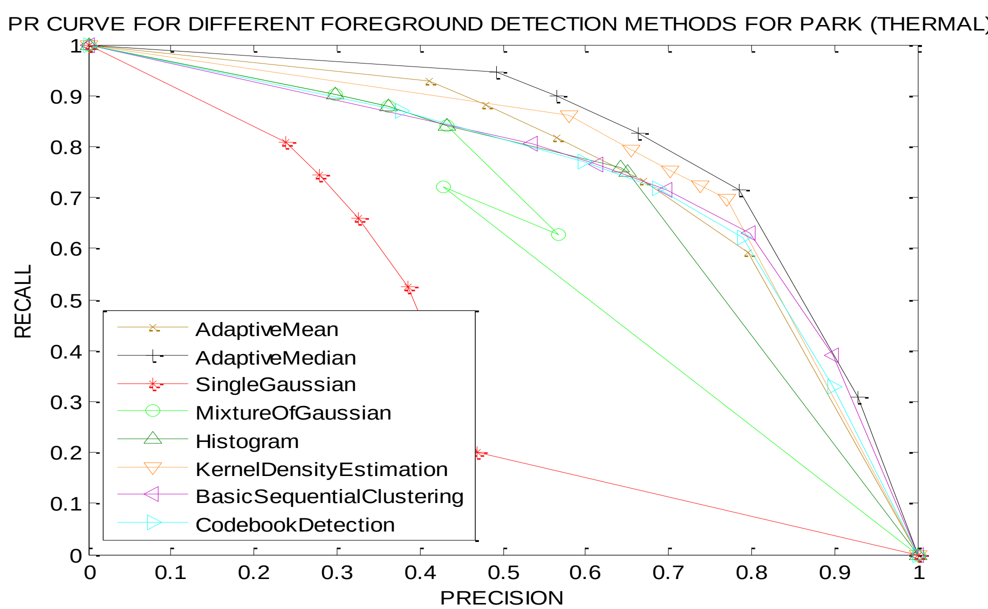


Fig. 4-6: Precision-Recall Curve for different object detection methods for the park data set

The bus station is a video sequence for shadow scenario. In this scenario, three algorithms namely KDE, AMD, and AM perform better than others. A high value of curve area suggests a better motion detection in this as compared to other scenarios. Park data set is related to thermal videos which are often appended with camouflage and reflections effects. AMD and KDE are the best-suited methods for such types of scenario. SG and MOG again fail miserably to address the problem. PR curves for these scenarios are shown in **Fig. 4-5** and **Fig. 4-6** respectively.

These results are also verified by the visual analysis of the outcomes of the various methods as shown in

Table 4-3. Here, results obtained by eight methods are shown for two frames from each scenario. Two frames are not randomly selected but judiciously distant in series to show the effect of an experiment on a variety of frames. For each frame, the F1 score is also measured with respect to its ground truth. The visuals and F1 score concretely provide evidence that KDE methods consistently perform better than all other methods in every scenario. HD method has also performed well in nearly all scenarios except in two frames of baseline and shadow cases. The third ranker is AMD which has given good results especially in thermal imagery and shadow scenes.

Overall AMD, AM, HD and KDE methods are better than others. But if we consider ET then clear-cut AMD is better than others followed by AM for real-time video surveillance. Researchers are advised to enhance these two simple but efficient methods to make them more effective in a different scenario. Also, spatial feature along with the temporal history of pixel need to be considered to design a robust but efficient method for object detection.

4.3 DISCUSSION ON RESULTS OF COMPARATIVE ANALYSIS

This chapter presents a comprehensive overview of different methods of object detection. It has also evaluated eight most cited algorithms on a benchmark dataset CDNet2012 consisting of six diverse scenarios. Some of the algorithms are very simple and fast in image processing. These methods are evaluated on five different parameters namely execution time, precision, recall, F1 score and PSNR to measure the efficiency and effectivity. P-R curves illustrated against different threshold values for every method in each scenario show that simple statistical method such as AM, AMD are both efficient and effective along with KDE method but later is slow in execution time. Therefore, researchers are suggested to find ways and means to enhance simple and efficient methods AM and AMD to make them more robust for real-time video surveillance.

CHAPTER 5

EDGE BASED OBJECT DETECTION USING CELLULAR LOGIC ARRAY

An edge, within an image, is defined as a sudden change or discontinuity in the intensity of a pixel. It is regarded as a boundary between the object and its background. The process of edge detection characterizes an image into important features which can be further used in image processing functions of a higher level. An ideal edge detection process is that which identifies less number of false edges or double edges and a maximum number of real edges (Canny, 1986). The edge detector should perform equally good in different environments and contexts.

5.1 EDGE BASED OBJECT DETECTION IN STILL IMAGES

Most of edge detection programs work on pixel level by identifying the convex region of same gray level intensity and extracting edges by isolating pixel having a differing level of intensity by thresholding. Two different images may be shot in quite a different setup and they can differ in background intensities greatly due to variation in the reflectance, illumination, orientation, and depth of scene surfaces necessitating different values of threshold in edge detection process. This gives an idea that instead of fixing a single value for all types of images and scenarios or empirically identifying threshold values each time during edge detection process; a formula must be devised which is dependent on background intensities so that good quality edge detection can materialize every time. This research work presents a model toward this direction.

changed to 0 marking a uniform intensity region. This procedure is repeated for every pixel of the image resulting in creating uniform regions having a similar gray level. Uniform intensity levels are labeled as background pixels which result in edge detection in the form of differing intensity level pixels. Prof. Rajan has used a fixed threshold of value 20 in his work (Rajan, 1993) but we have experimented with adaptive thresholds. The pseudo code for global thresholding and local threshold procedure is given in **Algorithm 5-1** and **Algorithm 5-2** respectively.

Step 1: Create a 3*3 neighborhood scanning structure consisting of only 5-neighborhood pixels

Step 2: for every central_pixel (i, j) of image {

Step 2.1: find the Mean Intensity Value of 5-neighborhood pixels, set $Th_l = \beta I_{lm}$; β varies between 0 and 1

Step 2.2: find G_{min} = Min(5-neighborhood structure)

Step 2.3: find G_{max} = Max(5-neighborhood structure)

Step 2.4: if diff(G_{max} , G_{min}) < Th_l then central_pixel (i, j) = 0

else Move structure to the next pixel as a central pixel.

end for

Algorithm 5-2: Local Threshold based edge detection

5.2 THRESHOLD MODELING IN EDGE BASED DETECTION

As discussed above, threshold plays a great deal in determining the quality of edges. Adaptive threshold values can be decided by taking Global Threshold as the percentage value of *Mean Intensity Value* of an image i.e. Global Mean I_{gm}. We need to find out the optimal percentage value for proposing a model for automatic global threshold values for quality edge detection.

$$Th_{global} = \alpha \times I_{gm} \qquad \text{Eqn. (5-1)}$$

Several edge detection algorithms use neighborhood principles for digital images to detect edges by locating edge points in the form of abrupt changes in gray levels. *The cellular logic array processing based algorithm also uses neighborhood principle but in a different way. Instead of finding points of abrupt changes in gray levels, it locates the regions of the image where gray levels remain static*. In this method, every pixel in digital image is investigated for its intensity difference with its surrounding neighbors as done in cellular logic array and a uniform rule is defined to work on its intensity and that rule is: "*if the difference between maximum and minimum intensity over the neighborhood region is less than a threshold than the pixel value is changed to 0 otherwise it is changed to 255*". The threshold value may be fixed empirically or it may be chosen as the percentage value of Mean/Median Intensity Value of an image.

Step 1: Find the Mean Intensity Value I_{gm} of image

Step 2: Set threshold $Th_g = \alpha I_{gm}$; α varies between 0 and 1

Step 3: Create a 3*3 neighborhood scanning structure consisting of only 5- neighborhood pixels

Step 4: for every central_pixel (i, j) of image {

Step 4.1: find G_{min} = Min(5-neighborhood structure)

Step 4.2: find G_{max} = Max(5-neighborhood structure)

Step 4.3: if diff(G_{max} , G_{min}) < Th_g then central_pixel (i, j) = 0 else Move structure to the next pixel as a central pixel.

end for

Algorithm 5-1: Global Threshold based edge detection

For practical implementation, the target digital image is scanned by a 3 x 3 pixels' window of five pixels forming convex region to find maximum and minimum gray level. If the difference between maximum and minimum intensities is less than a threshold value then the intensity of the center pixel is changed to 0 marking a uniform intensity region. This procedure is repeated for every pixel of the image resulting in creating uniform regions having a similar gray level. Uniform intensity levels are labeled as background pixels which result in edge detection in the form of differing intensity level pixels. Prof. Rajan has used a fixed threshold of value 20 in his work (Rajan, 1993) but we have experimented with adaptive thresholds. The pseudo code for global thresholding and local threshold procedure is given in **Algorithm 5-1** and **Algorithm 5-2** respectively.

Step 1: Create a 3*3 neighborhood scanning structure consisting of only 5-neighborhood pixels

Step 2: for every central_pixel (i, j) of image {

Step 2.1: find the Mean Intensity Value of 5-neighborhood pixels, set $Th_l = \beta I_{lm}$; β varies between 0 and 1

Step 2.2: find G_{min} = Min(5-neighborhood structure)

Step 2.3: find G_{max} = Max(5-neighborhood structure)

Step 2.4: if diff(G_{max} , G_{min}) < Th_l then central_pixel (i, j) = 0 else Move structure to the next pixel as a central pixel.

end for

Algorithm 5-2: Local Threshold based edge detection

5.2 THRESHOLD MODELING IN EDGE BASED DETECTION

As discussed above, threshold plays a great deal in determining the quality of edges. Adaptive threshold values can be decided by taking Global Threshold as the percentage value of *Mean Intensity Value* of an image i.e. Global Mean I_{gm}. We need to find out the optimal percentage value for proposing a model for automatic global threshold values for quality edge detection.

$$Th_{global} = \alpha \times I_{gm} \quad \text{Eqn. (5-1)}$$

Another threshold can be a local value taken as the percentage of a local mean value from 5-neighborhood of five pixels which are evaluated for finding maximum and minimum values in CLAP algorithm. This is taken as

$$Th_{local} = \beta \times I_{lm} \qquad \text{Eqn. (5-2)}$$

Six Berkeley segmentation database (BSD) images are taken for the empirical evaluation. There are various ground truths for a single image in BSD dataset, (Martin, Fowlkes, D., & Malik, 2001) but only one ground truth-extractor image is taken to maintain the consistency in evaluation. For every image, a hundred values of precision and recall are calculated using extracted edges and corresponding ground truths by varying global threshold or local threshold through percentage value α and β respectively. Precision and Recall based F1 score, performance ratio (PR) and precision-recall break-even-points (PR-BEP) are used to find out the optimum value of α and β separately for both methods. Performance is the ratio of the total correctly identified pixel (TP) to the total incorrect pixels (FP+FN) in the image. Precision, which is also called positive predictive value, is the percentage of retrieved instances that are relevant and on the other side recall (sensitivity) is pea rcentage of relevant instances that are retrieved. Both in combined form give a better measure which is called F1 score.

Another popular measure for a classifier is a Precision Recall Break-Even-Point (PR-BEP). Normally precision and recall are opposite to each other. At threshold zero, recall is at unity but as threshold increases, it starts decreasing and precision starts increasing. PR-BEP is that threshold point, where precision and recall are equal. Many papers reported that it is a better measure than the F1 score (William, 2016). The local and global threshold based CLAP methods are also compared by Average Recall (AR) and Average Precision (AP). Experiments are done using OpenCV software.

Another popular measure for a classifier is a Precision Recall Break-Even-Point (PR-BEP). Normally precision and recall are opposite to each other. At threshold zero, recall is at unity but as threshold increases, it starts decreasing and precision starts increasing. PR-BEP is that threshold point, where precision and recall are equal. Many papers reported that it is a better measure than the F1 score (William, 2016). The local and global threshold based CLAP methods are also compared by Average Recall (AR) and Average Precision (AP). Experiments are done using OpenCV software.

Table 5-1: Different Measures extracted by CLAP edge algorithm for Global Threshold, when applied on six BSD images

	Global Threshold					
BSD Image No.	**AR**	**AP**	**Mean F1 score**	**Mean PR**	**α Value for best measure α_{f1m}**	**α Value for PR-BEP α_{bep}**
35010	0.273	0.539	0.287	0.234	0.07	0.17
42049	0.235	0.576	0.274	0.225	0.03	0.11
118035	0.219	0.525	0.248	0.190	0.05	0.11
135069	0.255	0.645	0.312	0.272	0.05	0.8
189011	0.154	0.432	0.158	0.103	0.08	0.10
189080	0.175	0.474	0.169	0.110	0.10	0.13

Table 5-2: Different Measures extracted by CLAP edge algorithm for Local Threshold, when applied on six BSD images

	Local Threshold					
BSD Image No.	**AR**	**AP**	**Mean F1 score**	**Mean PR**	**β Value for best measure β_{f1m}**	**β Value for PR-BEP β_{bep}**
35010	0.294	0.509	0.297	0.243	0.08	0.22
42049	0.380	0.553	0.403	0.365	0.04	0.27
118035	0.222	0.429	0.245	0.190	0.06	0.12
135069	0.316	0.628	0.370	0.330	0.08	0.12
189011	0.190	0.386	0.186	0.119	0.02	0.21
189080	0.233	0.373	0.200	0.137	0.19	0.22

Table 5-1 and **Table 5-2** present the results of CLAP edge algorithm for taking threshold from a global average of the image and from local average of 5-neighborhood respectively. Average Recall (AR) and Average Precision (AP) measures and Mean Performance ratio (MPR) for both methods are displayed for evaluation. The local thresholding based CLAP method fares well as compared to global thresholding CLAP in all measures except in precision measure. From this fact, it can be inferred that although global thresholding helps in better retrieval of pixels that are relevant, local thresholding also retrieved a higher fraction of relevant instances. So, local thresholding is better than global thresholding, as it is less noisy. This can be easily verified from qualitative analysis of images in **Fig. 5-1** by visual inspection.

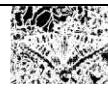
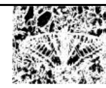
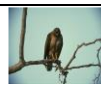
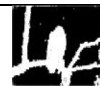
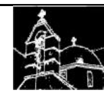
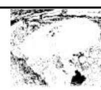
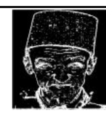
(a) BSD Sr. No.	(b) BSD Image	(c) Ground Truth	(d) Global Threshold with α_{f1m}	(e) Local Threshold with β_{f1m}	(f) Global Threshold with α_{bep}	(g) Local Threshold with β_{bep}
35010						
42049						
118035						
135069						
189011						
189080						

Fig. 5-1: Comparatively evaluation for CLAP Edge Detection with Global Threshold and CLAP edge detection with Local Threshold when best alpha is taken based on F1 score and PR-BEP.

It is also found that there is a considerable difference between the best values of alpha for the highest F1 score and for the PR-BEP for each method and image in both cases of thresholding. In **Fig. 5-1**, columns (d) to (g) shows the edge detection for all six images for these two best values of alpha and clearly, it can be established that PR-BEP measure works better than the F1 score in deciding the threshold value for automatic edge detection. Finally, for different images, it is found that there is a greater spread of alpha values for both measures F1 score and PR-BEP. Therefore, a single value can be deduced by taking an average of α_{bep} of different images for automatic thresholding in local thresholding and global thresholding based CLAP, but we need to work further on this area for better results. A hybrid of global and local thresholds may also give better result.

5.3 EDGE BASED OBJECT DETECTION IN VIDEO STREAMS

Object detection is a fundamental process in the analysis of video surveillance systems. Several methods for object detection have been proposed in the past but none presents a panacea to the various problems of object detection such as dynamics of the scene, occlusion, shadow, ghost, interleaved movements etc. These methods range from a simple and fast, yet ineffective frame difference method to effective but more complex, time-consuming parametric or nonparametric methods such as MoG and KDE respectively. Many algorithms such as optical flow, clustering based detection are just of academic interest due to their unsuitability in real video streams. In between to these contrasts, there remain many good statistical methods which produce good results in a varying situation within reasonable time limits which make them easily employable in real time video streams. There are methods such as KDE which provides good quality result in a different scenario but consumes much time. On the other side, statistical based adaptive average or adaptive median based background subtraction methods are not far behind than KDE in detection quality and simultaneously less time consuming making them better alternative than others in real time video surveillance application. These methods need to be improved further for better results. Edges in an image being less sensitive and robust to noise, shadow, dynamics of scene etc. can be effectively used in conjunction with these methods for better object detection.

Differential color or intensity against the background is sufficient to identify a moving object in an image. Due to this simple fact, edges act as an important tool in motion detection. Edges are high gradient features that easily help in identifying smallest of movement in the image thus provide accurate and robust motion information. There are other features such as texture, corners which can be used for motion detection but these are too few and costly on computation. Edges on the other side are macroscopic which provide enough motion information. Besides this, edges are photometric and geometric invariant to change between contiguous frames making matching and tracking effectively. They also provide reliable detection due to their long extent and continuous contour forcing all the pixels along an edge to follow object's motion. The edge pixels which are only around 5% in average of the total image also help to reduce the time taken for motion detection analysis. Moreover, human eyes are more sensitive to object edges than other image characteristics making them ideal for detection of objects in motion.

5.4 CELLULAR AUTOMATA BASED EDGED BACKGROUND SUBTRACTION METHODS FOR MOTION OBJECT DETECTION

Moving objects in a video can be obtained by taking the threshold of difference of contiguous frames but the slow frame rate and dynamic background may create problems. To avoid these, a reference background image is created from initial frames and then continuously updated with running mean (Adaptive Mean – AM) or median (Adaptive Median- AMD) methods. In edge based motion detection scheme, only edges of moving objects are detected. An edge by virtue of its characteristics is robust to noise but an edge should be continuously giving a clear contour for foreground image detection. In BGS methods of AM and AMD, the moving edges can be extracted at three points. One way of getting edge map is by taking the difference of two edge map of CF and BF respectively.

The second option is to get the difference first and then apply edge algorithm. The third option is to threshold the difference and then take edge map for post-processing functions of the median filter and fill function. Out of these three methods, the second option gave better result in experiments. BF is modeled as usual using running means and running median methods as the case may be.

The difference of edge map can be computed using standard edge detectors such as Canny or Sobel or CLAP edge detector and it is subsequently thresholded optimally to avoid noise in motion frame. Choosing an optimal threshold is crucial in these methods. The threshold value is modeled as a percentage value of global mean or local mean. Global thresholding can be assessed from the global mean of the image, while local thresholding can be determined separately for each pixel by taking it as the percentage (alpha) of the mean intensity value of local region surrounding the pixel. A single value of alpha can be determined for the automatic edge detection for all images but wider spread in the range of these values may limit us in this approach. Local thresholding based CLAP has given slightly better empirical results in still images as explained in the previous section.

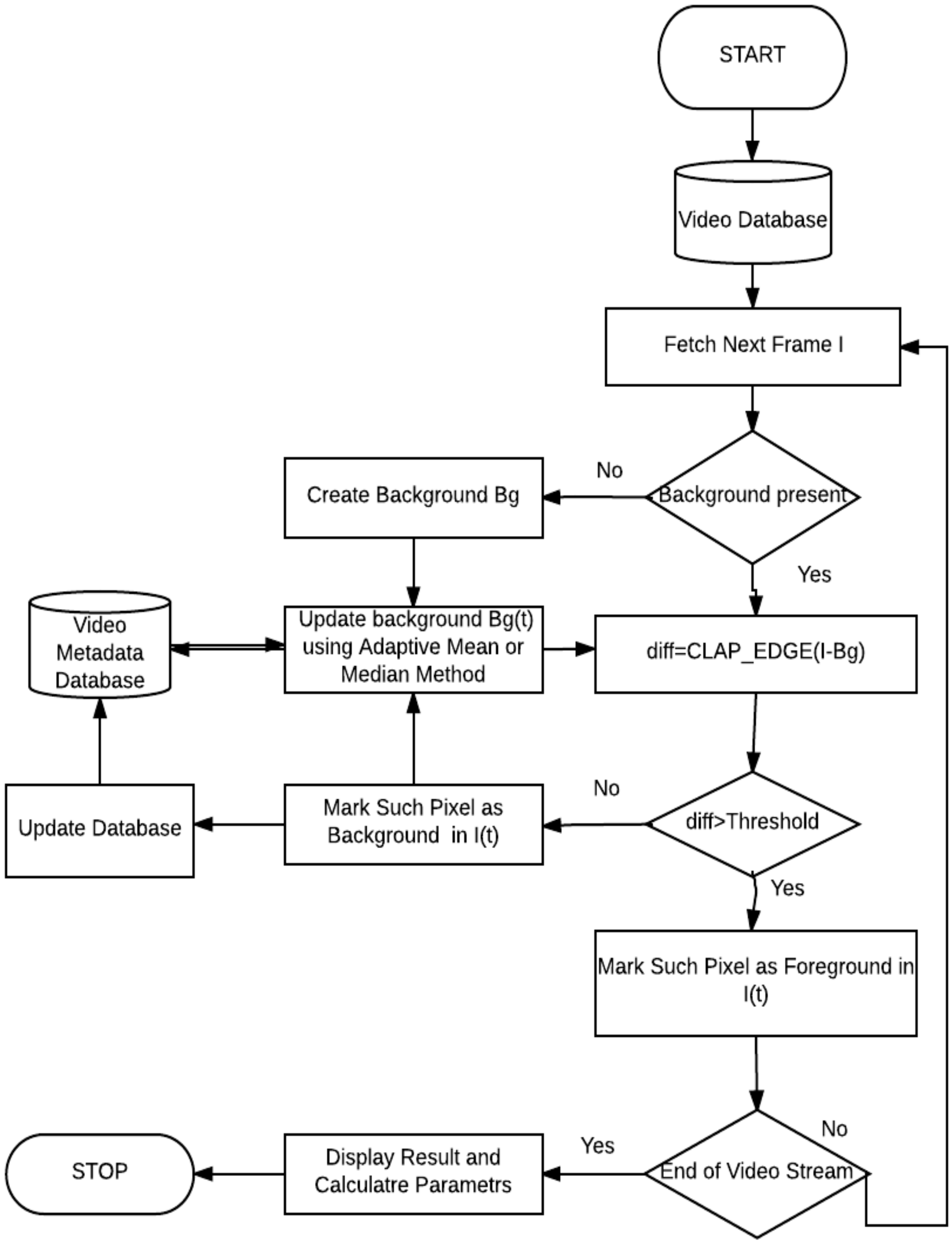


Fig. 5-2: Process Flow of CLAP-EDGE BGS Method

Step1:

(Fix Threshold)

Set $Th = 20$;   **OR**

(Global Thresholding- Th is defined as a percentage of global average)

Find the average of the image as I_{gm} and Set $Th = \alpha \times I_m$; where α ϵ [0, 1]

OR

(Local threshold- Th is defined as a percentage of local block average)

Create 16× 16 blocks of the image I and then replicate to recreate the image having uniform average intensity block of equal size called I_{lm}. and Set $Th = \beta \times I_{lm}$; where β ϵ [0, 1]

Step2: (CLAP_EDGE)

for the current frame image, scan 5-neighborhood for minimum and maximum intensity value G_{min} and G_{max} using the difference between *I* and background *B* in the following way:

Step2.1: Calculate I_N, I_E, I_S, I_W, by shifting I in four directions (viz. North, East, South, and West).

Step 2.2: Also calculate B_N, B_E, B_S, B_W by shifting B in four directions.

Step 2.3: Find $G_{min} = \min(abs(I - B), abs(I\_N - B\_N), abs(I\_E - B\_E), abs(I\_S - B\_S), abs(I\_W - B\_W))$

Step 2.4: Find $G_{max} = \max(abs(I - B), abs(I\_N - B\_N), abs(I\_E - B\_E), abs(I\_S - B\_S), abs(I\_W - B\_W))$

Step 2.5: if $diff(Gmax, Gmin) < Th$ then $I_{(i,j)} = 0$

else $I_{(i,j)} = 255$

end if

end for

Algorithm 5-3: The CLAP Algorithms for edge detection method

Step 1: Create an average image BG with some initial frames.
Step 2: For each of remaining frame say CF **Step 2.1:** Find Difference $D = (CF - BG)$ **Step 2.2:** Find edge map $EM = CLAP\_EDGE(abs\ (D))$ **Step 2.3:** Find foreground $FG(EM > Th) = 255$ **Step 2.4:** Update BG with the following equations: **(For AM)** $BG = \varepsilon \times CF + (1 - \varepsilon) \times BG$ **OR** **(For AMD)** $BG(D > 0) = BG(D > 0) + 1;$ $BG(D < 0) = BG(D < 0) - 1;$ End For

Algorithm 5-4: BGS algorithm with CLAP_EDGE

5.5 RESULTS OF CLAP-EDGE BASED EDGE DETECTION METHODS

5.5.1 Preliminary Investigation

In the preliminary investigation, the CLAP algorithm is implemented in MATLAB on an i3 processor 2.4GHz 4GB main memory system. For preliminary experimentation, one baseline scenario from CDnet2012 is taken to validate the results of the proposed algorithm with basic BGS algorithms. An adaptive edge detection method is used in which threshold for edge detection is not fixed but taken as a factor of average image intensities. This effect is represented and discussed in the results. Precision, Recall, and PSNR metrics are used to measure the quality of detection. Absolute time is also measured as frame processing time per second. Precision-Recall (PR) curve is drawn to find out the effect of different threshold on the outcome.

Table 5-3 Results for Adaptive Mean BGS Method and Global CLAP edge based Adaptive Mean Method

THRES-HOLD	Basic Mean Method					Global CLAP Mean Method				
	Prec.	Rec.	F1-Score	PSNR	Time (s)	Prec.	Rec.	F1-Score	PSNR	Time (s)
5	0.218	0.985	0.357	5.542	0.022	0.455	0.967	0.619	10.413	0.035
10	0.422	0.935	0.581	10.150	0.015	0.674	0.922	0.778	14.719	0.031
15	0.566	0.879	0.689	12.918	0.012	0.798	0.873	0.834	16.995	0.033
20	0.683	0.824	0.747	14.844	0.012	0.866	0.823	0.844	17.942	0.033
25	0.789	0.769	0.779	16.214	0.012	0.913	0.768	0.834	18.172	0.032
30	0.871	0.716	0.786	16.890	0.012	0.943	0.705	0.807	17.900	0.034
35	0.923	0.665	0.773	17.011	0.010	0.963	0.643	0.771	17.406	0.032
40	0.955	0.618	0.751	16.846	0.010	0.977	0.585	0.732	16.887	0.032
45	0.976	0.574	0.723	16.569	0.012	0.987	0.531	0.691	16.418	0.028
50	0.989	0.532	0.692	16.244	0.014	0.995	0.482	0.649	16.020	0.029

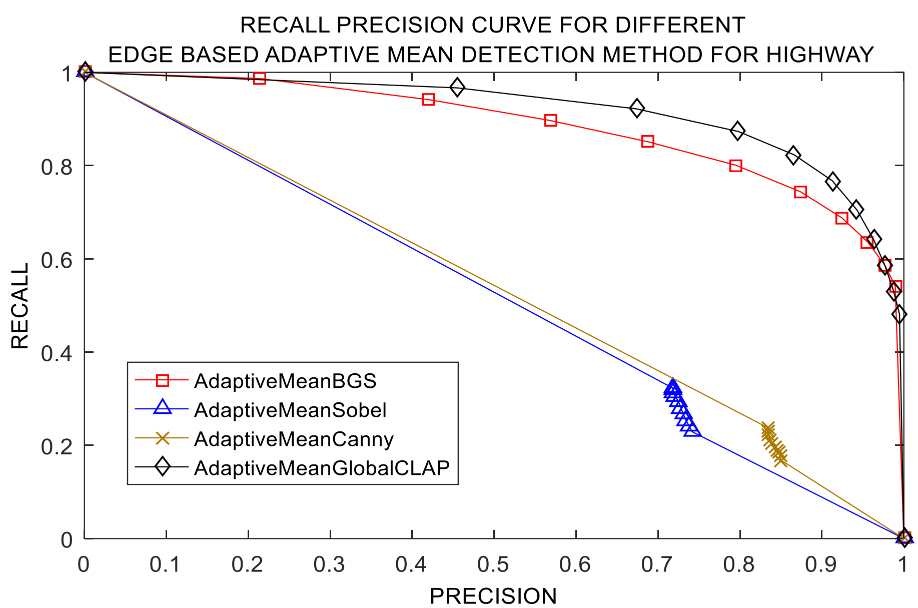


Fig. 5-3: P-R curves for Adaptive Mean BGS method and Edge-Based Adaptive Mean Methods

The research work implements and evaluates existing popular methods Sobel and Canny along with the proposed CLAP based edge detection algorithm for the purpose of detecting moving objects in video streams. The effectivity of these methods is measured on two scales, time and quality as explained in chapter 3. A post-processing median filter with the 5×5 neighborhood is used for spurious noise removal. Several challenging scenarios are used to test the proposed method on a larger scale with an adaptive threshold. The whole scheme of CLAP based edge detection BGS method is shown as a flowchart in **Fig. 5-2.** The pseudo code for the proposed CLAP algorithm is revealed in **Algorithm 5-3** and **Algorithm 5-4**.

Table 5-3 and **Table 5-4** show the results obtained for basic method AM and AMD respectively and CLAP methods for different thresholds when applied on highway data set. Based on these table data, the PR curves are drawn for AM and AMD methods both as shown in **Fig. 5-3** and **Fig. 5-4**. Clearly CLAP based edge detection methods are superior to other methods. Sobel and Canny based edge methods' results are also shown in the PR curves. Canny and Sobel methods are very high in precision but miserably fail to recall the true positives. The very reason for this may be due to its broken boundaries while the CLAP extracts continuous contours which are easy to fill as a segment and are better in results. The CLAP results have surpassed basic method where the whole image is extracted as it is, instead of edges but it consumes nearly 50% more processing time than. So preliminary investigation on a single scenario suggests better results but more experiments need to be done before making a general statement. A snapshot of the visual output of the AM BGS experiment is displayed in **Fig. 5-5.**

Table 5-4: Results for Median BGS Method and CLAP Median Method

TH	Basic Median Method					CLAP Median Method				
	Prec.	Rec.	F1-Score	PSNR	Time (s)	Prec.	Rec.	F1-Score	PSNR	Time (s)
5	0.517	0.964	0.673	11.902	0.018	0.594	0.953	0.732	13.257	0.032
10	0.751	0.903	0.820	16.193	0.013	0.810	0.899	0.852	17.315	0.031
15	0.851	0.853	0.852	17.795	0.012	0.894	0.857	0.875	18.772	0.029
20	0.913	0.806	0.856	18.454	0.012	0.937	0.817	0.873	19.227	0.032
25	0.951	0.759	0.844	18.523	0.011	0.963	0.771	0.857	19.086	0.031
30	0.974	0.713	0.823	18.231	0.012	0.978	0.721	0.830	18.598	0.028
35	0.985	0.669	0.797	17.773	0.012	0.987	0.670	0.798	17.957	0.030
40	0.991	0.627	0.768	17.294	0.013	0.992	0.620	0.763	17.354	0.028
45	0.994	0.586	0.738	16.845	0.011	0.995	0.571	0.726	16.791	0.028
50	0.997	0.547	0.707	16.439	0.011	0.997	0.527	0.689	16.333	0.029

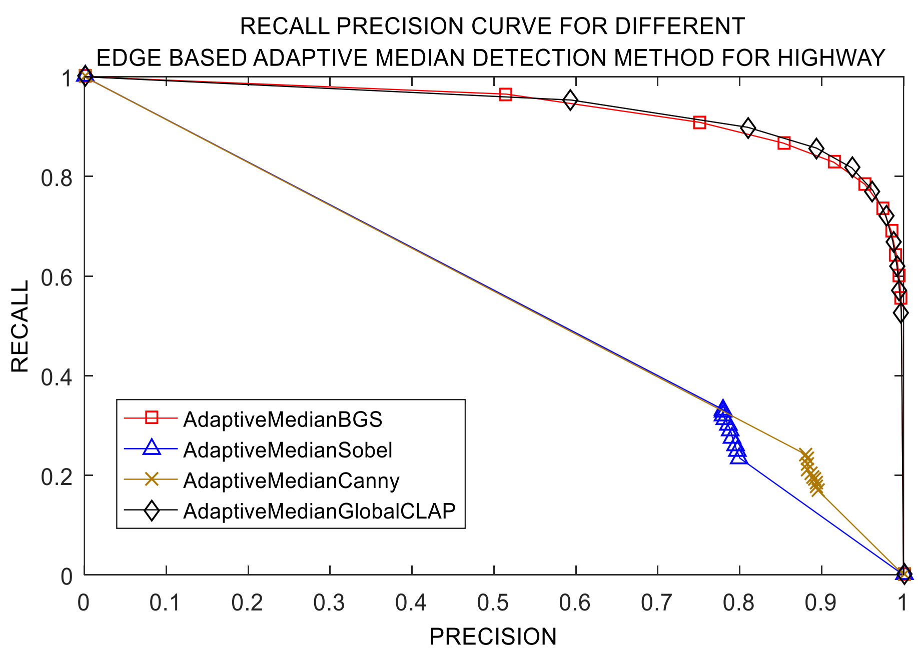


Fig. 5-4: Results for Adaptive Median BGS method and Edge-Based Adaptive Median Methods

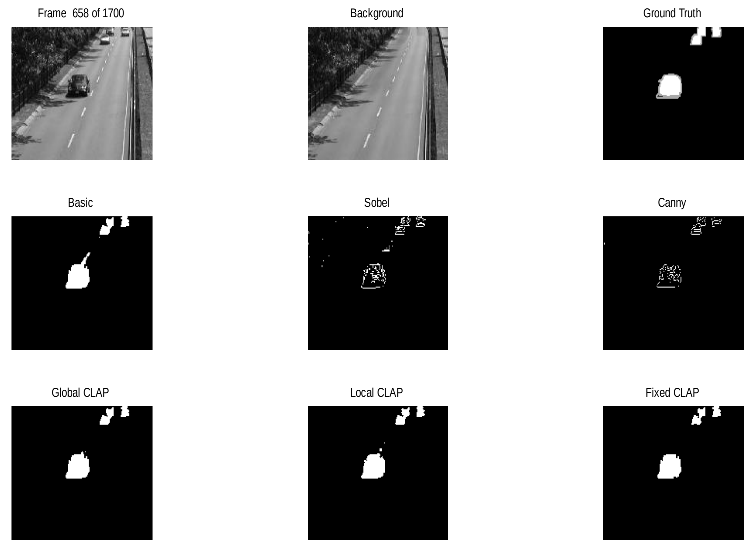


Fig. 5-5: A snapshot of BGS output with different Edge detection algorithms

5.5.2 Comparative Evaluation of Edge based Object Detection Algorithm

In detail investigation of edge-based detection, we have tested CLAP based BGS methods for different types of threshold values viz. fixed threshold, global adaptive threshold and local adaptive threshold. These three variants are compared with the basic method as well as with Sobel and Canny edge detectors. There are two BGS methods; AM and AMD, and six scenarios resulting total 12 PR curves charts. In each PR curve chart six algorithms namely basic BGS, Sobel Edge, Canny Edge and three flavors of CLAP edge (fixed, local and global threshold) are included**.** **Fig. 5-6** to **Fig. 5-10** are for edge-based AM method and **Fig. 5-11** to **Fig. 5-16** are for edge-based AMD method.

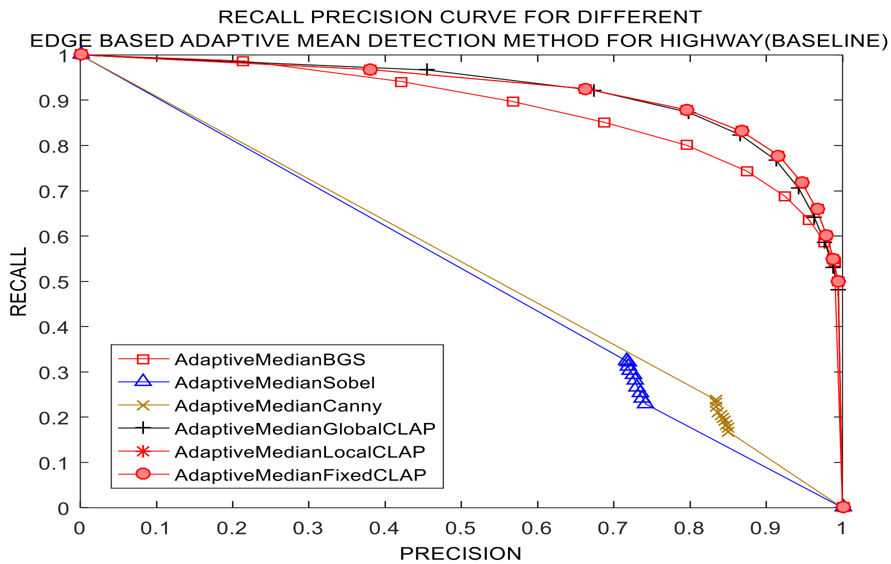


Fig. 5-6: Precision Recall Curve for different Edge based AM object detection methods for Highway data set

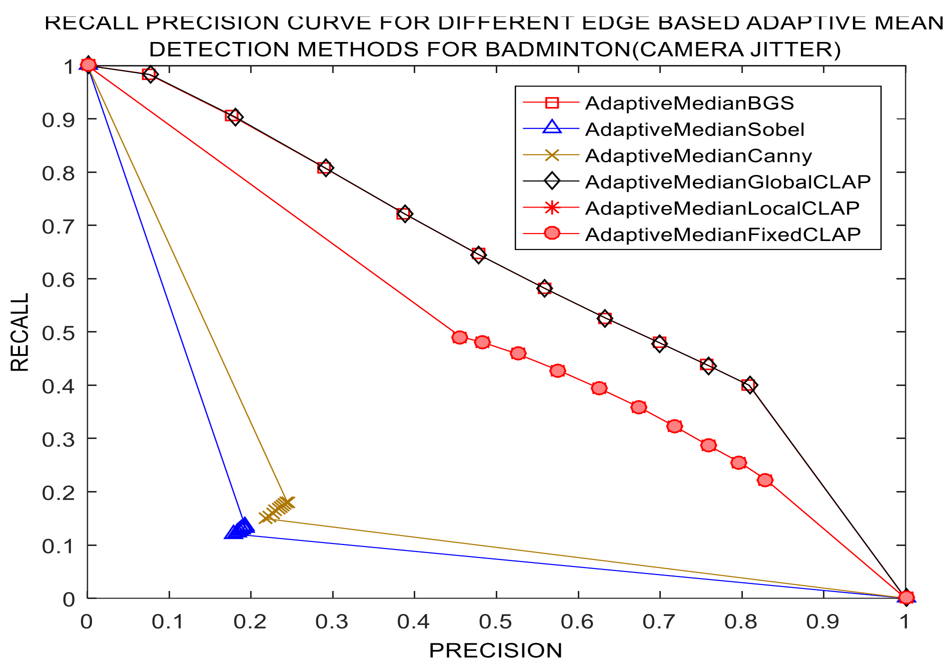


Fig. 5-7: Precision Recall Curve for different Edge based AM object detection methods for Badminton data set

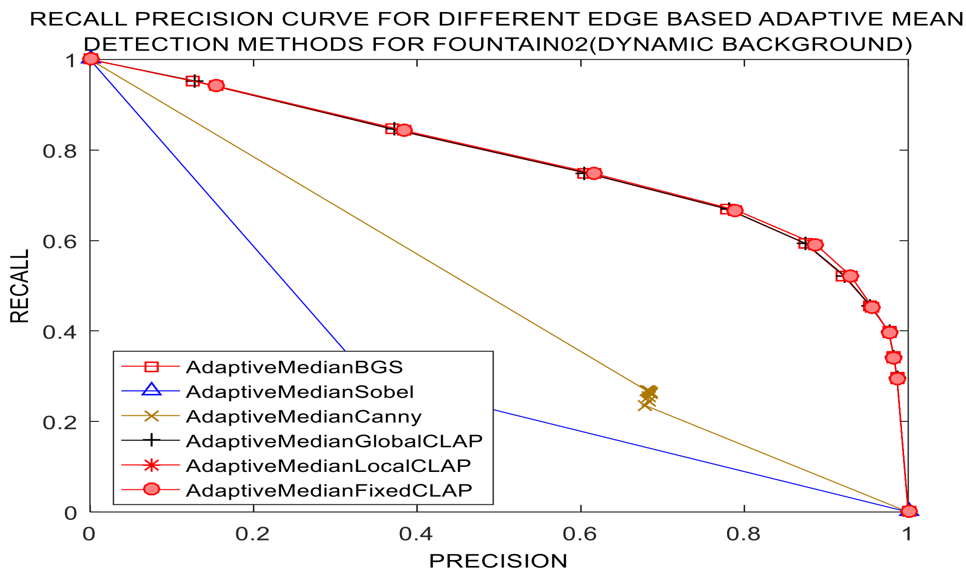


Fig. 5-8: Precision Recall Curve for different Edge based AM object detection methods for Fountain02 Dataset data set

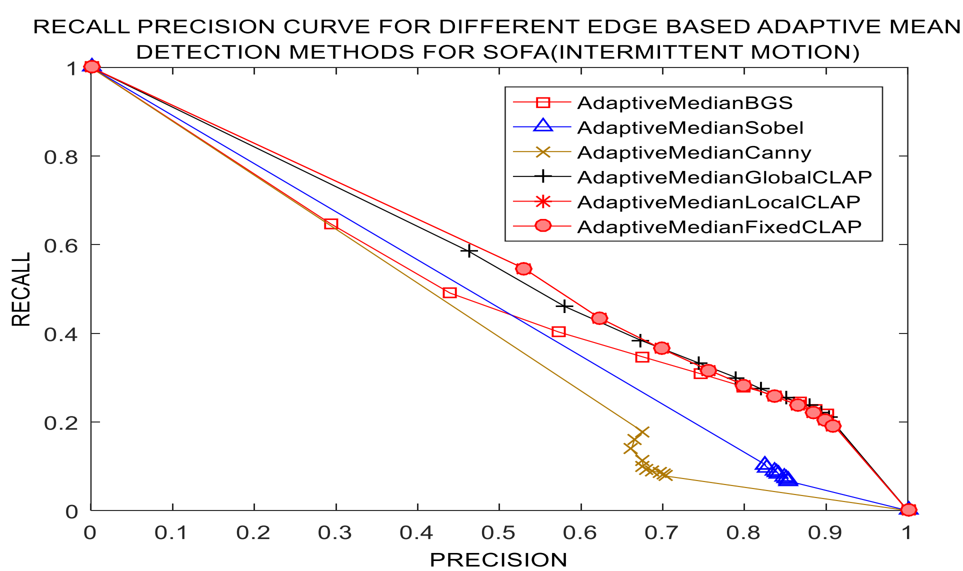


Fig. 5-9: Precision Recall Curve for different Edge based AM object detection methods for Sofa data set

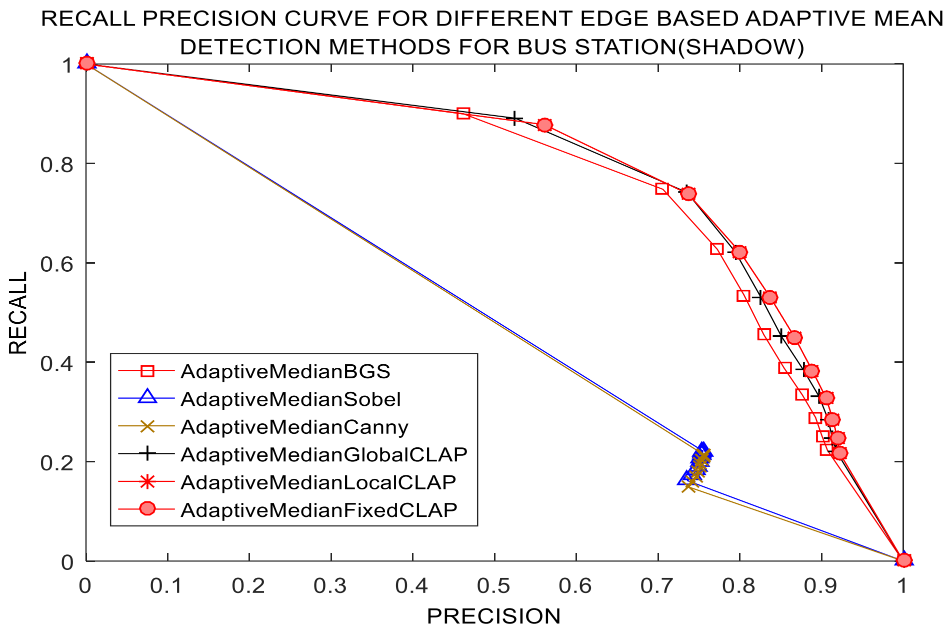


Fig. 5-10: Precision Recall Curve for different Edge based AM object detection methods for Bus Station data set

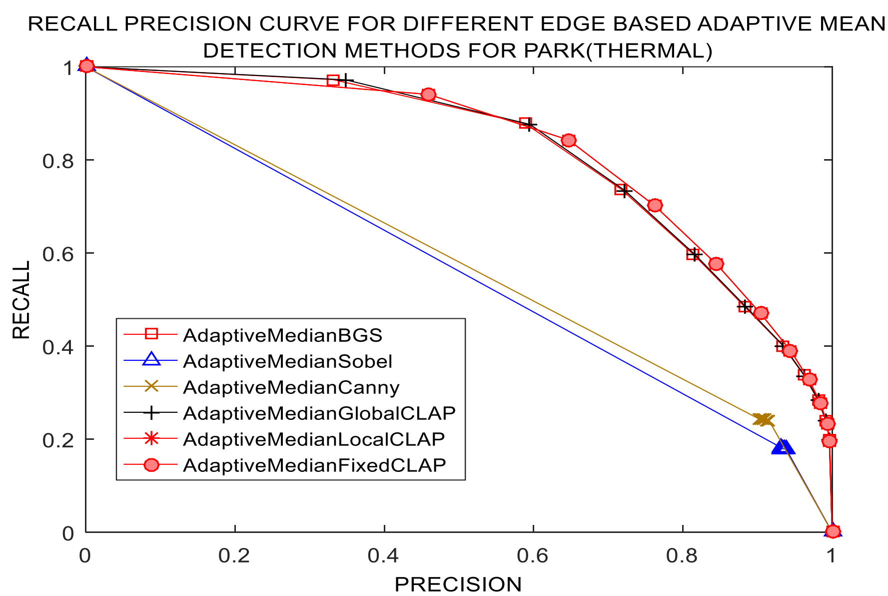


Fig. 5-11: Precision Recall Curve for different Edge based AM object detection methods for Park data set

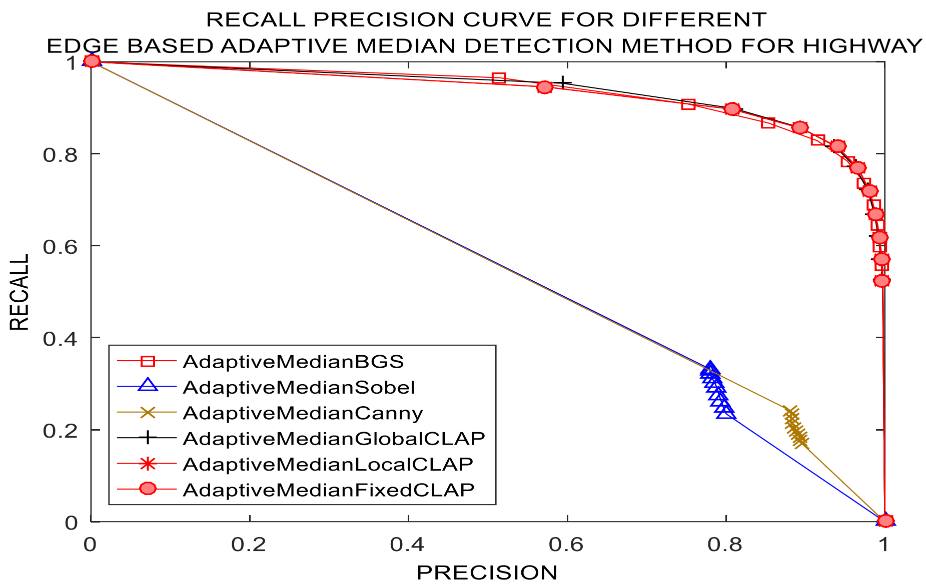


Fig. 5-12: Precision Recall Curve for different Edge based AMD object detection methods for Highway data set

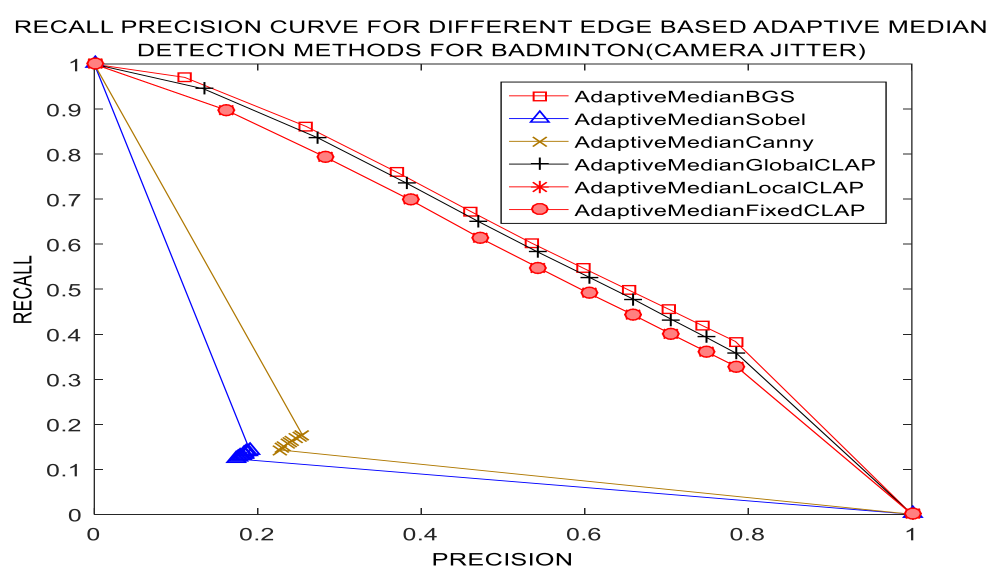


Fig. 5-13: Precision Recall Curve for different Edge based AMD object detection methods for Badminton data set

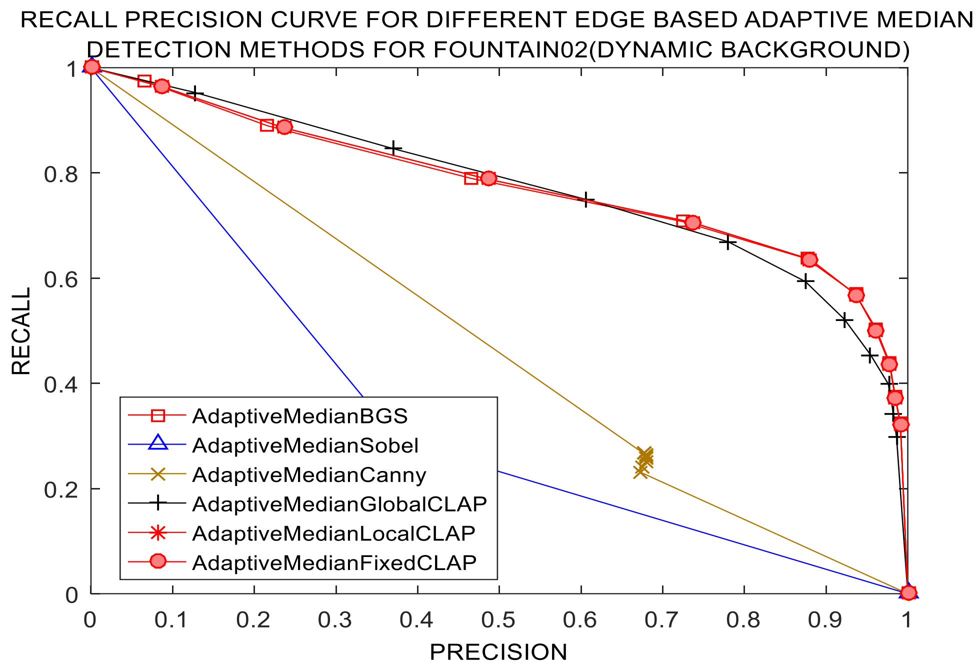


Fig. 5-14: Precision Recall Curve for different Edge based AMD object detection methods for Fountain02 Dataset data set

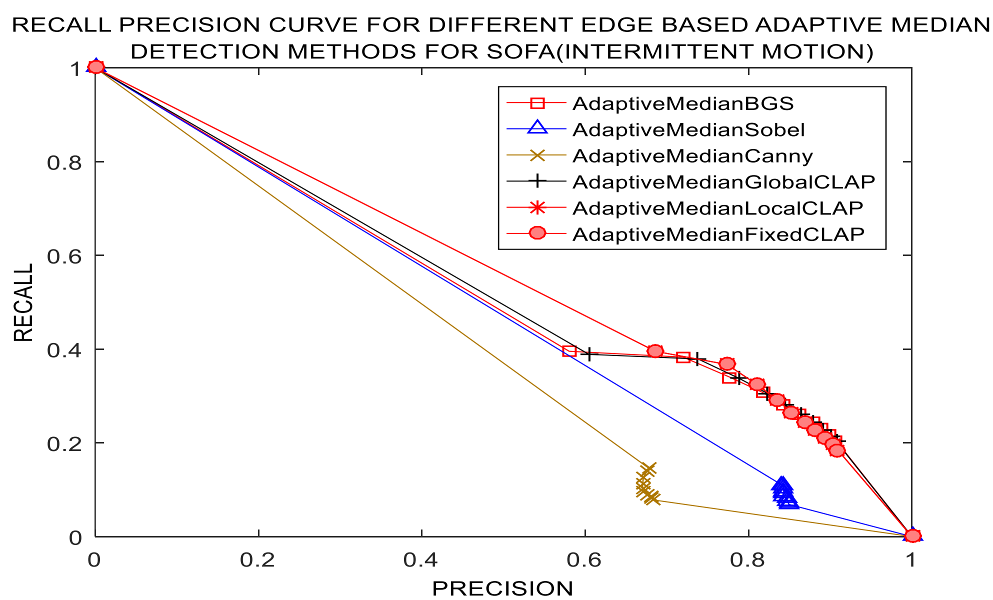


Fig. 5-15: Precision Recall Curve for different Edge based AMD object detection methods for Sofa data set

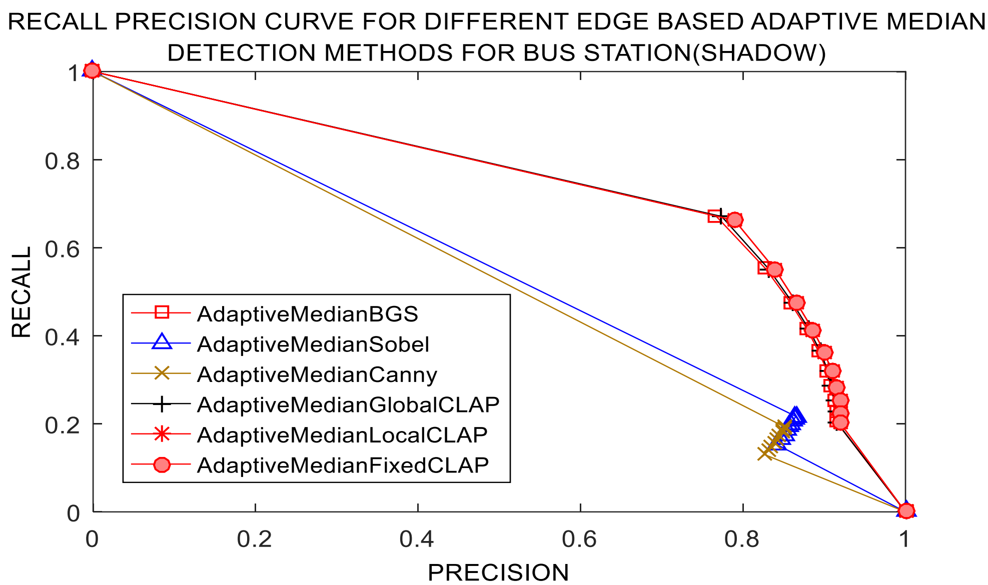


Fig. 5-16: Precision Recall Curve for different Edge based AMD object detection methods for Bus Station data set

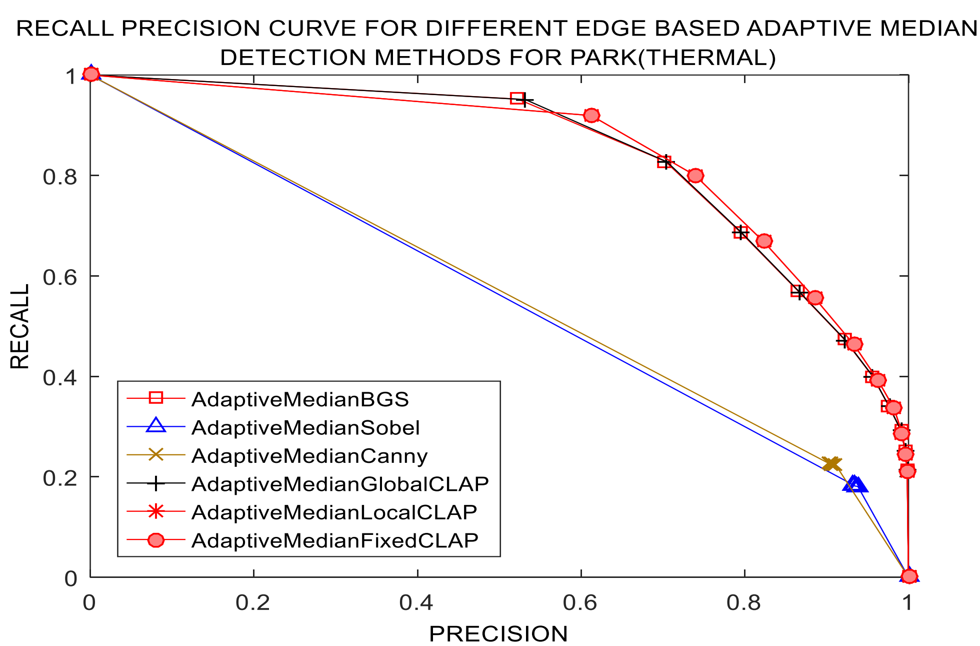


Fig. 5-17: Precision Recall Curve for different Edge based AMD object detection methods for Park data set

In most of the scenarios except camera jitter and dynamic background, there are improvements in quality detection if basic BGS and CLAP_EDGE BGS are compared, but when other standard edge extraction methods Sobel and Canny are considered and compared with CLAP_EDGE methods there is remarkable performance achieved by the CLAP based method. Besides this, we have also observed that Canny also takes nearly 2 to 3 times more average time of processing than CLAP method, making it unsuitable for real-time video surveillance environment. Another observation is that there is no remarkable difference in performance among three flavors of CLAP_EDGE detection methods. In paper (Singh, Prasad, Srivastava, & Bhattacharya, 2017), we have reported that local thresholding provided better result for edge detection from an image, where threshold is taken as percentage average of 5-neighborhood but in motion detection due to speed consideration of algorithm, we have taken average of larger block which may result in similar performance in all the three variants.

5.6 DISCUSSION ON RESULTS OF CLAP-EDGE BASED EDGE DETECTION METHODS

This chapter empirically evaluates five edge based background subtraction algorithms with basic AM and AMD BGS algorithm. Improvement in detection quality is noticed as compared to basic algorithms in baseline and intermittent object motion scenarios, but failed to register improvement in camera jitter and dynamic background scenarios. It is observed that CLAP-EDGE methods failed to model multimodal background. These algorithms are simply acting as de-noising filters, but in comparison to other traditional edge algorithms, the CLAP-EDGE based detection methods performed far better. It is also observed that local thresholding need smaller 5-neighborhood blocks for better performance which need to be evaluated further.

CHAPTER 6

LOCAL NEIGHBORHOOD DIFFERENCE BASED OBJECT DETECTION USING CELLULAR LOGIC ARRAY

This work proposes a novel scheme of "local neighborhood differencing" for finding the difference between the current frame and the current background image in BGS methods. It further validates the proposed methodology empirically on six sequences of a benchmarked dataset.

6.1 The Methodology- Local Neighborhood Differencing (LND)

In the methods described in previous chapters, the difference between two consecutive frames (in case of FD) or between the current frame I and background B is computed using the corresponding pixels (x, y) only as shown in **Eqn. (1-2).** The LND based methodology suggests to consider 9-neighborhood of a pixel while taking dithe fference. Instead of subtracting two corresponding pixels in I and B, it requires to subtract the average of 9-neighborhood of corresponding pixels to determine difference $Diff(x, y)$ as calculated in **Eqn. (6-1).**

$$LND(x,y) = \frac{1}{9}\sum_{i=x-1}^{i=x+1}\sum_{j=y-1}^{j=y+1} abs(I_t(i,j) - B_t(i,j))$$

Eqn. (6-1)

To make it more clear, let us take an example in which matrix I is the current image of 8×8 and B is Background of 8×8 and a duplicate border makes these images of 10×10 size for LND processing as shown in **Fig. 6-1** and **Fig. 6-2** respectively.

	1	2	3	4	5	6	7	8	9	10
1	104	104	58	60	49	52	58	61	59	59
2	104	104	58	60	49	52	58	61	59	59
3	87	87	41	42	38	39	45	54	48	48
4	83	83	34	32	32	34	36	46	43	43
5	155	155	65	31	37	36	39	40	42	42
6	118	118	31	55	58	60	64	62	62	62
7	121	121	58	66	73	76	79	81	76	76
8	128	128	106	98	103	104	101	109	101	101
9	138	138	130	124	120	119	118	123	121	121
10	138	138	130	124	120	119	118	123	121	121

Fig. 6-1: The Current frame *I*

	1	2	3	4	5	6	7	8	9	10
1	75	75	69	76	74	74	78	80	82	82
2	75	75	69	76	74	74	78	80	82	82
3	64	64	56	62	61	62	67	69	71	71
4	60	60	51	54	55	58	61	63	65	65
5	143	143	75	49	58	59	62	65	67	67
6	110	110	58	80	73	72	74	77	78	78
7	116	116	79	94	87	87	88	90	89	89
8	132	132	114	117	113	112	113	114	111	111
9	144	144	132	131	128	127	128	127	124	124
10	144	144	132	131	128	127	128	127	124	124

Fig. 6-2: The Current Background *B*

The simple difference can be achieved by pixel to pixel subtraction between two images, but LND suggests that in order to calculate the difference of a particular pixel say 3rd row and 4th column (3,4) we must find the difference of the corresponding 9-neighborhood of the pixel and then averaging it out to determine the difference. For example, the simple absolute difference d of the pixel (3, 4) is

$$Diff(3,4) = d(3,4) = abs(42 - 62) = 20 \qquad \text{Eqn. (6-2)}$$

	1	2	3	4	5	6	7	8	9	10
1										
2		22.11	18.89	18.00	21.33	22.44	20.22	20.44	21.22	
3		21.44	19.56	19.11	22.00	23.00	20.78	20.67	20.78	
4		17.56	17.78	18.78	21.89	23.00	21.89	21.89	21.89	
5		15.56	18.00	19.78	20.33	19.56	19.33	19.78	20.33	
6		12.00	17.11	19.89	18.56	15.33	15.22	16.11	17.44	
7		10.00	16.11	18.56	15.78	11.22	10.11	11.00	11.89	
8		6.78	11.11	13.00	12.56	10.00	8.44	8.33	7.78	
9		4.89	6.78	7.89	9.22	9.11	7.67	6.78	5.00	
10										

Fig. 6-3: LND of the current Frame *I* and the current Background *B*

LND is an average difference of 9-neighborhood of (3, 4) pixel which is shown in gray color in **Fig. 6-1** and **Fig. 6-2**. According to **Eqn. (6-1),** the average difference of pixel (3,4) is calculated as

$$LND(3,4) = \frac{d(2,3)+d(2,4)+d(2,5)+d(3,3)+d(3.4)+d(3.5)+d(4,3)+d(4,4)+d(4,5)}{9}$$

i.e. $$LND(3,4) = \frac{11+16+25+15+20+23+17+22+23}{9} = 19.11$$

Our hypothesis is that LND can effectively avoid sporadic noise to become a part of motion frame because if in a 9 neighborhood only one or two pixels are changed due to error, the average difference of the total neighborhood will be still less than threshold and pixel in consideration can't be designated as a part of foreground.

LOCAL_NEIGHBORHOOD_DIFF(*I, B*)

for every pixel (i, j) of current frame image *I* and background *B*

scan 9-neighborhood for finding average difference intensity value of the neighborhood

$$G_{avg} = \frac{1}{9}\sum_{r=i-1}^{i+1}\sum_{c=j-1}^{j+1} abs(I_{r,c} - B_{r.c})$$

end for

Algorithm 6-1: The LOCAL_NEIGHBORHOOD _DIFF Algorithm

Step 1: Create an average image *B* with some initial frames.

Step 2: For each of remaining frame say *I*

Step 2.1: Find Difference

$Diff = LOCAL\_NEIGHBORHOOD\_DIFF\ (I, B)$

Step 2.3: Find foreground $FG(Diff > Th) = 255$

Step 2.4: Update BG with the following equations:

(For AM) $B = \varepsilon\ \times I + (1 - \varepsilon) \times B$

where ε is frame refreshing rate.

OR

(For AMD) $B(Diff > 0) =\ B(Diff > 0) + 1;$

$B(Diff < 0) =\ B(Diff < 0) - 1;$

End For

Algorithm 6-2: BGS algorithm with CLAP_NEIGHBORHOOD _DIFF

Table 6-1: Comparison of Basic AM and LND_AM method with fixed Threshold (FT) and Adaptive Threshold(AT) for the best threshold value

Scenarios	Basic adaptive Mean with Fix Threshold	LND adaptive Mean with Fix Threshold	% Improvement	Basic adaptive Mean with Adaptive Threshold	LND adaptive Mean with Adaptive Threshold	% Improvement
Highway	0.7857	0.8225	4.7%	0.7869	0.8237	4.7%
Badminton	0.5719	0.5751	0.6%	0.5718	0.5750	0.6%
Fountain02	0.7275	0.7712	6.0%	0.7274	0.7719	6.1%
Sofa	0.4688	0.4825	2.9%	0.4692	0.4824	2.8%
Bus Station	0.7176	0.7214	0.5%	0.7155	0.7183	0.4%
Park	0.7052	0.7188	1.9%	0.7058	0.7211	2.2%

Table 6-2: Comparison of Basic AMD and LND_AMD method with fixed Threshold (FT) and Adaptive Threshold(AT) for the best threshold value

Scenarios	Basic adaptive Median with Fix Threshold	LND adaptive Median with Fix Threshold	% Improvement	Basic adaptive Median with Adaptive Threshold	LND adaptive Median with Adaptive Threshold	% Improvement
High Way	0.8558	0.8879	3.8%	0.8565	0.8891	3.8%
Badminton	0.5677	0.5679	0.0%	0.5675	0.5677	0.0%
Fountain02	0.6960	0.7436	6.8%	0.7096	0.7522	6.0%
Sofa	0.4948	0.5201	5.1%	0.4988	0.5222	4.7%
Bus station	0.7124	0.7357	3.3%	0.7130	0.7247	1.6%
Park	0.7435	0.7593	2.1%	0.7421	0.7569	2.0%

6.2 LND BASED OBJECT DETECTION

In order to extract motion from the current frame *I*, a BGS method subtracts the background frame *B* from the current frame *I*. The LND based object detection method requires to use local neighborhood difference as explained in the previous section. The algorithm for finding LND is given in **Algorithm 6-1** which is subsequently used in AM or AMD BGS method as depicted in **Algorithm 6-2**.

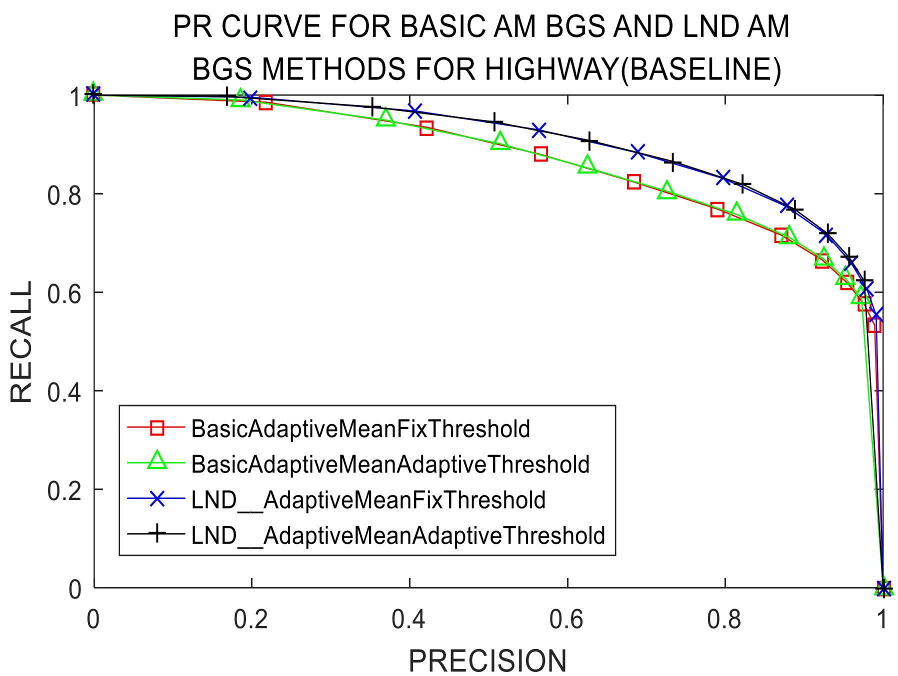


Fig. 6-4: Precision Recall Curve for LND based Adaptive Mean object detection methods for Highway data set

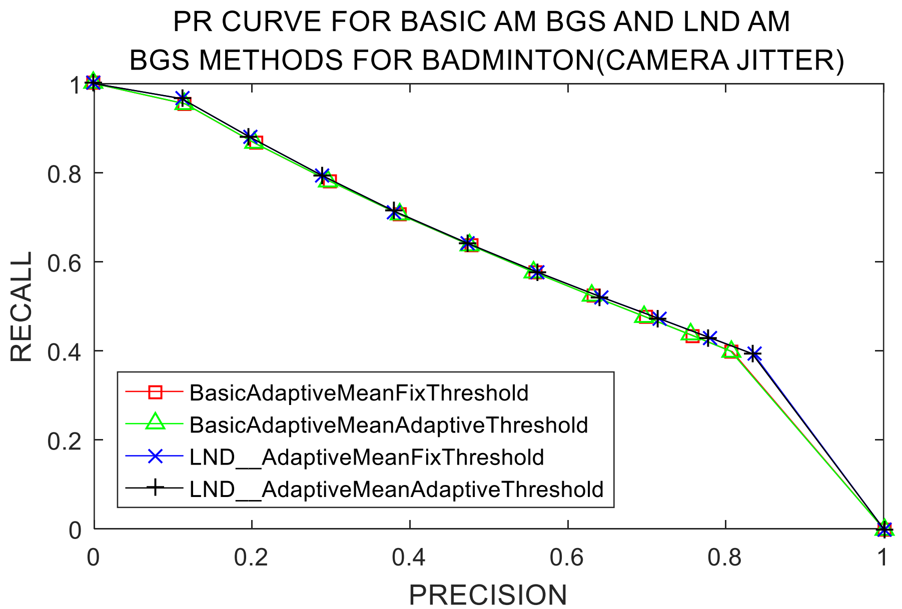


Fig. 6-5: Precision Recall Curve for LND based Adaptive Mean object detection methods for Badminton data set

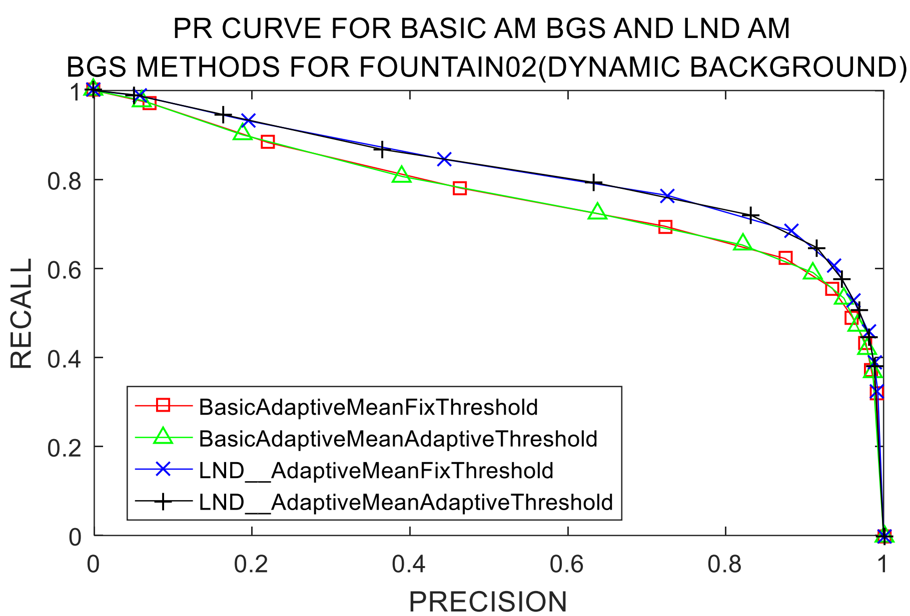


Fig. 6-6: Precision Recall Curve for LND based Adaptive Mean object detection methods for Fountain02 Dataset data set

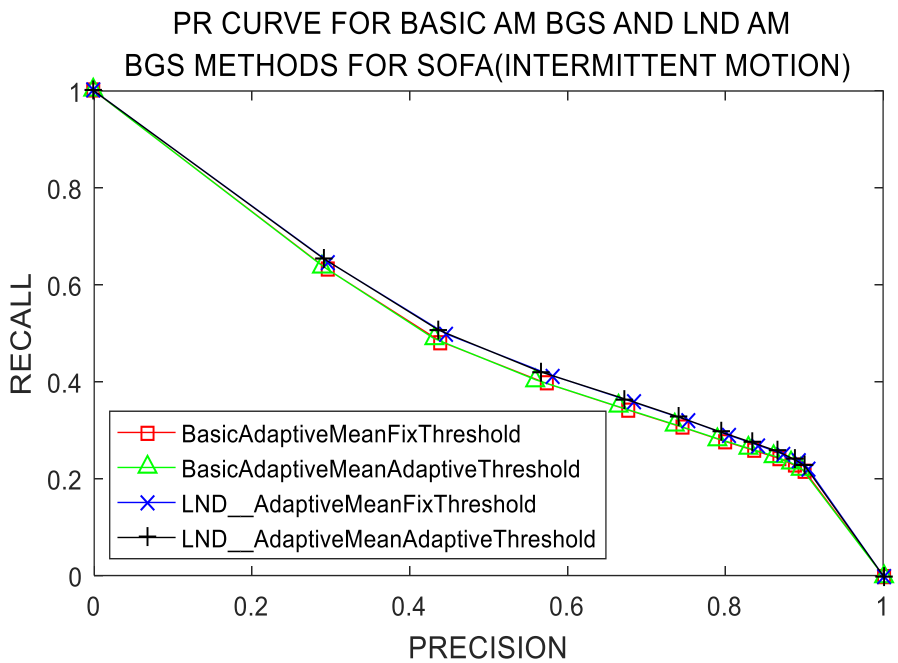


Fig 6-7: Precision Recall Curve for LND based Adaptive Mean object detection methods for Sofa data set

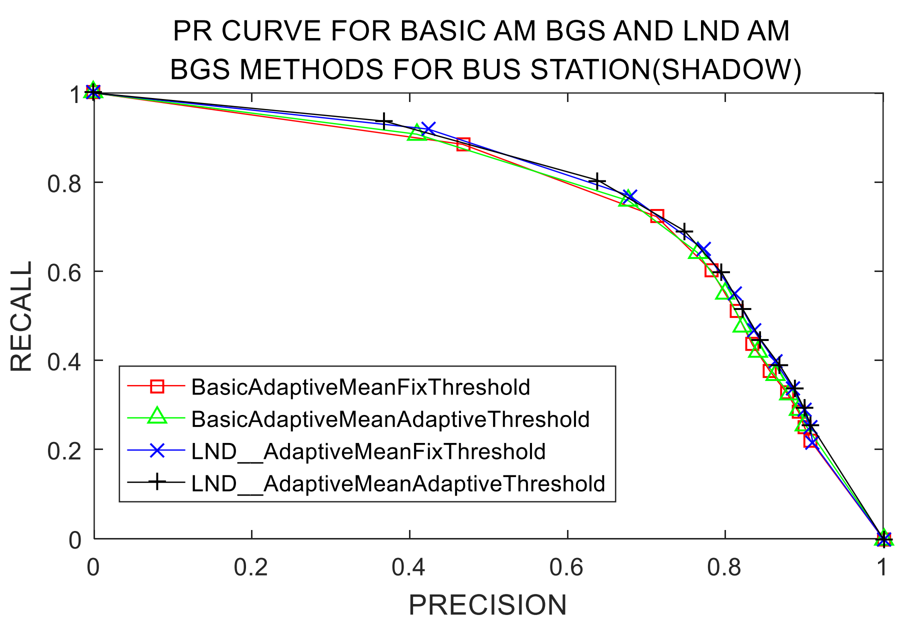


Fig. 6-8: Precision Recall Curve for LND based Adaptive Mean object detection methods for Bus Station data set

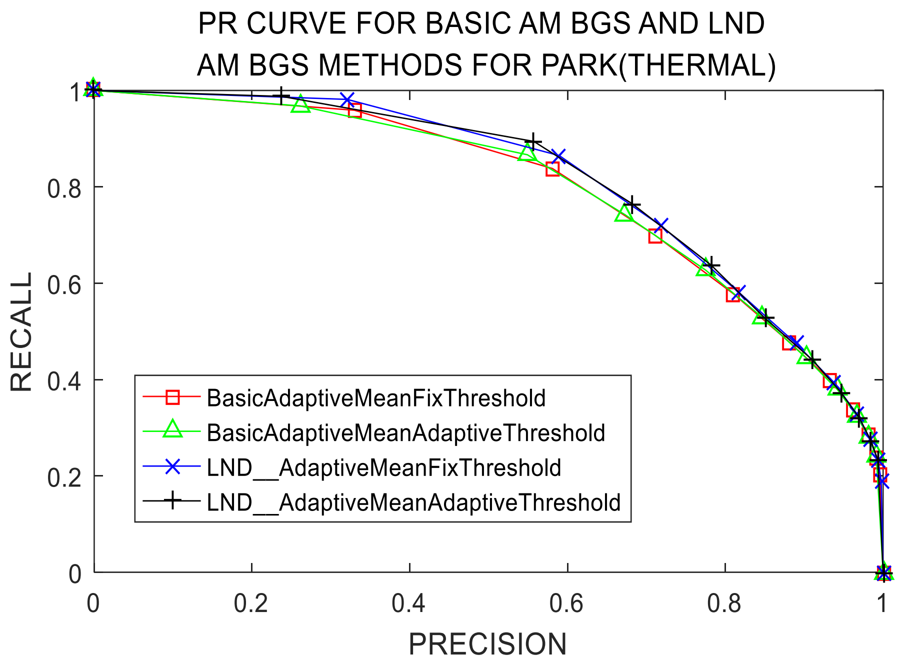


Fig. 6-9: Precision Recall Curve for LND based Adaptive Mean object detection methods for Park data set

6.3 RESULTS OF LND BASED METHODS

Experimental setup and benchmark test data are same as explained in chapter 3. Two basic BGS methods; AM and AMD are tested for six sequences for the proposed method to generate total 12 P-R charts. In each chart, four algorithms namely basic BGS with fixed thresholding (Basic_AM_FT), basic BGS with adaptive thresholding (Basic_AM_AT), LND based BGS with fixed thresholding (LND_AM_FT) and LND based BGS with adaptive thresholding (LND_AM_AT) are represented. **Fig. 6-4 to Fig. 6-9** are for AM methods and **Fig. 6-10** to **Fig. 6-15** are for AMD methods.

Table 6-1 and

Table 6-2 display F1 score for all the four type variations of Adaptive Mean and Adaptive Median BGS methods respectively. The last column of each table registers the percentage improvement achieved between basic method and LND based method with a fixed threshold.

In most of the scenarios, there is only a minor improvement if basic fixed threshold BGS and adaptive threshold basic BGS methods are compared. There is a very minor improvement with a global adaptive threshold, but when LND method is used in basic BGS method, notable improvement ranging from 4% to 7% is achieved in both Mean and Median methods for baseline and dynamic background sequence. For other sequences, the improvement is between 1% to 3%. Camera Jitter is the only scenario which doesn't register any improvement due to the definite multimodal background which is difficult to address by single modal methods such as AM and AMD.

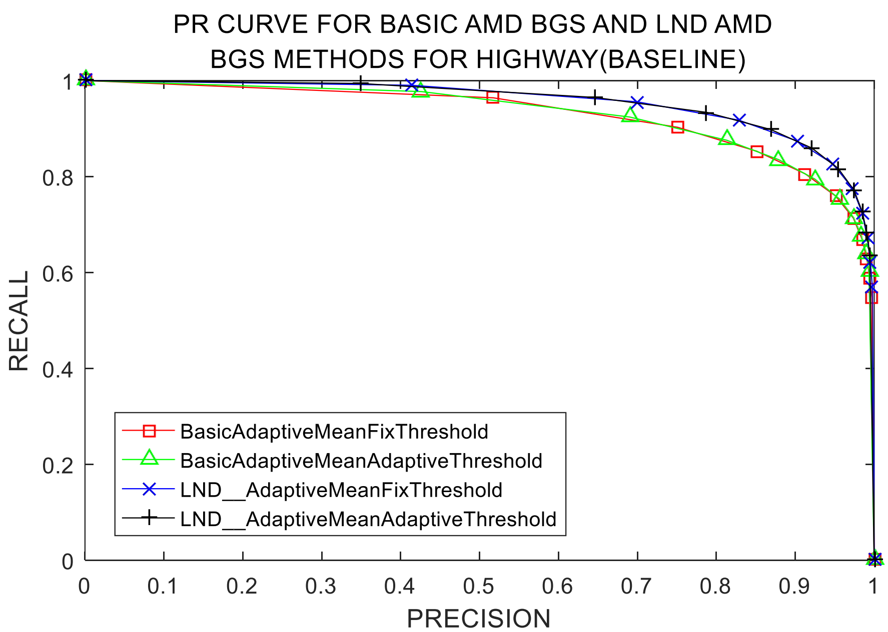


Fig. 6-10: Precision Recall Curve for LND based Adaptive Median object detection methods for Highway data set

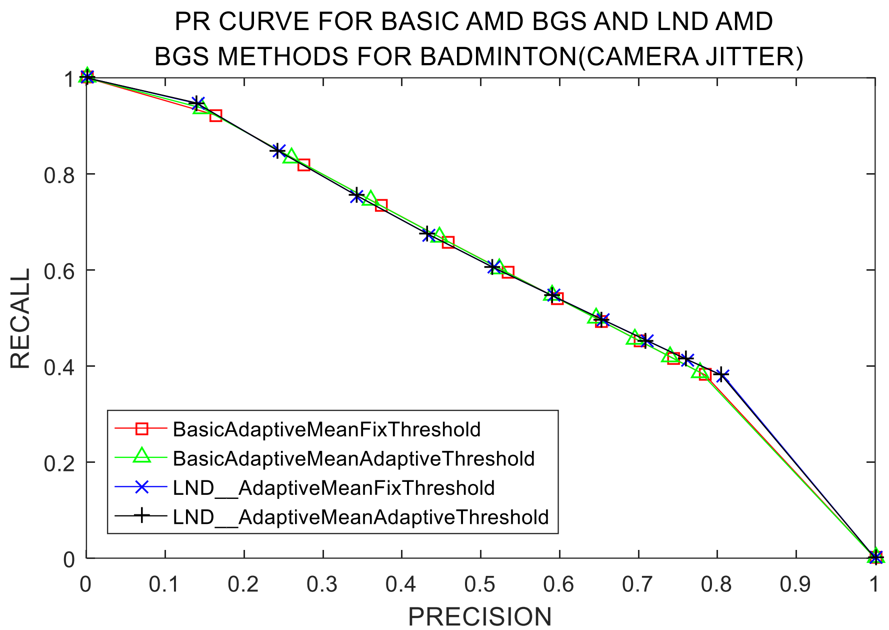


Fig. 6-11: Precision Recall for LND based Adaptive Median object detection methods for Badminton data set

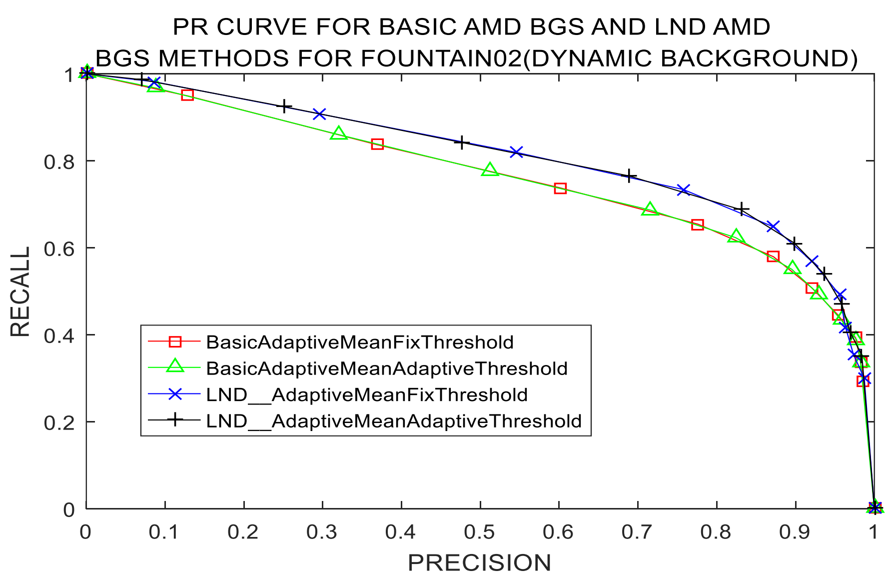


Fig. 6-12: Precision Recall Curve for LND based Adaptive Median object detection methods for Fountain02 Dataset data set

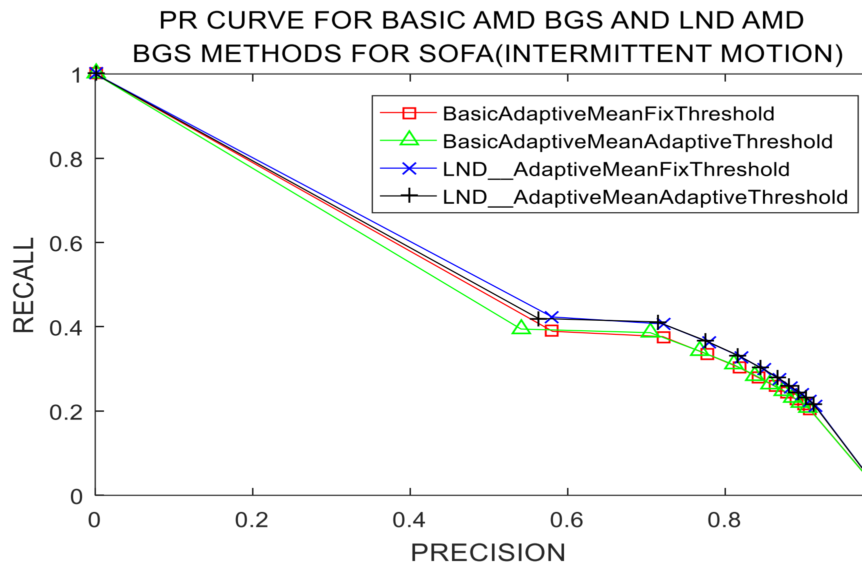


Fig. 6-13: Precision Recall Curve for LND based Adaptive Median object detection methods for Sofa data set

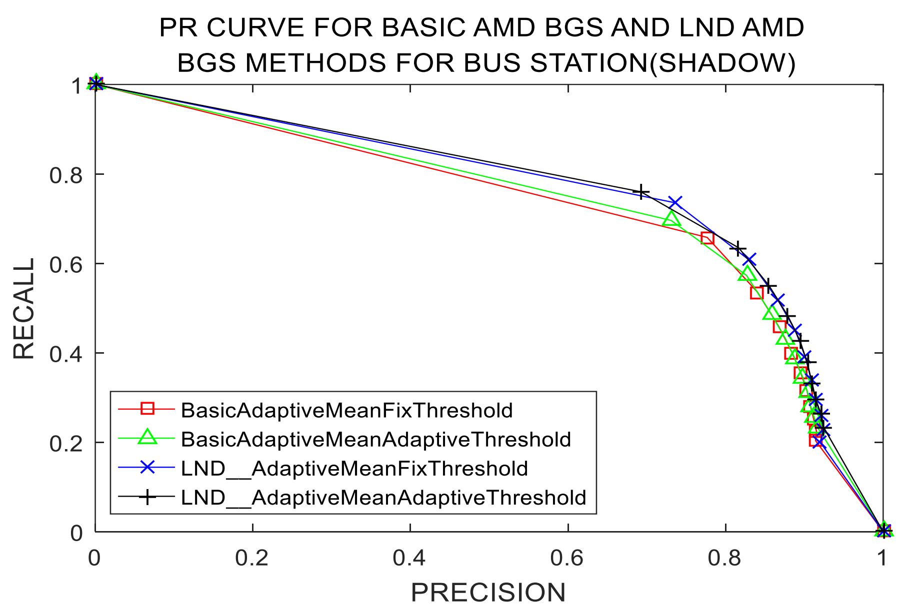


Fig. 6-14: Precision Recall Curve for LND based Adaptive Median object detection methods for Bus Station data set

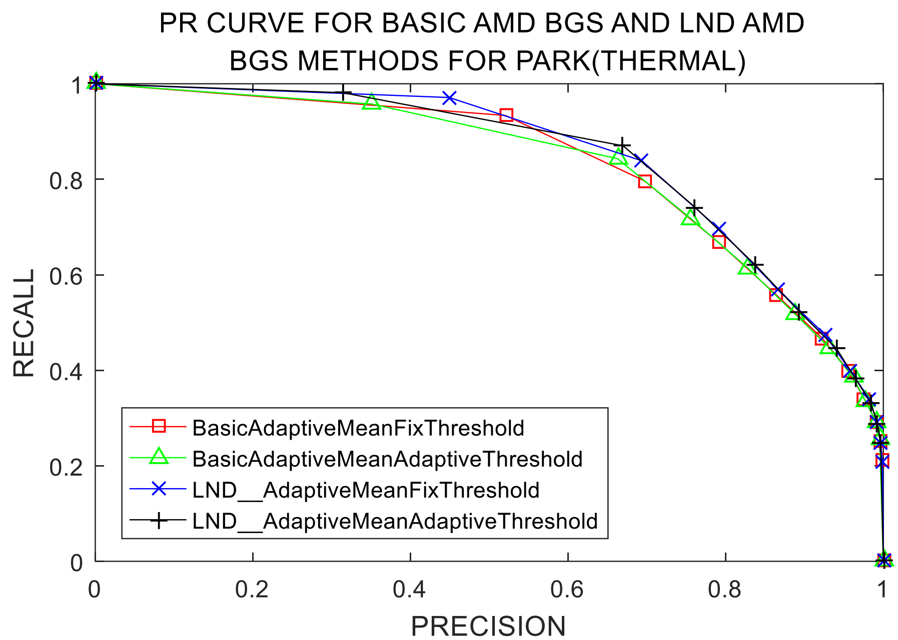


Fig. 6-15: Precision Recall Curve for LND based Adaptive Median object detection methods for Park data set

The proposed LND scheme is somewhat less efficient if time constraint is considered in the real time videos surveillance system. In these experiments, the LND difference has been calculated by processing frames on pixel level. To make this scheme efficient, we must devise a frame level processing of the LND scheme. Improvements in the range of 3% to 7% have been registered by the LND scheme as compared to basic methods in single modal background scenarios. It also investigated the effect of global adaptive threshold in foreground detection as compared to a fixed threshold. It also identified that the LND scheme needs to be further improved by devising a method which works at frame level processing to be employable in real-time video surveillance and analytics.

6.4 LND BASED OBJECT DETECTION USING CLAP

This section describes a methodology for implementing LND using CLAP. For proper understanding the proposed methodology, let us define some terms used in this section. Initially, we have a live streaming of video taken from surveillance cameras. Every camera takes pictures of the scene at a predefined frame rate. The current frame is denoted as CF. A digital image I is characterized by discrete space defined by the number of rows and columns. A space occupied by a particular row r and column c in I is called a pixel (r, c) and it contains a vector of three discrete values representing illumination intensity of basic colors; Red, Blue, and Green which is called pixel's intensity denoted by $I_{r,c}$. In a monochrome frame, a pixel contains only one value of gray intensity again represented by $I_{r,c}$. In object detection methods, we have used only monochrome intensity.

The proposed methodology modifies the difference computing procedure of basic BGS methods. In basic methods, only the absolute difference of corresponding pixels in the current frame and background is considered but no local neighborhood is considered. The pixel level processing scans every pixel to find the sum of absolute difference of corresponding local neighborhood of pixel (i, j) of the current frame and the background frame. This type of scanning is very time-consuming making the whole concept irrelevant to video surveillance.

The CLAP can help us in this because it does processing on the frame level. The frame I and frame B is shifted in all the eight directions by duplicating borders and then corresponding directional frames are subtracted as illustrated in **Algorithm 6-3**. Finally, the average difference is computed from this summation which is thresholded to detect objects in motion. This is the only difference between basic AM and CLAP modified AM which is deployed in **Algorithm 6-4.** The process flow of CLAP-LND method is shown in **Fig. 6-16**.

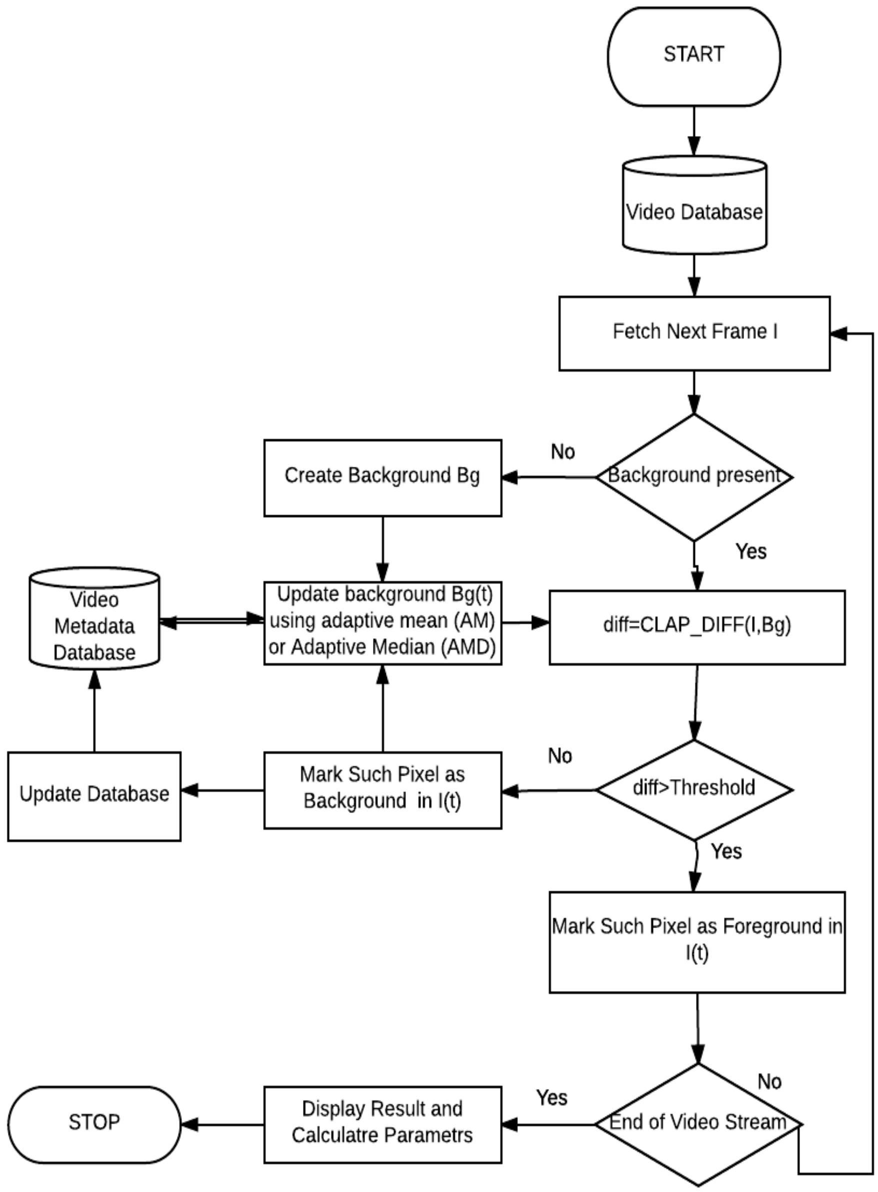


Fig. 6-16: Process Flow of CLAP based LND BGS method

CLAP__NEIGHBORHOOD_DIFF(I, B)

for the current frame image, I and background B

calculate I_N, I_NE, I_E, I_SE, I_S, I_SW, I_W, I_NW by shifting I in all eight directions.

calculate B_N, B_NE, B_E, B_SE, B_S, B_SW, B_W, B_NW by shifting B in all eight directions.

$$G_{avg} = \frac{1}{9}\sum_{d=N}^{NW} abs(I_d - B_d)$$

where d represents all eight directions including self in the 9-neighborhood.

Algorithm 6-3: The CLAP_NEIGHBORHOOD _DIFF Algorithm

Step 1: Create an average image BG with some initial frames.

Step 2: For each of remaining frame say CF

Step 2.1: Find Difference

$Diff = CLAP\_NEIGHBORHOOD\_DIFF\ (CF, BG)$

Step 2.3: Find foreground $FG(Diff > Th) = 255$

Step 2.4: Update BG with the following equations:

(For AM) $BG = \varepsilon\ \times CF + (1 - \varepsilon) \times BG$

where ε is frame refreshing rate.

OR

(For AMD) $BG(Diff > 0) = BG(Diff > 0) + 1;$

$BG(Diff < 0) = BG(Diff < 0) - 1;$

Algorithm 6-4: BGS algorithm with CLAP_NEIGHBORHOOD _DIFF

6.5 RESULTS OF CLAP-LND BASED METHODS

To prove the CLAP LND methodology, experiments are performed using MATLAB programming and image processing functions on an Intel i3 4GB system. “CDnet2012” is used as test data. Test data contains six scenarios involving several sequences having a different number of frames of different sizes and mimics the challenges thrown by a real video surveillance system such as camera jittering, shadow dynamic background, intermittent object motion and thermal imagery scenes. In each sequence, some of the video frames are identified as initial training frame used for initial background modeling. For our experimentations and result validation we have selected six video sequences; Highway (HW), Badminton (BM), Fountain02 (FT), Sofa (SF), bus station (BS) and park (PK), one each from every scenario. With every sequence, ground truths are also given to calculate Precision and Recall metrics for existing as well as the proposed methods.

Fig. 6-17: A snapshot of AM BGS output with basic and CLAP based Local Neighborhood Differencing Method

Table 6-3: Comparison of Basic AM and CLAP_LND_AM method

Scenario	Best Th	Basic_MEAN		CLAP_LND_MEAN_BGS		%Improvement in F1 Score
		F1-Score	Time (s)	F1-Score	Time (s)	
Highway	30	0.7857	0.011	0.8225	0.016	4.7%
Badminton	35	0.5719	0.050	0.5751	0.098	0.6%
Fountain02	25	0.7275	0.18	0.7712	0.032	6.0%
Sofa	15	0.4688	0.12	0.4825	0.019	2.9%
Bus Station	10	0.7176	0.14	0.7214	0.024	0.5%
Park	15	0.7052	0.17	0.7188	0.029	1.9%

Table 6-4: Comparison of Basic AMD and CLAP_LND_AMD method

Scenario	Best Th	Basic_MEDIAN_BGS		CLAP_LND_MEDIAN_BGS		%Improvement in F1 Score
		F1-Score	Time (s)	F1-Score	Time (s)	
Highway	20	0.8558	0.013	0.8879	0.019	3.8%
Badminton	35	0.5623	0.050	0.5646	0.089	0.4%
Fountain02	20	0.7100	0.020	0.7455	0.030	5.0%
Sofa	10	0.4948	0.012	0.5201	0.019	5.1%
Bus Station	5	0.7124	0.014	0.7357	0.022	3.3%
Park	10	0.7435	0.017	0.7593	0.025	2.1%

Total ten results in a set of {Pr, Re, F1, Time} are obtained for each method AM and AMD and data sequence against ten threshold values which are taken in the interval of 5 to 50 with a step size of 5. In addition to these threshold values, two extremes of 0 and 255 are also taken to draw Precision-Recall curves (PR curves). These curves represent the quality of motion detection. The higher area under PR curves indicates better method. These curves are shown in **Fig. 6-18** to **Fig. 6-23**. An integrated measure F1 which is harmonic mean of precision and recall is computed to show the combined effect. Higher values of F1 means better detection results. A snapshot of the visual output of the basic and CLAP based AM BGS method for the fountain02 sequence is displayed in **Fig. 6-17.**

Table 6-3 and **Table 6-4** presents the results for AM and AMD method respectively. In each table, basic and CLAP modified methods are compared with the help of F1 score metric and time taken. Percentage improvement for each scenario is also registered in both BGS methods. The CLAP based methods have registered noteworthy improvement in almost all scenarios. Only camera jitter scenario-based video sequence BM registers an insignificant improvement. This may be due to the limitation of basic BGS algorithms AM and AMD which are incapable of handling of multimodal background. PR curves are drawn for each scenario. Each chart draws PR curves for four algorithms namely AdaptiveMeanBasic, AdaptiveMeanCLAP, AdaptiveMedianBasic, AdaptiveMedian CLAP. The CLAP based algorithms always covered higher area under PR curves than corresponding basic methods. This validates our tabulated results.

Table 6-5: Comparison of execution time frame for basic, CLAP LND and Non-CLAP LND based AM method

Scenarios	**Parameters**	**Basic AM Method**	**CLAP based LND AM Method**	**Non-CLAP based LND AM Method**
Highway	Time Taken per frame(s)	0.0104	0.0154	0.0234
	Time in access of basic method	0%	49%	126%
Badminton	Time Taken per frame(s)	0.0501	0.0970	0.1203
	Time in access of basic method	0%	94%	140%
Fountain2	Time Taken per frame(s)	0.0183	0.0315	0.0424

	Time in access of basic method	0%	72%	131%
Sofa	Time Taken per frame(s)	0.0119	0.0191	0.0292
	Time in access of basic method	0%	60%	145%
Bus Station	Time Taken per frame(s)	0.0143	0.0236	0.0337
	Time in access of basic method	0%	64%	135%
Park	Time Taken per frame(s)	0.0167	0.0292	0.0401
	Time in access of basic method	0%	75%	140%

Table 6-6: Comparison of execution time frame for basic, CLAP LND and NON-CLAP LND based AMD method

Scenarios	**Parameters**	**Basic AMD Method**	**CLAP based LND AMD Method**	**Non_CLAP based LND AMD Method**
Highway	Time Taken per frame(s)	0.0130	0.0193	0.0259
	Time in access of basic method	0%	86%	150%
Badminton	Time Taken per frame(s)	0.0503	0.0892	0.1156
	Time in access of basic method	0%	78%	131%
Fountain2	Time Taken per frame(s)	0.0198	0.0295	0.0391
	Time in access of basic method	0%	61%	114%
Sofa	Time Taken per frame(s)	0.0124	0.0193	0.0251
	Time in access of basic method	0%	61%	110%
Bus Station	Time Taken per frame(s)	0.0137	0.0216	0.0289
	Time in access of basic method	0%	51%	102%
Park	Time Taken per frame(s)	0.0169	0.0252	0.0362
	Time in access of basic method	0%	51%	116%

The comparison is also made by observing the absolute time taken per frame to process using basic, CLAP and non-CLAP method for AM and AMD respectively in **Table 6-5** and

Table 6-6. In each case, CLAP method consumes high processing time (nearly 50%-70% in access of basic method) but this is still within the comfortable range (except BM sequence) of real-time video surveillance frame rate of 30frames/sec. The basic, as well as CLAP BGS method, takes very large processing time on BM sequence due to large frame size. On the other side, Simple LND methods consume nearly twice to thrice execution time per frame as compared to making these inapplicable for video surveillance systems.

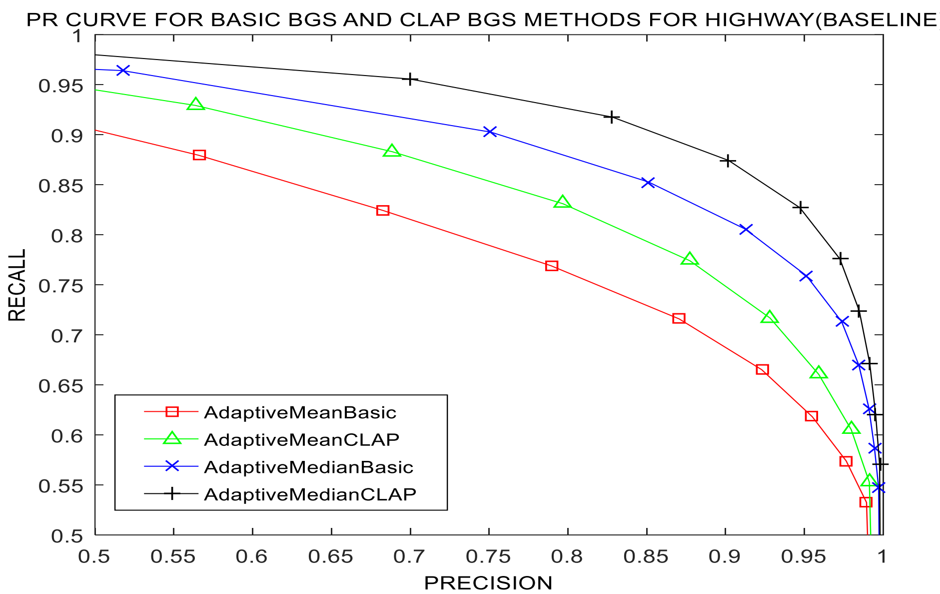


Fig. 6-18: Precision Recall Curve for basic BGS and CLAP-LND based object detection methods for Highway data set

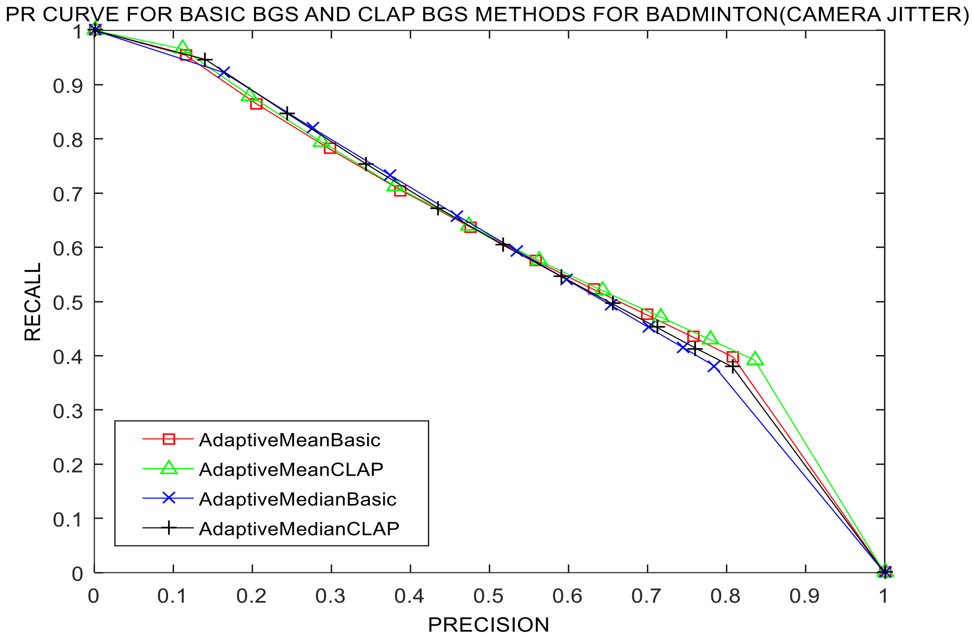


Fig. 6-19: Precision Recall Curve for basic BGS and CLAP-LND based object detection methods for Badminton data set

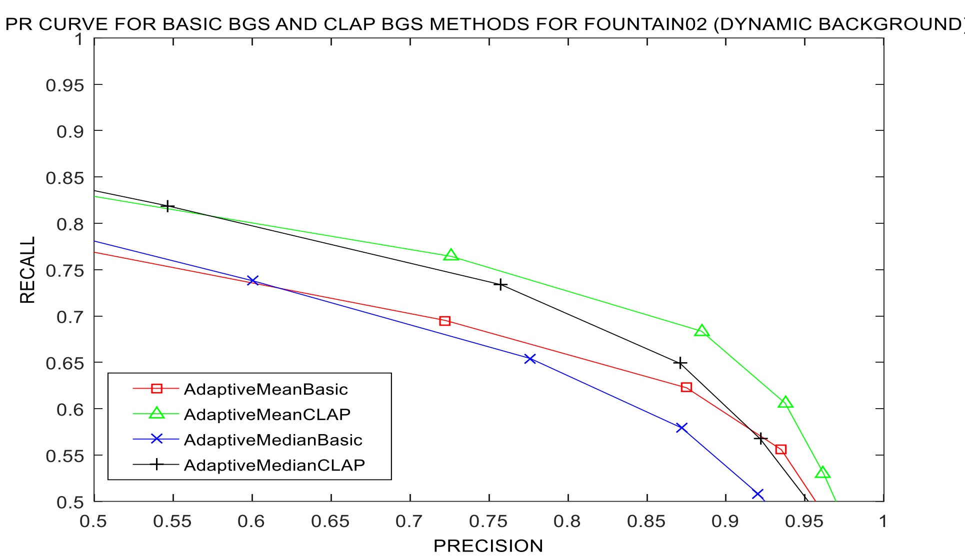


Fig. 6-20: Precision Recall Curve for basic BGS and CLAP-LND based object detection methods for Fountain02 Dataset data set

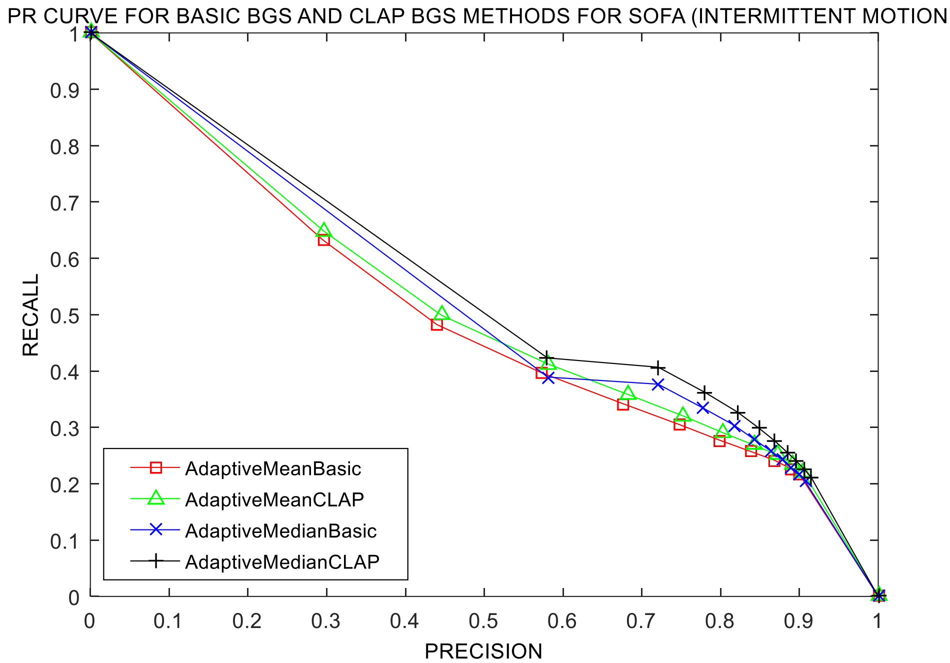


Fig. 6-21: Precision Recall Curve for basic BGS and CLAP-LND based object detection methods for Sofa data set

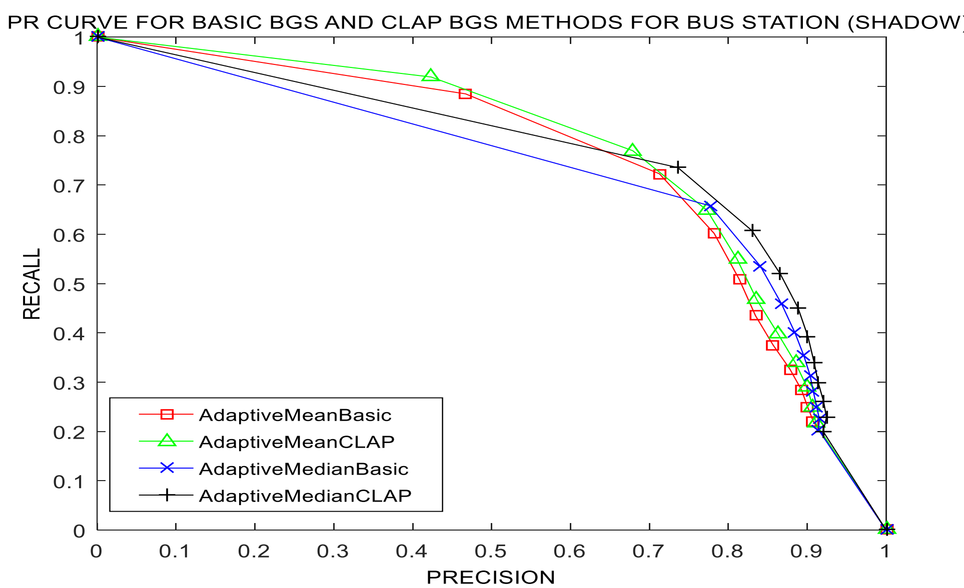


Fig. 6-22: Precision Recall Curve for basic BGS and CLAP-LND based object detection methods for Bus Station data set

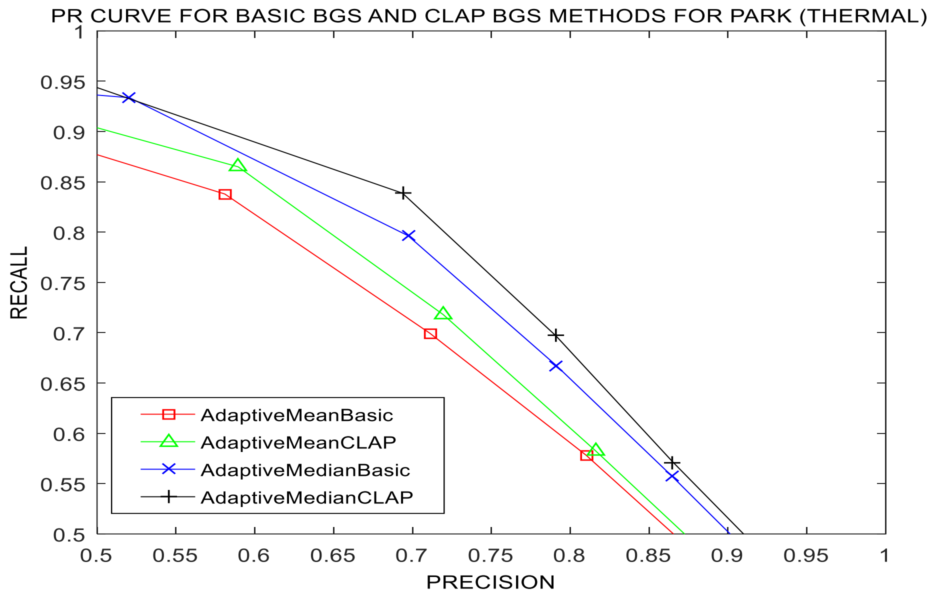


Fig. 6-23: Precision Recall Curve for basic BGS and CLAP-LND based object detection methods for Park data set

6.6 DISCUSSION ON CLAP-LND BASED METHODS

This chapter proposes a novel method of local neighborhood differencing which can be employed in BGS methods using cellular logic array processing for improved object detection. The proposed method is validated empirically by evaluating results on a standard benchmark test data. The CLAP based methods performed better than basic methods. Although the proposed method is a little time consuming, it produces better results as compared to the basic BGS method for all scenarios. The simple LND method which does not employ CLAP takes too much time in each scenario and can't be used for video surveillance systems where fast processing is required.

CHAPTER 7

RESULTS AND DISCUSSION

This chapter presents the integrated and comprehensive results obtained from experimentations in the previous chapters and also lists out the thesis contribution. The previous three chapters present experimental results of the individual concepts of object detection. The fourth chapter empirically evaluates different popular and existing methods of moving object detection used in video surveillance and helps in deducing that basic statistical methods such as AM and AMD are better in execution time and average in detection quality while more complex methods such as KDE is better in quality detection but too slow in execution time. This outcome motivates us to select these methods as a candidate for improvements. The chapter 5 and chapter 6 put forward the idea of edge-based detection and LND based detection using basic methods. Edge detection and LND detection were implemented and found to be too slow for pixel-based processing. This renders them unemployable in real-time video surveillance. These methods are further speeded up by the use of CLAP methodology and final results were compiled individually taking global adaptive, local adaptive and fixed threshold values. The present chapter sums up the previous best results from each chapter to compare these for final discussions and conclusions.

7.1 RESULTS

Table 7-1 presents the comparative results achieved by adaptive mean-based methods. We have compared four methods in the table; basic AM, CLAP_EDGE_AM_LT (CLAP based Edge Detection AM method with local threshold), CLAP_LND_AM_FT (CLAP based Edge Detection AM method with fixed threshold) and KDE method. The last three methods have performed best in their respective categories of edge-based detection, LND based detection and existing methods. The local threshold value in edge-based detection and the fixed threshold value in LND based detection produced best results. The KDE method's results are taken to find out the gaps remaining in the best performance. The test data and simulation environment is already described in chapter 3. The comparison between these methods are done through F1 score and execution time on six scenarios of test data abbreviated in the tables as HW (highway-baseline), BM (Badminton- Camera Jitter), FT2 (Fountain02-Dynamic Background), SOFA (Sofa- Intermittent Object Motion), BS (Bus Station- Shadow), PARK (Park- Thermal Image). **Table 7-2** represents the similar data for AMD based methods. The best threshold value for each scenario is also shown in the tables for analysis purpose. For each advanced method, a comparison column is also given to show the percentage improvement in F1 score with respect to the basic method in each table. Finally, the last row in each table shows the average metrics of all scenarios just to get a rough idea for the improvement achieved over basic method.

PR curves for AM based methods for each scenario are depicted in **Fig. 7-1** to **Fig. 7-6**. Each figure represents PR curves for four methods; basic AM, CLAP_EDGE_AM_LT, CLAP_LND_AM_FT, and KDE method. A P-R curve is a line graph drawn by taking the precision-recall pair for each threshold value. The more area a PR curve covers, the better it is. These curves also show the consistency of the method for object detection in different scenarios. PR curves for three AMD based methods; basic AMD, CLAP_EDGE_AMD_LT, CLAP_LND_AMD_FT and one KDE method are provided in **Fig. 7-7** up to **Fig. 7-12** for different scenarios.

Table 7-1: Comparative analysis of basic Adaptive Mean with CLAP_EDGE_AM, CLAP_LND_AM and KDE

Scenario	Basic_AM_BGS			CLAP_EDGE_AM_BGS_LT				CLAP_LND_AM_BGS_FT				KDE method			
	Best Th	F1 Score	Time(s)	Best Th	F1 Score	%improvement w.r.t basic method	Time(s)	Best Th	F1 Score	%improvement w.r.t basic method	Time(s)	Th prob	F1 Score	%improvement w.r.t basic method	Time(s)
HW	30	0.786	0.014	20	0.849	8%	0.028	30	0.822	5%	0.021	0.050	0.899	14%	0.039
BM	35	0.572	0.057	20	0.491	-14%	0.136	35	0.575	1%	0.098	0.030	0.720	26%	0.151
FT2	25	0.727	0.020	20	0.723	-1%	0.042	25	0.771	6%	0.030	0.030	0.753	4%	0.040
SOFA	15	0.469	0.013	5	0.538	15%	0.035	15	0.483	3%	0.021	0.250	0.660	41%	0.057
BS	10	0.718	0.018	10	0.738	3%	0.040	10	0.721	1%	0.027	0.150	0.801	12%	0.040
Park	15	0.705	0.017	10	0.732	4%	0.049	15	0.719	2%	0.026	0.200	0.730	4%	0.051
Average		0.663	0.023		0.679	2.4%	0.055		0.682	2.8%	0.037		0.760	16.6%	0.063

Table 7-2: Comparative analysis of basic Adaptive Median with CLAP_EDGE_AMD, CLAP_LND_AMD and KDE

Scenario	Basic_AMD_BGS			Best CLAP_EDGE_AMD_BGS_LT				CLAP_LND_AMD_BGS_FT				KDE method			
	Best Th	F1	Time(s)	Bes t Th	F1	%improvement w.r.t basic method	Time(s)	Best Th	F1	%improvement w.r.t basic method	Time(s)	Th Prob.	F1	%improvement w.r.t basic method	Time(s)
HW	20	0.856	0.013	15	0.874	2%	0.031	20	0.888	4%	0.019	0.050	0.899	5%	0.039
BM	35	0.562	0.055	25	0.545	-3%	0.131	35	0.565	0%	0.094	0.030	0.720	28%	0.151
FT2	20	0.710	0.019	20	0.721	2%	0.044	20	0.746	5%	0.030	0.030	0.753	6%	0.040
SOFA	10	0.495	0.014	10	0.498	1%	0.032	10	0.520	5%	0.021	0.250	0.660	33%	0.057
BS	5	0.712	0.019	5	0.720	1%	0.038	5	0.736	3%	0.023	0.150	0.801	12%	0.040
Park	10	0.744	0.022	10	0.800	8%	0.046	10	0.759	2%	0.035	0.200	0.730	-2%	0.051
Average		0.680	0.024		0.693	1.7%	0.054		0.702	3.3%	0.037		0.760	13.9%	0.063

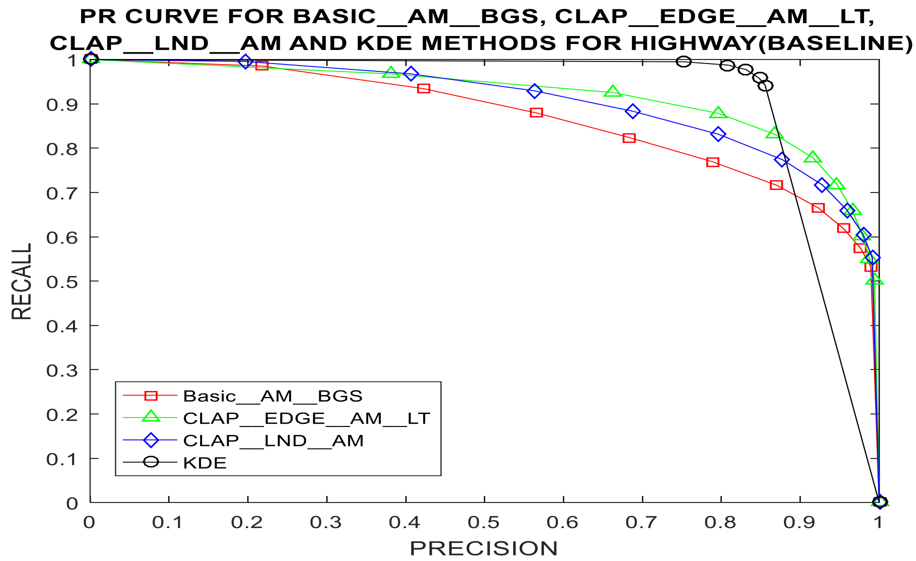


Fig. 7-1: PR curve for BASIC__AM__BGS, CLAP__EDGE__AM__LT, CLAP__LND__AM and KDE methods for highway dataset

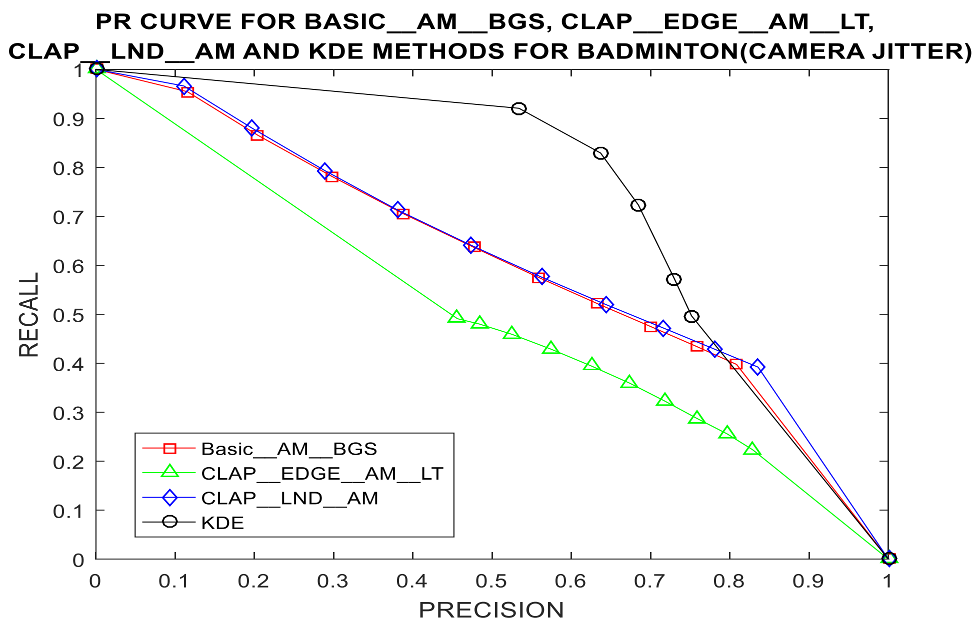


Fig. 7-2: PR curve for BASIC__AM__BGS, CLAP__EDGE__AM__LT, CLAP__LND__AM and KDE methods for badminton dataset

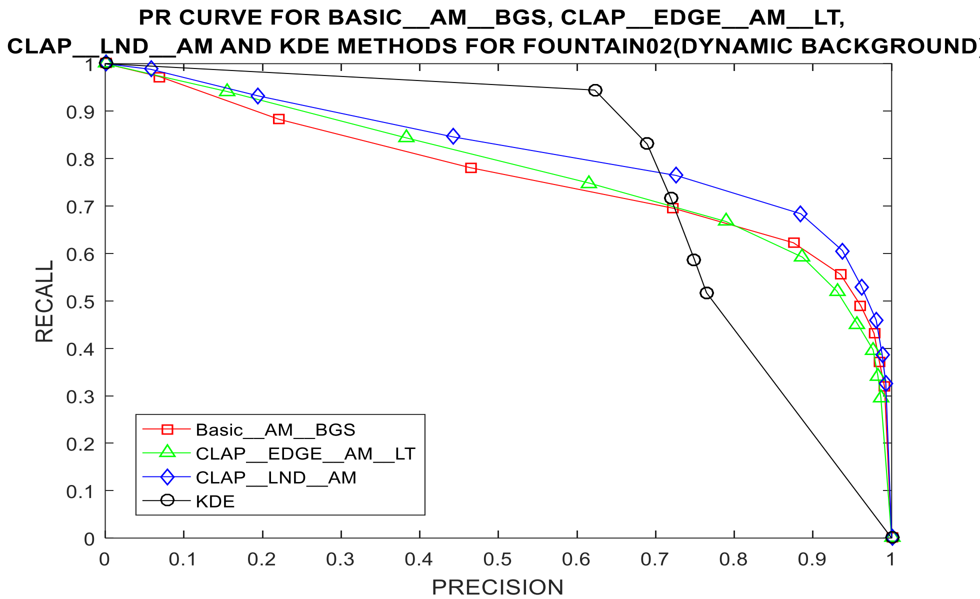


Fig. 7-3: PR curve for BASIC__AM__BGS, CLAP__EDGE__AM__LT, CLAP__LND__AM and KDE methods for fountain02 dataset

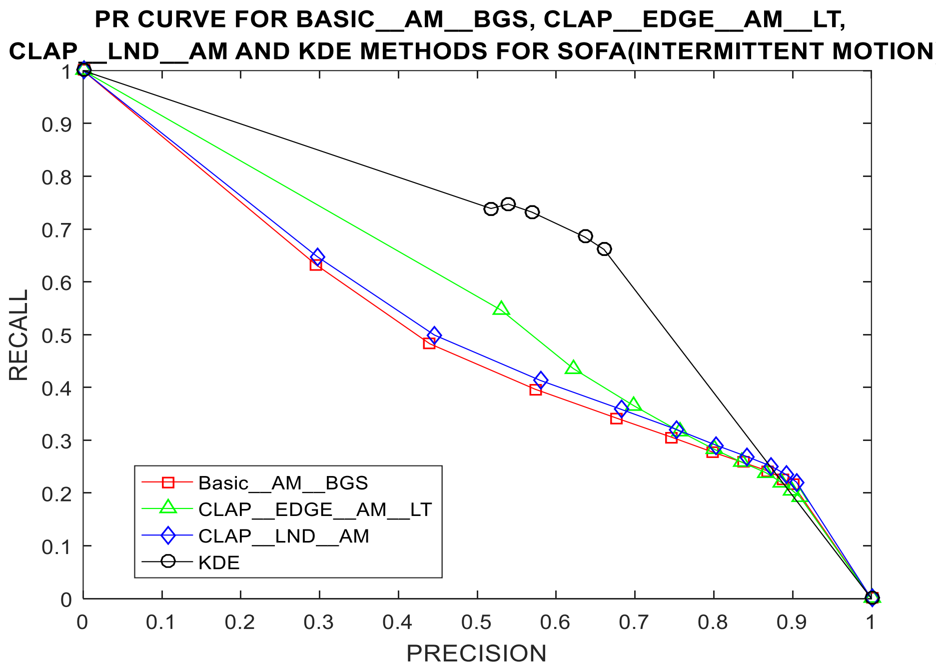


Fig. 7-4: PR curve for BASIC__AM__BGS, CLAP__EDGE__AM__LT, CLAP__LND__AM and KDE methods for sofa dataset

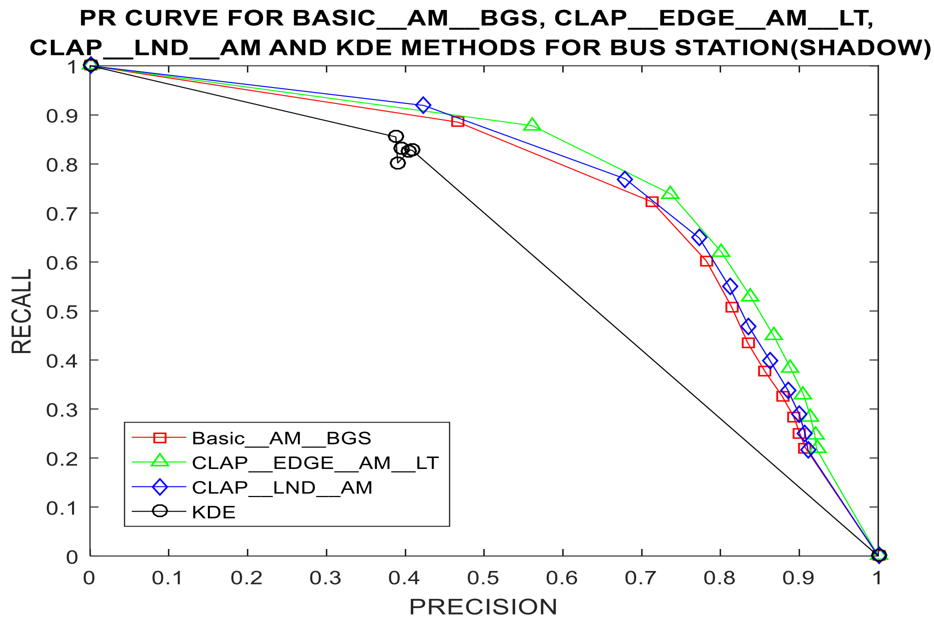


Fig. 7-5: PR curve for BASIC__AM__BGS, CLAP__EDGE__AM__LT, CLAP__LND__AM and KDE methods for bus station dataset

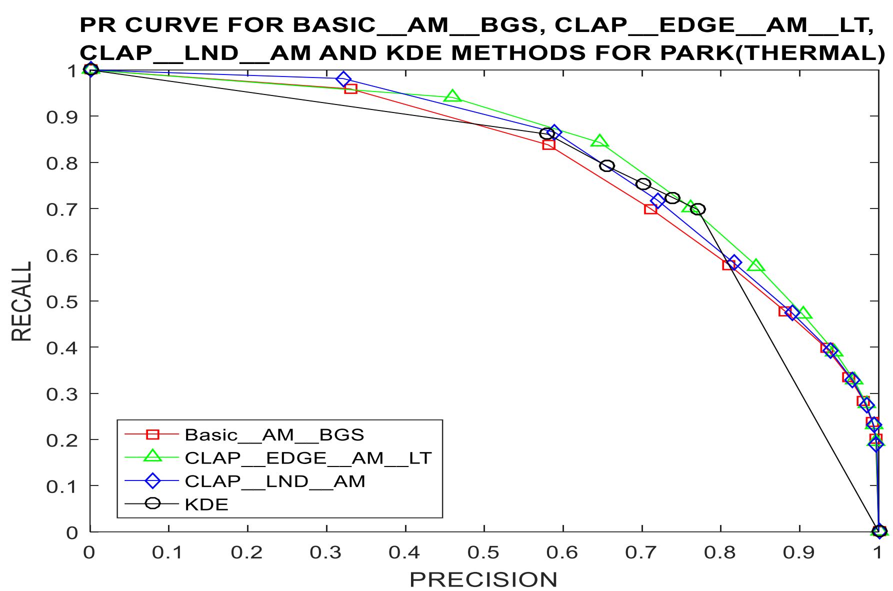


Fig. 7-6: PR curve for BASIC__AM__BGS, CLAP__EDGE__AM__LT, CLAP__LND__AM and KDE methods for park dataset

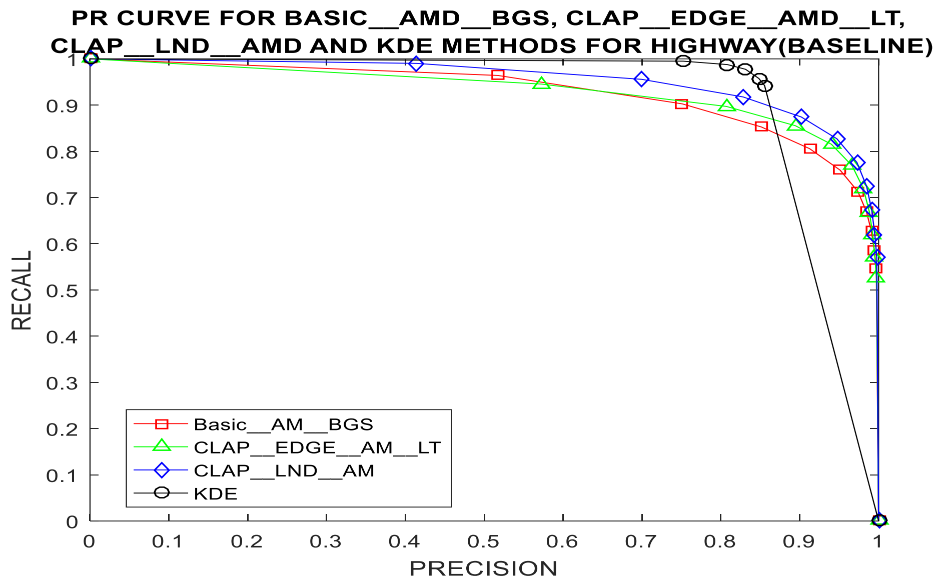


Fig. 7-7: PR curve for BASIC__AMD__BGS, CLAP__EDGE__AMD __LT, CLAP__LND__AMD and KDE methods for highway dataset

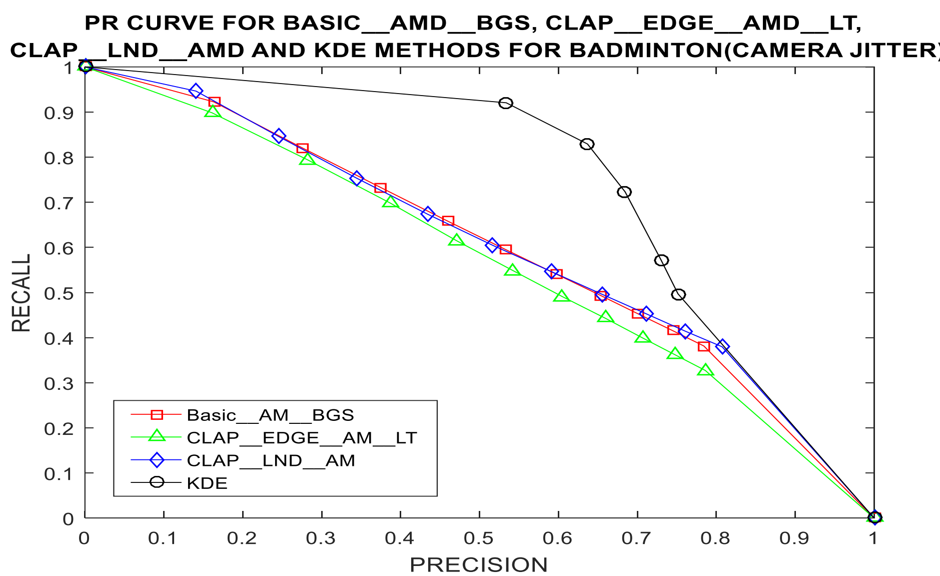


Fig. 7-8: PR curve for BASIC__AMD__BGS, CLAP__EDGE__AMD __LT, CLAP__LND__AMD and KDE methods for badminton dataset

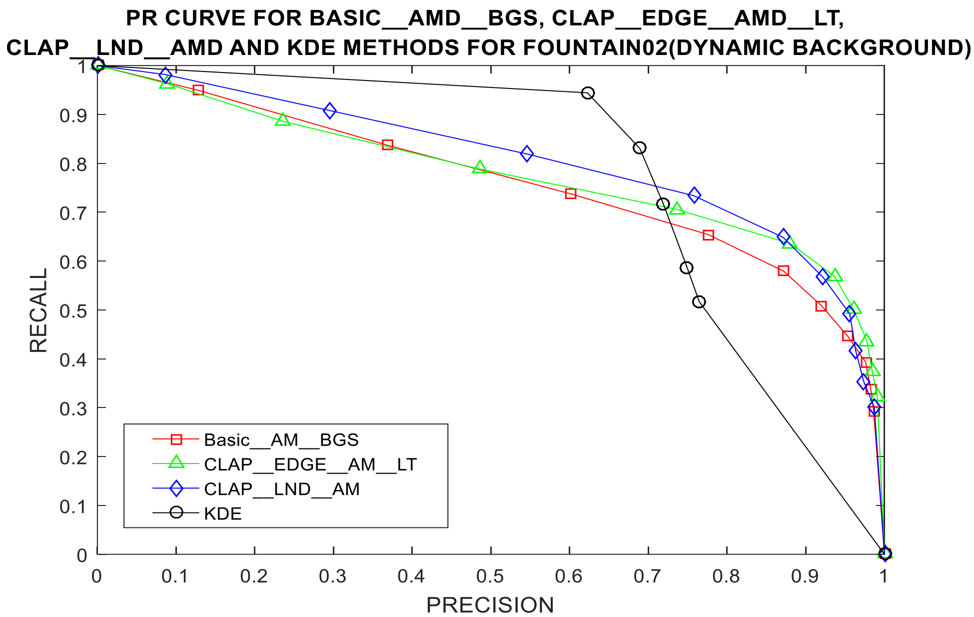


Fig. 7-9: PR curve for BASIC__AMD__BGS, CLAP__EDGE__AMD __LT, CLAP__LND__AMD and KDE methods for fountain02 dataset

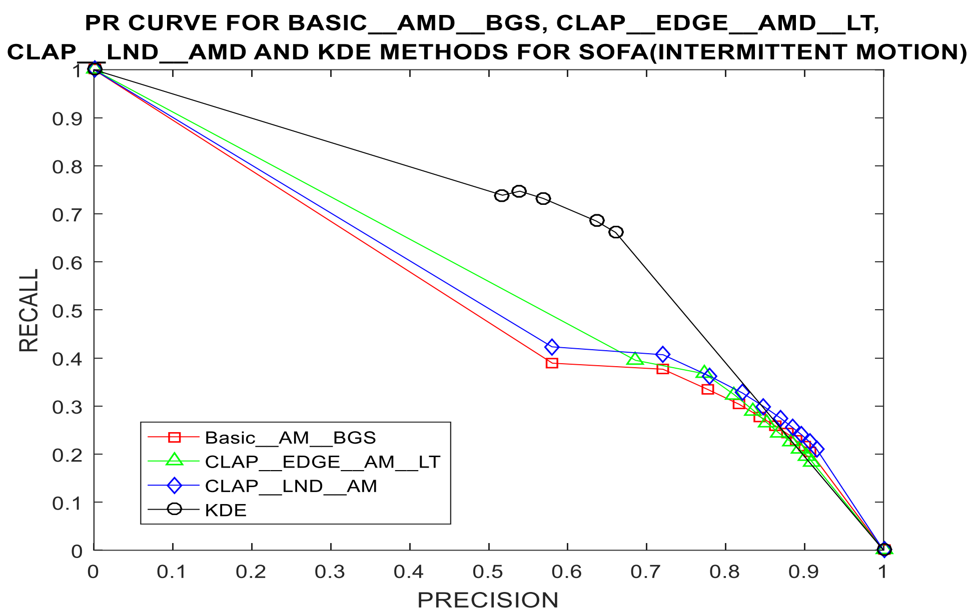


Fig. 7-10: PR curve for BASIC__AMD__BGS, CLAP__EDGE__AMD __LT, CLAP__LND__AMD and KDE methods for sofa dataset

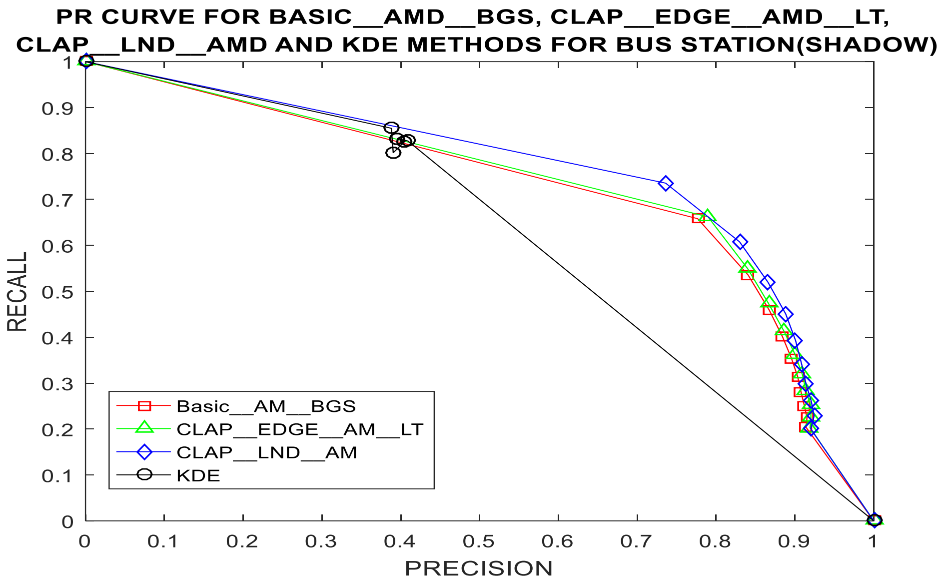


Fig. 7-11: PR curve for BASIC__AMD__BGS, CLAP__EDGE__AMD __LT, CLAP__LND__AMD and KDE methods for bus station dataset

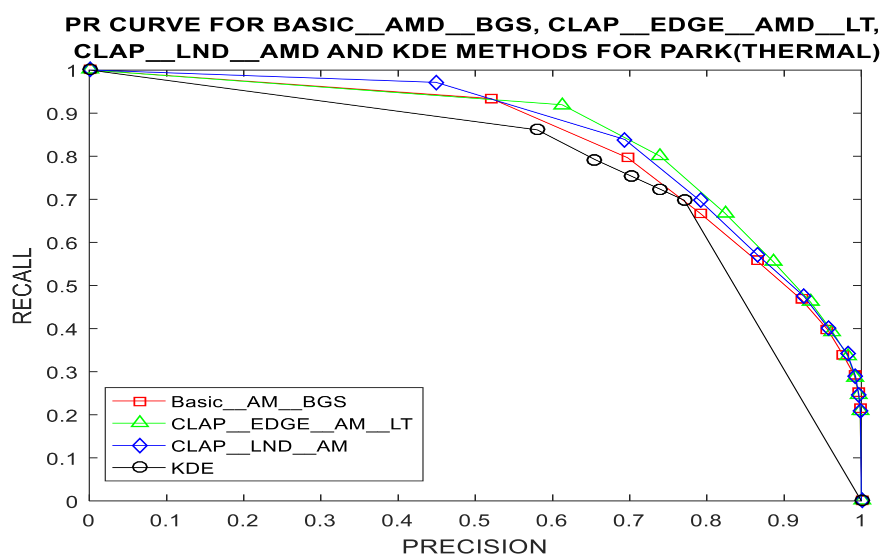


Fig. 7-12: PR curve for BASIC__AMD__BGS, CLAP__EDGE__AMD __LT, CLAP__LND__AMD and KDE methods for park dataset

7.2 DISCUSSION

In the analysis of **Table 7-1** and **Table 7-2**, It can be clearly deduced that in comparison to basic AM and AMD methods, the KDE method has achieved a very good improvement for the quality of object detection. Its performance is better than the basic method for every scenario especially in the complex and turbulent (noisy) scenarios such as BM (camera jitter) and SOFA (intermittent object motion). It's has performed nearly 15% better than basic method. If we compare CLAP_EDGE and CLAP_LND methods, the average improvement is in the range of 2 to 3 percent. AM based CLAP_EDGE method has registered very good result in SOFA dataset but AMD based method has failed to achieve any significant improvement in SOFA dataset which looks very inconsistent. the CLAP_EDGE method has totally failed in camera jitter scenario where it has registered a negative improvement as compared to basic method. This shows that the CLAP_EDGE method doesn't model multimodal background efficiently and is not consistent in its performance and seems to be context specific.

In case of CLAP_LND methods, improvement is within the range of 1% to 6% with an average figure of 2.8%. Its performance is consistent across all scenario and doesn't show any negative statistics. It has performed in all scenario except camera jitter which requires multimodal background modeling which is a limitation of AM and AMD based methods.

If we consider Execution time as one of the criteria of comparison and evaluation, the basic methods and CLAP_LND methods have performed better and are well within the range of 30frames/sec range. The other two techniques CLAP_EDGE and KDE methods are taking too much time to be suitable for the employment in video surveillance. The CLAP_LND method is taking almost 50% more time than the basic method while CLAP_EDGE and KDE methods are taking twice or thrice the time basic methods take to process one frame. Therefore, considering two factors of quality and time, we can conclude that CLAP_LND methods are better than CLAP_EDGE and KDE for the improvement in object detection in video surveillance.

We have also made an observation in our experiments related to the optimum threshold value which is very crucial for achieving better quality in object detection. We didn't observe any single fixed threshold value which provides the best F1 score. In chapter 5, we also devise a formula to model threshold value from the global average of the image and also from a local neighborhood of 16×16 pixels block but none has provided us an ideal solution. This also remains an open area for further research.

CHAPTER 8

CONCLUSION & FUTURE SCOPE

8.1 CONCLUSION

The new framework for data mining for object detection using cellular logic array has been designed by considering the specific requirement of video processing and the common flaws of the existing techniques. It has been observed that real-time surveillance requires object detection methods which provide optimum quality with smaller execution time, so that there is no time lag between video analytics. First, suitable algorithms which need improvement are identified through a literature survey and empirical evaluation. It was observed that simple statistical method such as AM, AMD are both efficient and effective along with KDE method but later is slow in execution time. Therefore, researchers are suggested to find ways and means to enhance simple and efficient methods AM and AMD to make them more robust for real-time video surveillance.

Subsequently, edge-based detection and local neighborhood difference based object detection methods are formulated and implemented. The proposed methods are validated by comparing with existing system quantitatively using benchmarked dataset by using popularly used metrics such as Recall and Precision. Integrated measures such as F1 score and PSNR are also used for result validations. Results are also compared with the average time taken per frame for determining the suitability of the proposed algorithms in real-time surveillance scenario.

Adaptive thresholding is introduced by taking threshold as fraction of global means and local neighborhood mean. The adaptive threshold modeling and edge maps are combined with CLAP to achieve better results in execution time. The research work empirically evaluates five edge based background subtraction algorithms viz. Canny, Sobel, CLAP-EDGE with fixed threshold, CLAP-EDGE with global threshold, CLAP-EDGE with local threshold for AM and AMD BGS algorithm. CLAP-EDGE methods are able to achieve improvement in object detection as compared to basic AM and AMD for all scenarios except camera jitter which requires multimodal background modeling, but in comparison to other traditional edge algorithms, CLAP edge-based detection methods performed far better. CLAP-EDGE algorithms acts as effective de-noising filters but failed to improve basic methods significantly. It is also observed that local thresholding need smaller 5-neighborhood blocks for better performance which need to be evaluated further. The CLAP_EDGE method is not found to be consistent in performance as it is also showing negative improvement in two scenarios. This method is also taking large execution time per frame even after using CLAP. Although CLAP has reduced time taken in generating edge map as compared to Sobel and Canny method, still it is not in manageable range for real-time video surveillance systems.

In most of the scenarios, there is only a minor improvement if basic fixed threshold BGS and adaptive threshold basic BGS methods are compared. There is a very minor improvement with a global adaptive threshold, but when local neighborhood difference LND method is used in basic BGS method, notable improvement ranging from 4% to 7% is achieved in both Mean and Median methods for baseline and dynamic background sequence. For other sequences, the improvement is between 1% to 3%. Camera Jitter is the only scenario which doesn't register any improvement due to the definite multimodal background which is difficult to address by single modal methods such as AM and AMD.

CLAP has also helped us to significantly reduce the time of execution per frame as compared to the non-CLAP edge-based and LND based methods because CLAP works on frame level instead of pixel level. CLAP is successfully implemented to manage execution time within reasonable range of real-time scenario.

So, we have attained the predefined objective of proposing and implementing a CLAP based novel data mining framework of object detection which improves the existing detection algorithms in video surveillance in all scenarios except camera jitter. It requires a necessary multi-modal background modeling method to reduce noise which can be addressed in future research.

8.2 FUTURE SCOPE

It has been observed that multimodal background is not efficiently modeled. There are several methods such as KDE which easily model multimodal background, but these take too much processing time rendering them ineligible for real-time scenarios. The CLAP which works on frame level can be explored in future to reduce execution time in these methods.

There is also a need to improve BGS algorithm by finding different ways of modeling and registering background to handle multimodal and noisier environment. There is need to identify the possibility of using CLAP for this purpose.

REFERENCES

Athanesious, J. J., & Suresh, P. (2012). Systematic Survey on Object Tracking Methods in Video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 242-247.

Badal, T., Nain, N., Ahmed, M., & Sharma, V. (2015). An improved multi-layer codebook model to eliminate ghost region. *Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, (pp. 1-4). Patna. doi:10.1109NCVPRIPG.2015.7490036

Baf, F. E., Bouwmans, T., & Vachon, B. (2008). Fuzzy integral for moving object detection. *International Conference on Fuzzy Systems, FUZZ-IEEE*, (pp. 1729-1736).

Bahadır, O., Selim, A., Sandra, E., Martino, P., & Daniele, E. (2010). Performance Measures for Object Detection Evaluation. *Pattern Recognition Letters, 31*(10), 1128–1137.

Basu, M. (2002). Gaussian Based Edge Detection Methods: A Survey. *IEEE Transactions on System, man, and cybernetics part c Application and Reviews, 32*(3).

Beevi, Y. C., & Natarajan, S. (2009). An efficient Video Segmentation Algorithm with Real-time Adaptive Threshold Technique. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 2*(4).

Benezeth, Y., Jodoin, P. M., Emile, B., Laurent, H., & Rosenberger, C. (2008). Review and evaluation of commonly-implemented background subtraction algorithms. *International Conference on Pattern Recognition (ICPR 2008)*, (pp. 1-4).

Blair, C. D., & Marion, M. E. (1985). An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System. *Communications of the ACM*, *28*(3), 289-99.

Borges, P., Conci, N., & Cavallaro, A. (2013). Video-based human behavior understanding: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, *11*, 1993-2008.

Boufares, O., Aloui, N., & Cherif, A. (2016). Adaptive Threshold for Background Subtraction in Moving Object Detection using Stationary Wavelet Transforms 2. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *7*(8), 29-36.

Bouwmans, T. (2011). Recent Advanced Statistical Background Modeling for Foreground Detection: A Systematic Survey. *Recent Patents on Computer Science*, *4*(3), 147-176.

Bouwmans, T. (2012). Background Subtraction For Visual Surveillance: A Fuzzy Approach. *Handbook on Soft Computing for Video Surveillance, CRC Press*, 103-134.

Bouwmans, T. (2014). Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview. *Computer Science Review*.

Brutzer, S., Hoferlin, B., & Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. *Computer Vision and Pattern Recognition*, 1937-1944.

Cancela, B., Ortega, M., & Penedo, M. (2014). Multiple human tracking systems for unpredictable trajectories. *Machine Vision and Applications*, *25*(2), 511-527.

Canfield, A. (2014). *Report: 68 percent want to use security cameras for operational improvement*. Security System News.

Canny, J. (1986). A Compuational Approach to Edge Detection. *IEEE transaction on Pattern Analysis and Machine Intelligence, 8*(6), 679-698.

Case, I. (2010). *Automatic object detection and tracking in the video.* Rochester Institute of Technology. Retrieved from https://cholarworks.rit.educgiviewcontent.cgiarticle=1247&context=theses

Chang, R. I., Wang, T. C., Wang, C. H., Liu, J. C., & Ho, J. M. (2012). Effective distributed service architecture for ubiquitous video surveillance. *Inf. Syst. Frontiers, 14*(3), 499–515.

Chang, S., & Aishy, A. (2006). Real-Time Adaptive Thresholding for video change detection. *IEEE International Conference on Image Processing (ICIP)*, (pp. 157-160).

Changick, K., & Jenq-Neng, H. (2002). Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology, 12*(2), 122-129.

Chauhan, A. K., & Krishan, P. (2013). Moving Object Tracking Using Gaussian Mixture Model and Optical Flow. *International Journal of Advanced Research in Computer Science and Software Engineering*.

Chen, S., Zhang, J., & Li, Y. (2012). A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction. *IEEE Trans. Ind. Inf., 8*(1), 118–127.

Cheng, F., Huang, S., & Ruan, S. (2010). Advanced Motion Detection for Intelligent Video Surveillance Systems. *ACM Symposium on Applied Computing*. Lausanne, Switzerland.

Ching, S., Cheung, S., & Chandrika, K. (2004). Robust techniques for background subtraction in urban traffic video. *Proc. SPIE 5308, Visual Communications and Image Processing*, (p. 881). doi:10.111712.526886

Cielniak, G., Treptow, A., & Duckett, T. (2005). Quantitative Performance Evaluation of a People Tracking System on a Mobile Robot. *European Conference on Mobile Robots (ECMR).* Ancona, Italy.

CodeBlocks. (2017, March 12). Retrieved from www.codeblocks.org: www.codeblocks.org

Colombari, A., & Fusiello, A. (2010). Patch-based background initialization in the heavily cluttered video. *IEEE Transactions on Image Processing, 19*(4), 926-933.

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 25*(5), 564–577.

Cropley, J. (2015). *Video Surveillance, Top ten trends in 2015*. IHS Technology.

Cucchiara, R., & Gualdi, G. (2010). Mobile video surveillance systems: An architectural overview. *Mobile Multimedia Process. Fundamental of Methods, Applications, 5960*, 89–109.

Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(10), 1337 - 1342.

Cui, Y. Y., Zeng, Z. Y., Cui, W. H., Fu, B. T., & Liu, W. (2011). Moving Object Detection Based on Edge Pair Difference. *Advanced Materials Research*, 1407-1410. doi:10.4028

Dammalapati, N., Gera, J., & Gera, I. P. (2013). An Intelligent Suspicious Activity Detection Framework (ISADF) for Video Surveillance Systems. *International Journal of Computer Applications (0975 – 8887), 84*(14).

Dhar, P., Khan, M., Hasan, D., & Kim, J. (2011). Enhanced Edge Localization and Gradient Directional Masking for Moving Object Detection. *Communications in Computer and Information Science, 260*. doi:10.1007978-3-642-27183-0_25

Dong, X., Wang, k., & Jia, G. (2009). Moving Object and Shadow Detection Based on RGB Color Space and Edge Ratio. *International Congress on Image and Signal Processing*, (pp. 1-5). Tianjin. doi:10.1109CISP.2009.5301770

Duff, M. J., Watson, D. M., Fountain, T. J., & Shaw, G. K. (1973). A cellular logic array for image processing. *Pat. Recog.*, *5*, 229–234.

Elgammal, A., Duraiswami, R., Harwood, D., & Davis, L. (2002). Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance. *IEEE Proceedings of Computer Vision, 90(7).* doi:0.1109/JPROC.2002.801448

Elliott, D. (2010). Intelligent video solution A definition. *Security*, 46–48.

El-Sayed, M. A. (2011). A New Algorithm Based Entropic Threshold for Edge Detection in Images. *IJCSI International Journal of Computer Science Issues*, *8*(1).

Ezhoosh, H. R. (2010). *Fast Fuzzy Edge Detection.* Waterloo, Ontario, N2L 3G1, Canada: Pattern Recognition and Machine Intelligence Lab, Systems Design Engineering, University of Waterloo .

Fieguth, P., & Terzopoulos, D. (1997). Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , (pp. 21-27).

Firdousi, R., & Parveen, S. (2014). Local Thresholding Techniques in Image Binarization, International. *Journal of Engineering and Computer Science, 3*(3), 4062- 4065.

Frejlichowski, D., Forczmański, P., Nowosielski, A., Gościewska, K., & Hofman, R. (2012). SmartMonitor: An Approach to Simple, Intelligent and Affordable Visual Surveillance System. In L. e. In Bolc, *ICCVG 2012, LNCS, vol. 7594* (pp. 726–734). Springer, Heidelberg.

Frejlichowski, D., Gościewska, K., Forczmański, P., Nowosielski, A., & Hofman, R. (2013). Extraction of the Foreground Regions by Means of the Adaptive Background Modeling Based on Various Color Components for a Visual Surveillance System. In R. e. In Burduk, *Advances in Intelligent Systems and Computing* (pp. 351–360). Springer International Publishing.

Gao, Q., Parslow, A., & Tan, M. (2001). Object motion detection based on perceptual edge tracking. *Second International Workshop on Digital and Computational Video*, (pp. 78-85). Tampa, FL. doi:10.1109DCV.2001.929945

Gemert, J. C., Veenman, C. J., Smeulders, A. W., & Geusebroek, J. M. (2010). Visual Word Ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(7), 1271-1283.

Goyette, N., Jodoin, P. M., Porikli, F., Konrad, J., & Ishwar, P. (2012). changedetection.net: A new change detection benchmark dataset. *IEEE Workshop on Change Detection (CDW-2012)*, (pp. 16-21).

Green, M. W., Travis, J., & Downs, R. (2014). *The Appropriate and Effective Use of Security Technologies in U.S. Schools.* U.S. Department of Justice, 810 Seventh Street N.W. Washington, DC 20531.

Hakeem, A., Gupta, H., Kanaujia, A., Choe, T. E., Gunda, K., Scanlon, A., . . . Haering, N. (2012). Video Analytics for Business Intelligence. In *Studies in Computational Intelligence* (pp. 309-354). Verlag Springer Berlin Heidelberg, Seite.

Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., . . . Pankanti, S. (2005). Smart Video Surveillance Exploring the Concept of Multiscale Spatiotemporal Tracking. *IEEE Signal Processing Magazine, 22*(2), 38–51.

Hu, Q., Qin, L., & Huang, Q. M. (2013). A survey on visual human action recognition. *Chinese Journal of Computers*, *36*, 2512-2524.

Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man and Cybernetics, Part C Applications and Reviews, 34*(3), 334–352.

Hua, Y. L., & Ruichun, L. (2014). Adaptive Thresholding Approach of Moving Object Detection. *Sensors & Transducers, 178*(9), 238-243.

Jabid, T., Mohammad, T., Ahsan, T., Abdullah-Al-Wadud, M., & Chae, O. (2011). An edge-texture based moving object detection for video content based application. *14th International Conference on Computer and Information Technology (ICCIT 2011)*, (pp. 112-116). Dhaka. doi:10.1109ICCITechn.2011.6164884

Jiang, S., & Zhao, Y. (2012). Background extraction algorithm based on Partition Weighed Histogram. *IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2012*, (pp. 433-437).

Johnson, J. (1958). Analysis of image forming systems. *Image Intensifier Symposium* (pp. 244-273). Ft. Belvoir, Va.: Warfare Electrical Engineering Department, U.S. Army Research and Development Laboratories.

Junejo, I., Bhutta, A., & Foroosh, H. (2011). Single Class Support Vector Machine (SVM) for Scene Modeling. *Journal of Signal, Image, and Video Processing, Springer-Verlag*.

Kafai, M., & Bhanu, B. (2012). Dynamic Bayesian networks for vehicle classification in video. *IEEE Trans. Ind. Inf.*, *8*(1), 100-109.

Kalirajan, K., & Sudha, M. (2015). Moving Object Detection for Video Surveillance. *Hindawi Publishing Corporation, Scientific World Journal*, 1-10. doi:10.11552015907469

Khan, Z., & Gu, I. (2010). Joint feature correspondences and appearance similarity for robust visual object tracking. *IEEE Trans. Inf. Forensics Security*, *5*(3), 591-606.

Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2004). Background Modeling and Subtraction by Codebook Construction. *IEEE International Conference on Image Processing, ICIP 2004.*

Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-Time Imag.*, *11*, 172-185.

Kumar, N. A., & Sureshkumar, C. (2013). Background subtraction based on threshold detection using modified K-means algorithm. *International Conference on Pattern Recognition, Informatics and Mobile Engineering,*, (pp. 378-382). Salem.

Kumar, T., & Sahoo, G. (2010). A Novel Method of Edge Detection using Cellular Automata. *International Journal of Computer Applications, 9*(4), 38-44.

Kuo, W., Chang, S., & Wang, C. L. (2009). An Efficient Histogram-Based Method for Background Modeling. *International Conference on Innovative Computing, Information and Control, ICICIC 2009*, (pp. 480-483). Kaohsiung, China.

Kwak, S., Bae, G., & Byun, H. (2010). Abandoned luggage detection using a finite state automaton in surveillance video. *Opt. Eng., 49*(2), 2007-10.

Lai, Y. K., & Rosin, P. L. (2014). Efficient Circular Thresholding. *IEEE Trans. on Image Processing, 23*(3), 992–1001. doi:10.1109TIP.2013.2297014

Lakis, C., Takis, K., & Oge, M. (2011). Advanced statistical and adaptive threshold techniques for moving object detection and segmentation. *IEEE DSP 2011*. doi:10.1109ICDSP.2011.6004875

Lakshmi, S., & Ranarayanan, V. S. (2010). A study of Edge Detection Techniques for Segmentation Computing Approaches. *IJCA Special Issue on "Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications" CASCT, 2010*.

Lee, B., & Hedley, M. (2002). Background Estimation for Video Surveillance. *Image and Vision Computing, IVCNZ 2002*, (pp. 315-320). NZ.

Lefter, I., Rothkrantz, L., Bouchner, P., Burghouts, G., & Wiggers, P. (2010). A Multimodal Car Driver Surveillance System in a Military Area. *Driver Car Interaction & Interface*.

Li, C. T., Yang, C. T., Yu-Wen, S., & Tzu-Kuei, S. (2010). An efficient and robust moving shadow removal algorithm and its applications in ITS. *EURASIP J. Adv. Signal Process.*

Li, M., & Qi, A. (2010). Skeleton extraction of the color cerebrovascular image based on topological nodes. *3rd International Congress on Image and Signal Processing*, (pp. 1133-1137). Yantai.

Li, Q., He, D., & Wang, B. (2008). Effective Moving Objects Detection Based on Clustering Background Model for Video Surveillance. *Congress on Image and Signal Processing, CISP 2008, 3*, pp. 656-660.

Li, X., Ng, M., & Yuan, X. (2015). Median filtering-based methods for static background extraction from surveillance video. *Numerical Linear Algebra with Applications, Wiley Online Library*.

Li, Y. H., Xiong, C. Z., Yin, Y. X., & Liu, Y. L. (2009). Moving Object Detection Based on Edged Mixture Gaussian Models. *International Workshop on Intelligent Systems and Applications*, (pp. 1-5). Wuhan. doi:10.1109IWISA.2009.5072961

Lin, D., Cao, D., & Zeng, H. (2014). Improving Motion State Change Object Detection by Using Block Background Context. *UCKI 2014.*

Lin, H. H., Liu, T. L., & Chuang, J. H. (2002). A probabilistic SVM approach for background scene initialization. *IEEE Int. Conf. Image Processing*, (pp. 893-896).

Liu, H., Chen, S., & Kubota, N. (2013). Intelligent Video Systems and Analytics: A Survey. *IEEE Transactions on industrial informatics, 9*(3), 1222-33.

Liu, Z., Lu, Y., & Zhang, Z. (2007). Real-time spatiotemporal segmentation of video objects in the H.264 compressed domain. *Journal of Visual Communication and Image Representation, 18*(3), 275–290.

Lu, X., & Manduchi, R. (2011). Fast image motion segmentation for surveillance applications. *Image Vis. Comput., 29*(2-3), 104-116.

Maddalena, L., & Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing, 17*(7), 1168–1177.

Marr, D., & Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences, 207*(1167), 187-217. doi:10.1098rspb.1980.0020

Martin, D., Fowlkes, C., D., T., & Malik, J. (2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *8th Int'l Conf. Computer Vision, 2*, pp. 416-423.

Mathur, N., Mathur, S., & Mathur, D. (2016). A Novel Approach To Improve Sobel Edge Detector. *6th International Conference On Advances In Computing & Communications, ICACC 2016*, (pp. 6-8). Cochin, India.

McFarlane, N., & Schofield, C. (1995). Segmentation and tracking of piglets in images. *Machine Vision and Applications, 8*(3), 187–193.

McManus, L., Renno, N. J.-P., Makris, D., & Jones, G. A. (2008). An object-based comparative methodology for motion detection based on the F-Measure. *Computer Vision and Image Understanding, 111*, 74-85.

Mehmet, S., & Bulent, S. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging, 13*(1), 146-165. doi:10.11171.1631315

Mehrotra, S., Das, S., Shikha, S., & Kar, C. (2015). Comparative Analysis of Traditional Methods with Game of Life in Cellular Automata for Edge Detection. *International Journal of Computer Applications, 118*(17), 18-24.

Mei, T., Guo, J. L., Hua, X. S., & Liu, F. L. (2010). AdOn Toward contextual overlay in-video advertising. *Multimedia Syst., 16*(4-5), 335–344.

Miezianko, R., & Pokrajac, D. (2008). Detecting and Recognizing Abandoned Objects in Crowded Environments. *Proc. of Computer Vision Systems*, (pp. 241-250).

Mingwu, R., & Han, S. (2005). A Practical Method for Moving Target Detection under Complex Background. *Computer Engineering*, 33-34.

Morioka, K., Kovacs, S., Joo-Ho, L., & Korondi, P. (2010). A cooperative object tracking system with fuzzy-based adaptive camera selection. *Int. J. Smart Sens. Intell. Syst.*, 338-358.

Mukherjee, A., & Kundu, D. (2013). Motion analysis in video surveillance using edge detection techniques. *IOSR Journal of Computer Engineering (IOSR-JCE), 12*(6), 10-15.

Murshed, M., Ramirez, A., & Chae, O. (2010). Statistical Background Modeling An Edge Segment Based Moving Object Detection Approach. *7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, (pp. 300-306). doi:10.1109AVSS.2010.18

NagaRaju, C. (2011). Morphological Edge Detection Algorithm Based on Multi-Structure Elements of Different Directions. *IJICT, 1*(1).

Nain, N., Jindal, G., A., G., & Jain, A. (2008). Dynamic Thresholding Based Edge Detection. *Proceedings of the World Congress on Engineering Vol I WCE 2008.* London, U.K.

Ng, G. S., & Chan, K. (2010). *Edge Detection using supervised Learning and voting scheme.* Singapore: Nanyang Technological University.

Oh, J., & Bandi, B. (2002). Multimedia data mining framework for raw video sequences. *In Proceedings of the third international workshop on multimedia data mining (MDMKDD'2002)*, (pp. 1-10).

Papari, G., & Petkov, N. (2011). Edge and line oriented contour detection State of the art. *Image and Vision Computing, 29*, 79-103.

Paragios, N., Rousson, M., & Ramesh, V. (2002). Marching distance functions A shape-to-area variational approach for global-to-local registration. *7th European Conference on Computer Vision*, (pp. 775-789). Copenhagen, Demark.

Parekh, H. S., Thakore, D. G., & Jaliya, U. K. (2014). A Survey on Object Detection and Tracking Methods. *International Journal of Innovative Research in Computer and Communication Engineering, 2*(2).

Pentagram. (2015, October 12). Retrieved from https://pentagramresearch.com/products.htm

Popovici, A., & Popovici, D. (2002). Cellular automata in image processing. *15th International Symposium on the Mathematical Theory of Networks and Systems* (pp. 1-6). electronic proceedings.

Powers, D. M. (2011). Evaluation From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Prewitt, J. (1970). Object Enhancement and Extraction In Picture Processing and Psychopictorics. *Academic Press*.

Priya, C. S., Mahesh, K., & Kuppusamy, K. (2014). Contour-based foreground moving object extraction in video segmentation, . *International journal of image processing and data visualization*.

Priyadarshini, S., & Sahoo, G. (2010). A New Edge Detection Method based on additions and Divisions. *International Journal of Computer Applications*, *9*(10).

Pulli, K., Baksheev, A., Kornyakov, K., & Eruhimov, V. (2012). Real-time Computer Vision with OpenCV. *Magazine Queue - Processors, 10*(4), 40. doi:10.11452181796.2206309

Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection- a survey. *IEEE Trans. Image Process.*, *14*(3), 294-307.

Rajan, E. G. (1993). Cellular logic array processing techniques for high-throughput image processing systems. *Trends In Computer Vision, Sadhana*, *18*(2), 279-300.

RamaBai, M., Krishna, V., & SreeDevi, J. (2010). A new Morphological Approach for Noise Removal cum Edge Detection. *International Journal of Computer Science Issues, 7*(6).

Ramirez-Rivera, A., Murshed, M., & Chae, O. (2011). Object detection through edge behavior modeling. *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, (pp. 273-278). Klagenfurt. doi:1109AVSS.2011.6027336

Raty, T. D. (2010). Survey on Contemporary Remote Surveillance Systems for Public Safety. *IEEE Transactions on Systems, Man, and Cybernetics - Part C Applications and Reviews, 40*(5), 493–515.

Roberts, L. (1963). *Machine Perception of Three-Dimensional Solids.* Massachusetts Institute of Technology: Department of Electrical Engineering, Massachusetts Institute of Technology.

Rosin, P. L., & Ioannidis, E. (2003). Evaluation of global image thresholding for change detection. *Pattern Recognition Letters, 24*, 2345-2356.

Rosin, P., Adamatzky, A., & Sun, X. (2014). Cellular Automata in Image Processing and Geometry. *Emergence, Complexity and Computation Series, Springer, 10*.

Samanta, D., & Sanyal, G. (2012). Novel Approach of Adaptive Thresholding Technique for Edge Detection in Videos. *Procedia Engineering, 30*, 283-288.

Sankari, M., & Meena, C. (2011). Estimation of Dynamic Background and Object Detection in Noisy Visual Surveillance. *International Journal of Advanced Computer Science and Applications, 2*(6).

Sappa, A., & Dornaika, F. (2006). An Edge-Based Approach to Motion Detection. *Lecture Notes in Computer Science, Springer, 3991*.

Sarhan, A., & Faheem, M. (2010). A proposed intelligent denoising technique for spatial video denoising for real-time applications. *Int. J. Mobile Computing Multimedia Commun.*, 20-39.

Scott, J., Pusateri, A. M., & Cornish, D. (2009). Kalman filter based video background estimation. *Applied Imagery Pattern Recognition Workshop (AIPRW)*, (pp. 1-7). doi:10.1109AIPR.2009.5466306

Serby, D., Koller-Meier, E., & Gool, L. J. (2004). Probabilistic object tracking using multiple features. *Int. Conf. on Pattern Recognition, 2*, pp. 184-187.

Shahbaz, A., Hariyono, J., & Jo, K. (2015). Evaluation of Background Subtraction Algorithms for Video Surveillance. *FCV 2015.*

Shao, Q., Zhou, Y., Li, L., & Chen, Q. (2015). Adaptive background subtraction approach of Gaussian mixture model. *Journal of Image and Graphics, 20*(6), 756-776.

Shivakumara, P., Sreedhar, R., Trung, Q. P., Shijian, L., & Chew, L. T. (2012). Multi-oriented video scene text detection through Bayesian classification and boundary growing. *IEEE Trans. Circuits Syst. Video Technol., 22*(8), 1227–1235.

Shivakumara, P., Trung, Q. P., & Chew, L. T. (2011). Laplacian approach to multi-oriented text detection in video. *IEEE Trans. Pattern Anal. Mach. Intell., 33*(2), 412-419.

Singh, S., Prasad, A., Srivastava, K., & Bhattacharya, S. (2016). A cellular logic array based data mining framework for object detection in video surveillance system. *2nd International Conference on Next Generation Computing Technologies (NGCT)*, (pp. 719-724). Dehradun, India. doi:10.1109/NGCT.2016.7877505

Singh, S., Prasad, A., Srivastava, K., & Bhattacharya, S. (2017). Empirical Evaluation of Edge based Background Subtraction Methods for Object Detection in Video Surveillance System. *International Journal of Applied Engineering Research (ISSN 0973-4562), 12*(22), 12036-12043.

Singh, S., Prasad, A., Srivastava, K., & Bhattacharya, S. (2017). Threshold modeling for cellular logic array processing based edge detection algorithm. *International Conference on Computing, Communication and Automation (ICCCA),* (pp. 1158-1162). Greater Noida, India. doi:10.1109/CCAA.2017.8229972

Singh, S., Prasad, A., Srivastava, K., & Bhattacharya, S. (2018). A Novel Method to Improve Basic Background Subtraction Methods for Object Detection in Video Surveillance System. *International Journal of Applied Engineering Research, (ISSN 0973-4562), 13*(4), 1866-1873.

Singh, S., Prasad, A., Srivastava, K., & Bhattacharya, S. (2018). Cellular Automata based Edged Background Subtraction Methods for Motion Object Detection in Video Streams. *Journal of Information Technology Project Management (IJITPM), special issue of Advancement and Imminent of IT*.

Singh, S., Prasad, A., Srivastava, K., & Bhattacharya, S. (2018). On the application of Cellular Logic Array Processing and Background Subtraction Methods for the Improved Object Detection in Video Surveillance using Local Neighborhood Difference. *International Journal of Imaging & Robotics (ISSN 2231-525X), 18*(4).

Smith, P., Drummond, T., & Cipolla, R. (2000). Segmentation of multiple motions by edge tracking between two frames. *11th British Machine Vision Conference*, (pp. 342-351). Bristol.

Smith, S. M., & Brady, J. M. (1997). SUSAN- a new approach to low-level image processing. *Intl. Journal of Computer Vision, 23*(1), 45-78.

Sobel, I., & Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. In D. R., & p. Hart, *Pattern Classification and Scene Analysis* (pp. 271-272). John Wiley & Sons.

Sobral, A., & Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding, CVIU 2014.*

Soh, Y., Hae, Y., Mehmood, A., Hadi Ashraf, R., & Kim, I. (2013). Performance Evaluation of Various Functions for Kernel Density Estimation. *Open Journal of Applied Sciences, 3*(18), 58-64.

Somhorst, M. (2012). *Multi-camera video surveillance system.* Master's Thesis, Man-Machine Interaction Group, Faculty of Electrical Engineering, Mathematics and Computer Science. Delf University.

Srinivasan, K., Porkumaran, K., & Sainarayanan, G. (2009). Intelligent human body tracking, modeling, and activity analysis of video surveillance system: A survey. *Journal of Convergence in Engineering, Technology and Science, 1*, 1-8.

Stauffer, C., & Grimson, W. E. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 747-757.

Subudhi, B. N., Ghosh, S., & Nanda, P. K. (2016). Moving object detection using spatiotemporal multilayer compound Markov Random Field and histogram thresholding based change detection. *Multimedia Tools and Applications*, 1-33.

Sun, Q., Hou, Y., Tan, Q., Li, C., & Liu, M. (2014). A robust edge detection method with sub-pixel accuracy. *Optik - International Journal for Light and Electron Optics, 125*(14), 3449-3453.

Suresh, S., & Deepak, P. C. (2014). An efficient low-cost background subtraction method to extract foreground object during human tracking. *International Conference on Circuit, Power and Computing Technologies*, (pp. 1432-1436).

Tzu-HengHenry, L. L. (2012). Edge Detection Analysis. *IJCSI International Journal of Computer Science Issues, 5*(6).

Venetianer, P. L., & Deng, H. L. (2010). Performance evaluation of an intelligent video surveillance system—A case study. *Comput. Vis. Image Understanding, 114*(11), 1292–1302.

Vijayakumar, V., & Nedunchezhian, R. (2012). A study on video data mining. *Int J Multimedia Info Retr*, 1153-1172.

Vishwakarma, S., & Agrawal, A. (2013). A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer, 29*(10), 983-1009.

Wang, B., & Dudek, P. (2014). A Fast Self-Tuning Background Subtraction Algorithm. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (pp. 401-404). Columbus, OH. doi:10.1109CVPRW.2014.64

Wang, B., Zhang, K., Shi, L., & Zhong, H. (2013). An Edge Detection Algorithm for Moving Object Based on Background Modeling and Active Contour Model. *Advanced Materials Research, 765-767*, 2393-2398.

Wang, J. J., & Singh, S. (2003). Video analysis of human dynamics - a survey. *Real-Time Imaging, 9*(5), 321-326.

Wang, p. (2010). Moving Object Segmentation Algorithm Based on Edge Detection. *International Conference on Computational Intelligence and Software Engineering*, (pp. 1-4). WuhaN. doi:10.1109CISE.2010.5677227

Wang, Y., Velipasalar, S., & Casares, M. (2010). Cooperative object tracking and composite event detection with wireless embedded smart cameras. *IEEE Trans. Image Process., 19*(10), 2614-2633.

Wani, F., Khan, A. A., & Patil, S. B. (2014). A Survey of Image Processing Techniques for Identification and Tracking of Objects from a Video Sequence. *International Journal of Computer & Organization Trends, 6*(1), 24-29.

Warade, P. P., Kale, S. N., & Thakare, V. M. (2015). Edge Detection for Moving Object Tracking. *International Journal of Advanced Research in Computer Science and Software Engineering, 5*(4).

Wengrovitz, M. S., & Janiszewski, T. (2014). *United States Patent No. US8773532B2.*

William, T. (2016). Logging Bioinformatics and Computational Biology in Drug Discovery and Development. *Cambridge University Press*.

Winkler, S., & Mohandas, P. (2008). The Evolution of Video Quality Measurement From PSNR to Hybrid Metrics. *IEEE TRANS. BROADCASTING, 54*(3), 660-668.

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 780–785.

Xiao, M. (2008). An Improved Background Reconstruction Algorithm Based on Basic Sequential Clustering. *Information Technology Journal, 7*, 522-527.

Xu, Y., Dong, J., Zhang, B., & Xu, D. (2016). Background modeling methods in video analysis A review and comparative evaluation. *CAAI Transactions on Intelligence Technology, 1(1),* pp. 43-60.

Yan, J., Yu, Y., Zhu, X., Lei, Z., & Li, S. Z. (2015). Object detection by labelling superpixels. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Boston, MA, USA.

Yang, X., Xu, Y., Zhang, R., Chen, E., Yan, Q., Xiao, B., . . . Huang, J. (2009). Surveillance Event Detection Task.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *Acm Computing Surveys, 38*(4).

Yilmaz, A., Li, X., & Shah, M. (2004). Object Contour Tracking Using Level Sets. *Asian Conf. on Computer Vision*, (pp. 1-5). Seoul.

Yin, F., Makris, D., & Velastin, S. (2007). Performance Evaluation of Object Tracking Algorithms. *International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2007).* Rio de Janeiro, Brazil .

Yokoi, K., Watanabe, T., & Ito, S. (Performers). (2009). *High-Level Feature Extraction and Surveillance Event Detection at TRECVID 2009.* Toshiba Shanghai Jiao Tong University, Shanghai.

Zablocki, M., Gosciewska, K., Frejlichowski, D., & Hofman, R. (2014). Intelligent video surveillance systems for public spaces – a survey. *Journal of Theoretical and Applied Computer Science, 8*(4), 13-27.

Zhan, C., Duan, X., Xu, S., Song, Z., & Luo, M. (2007). An Improved Moving Object Detection Algorithm Based on Frame Difference and Edge Detection. *Fourth International Conference on Image and Graphics (ICIG 2007)*, (pp. 519-523). Sichuan. doi:10.1109ICIG.2007.153

Zhang, J., Collins, R., & Liu, Y. (2009). Representation and matching of articulated shapes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, (pp. 342-349).

Zhang, J., Tian, Y., Yang, Y., & Zhu, C. (2009). Robust Foreground Segmentation Using Subspace-Based Background Model. *Asia-Pacific Conference on Information Processing, APCIP 2009, 2*, pp. 214-217.

Zhang, W., Wu, Q. M., & Yin, H. B. (2010). Moving vehicles detection based on adaptive motion histogram. *Digit. Signal Process., 20*(3), 793-805.

Zhu, S. (2011). Edge Detection Based on Multi-structure Elements Morphology and Image Fusion. *IEEE 2nd International Conference IE, 2.*

Zhu, S. C., & Yuille, A. (1996). Region Competition Unifying Snakes, Region Growing, and Bayes for Multiband Image Segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 18,* 884-900.

Zidek, K., & Hosovsky, A. (2014). Image Thresholding and Contour Detection with Dynamic Background Selection for Inspection Tasks in Machine Vision. *International Journal of Circuits, Systems and Signal Processing, 8*.

APPENDIX: A

WORKS PUBLISHED/PRESENTED/ACCEPTED FOR PUBLICATION BASED ON THE CURRENT RESEARCH

Published:

[1]. S. Singh, A. Prasad, K. Srivastava and S. Bhattacharya, "A cellular logic array based data mining framework for object detection in video surveillance system," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 2016, pp. 719-724. doi: 10.1109/NGCT.2016.7877505 (**IEEE Conference SCOPUS Indexed)**

URL link: ieeexplore.ieee.org/document/7877505/

[2]. S. Singh, A. Prasad, K. Srivastava and S. Bhattacharya, "Threshold modeling for cellular logic array processing based edge detection algorithm," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 1158-1162. doi: 10.1109/CCAA.2017.8229972 (**IEEE Conference SCOPUS Indexed)**

URL link: ieeexplore.ieee.org/document/8229972/

[3]. Surender Singh, Ajay Prasad, Kingshuk Srivastava, Suman Bhattacharya, " Empirical Evaluation of Edge based Background Subtraction Methods for Object Detection in Video Surveillance System," International Journal of Applied Engineering Research (ISSN 0973-4562) Volume 12, Number 22 (2017) pp. 12036-12043 (**SCOPUS indexed Journal**)

URL link: https://www.ripublication.com/ijaer17/ijaerv12n22_37.pdf

[4]. Surender Singh, Ajay Prasad, Kingshuk Srivastava, Suman Bhattacharya, " A Novel Method to Improve Basic Background Subtraction Methods for Object Detection in Video Surveillance System," International Journal of Applied Engineering Research, (ISSN 0973-4562) Volume 13, Number 04 (2018) pp. 1866-1873 (**SCOPUS indexed Journal)**

URL link: https://www.ripublication.com/ijaer18/ijaerv13n4_10.pdf

Accepted for Publication:

[5]. Surender Singh, Ajay Prasad, Kingshuk Srivastava, Suman Bhattacharya, "On the application of Cellular Logic Array Processing and Background Subtraction Methods for the Improved Object Detection in Video Surveillance using Local Neighborhood Difference," International Journal of Imaging & Robotics (ISSN 2231-525X), 2018. (**SCOPUS Indexed Journal - Accepted for Publication in Vol. 18 and issue No. 4**)

[6]. Surender Singh, Ajay Prasad, Kingshuk Srivastava, Suman Bhattacharya, "Cellular Automata based Edged Background Subtraction Methods for Motion Object Detection in Video Streams", Doctoral Colloquium in Management, Economics and Information Technology (DCMEIT-2017) on 18th November 2017. (**Accepted for publication in SCOPUS indexed International Journal of Information Technology Project Management (IJITPM) special issue of Advancement and Imminent of IT**)

Communicated for Publication:

[7]. Surender Singh, Ajay Prasad, Kingshuk Srivastava, Suman Bhattacharya, "Object Motion Detection Methods for Real-Time Video Surveillance: A Survey with Empirical Evaluation", **(Communicated for publication)**

APPENDIX: B

BRIEF BIODATA

SURENDER SINGH

B2, Ambala College of Engg. & Applied Research,
Devsthali, Mithapur, Sambhalkha, Ambala (HR)
Ph: 8295300367, 9729154367.
surendahiya@gmail.com

Summary of Qualification and Experience:

- **Qualification:** M. Tech. (CSE), MFC (Finance) B. Tech. (ECE)
- **Total Experience:** 20 years, Academic : 16½ Yrs & Industrial: 3½ Yrs

Professional work Experience:

- Associate Professor in Computer Science & Engineering (July 2013 onward) ACE, Mithapur, Ambala, HR
- Associate Professor in Computer Science & Engineering (June 2011 to July 2013) SIET, Aliyaspur, Ambala, HR
- Associate Professor in Information Technology (May 2010 to June 2011) ISTK, Kalawad, Yamunanagar, HR
- Associate Professor in Computer Science & Engineering (August 2008 to May 2010), APIIT SD INDIA, Panipat, HR
- Associate Professor in Information Technology (July 2007 to July 2008), MMEC, Mullana, Ambala, HR
- Assistant Professor in Computer Engineering (July 2003 to July 2007) HEC, Jagadhri, Yamunanagar, HR
- Lecturer in Information Technology **(**January 2000 to July 2003) JIET, Jind, HR
- System Engineer June 1997 to January 2000 BSSL, Sahibabad UP.

- Management Trainee July 1994 to June 1995 BSSL, Sahibabad UP

Education:

- M. Tech. in Computer Science and Engineering (2001–2003)
  Dept. of CSE, GJUST, Hisar, HR
- Master of Finance and Control (1995–1997)

  DBS, KUK, Kurukshetra, HR
- B. Tech. in Electronics & Communication Engineering (1990–1994)
  NIT, Kurukshetra, HR

Certificates Earned:

- **Machine Learning** by Stanford University on Coursera.
  Certificate earned on June 9, 2017 (Grade Achieved:96.9%)
  https://www.coursera.org/account/accomplishments/certificate/3G58EPJK98TK
- Microsoft certified System Engineer (MCSE)

Research Consultancy:

- An MNC project named: "An intelligent Web Advertisement Scheduling" completed in 2006.
-

Research Supervised:

- Supervised more than Fourteen M. Tech. Dissertations on Soft Computing, Software Testing, Wireless Sensor Networks, Digital Image Processing, Machine Learning etc.

Research Interest:

Machine Learning, Digital Image Processing, Software Engineering, Soft Computing, Computer Networks.

Research Publication: 33 (Communicated:1), Book Chapter:1, National Journal:2, National Conference:3, International Journal:14, International Conference:13

(Surender Singh)