VIDEO SURVEILLANCE: EXPLOSION DETECTION

By

Lee Shao Yuan

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Leung Kar Hang, for granting me the opportunity to be involved in a computer vision project. His invaluable support and insightful guidance have greatly enriched my learning experience. This project marks an important first step in my journey into the field of computer vision, and I am deeply thankful for his mentorship throughout.

I would also like to take this opportunity to thank my immediate supervisor, who stepped in to provide continuous guidance and advice during the period when my main supervisor was on medical leave. Their dedicated support, timely feedback, and encouragement ensured that I was able to remain focused and make consistent progress in my project despite the challenging circumstances.

I would also like to extend my heartfelt appreciation to my parents and family for their unconditional love, unwavering support, and continuous encouragement throughout the course. Their presence has been a constant source of strength and motivation.

ABSTRACT

The proliferation of surveillance technologies has emphasised their pivotal role in enhancing public safety by monitoring and detecting anomalies in real time. Among the anomalies, explosions present a grave threat due to the potentially resulting major loss of life, widespread panic and significant destruction of property. However, traditional surveillance systems are limited by their reliance on human monitoring, which is susceptible to oversight due to fatigue or distractions. Thus, this research focusses on developing an intelligent explosion detection surveillance system that is capable of early and accurate explosion detection. Explosions typically happen in a very short timeframe, often just a few seconds, leading to significant challenges for the rapid and accurate identification of explosions' unique visual patterns immediately. Hence, this study proposes to embed computer vision and advanced image processing algorithms, such as Motion History Images (MHI) and Motion Energy Images (MEI), into the intelligent explosion detection video surveillance system. By leveraging three motion-based variables, including motion ratio, new pixel ratio and optical flow values, together with three detection approaches, namely global detection, non-eroded detection and eroded detection, the system demonstrates the effectiveness of motion-based methods in detecting explosions at an early stage with acceptable performance. Eventually, this approach aims to minimise explosions' damage by enabling immediate responses, preventing the spread of fires and the occurrence of secondary explosions.
Area of Study: Computer Vision, Image Processing

Keywords: Explosion Detection, Intelligent Video Surveillance System, Motion History Images (MHI), Motion Energy Images (MEI), Real Time Detection, Motion-Based Methods

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

τ	Tau, number of past frames or the maximum motion intensity value

LIST OF ABBREVIATIONS

CCTV	Close-Circuit Televisions
MIL	Multiple Instance Learning
AUC	Area Under the Curve
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
Adam	Adaptive Moment Estimation
I3D	Inflated 3D Convnet
BCE	Binary Cross-Entropy
CTR	Causal Temporal Relation
ANFs	Anomalous-Near Features
AAFs	Anomalous-Away Features
RTFM	Robust Temporal Feature Magnitude Learning
TCA	Temporal Context Aggregation
PEL	Prompt-Enhanced Learning
DPE	Dynamic Position Encoding
CLIP	Contrastive Language-Image Pretraining
FAR	False Alarm Rate
UMIL	Unbiased Multiple Instance Learning
PCA	Principal Component Analysis
MEI	Motion-Energy Images
MHI	Motion History Images
HAR	Human Action Recognition
ST-MEI	Short-Time Motion Energy Image
ROI	Region Of Interest
IDE	Integrated Development Environment

CHAPTER 1

Introduction

In this chapter, we present a brief introduction to the problem we aim to address, the background and motivation of our research, our expected contributions to the field, and the outline of the thesis.

1.1 Problem Statement and Motivation

An explosion is a sudden and violent burst of energy or rapid expansion that can generate severe threats to human life, infrastructure, and the environment. It is often triggered by the detonation of a bomb or by a chemical or nuclear reaction, generating enormous amounts of heat, light, sound, and pressure waves. The explosion damage is often disastrous, which is usually accompanied by side effects such as fires and secondary explosions.

Traditional video surveillance systems rely heavily on human monitoring to detect abnormal events such as explosions. However, the dramatic increase in the number of Closed-Circuit Television (CCTV) cameras has introduced new challenges. Currently, there are millions of CCTV cameras installed in developed countries—approximately 200 million in China and 50 million in the United States [1]. With the widespread installation of closed-circuit television (CCTV) in public and private spaces, this imposes an excessive workload on human operators. Meanwhile, operators usually become fatigued and lose concentration over prolonged monitoring periods, causing them to have a high chance of missing detection for sudden events such as explosions. This inefficiency in detection and response leads to increased casualties, longer rescue times, and greater damage.

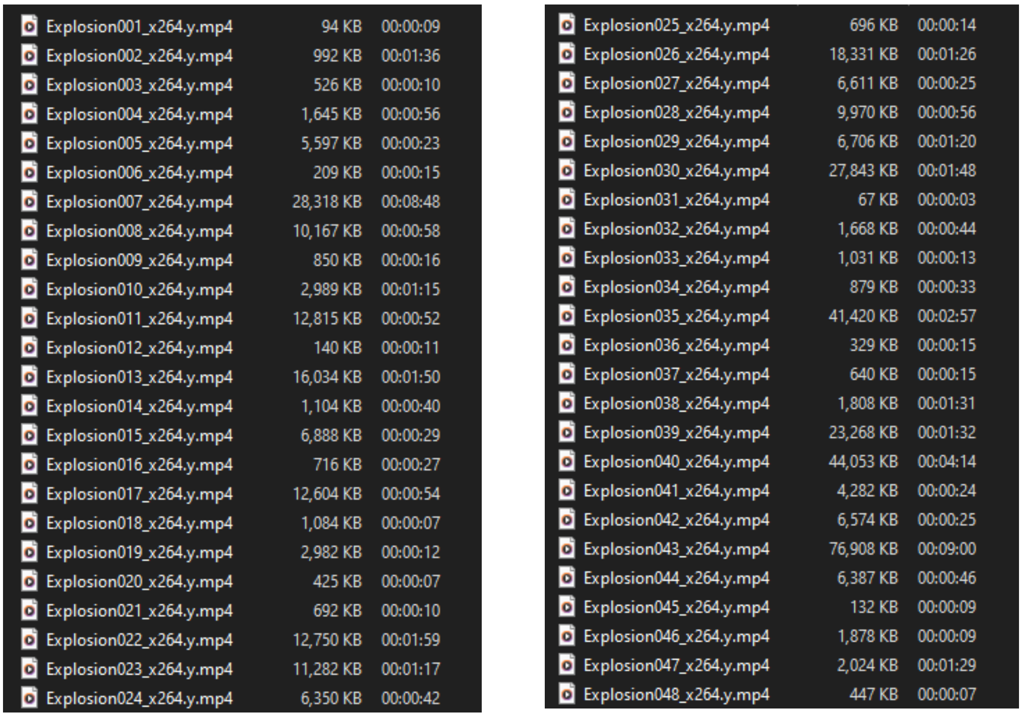
For instance, the explosion of the Piper Alpha oil platform operated by Occidental Petroleum in 1988 is an unforgettable industrial explosion event. It was started with the leakage of gas from a faulty pump, causing the first explosion that resulted in a huge fire. The fire then led to a series of subsequent explosions, resulting in more fire spreading around the platform. Later, the massive fire gradually engulfed the platform due to the existence of the huge quantities of oil and gas on the platform. This explosion not only resulted in the complete destruction of the platform but also regrettably caused 167 deaths [2]. This tragedy shows the devastating consequences of explosions, making immediate explosion detection a critical area of focus so such a tragedy will not happen again in the future.

Thus, an intelligent video surveillance system leveraging image processing techniques is necessary to develop for detecting explosion events automatically. The intelligent video surveillance system for explosion detection can provide real-time alarms to authorities whenever an explosion happens with the aim of safeguarding lives and property.

1.2 Project Scope

The project scope covers the analysis and detection of explosion event based on the UCF-Crime dataset with the implementation of computer vision and image processing techniques.

There are total of 48 videos containing the scenes of explosion event from the UCR-Crime dataset listed in Figure 1.1. The videos range in length from 9 seconds to 8 minutes, with the explosion appearing within a few seconds. To ensure consistency and reduce noise in motion analysis, all selected videos are required to have a fixed scene with no camera movement or scene transitions.



Explosion001_x264.y.mp4	94 KB	00:00:09
Explosion002_x264.y.mp4	992 KB	00:01:36
Explosion003_x264.y.mp4	526 KB	00:00:10
Explosion004_x264.y.mp4	1,645 KB	00:00:56
Explosion005_x264.y.mp4	5,597 KB	00:00:23
Explosion006_x264.y.mp4	209 KB	00:00:15
Explosion007_x264.y.mp4	28,318 KB	00:08:48
Explosion008_x264.y.mp4	10,167 KB	00:00:58
Explosion009_x264.y.mp4	850 KB	00:00:16
Explosion010_x264.y.mp4	2,989 KB	00:01:15
Explosion011_x264.y.mp4	12,815 KB	00:00:52
Explosion012_x264.y.mp4	140 KB	00:00:11
Explosion013_x264.y.mp4	16,034 KB	00:01:50
Explosion014_x264.y.mp4	1,104 KB	00:00:40
Explosion015_x264.y.mp4	6,888 KB	00:00:29
Explosion016_x264.y.mp4	716 KB	00:00:27
Explosion017_x264.y.mp4	12,604 KB	00:00:54
Explosion018_x264.y.mp4	1,084 KB	00:00:07
Explosion019_x264.y.mp4	2,982 KB	00:00:12
Explosion020_x264.y.mp4	425 KB	00:00:07
Explosion021_x264.y.mp4	692 KB	00:00:10
Explosion022_x264.y.mp4	12,750 KB	00:01:59
Explosion023_x264.y.mp4	11,282 KB	00:01:17
Explosion024_x264.y.mp4	6,350 KB	00:00:42
Explosion025_x264.y.mp4	696 KB	00:00:14
Explosion026_x264.y.mp4	18,331 KB	00:01:26
Explosion027_x264.y.mp4	6,611 KB	00:00:25
Explosion028_x264.y.mp4	9,970 KB	00:00:56
Explosion029_x264.y.mp4	6,706 KB	00:01:20
Explosion030_x264.y.mp4	27,843 KB	00:01:48
Explosion031_x264.y.mp4	67 KB	00:00:03
Explosion032_x264.y.mp4	1,668 KB	00:00:44
Explosion033_x264.y.mp4	1,031 KB	00:00:13
Explosion034_x264.y.mp4	879 KB	00:00:33
Explosion035_x264.y.mp4	41,420 KB	00:02:57
Explosion036_x264.y.mp4	329 KB	00:00:15
Explosion037_x264.y.mp4	640 KB	00:00:15
Explosion038_x264.y.mp4	1,808 KB	00:01:31
Explosion039_x264.y.mp4	23,268 KB	00:01:32
Explosion040_x264.y.mp4	44,053 KB	00:04:14
Explosion041_x264.y.mp4	4,282 KB	00:00:24
Explosion042_x264.y.mp4	6,574 KB	00:00:25
Explosion043_x264.y.mp4	76,908 KB	00:09:00
Explosion044_x264.y.mp4	6,387 KB	00:00:46
Explosion045_x264.y.mp4	132 KB	00:00:09
Explosion046_x264.y.mp4	1,878 KB	00:00:09
Explosion047_x264.y.mp4	2,024 KB	00:01:29
Explosion048_x264.y.mp4	447 KB	00:00:07

Figure 1.1 List of explosion videos from UCF-Crime dataset

In addition, the explosion can be classified into three categories: Mild Explosion (Figure 1.2) and Massive Explosion (Figure 1.3) as shown below.

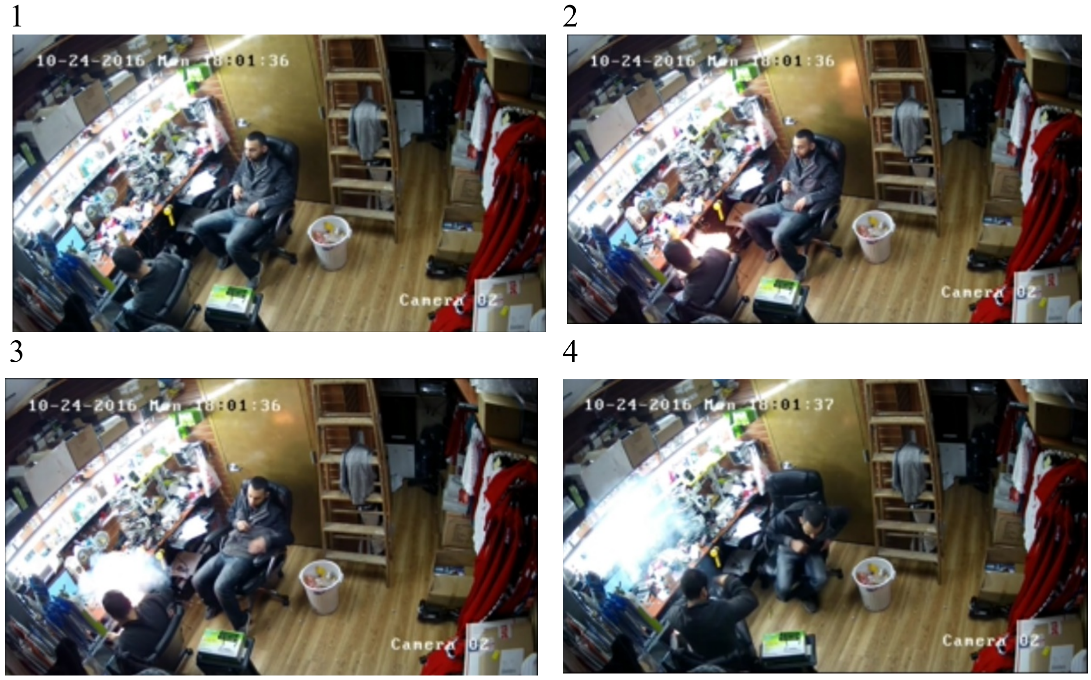


Figure 1.2 Sample of Mild Explosion

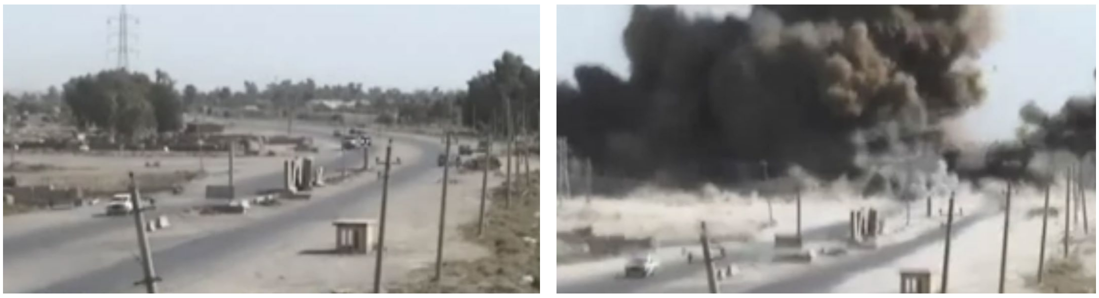


Figure 1.3 Sample of Massive Explosion

1.3 Project Objectives

The objectives of the project are:

I. **To develop a real-time explosion detection video surveillance system**

The system is embedded with technologies such as computer vision and image processing techniques to replace the traditional video surveillance system that rely heavily on human monitoring.

II. To detect significant visual changes in the environment

Explosion often happens with a few characteristics such as a blast wave, a dense plume of smoke or the presence of fire. By focusing on these specific visual changes, the system will be able to recognize explosions quickly and effectively.

III. To analyse temporal motion patterns for identifying explosion characteristics

The system will integrate Motion History Images (MHI) and Motion Energy Images (MEI) to capture and present motion in the video footage. By analysing these motion patterns, the system can differentiate explosion events from regular activities.

IV. To minimise false explosion event detections

The system will implement key motion parameters and thresholding to reduce false alarms caused by non-explosive activities such as human and vehicle movement. This ensures the accuracy and effectiveness of the detection system in real-world conditions.

1.4 Impact, Significance and Contribution

The real-time video surveillance system for explosion detection is a crucial enhancement in national security, public safety, and disaster response. The authorities will receive early notifications and alerts from this system whenever it detects explosion events. It significantly reduces response time for detecting explosion events, including motion detection and pattern recognition. Consequently, it increases the likelihood of saving lives and minimising explosion damage to property and infrastructure. Other than that, the system can reduce human errors by leveraging image processing techniques, enabling it to distinguish between true explosion events and false alarms. The capability to detect explosions accurately plays a key role in protecting critical locations such as industrial sites.

In addition, the implementation of a real-time explosion detection system optimises human resource utilisation. Traditional surveillance systems require continuous human monitoring, which is resource-intensive and costly. In contrast, the proposed video surveillance system eliminates the need for continuous human supervision and reduces the risk of human errors. The constant freedom from the monitoring system allows the security personnel to concentrate on higher-priority tasks. The high accuracy in detecting explosion events also allows it to notify authorities when necessary. Therefore, it effectively and efficiently utilises human resources. This project presents a proactive and innovative approach in surveillance technology, with the potential to revolutionise public safety protocols.

1.5 Project Background

1.5.1 Intelligent Video Surveillance System

Intelligent video surveillance systems are embedded with monitoring systems that utilise artificial intelligence (AI) to automatically analyse video footage and detect anomalies, such as explosions, vandalism, accidents, etc. Unlike traditional surveillance systems that rely heavily on manual monitoring, intelligent video surveillance systems can execute these detection jobs with better performance and efficiency while significantly reducing human workload by autonomously identifying, tracking, and classifying anomalies.

These intelligent video surveillance systems often adapt machine learning and deep neural network architectures to process and learn the large amounts of visual data effectively. As the systems are trained by continuously analysing video streams, they become capable of recognising specific characteristics that distinguish normal from abnormal events. After the training session, the systems can detect anomalies by detecting the specific visual patterns to indicate the event of an anomaly. However, these intelligent video surveillance systems usually detect variance of anomalies in one system, lacking a sophisticated explosion detection system in the market.

1.5.2 Image Processing Techniques and Motion Detection

Image processing techniques allow intelligent video surveillance systems to interpret and understand visual data from video footage by involving the adaptation of algorithms. These algorithms can enhance, filter, and analyse images to extract meaningful information from each frame. By analysing pixels from each frame, image processing helps the system identify specific patterns or characteristics that may indicate abnormal events, such as fire or smoke. This ability is essential to detect explosions, as explosions show unique visual features like sudden flashes or rapid expansion.

On the other hand, motion detection focuses on identifying and tracking changes in movement within the video footage. Motion is indicated if massive differences are detected between sequential frames. These algorithms can immediately recognise sudden and high-intensity motion, enabling it to detect characteristics of explosion events well. Motion detection algorithms are vital for capturing the extensive motion from an explosion event, such as a rapid increase in the number of motion pixels within a short time frame.

In short, image processing techniques and motion detection are fundamental components of modern intelligent surveillance systems which allow computers to analyse video footage and detect specific events such as explosions.

1.6 Report Organisation

The report is organised into 7 chapters. The subsequent Chapter 2 presents a review of related works relevant to this project. Chapter 3 outlines the system methodology and discusses selected real-world case scenarios. Chapter 4 explains the workflow of the proposed system in detail. Chapter 5 describes the system operation and its overall functionality. Chapter 6 discusses the testing of real-world case scenarios and presents the results along with an objective evaluation. Finally, Chapter 7 concludes the report and offers recommendations for future work.

CHAPTER 2

Literature Reviews

This chapter presents the details of previous work that are relevant to the development project.

2.1 Previous Work on Anomaly Detection

2.1.1 Real-world Anomaly Detection in Surveillance Videos

In this case study, Sultani et al. [3] proposed a deep multiple instances ranking framework to detect real-world anomalies such as explosion, accident, burglary, etc. Their system proposed to train their system using the weakly label training videos due to convenient annotation of large number of videos and improve time efficiency. The weakly label training videos indicated that the anomalous videos were known to be normal or containing anomaly somewhere. However, the location of the anomaly was not known during the training process.

Furthermore, they leveraged deep multiple instance learning (MIL) to learn anomaly, thereby differentiating the anomalous and normal cases. The system first considered normal and anomalous surveillance videos as bags and temporal video segments as instances. The bags containing anomalous video were labelled as positive else negatives for bags containing normal video.

Afterward, they applied a pre-trained 3D convolutional Neural Network that would extract C3D features from the video segments. An anomaly ranking model was learning by the deep MIL throughout the training process where it could compute the ranking loss function between the highest scored instances (shown in red) in the positive bags and negative bags. In short, the segments in the positive bags with highest anomaly score were most likely to contain anomaly. On converse, the segments in the negative bags possessed a highest anomaly score were described most similar to an anomalous segment, classifying as false positive examples. Nonetheless, these false positive examples were utilized to aid the deep MIL in understanding and resolving some ambiguity.

Figure 2.1 showed how the deep MIL was trained to predict the anomaly.

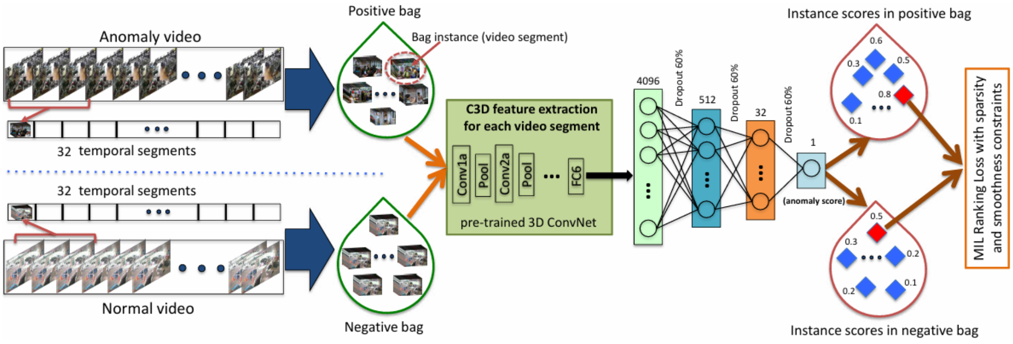


Figure 2.1 Flow diagram of the proposed anomaly detection system [3]

They also created a new, extensive dataset, dividing a total of 1900 surveillance videos into anomalous and normal categories. Those surveillance anomalous videos revealed clear anomalies, including abuse, arrest, arson, assault, accident, burglary, explosion, fighting, robbery, shooting, stealing, shoplifting, and vandalism. Figure 2.2 showed some examples of the anomalies contained in the dataset.

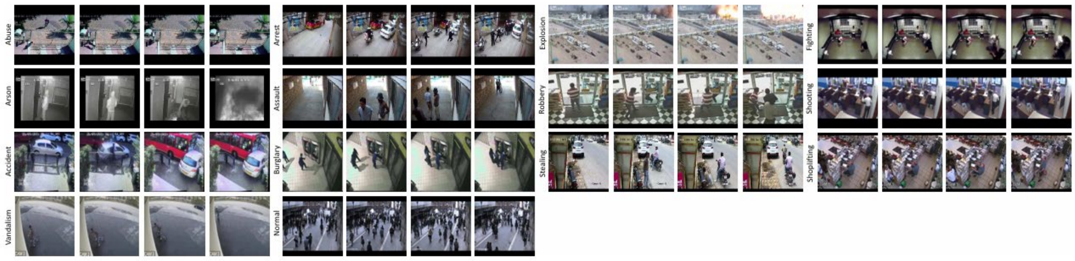
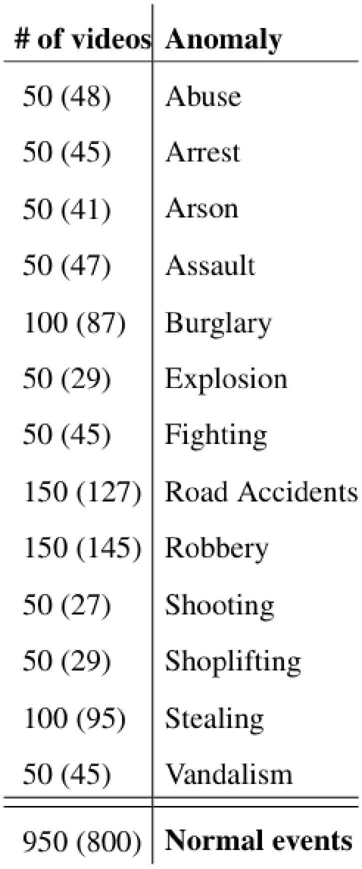


Figure 2.2 Example of different anomalies in dataset [3]

Table 2.1 showed the total number of each category anomalous video in the dataset. The numbers in the brackets indicate the total number of videos that underwent deep MIL training.



# of videos	Anomaly
50 (48)	Abuse
50 (45)	Arrest
50 (41)	Arson
50 (47)	Assault
100 (87)	Burglary
50 (29)	Explosion
50 (45)	Fighting
150 (127)	Road Accidents
150 (145)	Robbery
50 (27)	Shooting
50 (29)	Shoplifting
100 (95)	Stealing
50 (45)	Vandalism
950 (800)	**Normal events**

Table 2.1 Total number of videos of each anomaly in dataset [3]

During the experiments, 800 normal and 810 anomalous videos were utilized to train the system, while the remaining 150 videos from the dataset were used for testing purposes. Figure 2.3 depicted the development of the system on the training videos over iterations. The coloured windows illustrate an anomalous region. The system demonstrated effective anomaly detection after the 8000th iteration, exhibiting a higher anomaly score for abnormal frames and nearly zero for normal frames.

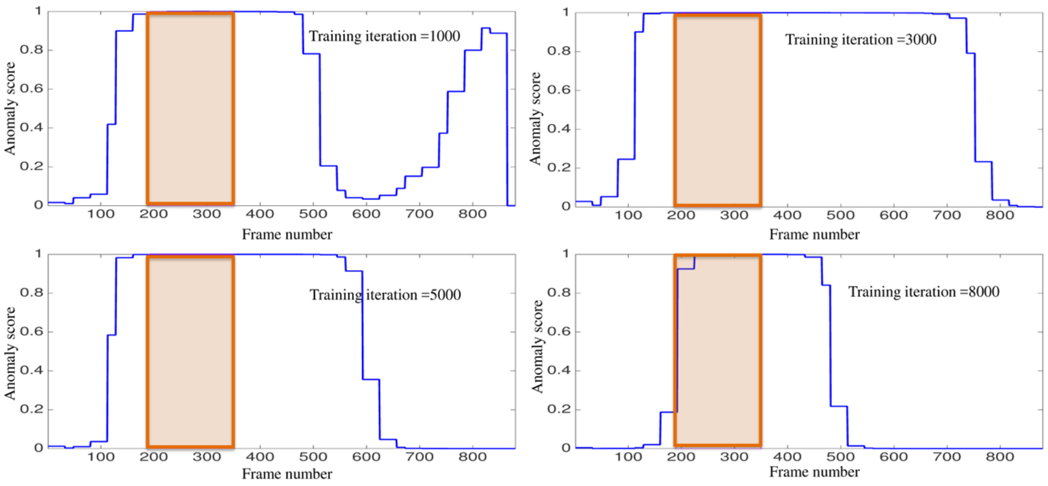


Figure 2.3 Development of the system on the training videos over iterations [3]

After the training phase, the system tested with the testing set available, and the experimental results presented were outstanding as the system could understand the frames better and automatically localise anomalies precisely. The proposed method in this case study outperformed deep auto-encoder and dictionary-based techniques. The proposed method acquired an area under the curve (AUC) of 75.41, while the other 2 techniques obtained AUC of 65.51 and 50.6, respectively.

Other than that, the proposed method achieved a lower false alarm rate than the other two techniques, proving that it was more practically robust. The proposed method triggered a false alarm with a rating of 1.9, while deep auto-encoder and dictionary-based techniques had false alarm rates of 27.2 and 3.1 separately. This demonstrated that the deep MIL ranking model was able to perform better by utilising both anomalous and normal videos during the training phase.

Nonetheless, a few weaknesses were shown by the proposed method. First, the proposed method appeared to have a weakness in the dark scene, leading to a failure of anomaly detection. Moreover, the system would likely generate false alarms due to blockage of flying insects in front of the camera and sudden people gathering.

2.1.2 Real-Time Anomaly Recognition Through CCTV Using Neural Networks

Singh et al. [4] proposed an anomaly recognition system, which was a real-time surveillance program that employed two distinct deep learning models to identify and report anomalies in CCTV footage. They specifically used these models to detect and categorise instances of significant movement within the frame. The detection of suspicious activities triggered an alarm, alerting users to the possibility of anomalous events. This case study successfully recognised 12 types of anomalous activities, such as explosions, vandalism, and road accidents.

Figure 2.4 depicted the proposed framework for the anomaly recognition system, which implemented convolutional neural network (CNN) and recurrent neural network (RNN). The CNN played a crucial role in extracting key features and its output was subsequently fed into the RNN for final classification.

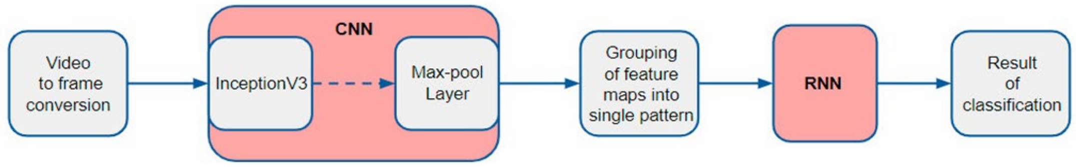

Video to frame conversion
CNN
InceptionV3
Max-pool Layer
Grouping of feature maps into single pattern
RNN
Result of classification

Figure 2.4 Workflow of anomaly recognition system [4]

The process of the system began with the extraction of frames from captured CCTV recordings. The system typically extracted these frames one second long, transforming the dynamic video content into a series of static images suitable for frame-by-frame analysis. They preprocessed these frames to the required format for further analysis by CNN layer. This preprocessing included resizing each frame to 299x299 pixels, the standard input dimension for the InceptionV3 model.

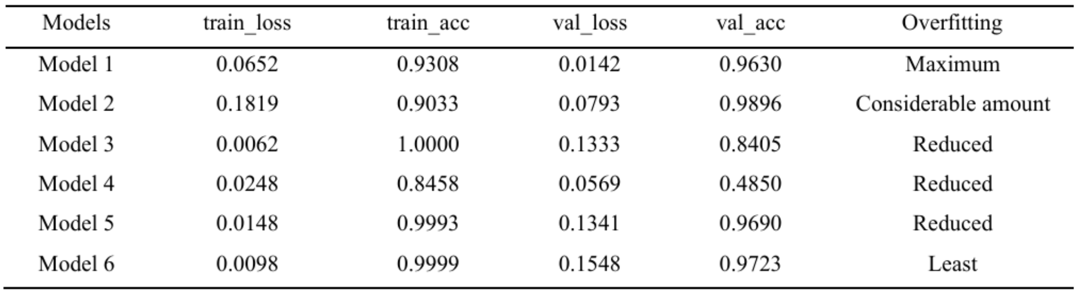
Feature extraction was conducted using InceptionV3, a CNN pre-trained on the ImageNet dataset. This model identified a wide range of objects and scenarios within the frames, focussing on extracting general features from the images in the initial half. In the second half, these images were subsequently classified based on the extracted features with greater specificity. Following feature extraction, the process continued with feature aggregation and refinement. The extracted features then passed through the Max-pool Layer, which was vital for efficiently processing large volumes of data and resulted in a condensed feature map consisting of 2,048 features.

Furthermore, multiple pre-processed frames were aggregated into chunks to provide a sequential understanding of the footage. These frames encapsulated temporal segments of the video, offering insights into motion patterns. During this phase, certain feature maps predicted by the Inception model were stored and analysed to create a high-level feature map, aiding in the identification of objects and shapes. This combined feature map was then passed to the RNN, simplifying the training process and reducing the complexity.

The final phase involved sequence analysis using RNN, processing the concatenated collection of high-level feature maps generated by the prior step. This network utilises a Long Short-Term Memory (LSTM) cell, containing 5727 neurons in its primary layer, followed by two hidden layers. The first hidden layer comprises 1,024 neurons using the Relu activation function, and the subsequent layer includes 50 neurons with the sigmoid activation function. The LSTM processed the concatenated feature maps to detect anomalies by learning the patterns of motion and object interactions. The output layer which consists of thirteen neurons with a softmax activation function, performs the probabilistic classification, classifying the videos into 13 predefined anomaly groups.

To enhance the performance of the anomaly recognition system, they collected 950 unaltered real-world surveillance videos containing anomalies and 940 normal scenarios, making a total of 1800 videos in their dataset from the UCF-Crime Dataset. They trained the model exclusively on labelled recordings to identify deviations from typical behaviour. To evaluate the model, the start and end frames of abnormal events were established with the implementation of multiple annotators, and these annotations were averaged to determine the temporal extent of each anomalous behaviour. The dataset was divided in a 3:1 ratio for training and validation, with normal videos shuffled among those containing anomalies.

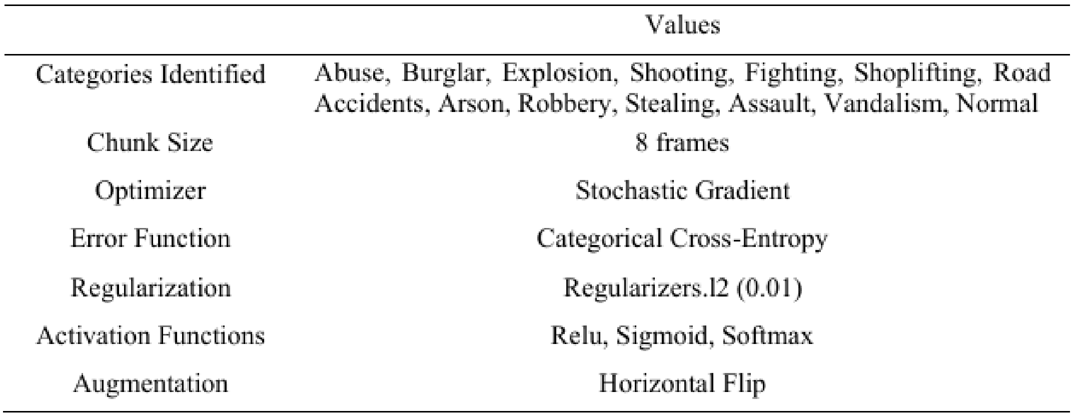
To achieve optimal results, six models were trained using various parameters and refining the dataset. After multiple trials and adjustments, Model 6 demonstrated the desired performance, with a training set accuracy of 0.9999, which was nearly perfect, and a validation set accuracy of 0.9723, which is also exceptionally high. Additionally, Model 6 exhibited the least amount of overfitting among all the models evaluated. Table 2.2 displayed the performance comparison of the six models that were trained during the experiments.



Models	train_loss	train_acc	val_loss	val_acc	Overfitting
Model 1	0.0652	0.9308	0.0142	0.9630	Maximum
Model 2	0.1819	0.9033	0.0793	0.9896	Considerable amount
Model 3	0.0062	1.0000	0.1333	0.8405	Reduced
Model 4	0.0248	0.8458	0.0569	0.4850	Reduced
Model 5	0.0148	0.9993	0.1341	0.9690	Reduced
Model 6	0.0098	0.9999	0.1548	0.9723	Least

Table 2.2 Performance Comparison of all models [4]

The optimised model in this case study was Model 6, which successfully classified all 13 categories, which included 12 groups of anomalies and 1 normal scenario. The model was configured to concatenate 8 frames to form a chunk for analysis. It utilised Stochastic Gradient with Adaptive Moment Estimation (Adam) parameter as optimiser and categorical cross-entropy for the error function. Additionally, it implemented three different types of activation functions: Relu, Sigmoid, and Softmax. The video was flipped horizontally, where the amount of data available for testing was doubled, further increasing the testing accuracy of the model. The details of the optimised model were illustrated in Table 2.3.



	Values
Categories Identified	Abuse, Burglar, Explosion, Shooting, Fighting, Shoplifting, Road Accidents, Arson, Robbery, Stealing, Assault, Vandalism, Normal
Chunk Size	8 frames
Optimizer	Stochastic Gradient
Error Function	Categorical Cross-Entropy
Regularization	Regularizers.l2 (0.01)
Activation Functions	Relu, Sigmoid, Softmax
Augmentation	Horizontal Flip

Table 2.3 Details about the Optimized Model [4]

The authors stated that the proposed models required fast Internet connectivity and were memory intensive. Consequently, they recommended modifications to the models to create a more effective and cost-efficient solution in the future. Additionally, it was suggested that the models should be enhanced to predict potential incoming threats and alert authorities, further improving public safety.

2.1.3 Real-World Video Anomaly Detection by Extracting Salient Features in Videos

Watanabe et al. [5] reviewed recent anomaly detection methods that relied on MIL or unsupervised learning and found that they emphasised the importance of learning temporal order relationships among video segments. Nevertheless, they challenged this fundamental assumption in video anomaly detection by proposing a lightweight and accurate method with a self-attention mechanism. The self-attention mechanism allows the system to automatically extract salient features from video segments, which are crucial for determining normal and abnormal status. Based on their findings, they argued that the temporal order of segments was irrelevant for accurately detecting anomalies.

Figure 2.5 presented the overview diagram of their proposed method.

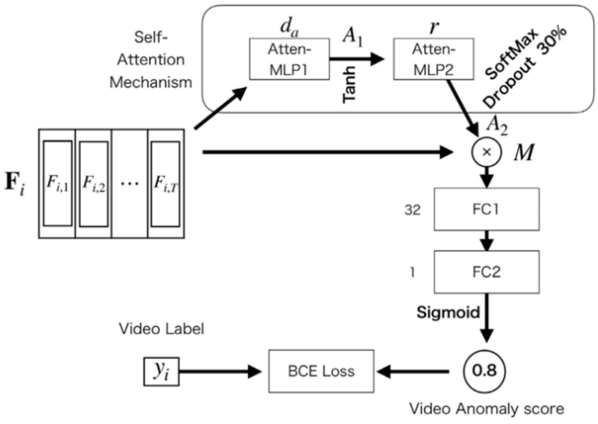
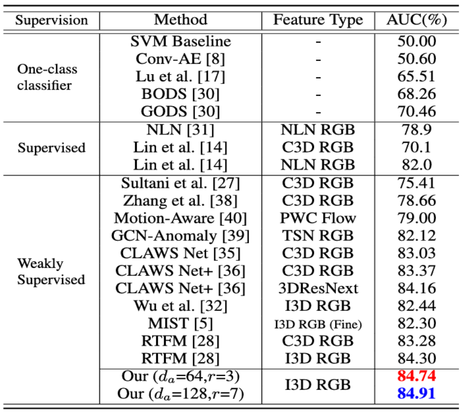


Figure 2.5 Overview diagram of proposed method [5]

The method analysed the entire video and divided it into T segments, with each segment being transformed into a D-dimensional feature vector $F_{i,j}$ using Inflated 3D ConvNet (I3D), which had been trained on the Kineticts dataset. The extracted feature vectors were then fed into the self-attention mechanism, which was made up of an MLP with two layers. An attention map was generated as a result of the self-attention mechanism, ensuring that only critical features contribute to the final classification.

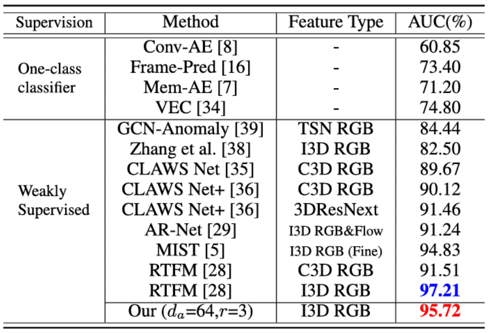
Next, these extracted features were passed through two fully connected layers to compute the anomaly score. The first layer (FC1) applies a transformation to the extracted features, while the second layer (FC2) utilises a sigmoid activation function to classify the video as either normal or abnormal. The binary cross-entropy (BCE) loss function is used to optimise the model during training. Moreover, it was notable that their self-attention mechanism was not dependent on temporal order, where shuffling the input feature vectors would not change the video anomaly score. This characteristic demonstrated that temporal order between segments is not essential for accurate anomaly detection.

The proposed method was evaluated on three widely used benchmark datasets: UCF-Crime, ShanghaiTech and XD-Violence. The comparison between the results of their proposed method and existing methods is listed in table 2.4, table 2.5 and table 2.6 respectively. The blue colour highlighted value is the highest value while the red colour highlighted value is the second highest value.



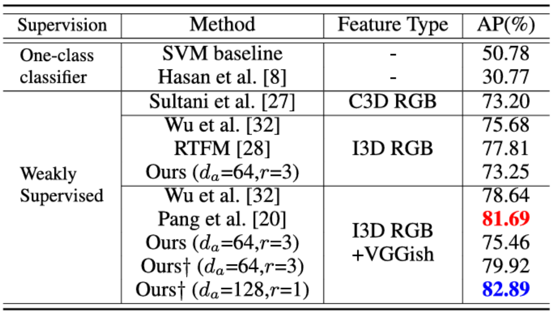
Supervision	Method	Feature Type	AUC(%)
One-class classifier	SVM Baseline	-	50.00
	Conv-AE [8]	-	50.60
	Lu et al. [17]	-	65.51
	BODS [30]	-	68.26
	GODS [30]	-	70.46
Supervised	NLN [31]	NLN RGB	78.9
	Lin et al. [14]	C3D RGB	70.1
	Lin et al. [14]	NLN RGB	82.0
Weakly Supervised	Sultani et al. [27]	C3D RGB	75.41
	Zhang et al. [38]	C3D RGB	78.66
	Motion-Aware [40]	PWC Flow	79.00
	GCN-Anomaly [39]	TSN RGB	82.12
	CLAWS Net [35]	C3D RGB	83.03
	CLAWS Net+ [36]	C3D RGB	83.37
	CLAWS Net+ [36]	3DResNext	84.16
	Wu et al. [32]	I3D RGB	82.44
	MIST [5]	I3D RGB (Fine)	82.30
	RTFM [28]	C3D RGB	83.28
	RTFM [28]	I3D RGB	84.30
	Our (d_a=64,r=3)	I3D RGB	**84.74**
	Our (d_a=128,r=7)		**84.91**

Table 2.4 Comparison of frame-level AUC performance on the UCF-Crime dataset [5]



Supervision	Method	Feature Type	AUC(%)
One-class classifier	Conv-AE [8]	-	60.85
	Frame-Pred [16]	-	73.40
	Mem-AE [7]	-	71.20
	VEC [34]	-	74.80
Weakly Supervised	GCN-Anomaly [39]	TSN RGB	84.44
	Zhang et al. [38]	I3D RGB	82.50
	CLAWS Net [35]	C3D RGB	89.67
	CLAWS Net+ [36]	C3D RGB	90.12
	CLAWS Net+ [36]	3DResNext	91.46
	AR-Net [29]	I3D RGB&Flow	91.24
	MIST [5]	I3D RGB (Fine)	94.83
	RTFM [28]	C3D RGB	91.51
	RTFM [28]	I3D RGB	**97.21**
	Our (d_a=64,r=3)	I3D RGB	**95.72**

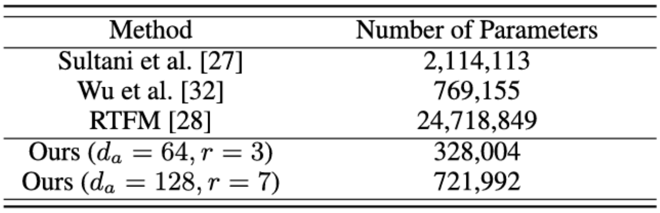
Table 2.5 Comparison of frame-level AUC performance on the ShanghaiTech dataset [5]



Supervision	Method	Feature Type	AP(%)
One-class classifier	SVM baseline	-	50.78
	Hasan et al. [8]	-	30.77
Weakly Supervised	Sultani et al. [27]	C3D RGB	73.20
	Wu et al. [32]	I3D RGB	75.68
	RTFM [28]		77.81
	Ours (d_a=64,r=3)		73.25
	Wu et al. [32]	I3D RGB +VGGish	78.64
	Pang et al. [20]		**81.69**
	Ours (d_a=64,r=3)		75.46
	Ours† (d_a=64,r=3)		79.92
	Ours† (d_a=128,r=1)		**82.89**

Table 2.6 Comparison of frame-level AUC performance on the XD-Violance dataset [5]

Table 2.7 showed the comparison of the number of trainable parameters between proposed methods and existing methods.



Method	Number of Parameters
Sultani et al. [27]	2,114,113
Wu et al. [32]	769,155
RTFM [28]	24,718,849
Ours ($d_a = 64, r = 3$)	328,004
Ours ($d_a = 128, r = 7$)	721,992

Table 2.7 Comparison of the number of trainable parameters [5]

According to table 2.6, it was unarguably true that the proposed approach was better in computational efficiency due to its achievement in higher accuracy while using a smaller number of parameters.

Figure 2.6 illustrated the AUC performance by anomaly classes in UCF-Crime dataset, providing deeper insights into the model’s effectiveness in detecting different types of anomalies.

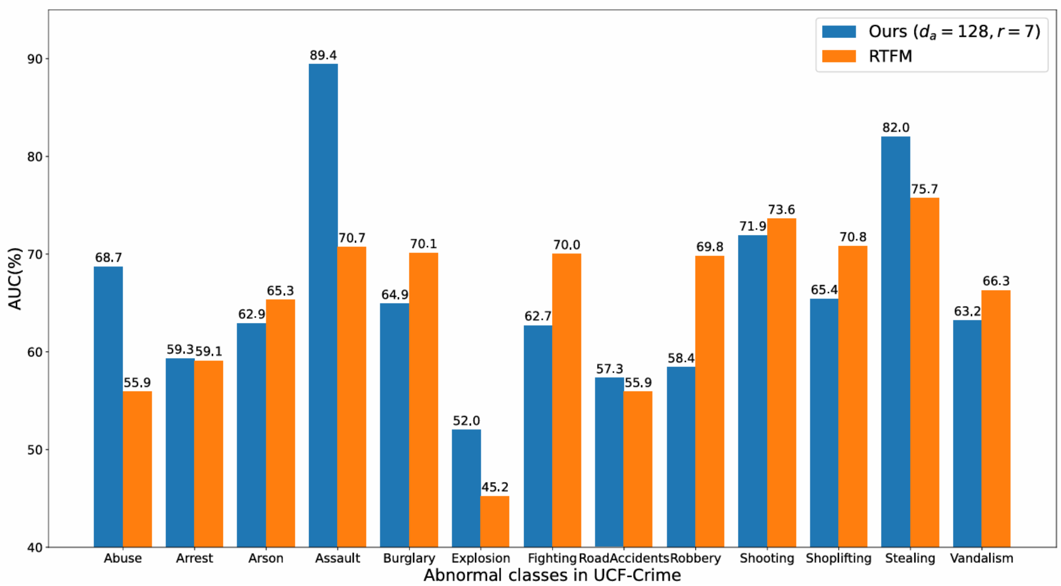


Figure 2.6 AUC performance by anomaly classes in UCF-Crime dataset [5]

Based on figure 2.6, the proposed method outperformed on long-duration anomalies (e.g. Abuse, Arrest, Assault, Explosion, Road Accidents, Stealing) but struggles with short-duration anomalies (e.g. Burglary, Fighting, Robbery, and Shoplifting). Nonetheless, the AUC for explosion detection (52%) suggests that the model struggles to distinguish explosion-related anomalies from normal events, highlighting a potential weakness in feature extraction or classification.

2.1.4 Learning Causal Temporal Relation and Feature Discrimination for Anomaly Detection

Wu and Liu [6] proposed a weakly supervised anomaly detection method that leveraged two ignored critical factors, temporal cue and feature discrimination. They declared that the absence of utilisation of these two key factors in previous methods led to the undesired performance in anomaly detection.

In order to boost the performance of their proposed system, they introduced four key modules to enhance feature representation and discrimination. First, the causal temporal relation (CTR) module helped in capturing local-range temporal dependencies, enabling the model to learn meaningful relationships across consecutive frames. It used a temporal attention mechanism to aggregate useful information from historical and current features, ensuring that no future information was leaked. This is further reinforced with the application of the Classifier (CL) module, projecting the enhanced features into a category space using a causal convolution operation. Additionally, the CL was used to compute the final anomaly score after collecting the needed information.

The discriminative power of the proposed system was aided with two auxiliary modules, the compactness (CP) module and the dispersion (DP) module. The CP modules pulled normal features closer to their centres so they could remain tightly grouped, reducing intraclass variation. On the other hand, the DP module enhanced intraclass dispersion by separating normal and anomalous features, thus increasing the separability between normal and abnormal regions. This enhanced the model's discriminative power. Figure 2.7 depicted the principles of CP and DP modules, where the CP module attempted to compress normal features (NFs) while the DP module pushed anomalous-near features (ANFs) to the middle of NFs and moved anomalous-away features (AAFs) from the middle. So, this mechanism made it easier to find anomalies because it reduced variation within classes for NFs and increased separation between classes for anomalies (ANF and AAF).

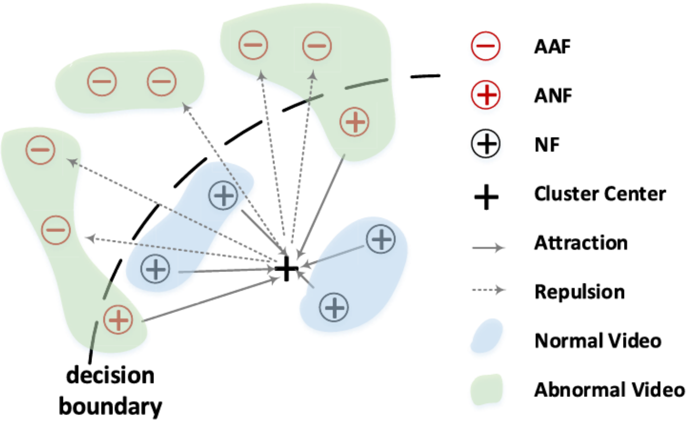
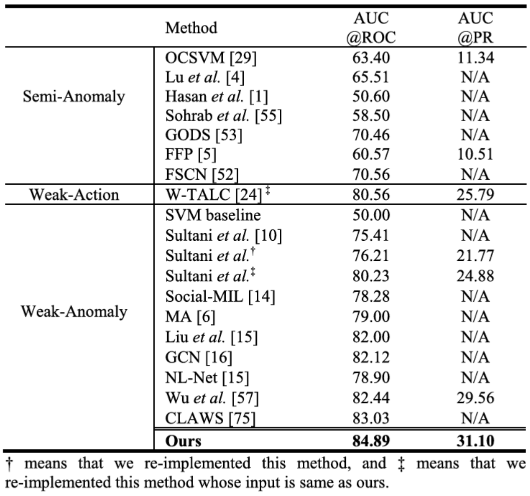


Figure 2.7 Principles of CP and DP modules [6]

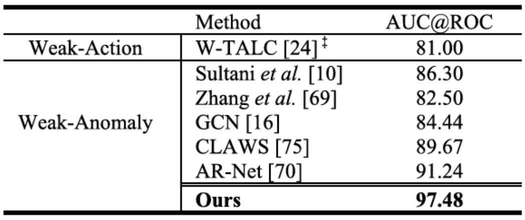
The performance comparison between proposed method and existing methods on UCF-Crime dataset, ShanghaiTech dataset and XD-Violence dataset were revealed in table 2.8, table 2.9 and table 2.10 respectively.



	Method	AUC @ROC	AUC @PR
Semi-Anomaly	OCSVM [29]	63.40	11.34
	Lu *et al.* [4]	65.51	N/A
	Hasan *et al.* [1]	50.60	N/A
	Sohrab *et al.* [55]	58.50	N/A
	GODS [53]	70.46	N/A
	FFP [5]	60.57	10.51
	FSCN [52]	70.56	N/A
Weak-Action	W-TALC [24]‡	80.56	25.79
Weak-Anomaly	SVM baseline	50.00	N/A
	Sultani *et al.* [10]	75.41	N/A
	Sultani *et al.*†	76.21	21.77
	Sultani *et al.*‡	80.23	24.88
	Social-MIL [14]	78.28	N/A
	MA [6]	79.00	N/A
	Liu *et al.* [15]	82.00	N/A
	GCN [16]	82.12	N/A
	NL-Net [15]	78.90	N/A
	Wu *et al.* [57]	82.44	29.56
	CLAWS [75]	83.03	N/A
	Ours	**84.89**	**31.10**

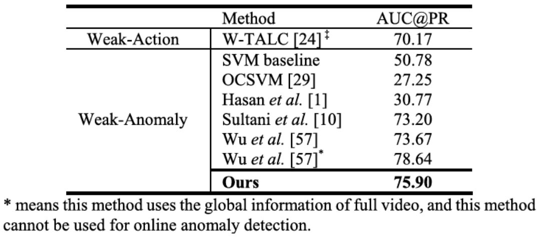
† means that we re-implemented this method, and ‡ means that we re-implemented this method whose input is same as ours.

Table 2.8 Performance comparison between proposed method and existing methods on the UCF-Crime dataset [6]



	Method	AUC@ROC
Weak-Action	W-TALC [24]‡	81.00
Weak-Anomaly	Sultani *et al.* [10]	86.30
	Zhang *et al.* [69]	82.50
	GCN [16]	84.44
	CLAWS [75]	89.67
	AR-Net [70]	91.24
	Ours	**97.48**

Table 2.9 Performance comparison between proposed method and existing methods on the ShanghaiTech dataset [6]



	Method	AUC@PR
Weak-Action	W-TALC [24]‡	70.17
Weak-Anomaly	SVM baseline	50.78
	OCSVM [29]	27.25
	Hasan *et al.* [1]	30.77
	Sultani *et al.* [10]	73.20
	Wu *et al.* [57]	73.67
	Wu *et al.* [57]*	78.64
	Ours	**75.90**

* means this method uses the global information of full video, and this method cannot be used for online anomaly detection.

Table 2.10 Performance comparison between proposed method and existing methods on the XD-Violence dataset [6]

Accordance with the results, the proposed method outperformed all existing methods on the UCF-Crime dataset and ShanghaiTech dataset with achievement of AUC@ROC 84.89% and 97.48%, respectively. However, it trailed the best existing approach by approximately 3%. on the XD-Violence dataset.

In summary, the authors demonstrated a novel method for anomaly detection that leverages causal temporal dependencies and feature discrimination, significantly enhancing weakly supervised anomaly detection significantly.

2.1.5 Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning

Tian et al. [7] introduced a novel method for weakly supervised video anomaly detection called Robust Temporal Feature Magnitude Learning (RTFM). Its goal is to make it easier to find abnormal snippets in abnormal videos within MIL framework. Traditional MIL-based approaches' ability was limited because of failing to accurately identify subtle anomalies due to the dominance of normal snippets in abnormal videos, increasing the difficulty to classify anomalies properly. Meanwhile, RTFM utilised dilated convolutions and self-attention mechanisms to capture both long- and short-range temporal relationships to have a more robust understanding of feature magnitude for improved anomaly detection.

Figure 2.8 displayed the workflow the proposed RTFM method.

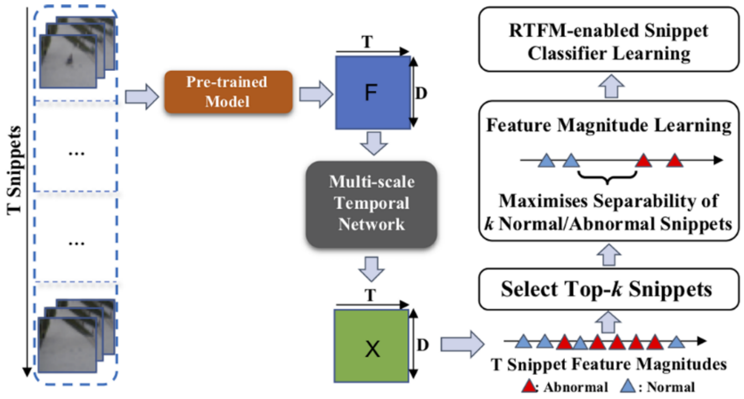
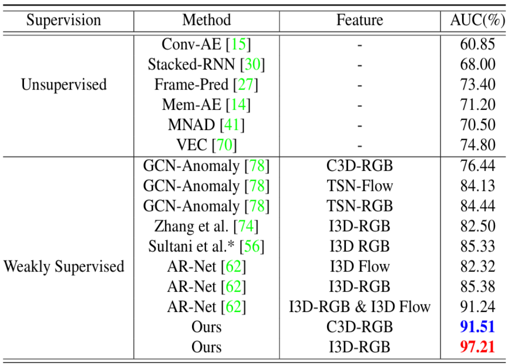

T Snippets
...
...
Pre-trained Model
T
F
D
Multi-scale Temporal Network
T
X
D
RTFM-enabled Snippet Classifier Learning
Feature Magnitude Learning
Maximises Separability of k Normal/Abnormal Snippets
Select Top-k Snippets
T Snippet Feature Magnitudes
: Abnormal
: Normal

Figure 2.8 Workflow of the proposed RTFM method [7]

First of all, T video snippets that are extracted from input videos were fed to a pre-trained model to create a feature representation, denoted as F (a feature matrix with dimensions T × D, where T is the number of snippets and D is the feature dimension). Next, F passed through a multi-scale temporal network, which utilised dilated convolutions and self-attention mechanisms to capture both short- and long-range temporal dependencies, producing an enhanced feature representation X.

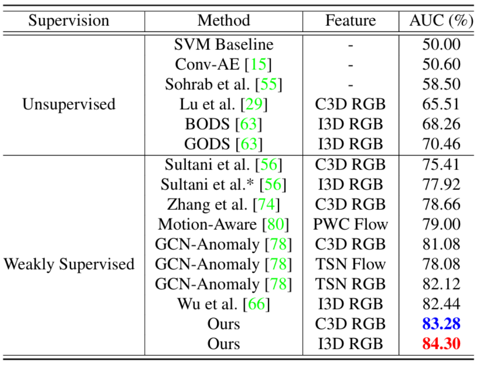
RTFM then selected Top-k snippets based on feature magnitude with an assumption of abnormal snippets exhibiting higher feature magnitudes than normal ones. The selected snippets were further utilised in the feature magnitude learning step, maximising the separability between normal and abnormal snippets. Lastly, this refined feature representation was fed into the RTFM-enabled snippet classifier learning module to enhance the anomaly classification ability by leveraging the learnt feature magnitudes so it could distinguish between normal and abnormal video snippets accurately.

The authors evaluated their model on 4 datasets: ShanghaiTech, UCF-Crime, XD-Violence, and UCSD-Peds. The results were listed in table 2.11, table 2.12, 2.13, 2.14 respectively and the blue colour highlighted value is the highest value while the red colour highlighted value is the second highest value.



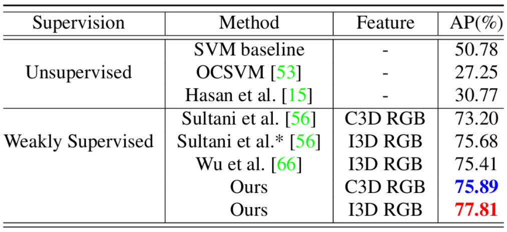
Supervision	Method	Feature	AUC(%)
Unsupervised	Conv-AE [15]	-	60.85
	Stacked-RNN [30]	-	68.00
	Frame-Pred [27]	-	73.40
	Mem-AE [14]	-	71.20
	MNAD [41]	-	70.50
	VEC [70]	-	74.80
Weakly Supervised	GCN-Anomaly [78]	C3D-RGB	76.44
	GCN-Anomaly [78]	TSN-Flow	84.13
	GCN-Anomaly [78]	TSN-RGB	84.44
	Zhang et al. [74]	I3D-RGB	82.50
	Sultani et al.* [56]	I3D RGB	85.33
	AR-Net [62]	I3D Flow	82.32
	AR-Net [62]	I3D-RGB	85.38
	AR-Net [62]	I3D-RGB & I3D Flow	91.24
	Ours	C3D-RGB	**91.51**
	Ours	I3D-RGB	**97.21**

Table 2.11 Comparison of frame-level AUC performance on ShanghaiTech dataset [7]



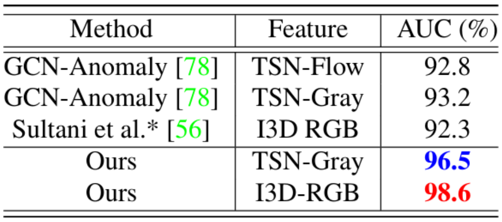
Supervision	Method	Feature	AUC (%)
Unsupervised	SVM Baseline	-	50.00
	Conv-AE [15]	-	50.60
	Sohrab et al. [55]	-	58.50
	Lu et al. [29]	C3D RGB	65.51
	BODS [63]	I3D RGB	68.26
	GODS [63]	I3D RGB	70.46
Weakly Supervised	Sultani et al. [56]	C3D RGB	75.41
	Sultani et al.* [56]	I3D RGB	77.92
	Zhang et al. [74]	C3D RGB	78.66
	Motion-Aware [80]	PWC Flow	79.00
	GCN-Anomaly [78]	C3D RGB	81.08
	GCN-Anomaly [78]	TSN Flow	78.08
	GCN-Anomaly [78]	TSN RGB	82.12
	Wu et al. [66]	I3D RGB	82.44
	Ours	C3D RGB	**83.28**
	Ours	I3D RGB	**84.30**

Table 2.12 Comparison of frame-level AUC performance on UCF-Crime dataset [7]



Supervision	Method	Feature	AP(%)
Unsupervised	SVM baseline	-	50.78
	OCSVM [53]	-	27.25
	Hasan et al. [15]	-	30.77
Weakly Supervised	Sultani et al. [56]	C3D RGB	73.20
	Sultani et al.* [56]	I3D RGB	75.68
	Wu et al. [66]	I3D RGB	75.41
	Ours	C3D RGB	**75.89**
	Ours	I3D RGB	**77.81**

Table 2.13 Comparison of frame-level AUC performance on XD-Violence dataset [7]



Method	Feature	AUC (%)
GCN-Anomaly [78]	TSN-Flow	92.8
GCN-Anomaly [78]	TSN-Gray	93.2
Sultani et al.* [56]	I3D RGB	92.3
Ours	TSN-Gray	**96.5**
Ours	I3D-RGB	**98.6**

Table 2.14 Comparison of frame-level AUC performance on UCSD-Peds dataset [7]

Based on the results, it was evident that the proposed RTFM method outperformed all state-of-the-art approaches in weakly supervised video anomaly detection. The model achieved a higher AUC score across 4 benchmark datasets, proving its capabilities in classifying abnormal events from normal activities.

Figure 2.9 illustrated the AUC result with respect to individual classes on the UCF-Crime dataset.

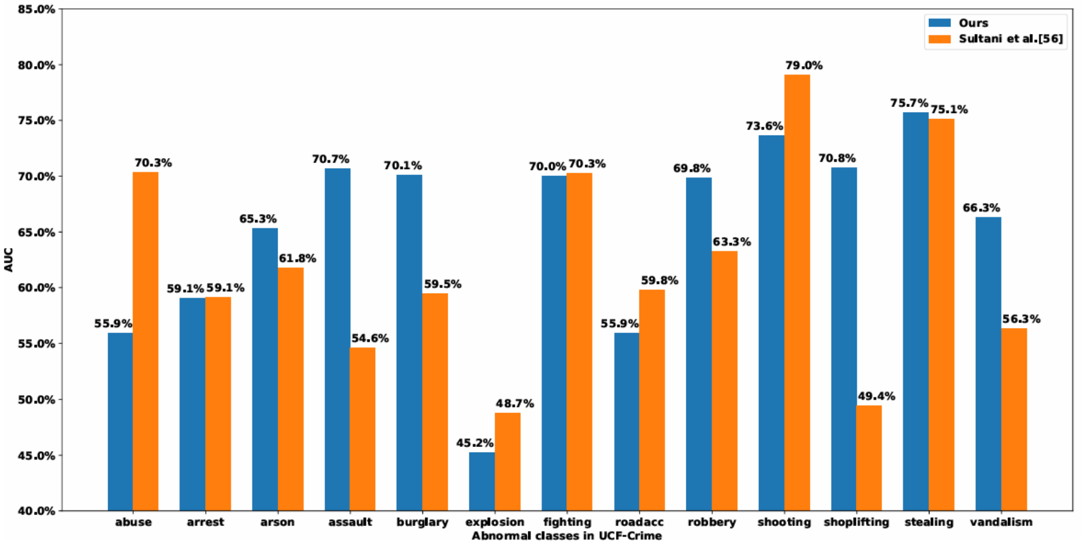


Figure 2.9 AUC results with respect to individual classes on the UCF-Crime dataset [7]

However, the proposed method still achieved a low AUC value on the explosion category on UCF-Crime according to figure 2.9.

2.1.6 Learning Prompt-Enhanced Context Features for Weakly-Supervised Video Anomaly Detection

Pu et al. [8] proposed a novel framework for weakly supervised video anomaly detection to address two key limitations, which were high computational cost and the lack of fine-grained discriminability within anomalous classes. They aimed to improve temporal modelling efficiency and anomaly class discriminability with the application of two innovative modules, the temporal context aggregation (TCA) module and the prompt-enhanced learning (PEL) module.

Figure 2.10 presented the overview of the proposed framework.

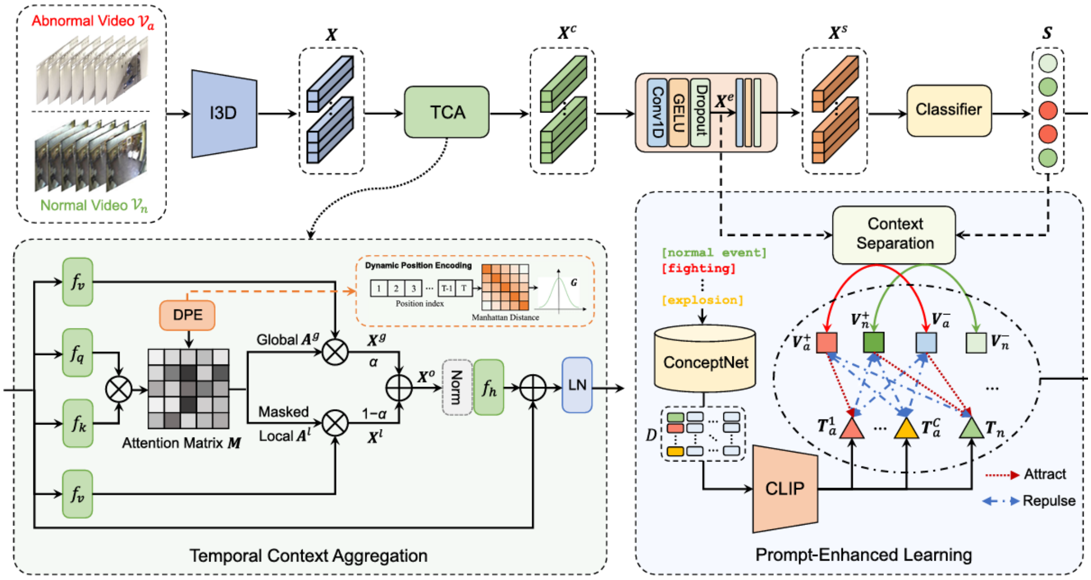


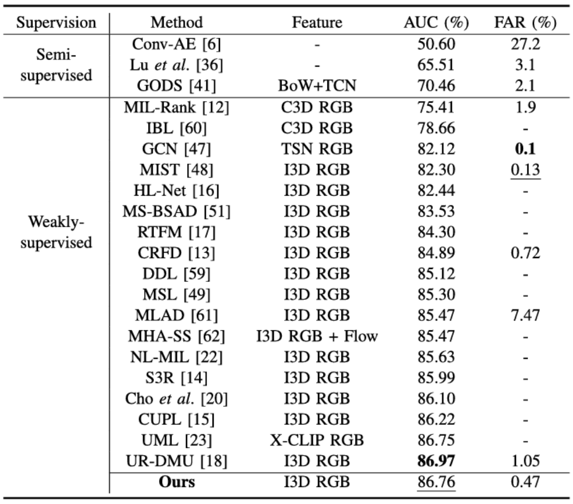
Figure 2.10 Overview of the proposed framework [8]

The system began by inputting a sequence of untrimmed video frames, containing either normal or abnormal events. These frames were then processed using I3D to extract features from the video frames. The extracted features were denoted as X, then fed into TCA module for further processing.

The TCA module mainly focused on capturing both local and global dependencies. Additionally, it employed dynamic position encoding (DPE), which uses Manhattan Distance to represent the positional relationships between frames dynamically. This helped in easing the long-range noise that appeared in anomaly detection. Lastly, the TCA module computed Global Attention (A^g) and Masked Local Attention (A^l). Ag captured long-range dependencies in the video sequence while A^l focused on short-term dependencies to retain fine-grained motion information. A^g and A^l were then fused, producing a refined feature representation to be used in PEL module later.

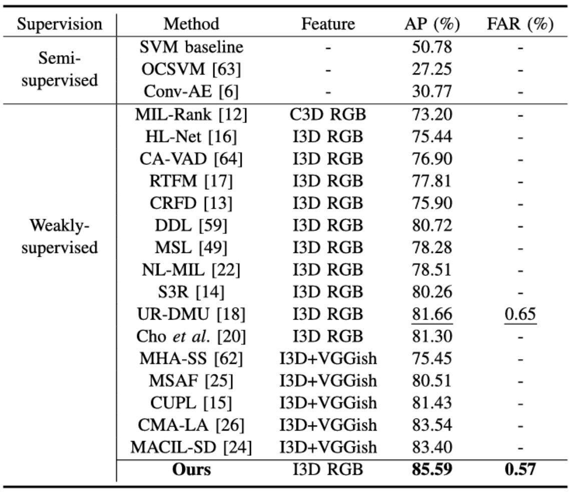
Within the PEL module, it leveraged ConceptNet, a structured knowledge graph containing relationships between different concepts. Anomaly-related prompts such as "fighting" or "explosion" were extracted from ConceptNet and processed using Contrastive Language-Image Pretraining (CLIP) to generate feature representations. Next, the module applied the context separation to differentiate normal and abnormal contexts, enhancing system's ability in discrimination. Finally, the Classifier processed the refined and enriched features to generate an anomaly score(S), indicating whether a video segment is normal or anomalous.

The performance comparison between proposed method and existing methods on UCF-Crime dataset, XD-Violence dataset and ShanghaiTech dataset were revealed in table 2.15, table 2.16 and table 2.17 respectively.



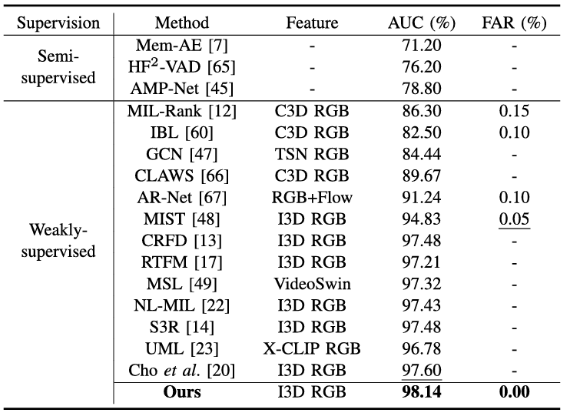
Supervision	Method	Feature	AUC (%)	FAR (%)
Semi-supervised	Conv-AE [6]	-	50.60	27.2
	Lu *et al.* [36]	-	65.51	3.1
	GODS [41]	BoW+TCN	70.46	2.1
Weakly-supervised	MIL-Rank [12]	C3D RGB	75.41	1.9
	IBL [60]	C3D RGB	78.66	-
	GCN [47]	TSN RGB	82.12	**0.1**
	MIST [48]	I3D RGB	82.30	0.13
	HL-Net [16]	I3D RGB	82.44	-
	MS-BSAD [51]	I3D RGB	83.53	-
	RTFM [17]	I3D RGB	84.30	-
	CRFD [13]	I3D RGB	84.89	0.72
	DDL [59]	I3D RGB	85.12	-
	MSL [49]	I3D RGB	85.30	-
	MLAD [61]	I3D RGB	85.47	7.47
	MHA-SS [62]	I3D RGB + Flow	85.47	-
	NL-MIL [22]	I3D RGB	85.63	-
	S3R [14]	I3D RGB	85.99	-
	Cho *et al.* [20]	I3D RGB	86.10	-
	CUPL [15]	I3D RGB	86.22	-
	UML [23]	X-CLIP RGB	86.75	-
	UR-DMU [18]	I3D RGB	**86.97**	1.05
	Ours	I3D RGB	86.76	0.47

Table 2.15 Performance comparison between proposed method and existing methods on the UCF-Crime dataset [8]



Supervision	Method	Feature	AP (%)	FAR (%)
Semi-supervised	SVM baseline	-	50.78	-
	OCSVM [63]	-	27.25	-
	Conv-AE [6]	-	30.77	-
Weakly-supervised	MIL-Rank [12]	C3D RGB	73.20	-
	HL-Net [16]	I3D RGB	75.44	-
	CA-VAD [64]	I3D RGB	76.90	-
	RTFM [17]	I3D RGB	77.81	-
	CRFD [13]	I3D RGB	75.90	-
	DDL [59]	I3D RGB	80.72	-
	MSL [49]	I3D RGB	78.28	-
	NL-MIL [22]	I3D RGB	78.51	-
	S3R [14]	I3D RGB	80.26	-
	UR-DMU [18]	I3D RGB	81.66	0.65
	Cho *et al.* [20]	I3D RGB	81.30	-
	MHA-SS [62]	I3D+VGGish	75.45	-
	MSAF [25]	I3D+VGGish	80.51	-
	CUPL [15]	I3D+VGGish	81.43	-
	CMA-LA [26]	I3D+VGGish	83.54	-
	MACIL-SD [24]	I3D+VGGish	83.40	-
	Ours	I3D RGB	**85.59**	**0.57**

Table 2.16 Performance comparison between proposed method and existing methods on the XD-Violence dataset [8]



Supervision	Method	Feature	AUC (%)	FAR (%)
Semi-supervised	Mem-AE [7]	-	71.20	-
	HF^2-VAD [65]	-	76.20	-
	AMP-Net [45]	-	78.80	-
Weakly-supervised	MIL-Rank [12]	C3D RGB	86.30	0.15
	IBL [60]	C3D RGB	82.50	0.10
	GCN [47]	TSN RGB	84.44	-
	CLAWS [66]	C3D RGB	89.67	-
	AR-Net [67]	RGB+Flow	91.24	0.10
	MIST [48]	I3D RGB	94.83	0.05
	CRFD [13]	I3D RGB	97.48	-
	RTFM [17]	I3D RGB	97.21	-
	MSL [49]	VideoSwin	97.32	-
	NL-MIL [22]	I3D RGB	97.43	-
	S3R [14]	I3D RGB	97.48	-
	UML [23]	X-CLIP RGB	96.78	-
	Cho *et al.* [20]	I3D RGB	97.60	-
	Ours	I3D RGB	**98.14**	**0.00**

Table 2.17 Performance comparison between proposed method and existing methods on the ShanghaiTech dataset [8]

The proposed model achieved superior performance on both XD-Violence and ShanghaiTech datasets, surpassing all the existing methods in terms of average precision (AP), AUC and false alarm rate (FAR). However, despite its strong performance, the proposed model failed to surpass UR-DMU in all aspects, ranking as the second-best method behind UR-DMU. In detail, the proposed method fell behind the best method by 0.21% in AUC and 0.37% in FAR. Figure 2.11 illustrated the AUC result with respect to individual classes on the UCF-Crime dataset.

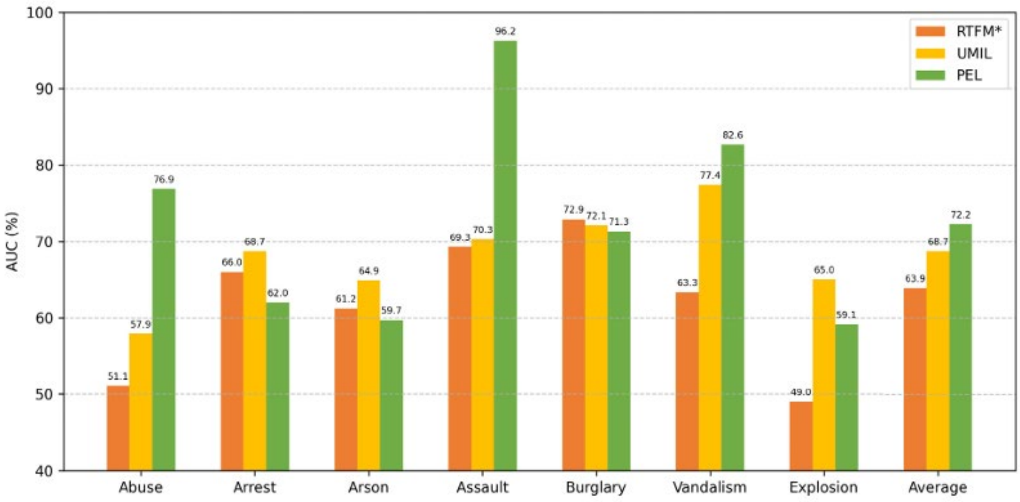


Figure 2.11 AUC results with respect to individual classes on the UCF-Crime dataset [8]

2.1.7 Unbiased Multiple Instance Learning for Weakly Supervised Video Anomaly Detection

Lv et al. [9] proposed a new MIL framework, Unbiased Multiple Instance Learning (UMIL), to improve weakly supervised video anomaly detection. Traditional MIL methods tend to be biased due to focusing only on the most confident normal and abnormal snippets, leading to a high false alarm rate and low accuracy. Hence, UMIL incorporates both confident and ambiguous snippets during training, allowing it to learn anomaly features in an unbiased manner which would improve detection performance.

Figure 2.12 depicted the overview of UMIL framework.

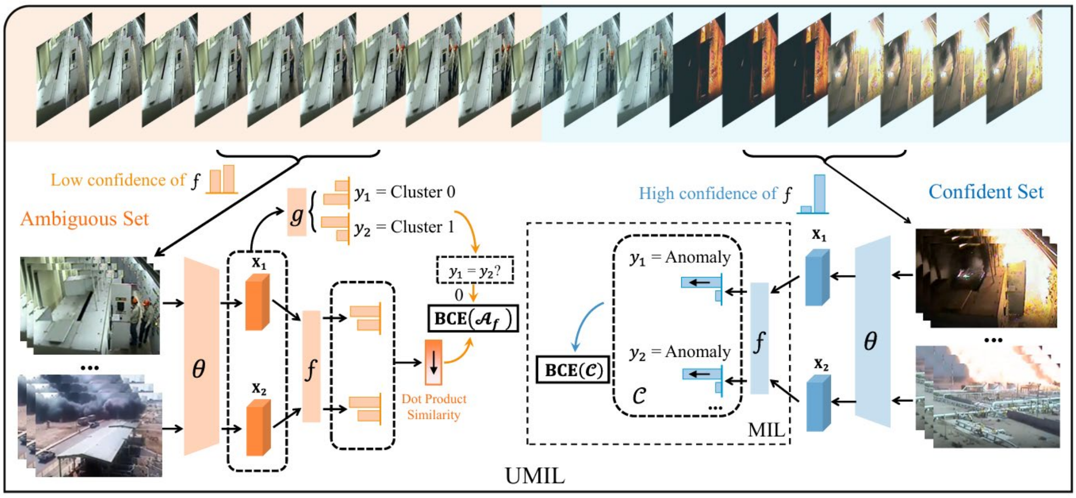


Figure 2.12 Overview of UMIL framework [9]

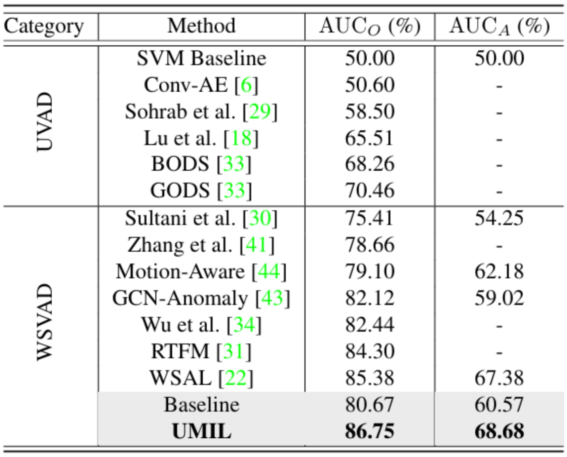
The framework consisted of three main components: a backbone model (θ), an anomaly head (f), and a cluster head (g). θ extracted features from the input video snippets, which were then processed by f to predict an anomaly score. g further separated ambiguous snippets into clusters to refine the anomaly detection process.

First, video snippets were categorised into two sets, a confident set and an ambiguous set. The confident set consisted of snippets with high-confidence anomaly scores, which were directly used in the training process according to traditional MIL principles. In contrast, the ambiguous set contained snippets where the anomaly predictions were uncertain.

For the ambiguous set, g further separated snippets into two clusters, with one likely containing normal snippets and the other containing anomalous snippets. BCE loss (A_f) was applied based on the dot-product similarity between the snippets, assigning distinct anomaly scores to different clusters to reduce context bias while improving the ability to detect subtle anomalies.

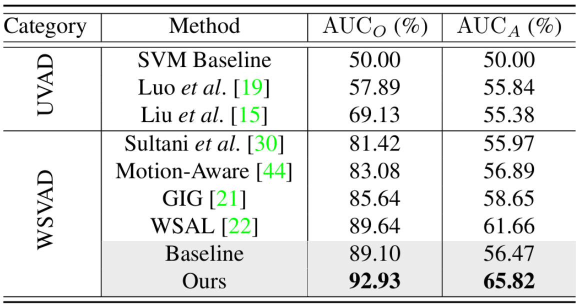
On the other hand, f was trained using a standard MIL framework with the BCE loss (C) to the confident set. The black arrows in the probability bar indicated that the UMIL increased the confidence of anomaly prediction over time. Precisely, UMIL enhanced detection accuracy by learning from both confident and ambiguous instances, distinguishing it from traditional MIL methods that only focused on high-confidence snippets.

The performance comparison between proposed method and existing methods on UCF-Crime dataset and TAD dataset were presented in table 2.18 and table 2.19 respectively.



Category	Method	AUC$_O$ (%)	AUC$_A$ (%)
UVAD	SVM Baseline	50.00	50.00
	Conv-AE [6]	50.60	-
	Sohrab et al. [29]	58.50	-
	Lu et al. [18]	65.51	-
	BODS [33]	68.26	-
	GODS [33]	70.46	-
WSVAD	Sultani et al. [30]	75.41	54.25
	Zhang et al. [41]	78.66	-
	Motion-Aware [44]	79.10	62.18
	GCN-Anomaly [43]	82.12	59.02
	Wu et al. [34]	82.44	-
	RTFM [31]	84.30	-
	WSAL [22]	85.38	67.38
	Baseline	80.67	60.57
	UMIL	**86.75**	**68.68**

Table 2.18 Performance comparison between proposed method and existing methods on the UCF-Crime dataset [9]



Category	Method	AUC$_O$ (%)	AUC$_A$ (%)
UVAD	SVM Baseline	50.00	50.00
	Luo *et al.* [19]	57.89	55.84
	Liu *et al.* [15]	69.13	55.38
WSVAD	Sultani *et al.* [30]	81.42	55.97
	Motion-Aware [44]	83.08	56.89
	GIG [21]	85.64	58.65
	WSAL [22]	89.64	61.66
	Baseline	89.10	56.47
	Ours	**92.93**	**65.82**

Table 2.19 Performance comparison between proposed method and existing methods on the TAD dataset [9]

AUC$_O$ represents AUC computed on the entire test set, including both normal and abnormal videos. On the contrary, AUC$_A$ referred to the AUC computed only on abnormal test videos, providing a more focused evaluation of anomaly detection performance.

The proposed method outperformed existing methods on both the UCF-CRIME dataset and TAD dataset due to its better anomaly localisation. UMIL aided in enhancing subtle anomaly detection compared to traditional MIL-based approaches, which were biased. Despite achieving solid overall performance, the effectiveness of UMIL in detecting explosion-related anomalies remains unknown. As shown in Figure 2.13, the AUC$_A$ for the explosion category was only 65%, raising the question of whether further improvements could be made in this specific category.

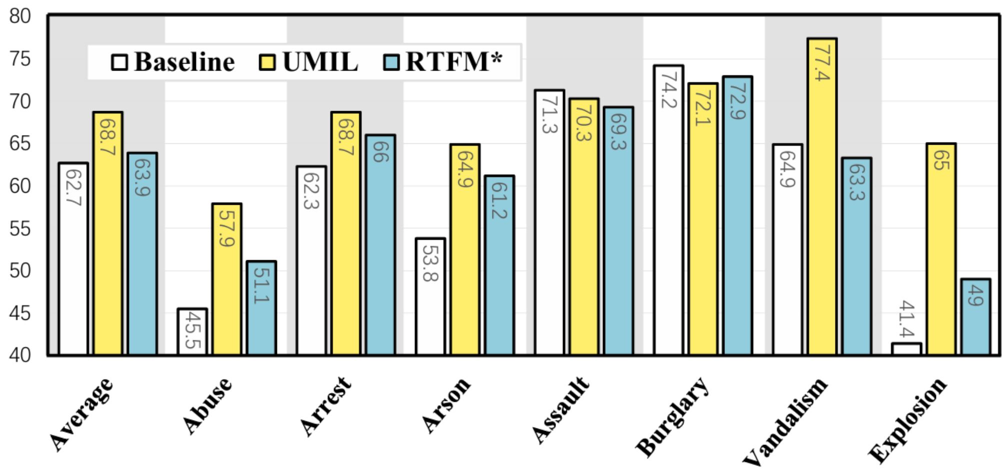


Figure 2.13 Class-wise AUC$_A$ of three methods on UCF-Crime [9]

2.2 Previous work on Explosion Detection

2.2.1 A Novel Vision-Based Classification System for Explosion Phenomena

Abusaleh et al. [10] proposed an innovative vision-based surveillance system for detecting explosion events. The system utilised pattern recognition techniques involving advanced feature extraction methods and a classification strategy. Moreover, supervised learning was employed to effectively separate the data into distinct regions or classes, mapping input frames to specific outputs based on identified patterns.

The system started by preprocessing training data. Firstly, 2D RGB images were resized to a uniform dimension of 64x64 pixels, which facilitated a reduction in computational demands. These images were then converted to greyscale so the system could focus on intensity variations rather than colour, effectively reducing the complexity for the subsequent feature extraction phase.

Feature extraction was conducted using several techniques, including principal component analysis (PCA), YC_bC_r amplitude features in the spatial domain and feature extraction in the transform domain.

Figure 2.14 summarized the steps of the PCA algorithm during training process.

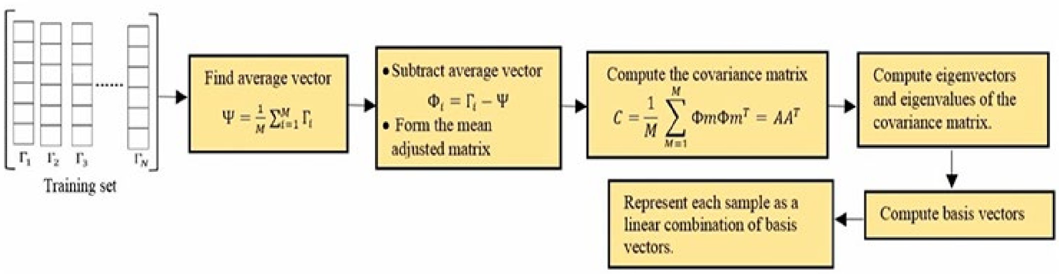

Training set
Γ_1 Γ_2 Γ_3 Γ_N
Find average vector $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$
• Subtract average vector $\Phi_i = \Gamma_i - \Psi$
• Form the mean adjusted matrix
Compute the covariance matrix $C = \frac{1}{M}\sum_{M=1}^{M}\Phi m\Phi m^T = AA^T$
Compute eigenvectors and eigenvalues of the covariance matrix.
Compute basis vectors
Represent each sample as a linear combination of basis vectors.

Figure 2.14 Steps of the PCA algorithm during training process [10]

The PCA algorithm was employed as a mathematical technique to reduce high-dimensional image data into a lower-dimensional space, focusing on the apparent difference between images in the dataset. Initially, each image was represented as a vector Γ_i, and an average vector was computed. This average was then subtracted from each image vector to form a mean-adjusted matrix. The covariance matrix of this adjusted matrix was then calculated, followed by the computation of its eigenvectors and eigenvalues. Next, the system sorted the magnitude of their corresponding eigenvalues, storing only the top 100 eigenvalues to capture the most significant features of the images. These eigenvectors were assembled into an eigenvector matrix and multiplied by the previously calculated mean-adjusted matrix to form the basis vectors. This method ensured that the key features extracted remained the same regardless of translation and illumination, providing a strong foundation for further analysis and classification.

Other than that, the authors acknowledge colour as a significant physical characteristic for differentiating between explosion and non-explosion events, which are influenced by the temperature and composition of each event. Figure 2.15 demonstrated the process of obtaining YC_bC_r amplitude features in the spatial domain.

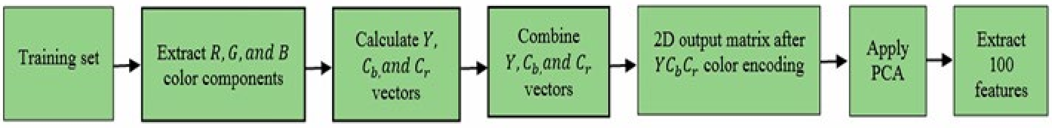

Training set
Extract $R, G, and\ B$ color components
Calculate Y, C_b and C_r vectors
Combine $Y, C_b, and\ C_r$ vectors
2D output matrix after YC_bC_r color encoding
Apply PCA
Extract 100 features

Figure 2.15 Process of obtaining YC_bC_r amplitude features in the spatial domain [10]

The process applied a linear transformation of bitmap pixel component intensities, commonly known as RGB, to the YC_bC_r vector. This method was preferred for its sensitivity to luminance (Y) over chrominance changes (C_b and C_r), focusing on brightness variations rather than color changes. The transformation from RGB to YC_bC_r was defined by the ITU-R BT.601 standard, as shown in Figure 2.16. The YC_bC_r vectors for all images were then aggregated into a 2D output matrix. The PCA algorithm subsequently processed this matrix to extract the 100 most significant eigenvectors, allowing the system to distinguish between different types of explosive events based on their luminance and chrominance characteristics.

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Figure 2.16 Formula for conversion of RGB color space into YC_bC_r color space [10]

Furthermore, features in the transform domain in the images were taken into consideration and extracted using the Radix-2 FFT algorithm illustrated in Figure 2.17. This algorithm transformed spatial-domain images into the frequency domain, decomposing each image into a weighted sum of complex exponential functions termed spectral components.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m}e^{-}\frac{2\pi i}{N}(2m)^{k} + \sum_{m=0}^{N/2-1} x_{2m+1}e^{-}\frac{2\pi i}{N}(2m+1)^{k}$$

Figure 2.17 Radix-2 FFT algorithm [10]

Figure 2.18 presented the process of extracting features in the transform domain in the images.

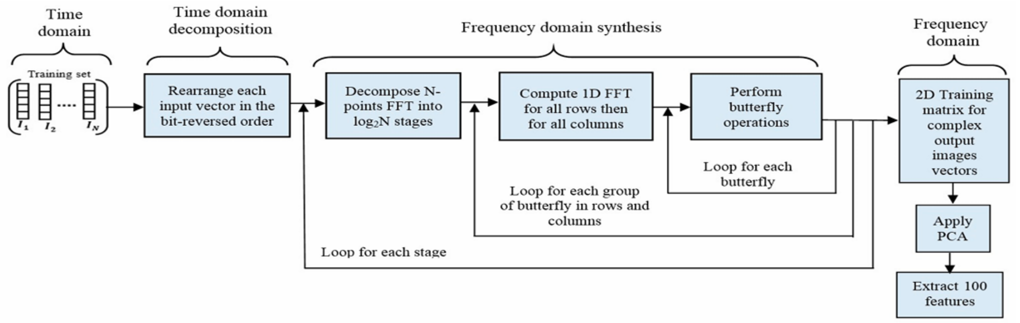
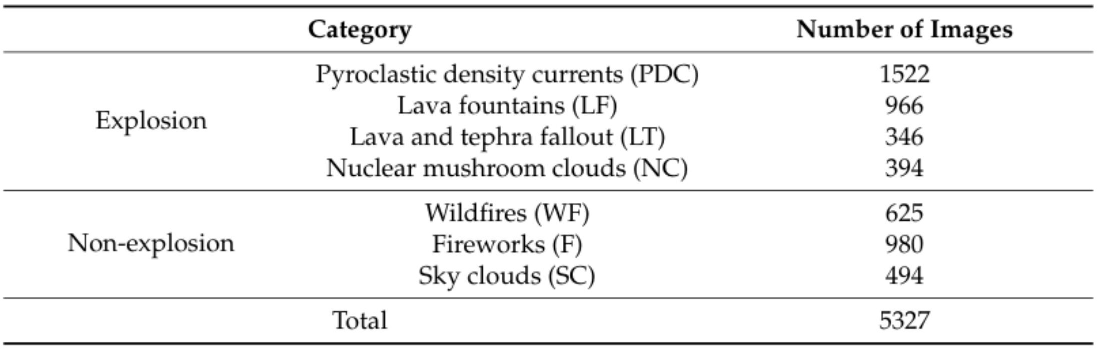

Time domain
Training set
I_1 I_2 I_N
Time domain decomposition
Rearrange each input vector in the bit-reversed order
Frequency domain synthesis
Decompose N-points FFT into log₂N stages
Compute 1D FFT for all rows then for all columns
Perform butterfly operations
Loop for each butterfly
Loop for each group of butterfly in rows and columns
Loop for each stage
Frequency domain
2D Training matrix for complex output images vectors
Apply PCA
Extract 100 features

Figure 2.18 Process of extracting features in the transform domain in the images [10]

The process began with a time-domain decomposition using a bit-reversal sorting algorithm, rearranging the input vectors into a bit-reversed order. Each rearranged input underwent N-point FFT decomposition, broken down into $\log_2N$ stages. Then, 1D FFT was computed for each row and subsequently for each column of the decomposed data. Butterfly operations were then implemented, where the process involved looping through groups of butterfly operations in both rows and columns, combining individual FFT outputs to finalise the frequency domain. Afterward, the model compiled the frequency domain into a 2D training matrix for complex image vectors, subsequently applying PCA to extract the top 100 significant features. With the application of the Radix-2 FFT algorithm, rapid or gradual changes in image intensity could be detected effectively, making it ideal for analysing features contained within image data.

Lastly, supervised classification was performed using a Support Vector Machine (SVM), originally a binary classification method. Nevertheless, it implemented a one-against-one technique with a degree-3 polynomial kernel in this project in order to handle a multi-class categorization. This involved creating individual classifiers for each pair of classes.

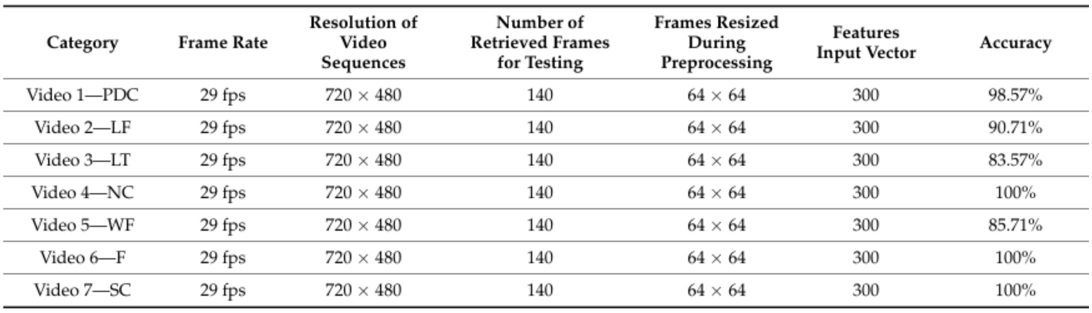
In order to train their system, they advocated the use of 2D RGB images to train their classifier, a strategy aimed at reducing computational costs. The training dataset was made up of four categories of explosion phenomena and three categories of non-explosion phenomena, consisting of a total of 5327 images. Table 2.20 presented the details of the image dataset. Subsequently, videos were used as the testing set to evaluate the system's performance, with each category contributing 140 frames, resulting in a total of 980 frames used for testing purposes.



	Category	Number of Images
Explosion	Pyroclastic density currents (PDC)	1522
	Lava fountains (LF)	966
	Lava and tephra fallout (LT)	346
	Nuclear mushroom clouds (NC)	394
Non-explosion	Wildfires (WF)	625
	Fireworks (F)	980
	Sky clouds (SC)	494
	Total	5327

Table 2.20 Number of images in each category of explosion and non-explosion phenomenon [10]

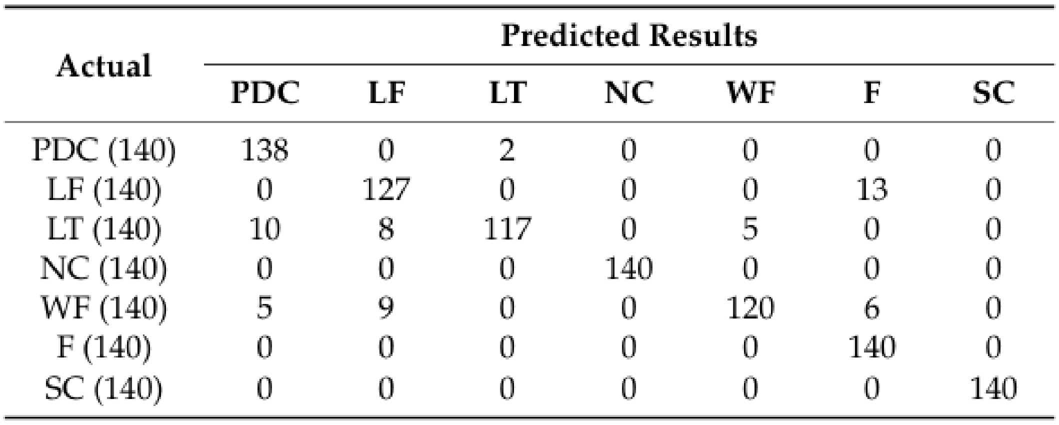
During the testing phase of the classification system, each frame was characterized by 300 features, with each set of 100 features derived from three distinct feature extraction phases. Then, the SVM classifier analysed these combined 300 features, treating them as a single input vector, and assigned the input into a specific category. The proposed vision-based explosion categorization system attained an overall accuracy of 94.08%, with the NC, F, and SC samples being classified correctly with 100% accuracy. This was followed by the categories PDC, LF, WF, and LT, which achieved accuracies of 98.57%, 90.71%, 85.71%, and 83.57%, respectively. Table 2.21 contained the details of the testing set's results.



Category	Frame Rate	Resolution of Video Sequences	Number of Retrieved Frames for Testing	Frames Resized During Preprocessing	Features Input Vector	Accuracy
Video 1—PDC	29 fps	720 × 480	140	64 × 64	300	98.57%
Video 2—LF	29 fps	720 × 480	140	64 × 64	300	90.71%
Video 3—LT	29 fps	720 × 480	140	64 × 64	300	83.57%
Video 4—NC	29 fps	720 × 480	140	64 × 64	300	100%
Video 5—WF	29 fps	720 × 480	140	64 × 64	300	85.71%
Video 6—F	29 fps	720 × 480	140	64 × 64	300	100%
Video 7—SC	29 fps	720 × 480	140	64 × 64	300	100%

Table 2.21 Result of the testing set [10]

Table 2.22 presented the confusion matrix for multi-class degree-3 polynomial kernel SVM classifier, providing a more in-depth result.



Actual	Predicted Results						
	PDC	LF	LT	NC	WF	F	SC
PDC (140)	138	0	2	0	0	0	0
LF (140)	0	127	0	0	0	13	0
LT (140)	10	8	117	0	5	0	0
NC (140)	0	0	0	140	0	0	0
WF (140)	5	9	0	0	120	6	0
F (140)	0	0	0	0	0	140	0
SC (140)	0	0	0	0	0	0	140

Table 2.22 Confusion matrix for multi-class degree-3 polynomial kernel SVM classifier [10]

The classification system's weaknesses were primarily observed through errors due to visual similarities among different phenomena, leading to frequent misclassifications across categories. For instance, the LT category, which was comprising lava and tephra fallout, was frequently confused with PDC, LF, and WF due to the resemblance of its luminous (lava) and non-luminous (tephra) regions to flames and dark smoke, respectively. Furthermore, the presence of flames and smoldering smoke in WF caused them to be misclassified as PDC, LF, or F. These classification errors stemmed from the system’s reliance on visual cues, causing the accuracy to drop in complex scenarios showing multiple similar key features.

2.3 Motion Detection

2.3.1 The Recognition of Human Movement Using Temporal Templates

Bobick and Davis [11] proposed a pioneering approach to human movement analysis and recognition through the development of temporal templates. They conceptualized the motion analysis into two primary categories: where is the motion happening and how is the motion moving. This information was effectively extracted by integrating motion-energy images (MEI) and motion history images (MHI), which together offered a static vector-image representation of motion properties over time at each spatial location within a sequence of images. Figure 2.19 displayed some selected frames from a video of an individual sitting. Despite the blurriness of the frames, they stated that humans can trivially recognize the motion. This observation revealed that the temporal template was crucial in capturing essential motion properties, therefore simplifying the process of activity recognition.

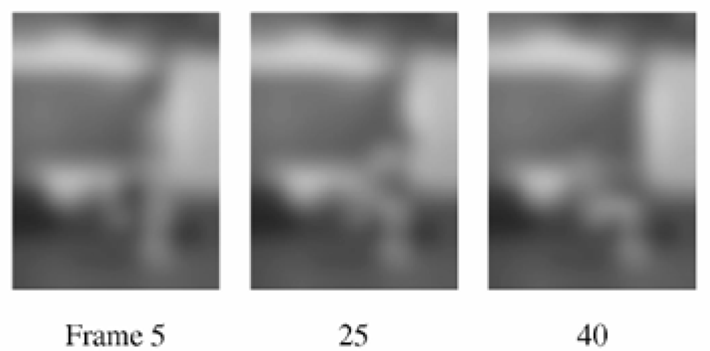


Figure 2.19 Selected frames from a video of an individual sitting [11]

MEI stored cumulative binary motion images which was designed to illustrate where motion occurs within a frame sequence. These images were derived by superimposing binary images that signal regions of motion, captured over the duration of a movement. Figure 2.20 showed an example of MEI of someone sitting, highlighting the comprehensive motion undertaken by the person to sit on a chair.

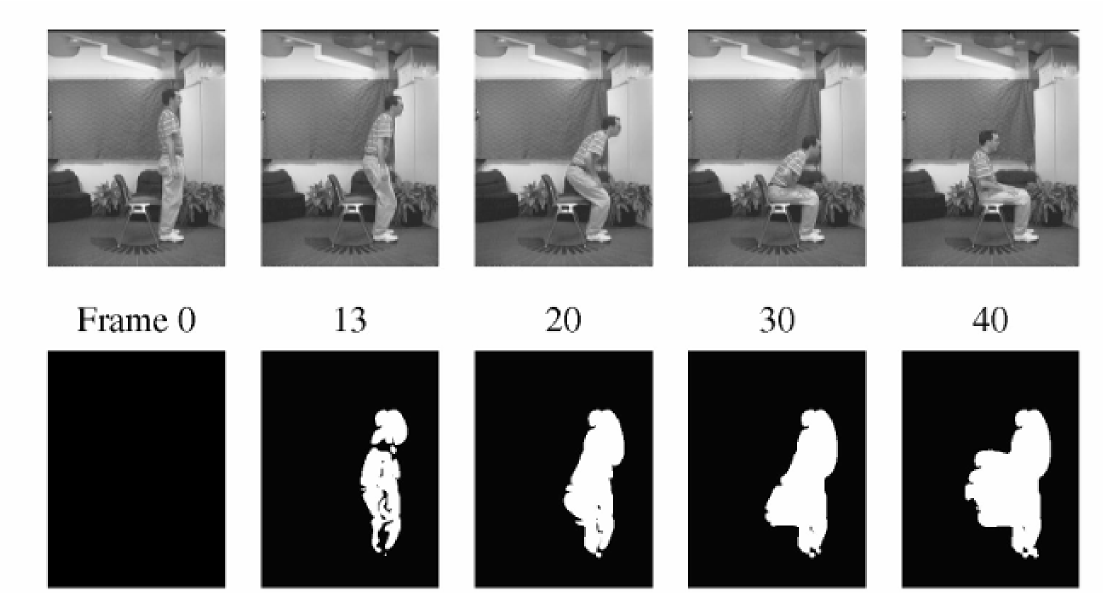


Figure 2.20 Example of MEI of someone sitting [11]

In contrast, MHI was described as tools to depict how motion unfolds over time within the same spatial sequence. MHI differ from MEI by integrating the temporal dimension of movement, where pixel intensity within an MHI was defined as a function of the motion's historical presence at that point. Figure 2.21 depicted simple movements along with their MHIs used in a real time system.

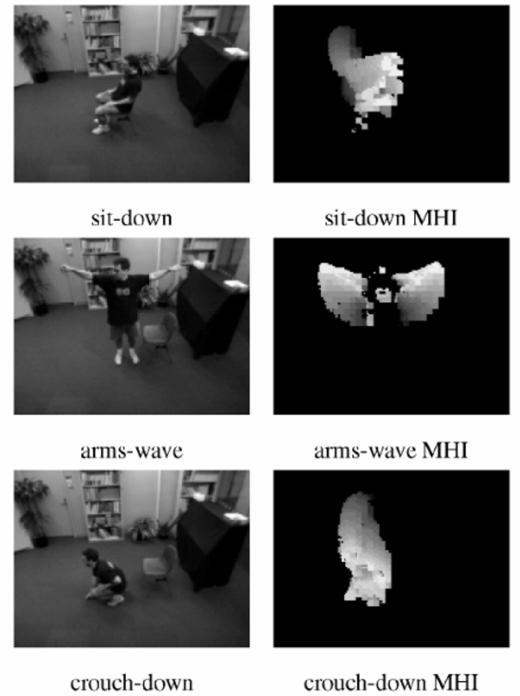


Figure 2.21 Simple movements along with their MHIs used in a real time system [11]

In the experiments, they examine the motion recognition method using a set of 18 aerobic exercises. Figure 2.22 presented a single key frame and MEI from the frontal view of each of the 18 aerobic exercises.

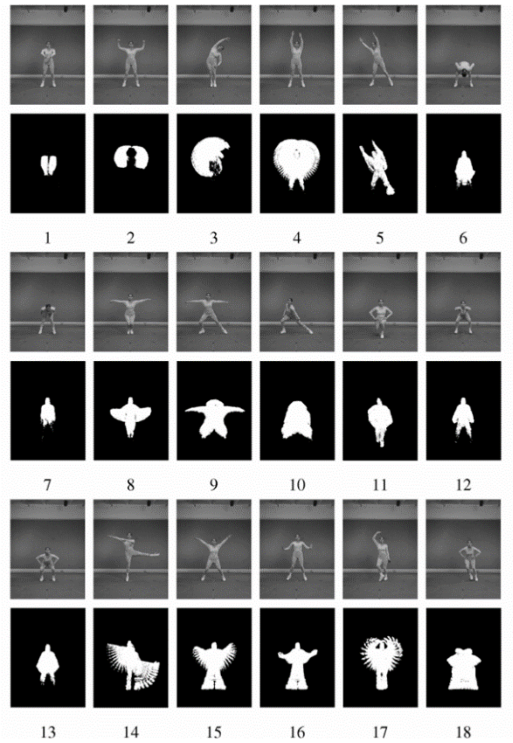


Figure 2.22 A single key frame and MEI from the frontal view of each of the 18 aerobic exercises [11]

Figure 2.23 demonstrates a comparison between MEI and MHI.

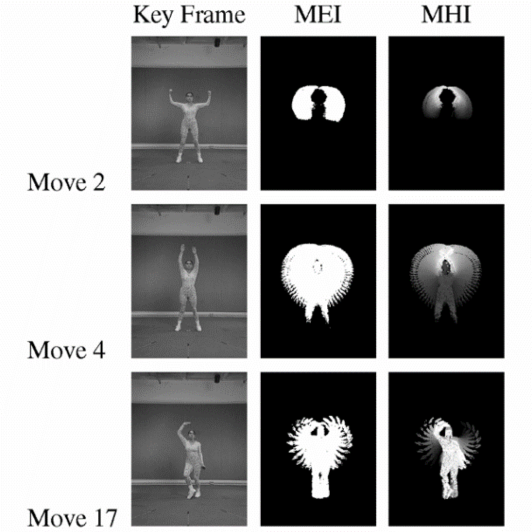


Figure 2.23 A comparison between MEI and MHI [11]

This comparison revealed that MEIs could occasionally cause confusion between specific movements, like moves 4 and 17. Both MEIs exhibited a similar motion, making it easy to mistake them for the same activity. On the contrary, MHI was capable of showing temporal movement, where more recently moving pixels were brighter. The ability of MHI to depict the temporal sequence of movement enables a more detailed and accurate depiction of motion differences, clearly identifying moves 4 and 17. However, the MHIs of moves 2 and 4 appeared similar, which could potentially cause confusion as they represent the same movement. Thus, it was recommended to combine both MEI and MHI in order to provide more robust discrimination. This is because the global shape descriptions in these images are weighted by the pixel values, enhancing the precision in distinguishing between different movements.

2.3.2 Human action recognition using short-time motion energy template images and PCANet features

Abdelbaky and Aly [12] proposed a human action recognition (HAR) method that utilised short-time motion energy image (ST-MEI) templates and PCANet, a deep learning architecture. They mentioned that CNN-based HAR commonly employed either spatial or spatiotemporal information to illustrate actions in video. Nonetheless, this was computationally intensive and primarily designed to process static 2D images. Hence, their proposed method combined the temporal information of video sequences with PCANet, offering a simpler and unsupervised deep learning method. Figure 2.24 depicted the proposed method for HAR using ST-MEI and PCANet.

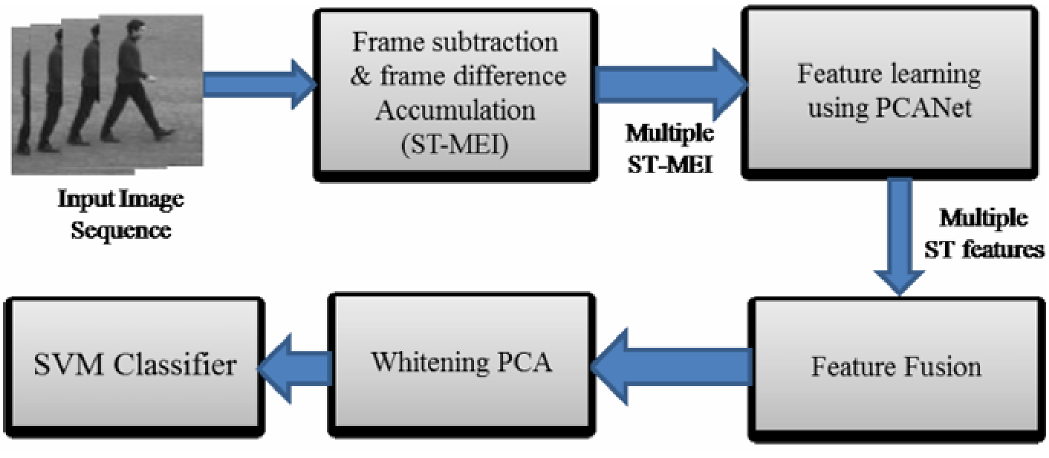


Figure 2.24 Proposed method for HAR using ST-MEI and PCANet [12]

The proposed method began by computing multiple ST-MEI templates for the input video. A simple frame differencing technique was utilised to compute these templates, calculating the differences between consecutive frames to capture the motion occurring in the video sequence. This method effectively represented the motion information at various stages of the action, including the beginning, middle and end. Figure 2.25 presented a sample of walking action with different numbers of durations for each ST-MEI template.

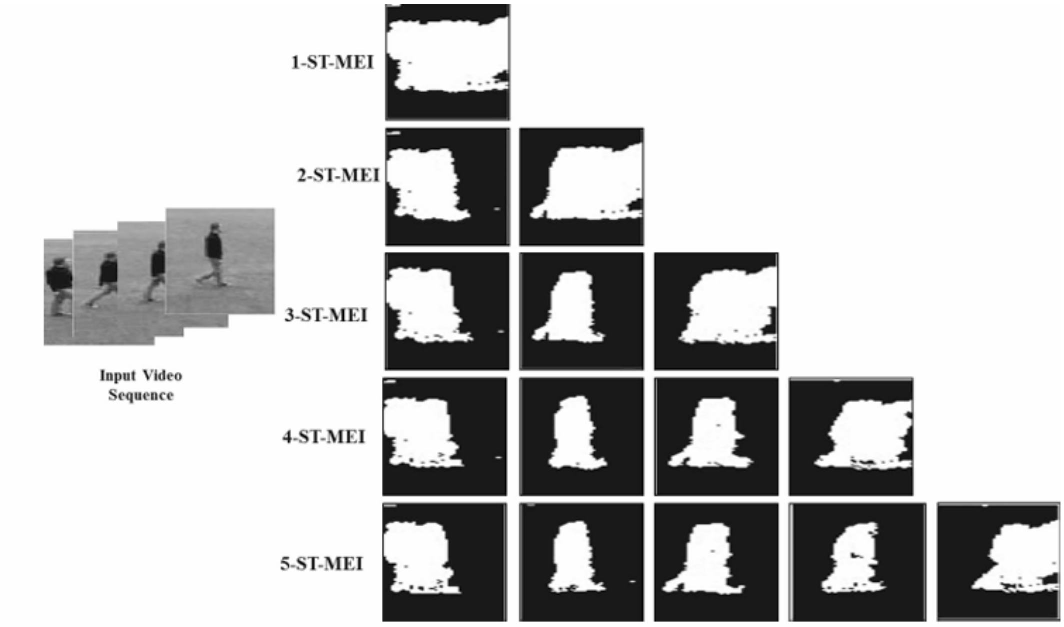


Figure 2.25 Sample of walking action with different numbers of durations for each ST-MEI template [12]

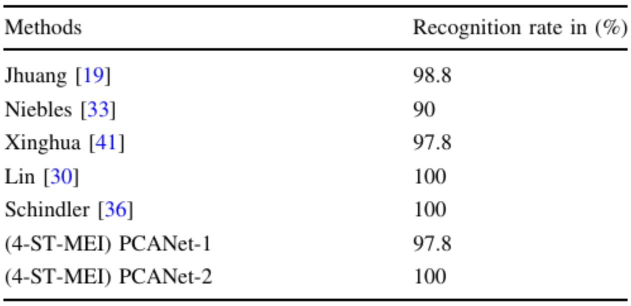
PCANet was then implemented to extract the features from the generated ST-MEI templates. It consisted of three processing layers to mimic the structure of traditional convolutional networks. Initially, PCA filters, which had been trained using the PCA algorithm, were applied as the convolution filter bank in the convolutional layers. Next, a simple binary hashing technique was used for the nonlinear layer. This was followed by the block-wise histograms of the binary codes, serving as the feature pooling layer. This phase yielded multiple ST features for further analysis in the subsequent stages.

The local features extracted from multiple ST-MEI templates were then fused by summing the feature vectors of all the templates corresponding to the same video. This process generated a single, comprehensive feature vector for each video, ensuring the temporal information from different phases of the action was combined to enhance the overall representation of the action.

Afterward, these resulting feature vectors underwent a whitening process due to the high level of redundancy caused by the correlations between local histograms. The whitening PCA algorithm was applied to reduce both the dimensionality and redundancy of the feature vectors, removing unnecessary information while retaining the key features necessary for action recognition.

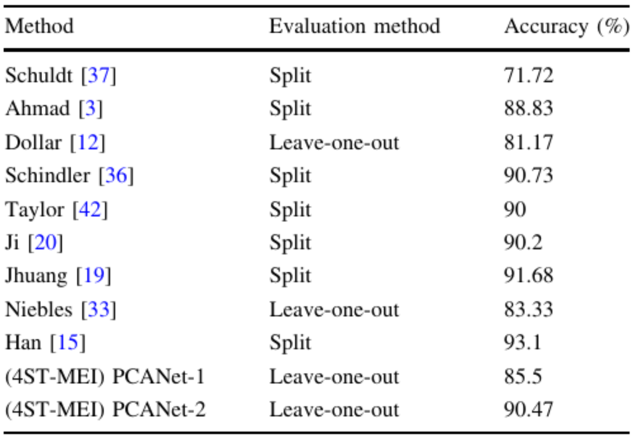
Finally, the processed feature vectors were fed into a Support Vector Machine (SVM) classifier for action recognition, which was trained to classify the actions based on the feature vectors. However, the SVM's overall computational time was dependent on the length of the final feature vector.

They evaluated their proposed HAR approach by testing it on three popular action datasets: Weizmann, KTH, and UCF Sports Action. To improve the recognition rate, they optimised the PCANet parameters for both the single-stage PCANet (PCANet-1) and two-stage PCANet (PCANet-2) for each dataset. These parameters included the filter size (k1, k2), the number of filters in each stage (L1, L2), the number of stages, the block size of the local histograms, and the ratio of overlapping between blocks. The recognition rate for each dataset was listed in Table 2.23, Table 2.24 and Table 2.25 respectively.



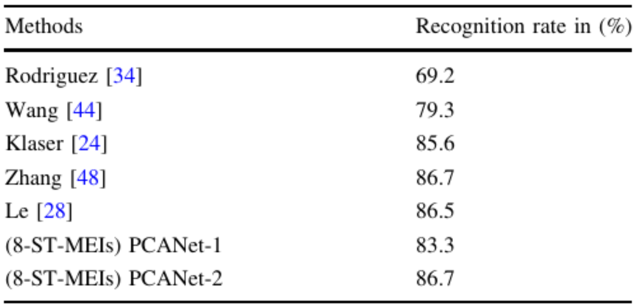
Methods	Recognition rate in (%)
Jhuang [19]	98.8
Niebles [33]	90
Xinghua [41]	97.8
Lin [30]	100
Schindler [36]	100
(4-ST-MEI) PCANet-1	97.8
(4-ST-MEI) PCANet-2	100

Table 2.23 Comparison of the proposed method with state-of-the-art results on Weizmann dataset [12]



Method	Evaluation method	Accuracy (%)
Schuldt [37]	Split	71.72
Ahmad [3]	Split	88.83
Dollar [12]	Leave-one-out	81.17
Schindler [36]	Split	90.73
Taylor [42]	Split	90
Ji [20]	Split	90.2
Jhuang [19]	Split	91.68
Niebles [33]	Leave-one-out	83.33
Han [15]	Split	93.1
(4ST-MEI) PCANet-1	Leave-one-out	85.5
(4ST-MEI) PCANet-2	Leave-one-out	90.47

Table 2.24 Comparison of the proposed method with state-of-the-art results on KTH dataset [12]



Methods	Recognition rate in (%)
Rodriguez [34]	69.2
Wang [44]	79.3
Klaser [24]	85.6
Zhang [48]	86.7
Le [28]	86.5
(8-ST-MEIs) PCANet-1	83.3
(8-ST-MEIs) PCANet-2	86.7

Table 2.25 Comparison of the proposed method with state-of-the-art results on UCF sports dataset [12]

The proposed method's experimental results were both effective and accurate. It successfully attained a 100% recognition rate on the Weizmann human action dataset and a 90.47% recognition rate on the KTH dataset, equivalent to the performance of other state-of-the-art results. Although the recognition rate shown in the UCF Sports dataset was not as high as the other 2, the results were still considered adequate due to the significant challenges in the dataset.

CHAPTER 3

System Methodology/Approach

3.1 Methodologies and General Work Procedures

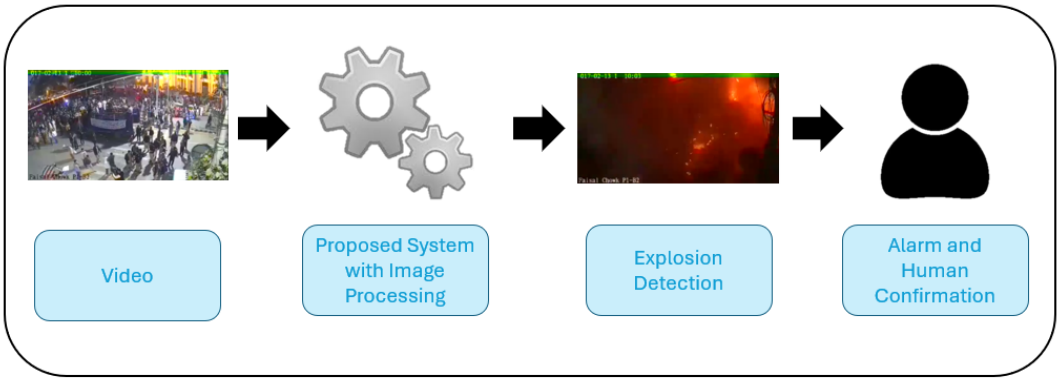

Video
Proposed System with Image Processing
Explosion Detection
Alarm and Human Confirmation

Figure 3.1 Project Methodology

The explosion detection system follows a structured methodology to process video footage and detect explosion events accurately. First, the system begins with a video input where the video can either contain an explosion or normal activities. The video is then analysed using image processing techniques, including frame extraction and motion detection. These techniques are believed to be the key components to help identify explosion-related features. Once an explosion event is detected, the system will trigger an alarm notification to alert humans. To avoid unnecessary panic, a human verification step is necessary to confirm whether the detected event is indeed an explosion.

The system is developed using the C++ programming language and implemented in Visual Studio 2022. The system is designed to achieve an explosion detection accuracy exceeding 80%, providing a promising performance to replace traditional video surveillance systems . In order to evaluate its performance, the system will be tested using videos from the UCF-Crime dataset. Based on the results, the system's accuracy and effectiveness can be analysed, allowing for further refinements to enhance its real-world applicability.

3.2 Real Case Scenario

Five scenarios are discussed in this section. Explosion events that happen within the scene are classified into massive or mild categories. In addition, each scenario is further grouped as either an excessive motion scene or a static scene.

Excessive motion scenes refer to the environments that are filled with significant background activity, such as vehicle motion or pedestrian movement, creating a more challenging context for explosion detection due to the abundance of motion. Conversely, static scenes contain little or no background motion, making them comparatively easier to analyse but still requiring precise detection to avoid false alarms.

These five scenarios will be further discussed in Chapter 5 later.

3.2.1 Scenario 1

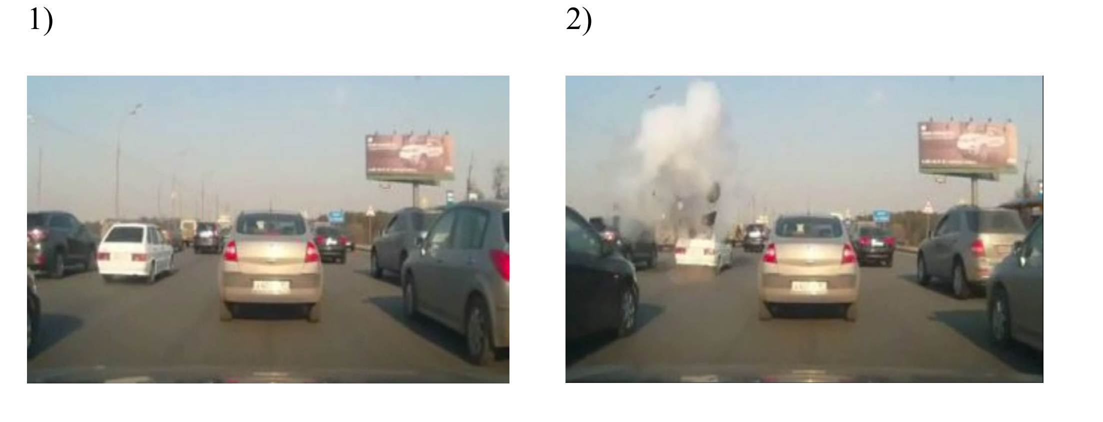


Figure 3.2 Video Footage of Scenario 1

Event: Massive Explosion, Excessive Motion Scenes

1) Normal vehicle motion where cars are moving smoothly in traffic.
2) A car suddenly explodes

3.2.2 Scenario 2

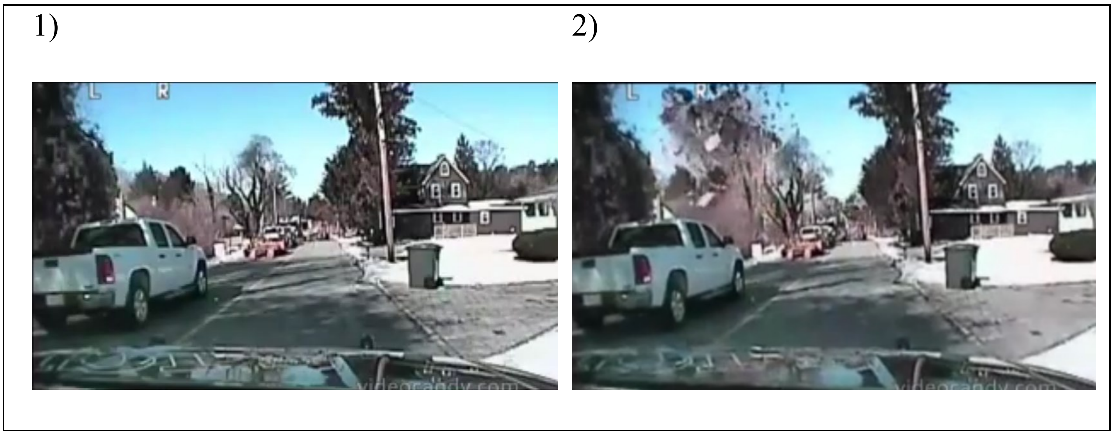


Figure 3.3 Video Footage of Scenario 2

Event: Massive Explosion, Static Scenes

1) The scene is static with no visible motion.
2) A sudden explosion occurs nearby.

3.2.3 Scenario 3

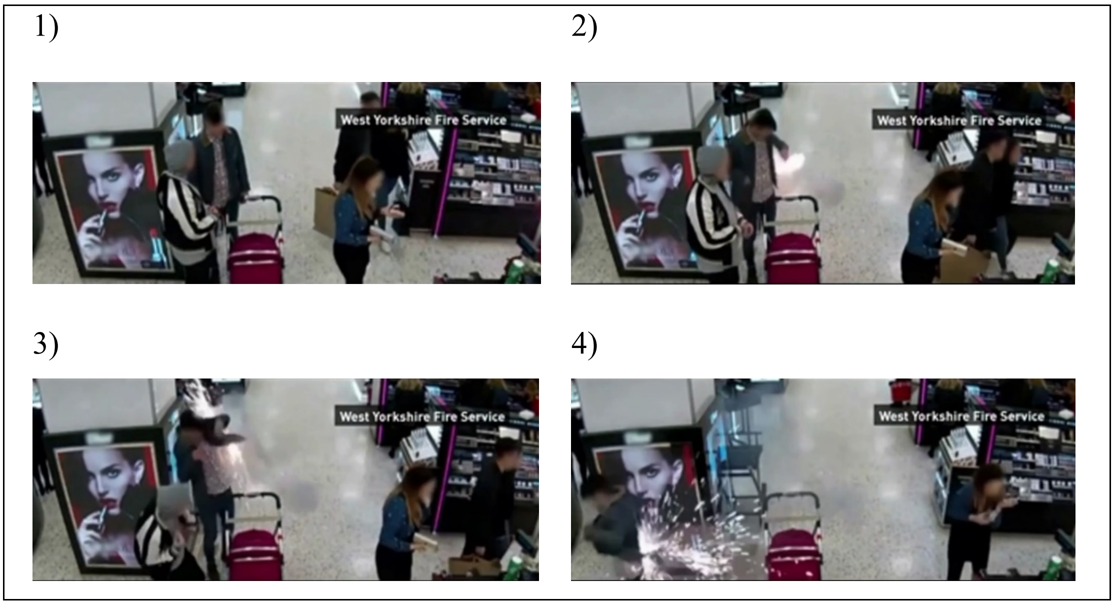


Figure 3.4 Video Footage of Scenario 3

Event: Mild Explosion, Excessive Motion Scenes

1) Two men are chatting inside a mall while other people walk nearby.
2) A sudden spark appears in a man’s pocket.
3) The spark grows stronger.

4) The explosion bursts with visible sparks, scattering around.

3.2.4 Scenario 4

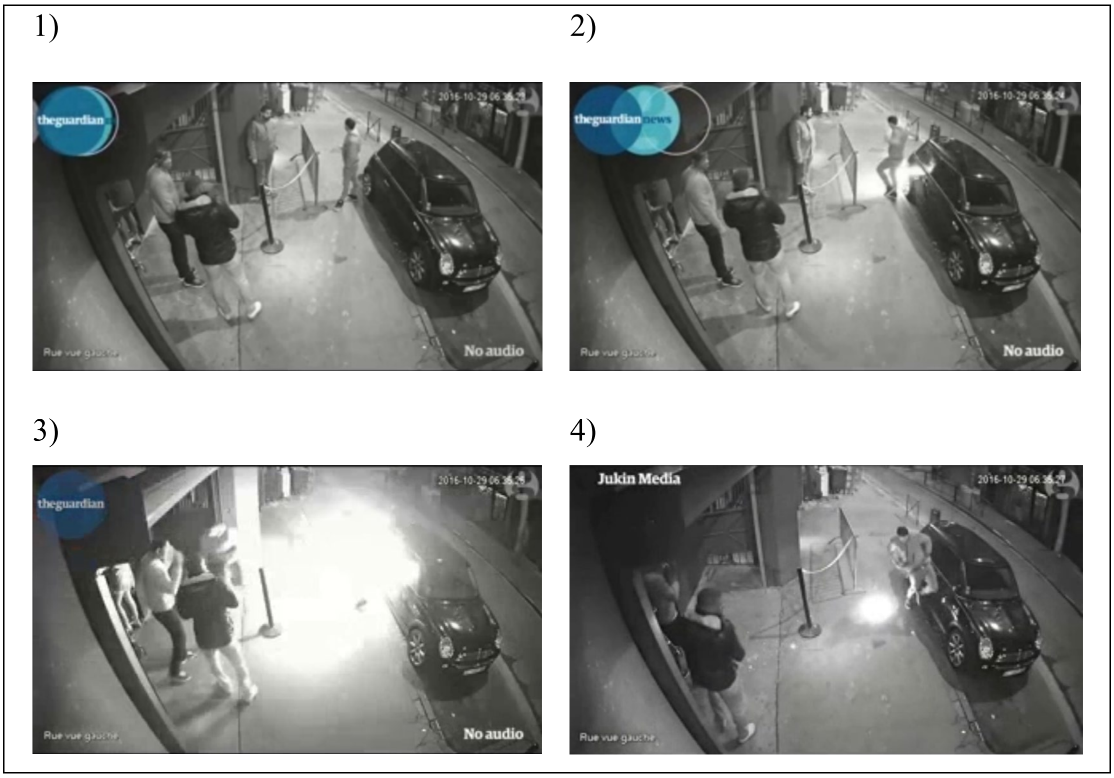


Figure 3.5 Video Footage of Scenario 4

Event: Mild Explosion, Static Scenes

1) A few men are chatting on the street.
2) A sudden explosion occurs inside a man’s pocket.
3) The explosion grows bigger.
4) The explosive object falls to the ground.

3.2.5 Scenario 5

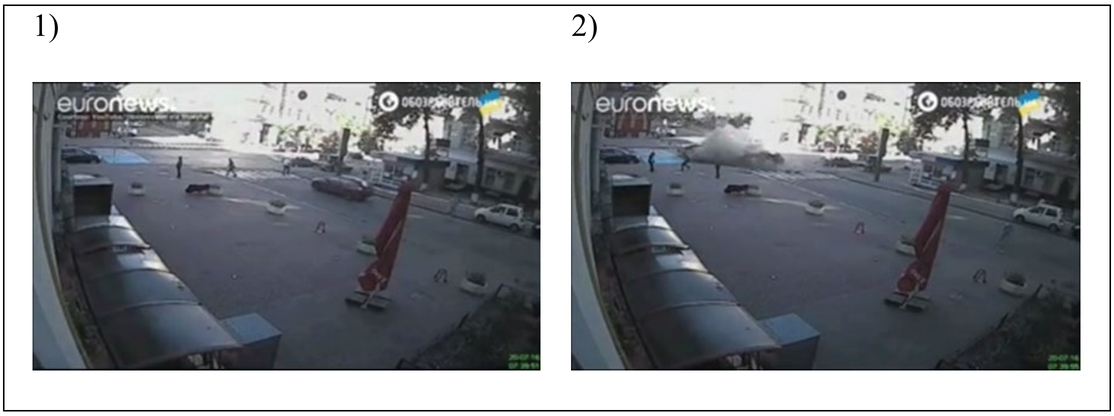


Figure 3.6 Video Footage of Scenario 5

Event: Mild Explosion, Excessive Motion Scenes

1) Normal vehicle motion on a road.
2) A car suddenly explodes.

3.3 Discussion

Based on the above scenarios, several key observations can be made:

- The camera remains fixed in most cases, ensuring that the footage is stable and free from unnecessary movement.
- The explosions are clearly visible to the human eye, which implies that the detection system should also be able to capture these visual cues through computer vision techniques.
- An explosion is often accompanied by sudden and excessive motion, making them distinct from regular motion patterns.
- Explosions can occur in different contexts, such as within static scenes with little to no prior motion (Scenario 2 and 3), or in a dynamic scene with significant background activity already present (Scenario 1, 4 and 5).
- The explosion is likely preceded by a normal or static scene, followed by a sudden and intense burst of activity.

CHAPTER 4

System Design

4.1 System Overview

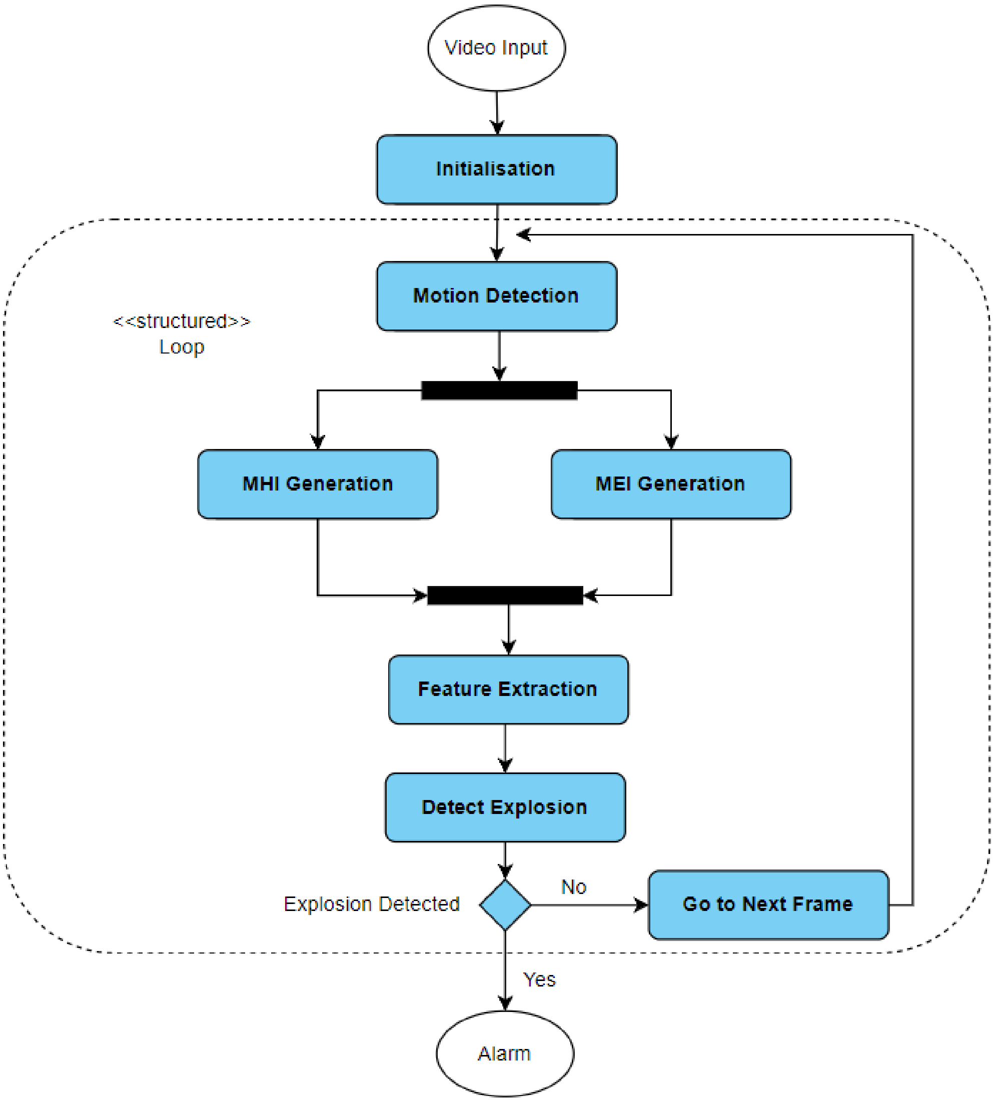


Figure 4.1 System Overview

Figure 4.1 shows the system overview diagram for explosion detection through video analysis with the implementation of image processing techniques and motion detection. The system begins with capturing the video input and then proceeds to the initialisation process, where each video frame will go through frame preprocessing to enhance its quality. For instance, frames are converted to greyscale to suppress insignificant motion and reduce noise.

The system then processes the pre-processed video frames through several key modules. First, the motion detection module identifies movements within the frames. If the pixel differences are big enough, it will be recorded as a valid motion instead of being classified as noise. Subsequently, the generation of MHI and MEI comes into play. The MHI module will be used to capture temporal motion information, while the MEI module will highlight the spatial distribution of motion within the scene.

Next, the system performs feature extraction from both MHI and MEI to distinguish the specific patterns that may indicate an explosion. Finally, the system analyses these extracted features to determine if an explosion is present. If an explosion is detected, the system triggers an alarm to notify the users for further confirmation.

4.2 System Component Specification

4.2.1 Motion Detection

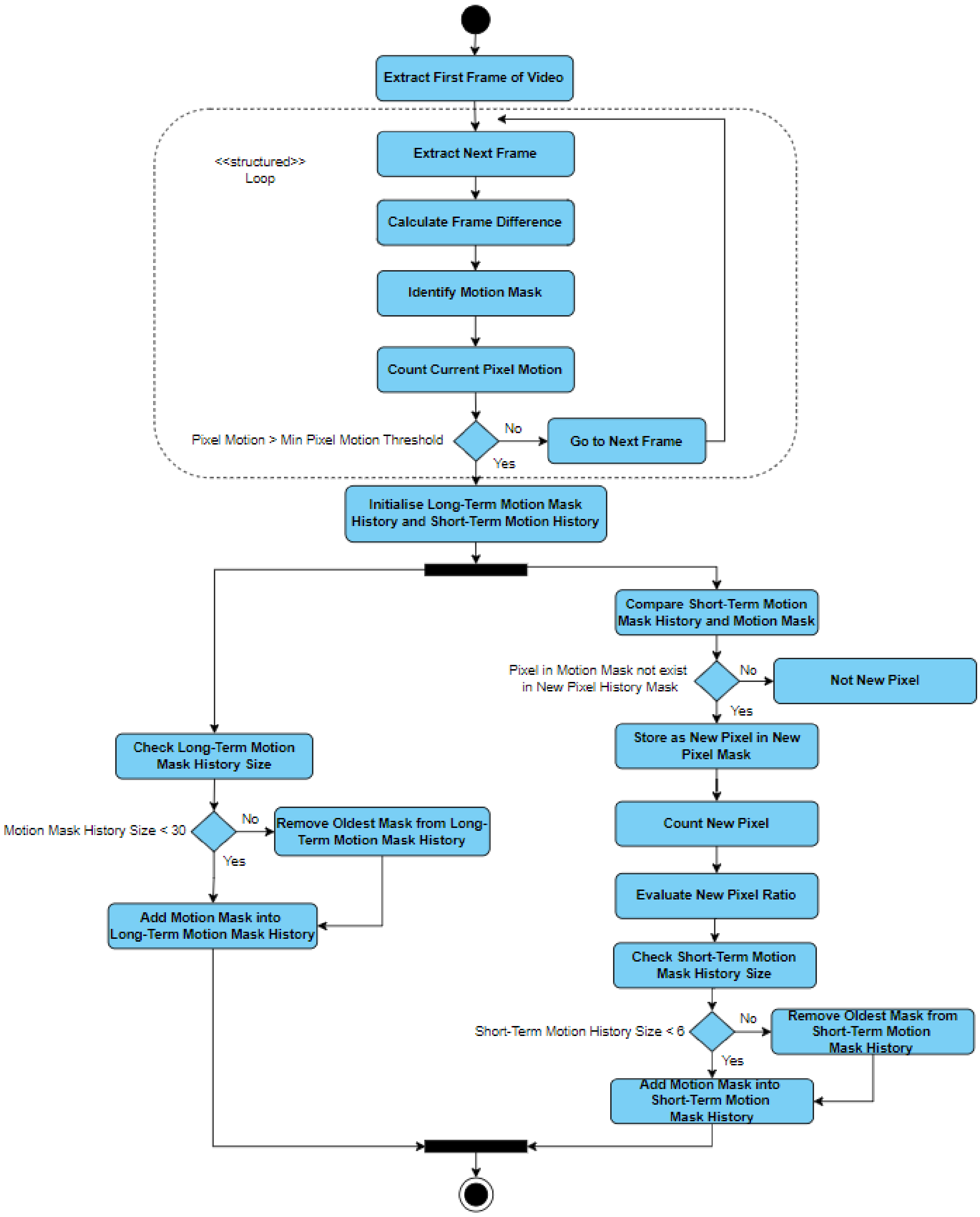

Extract First Frame of Video
<<structured>>
Loop
Extract Next Frame
Calculate Frame Difference
Identify Motion Mask
Count Current Pixel Motion
Pixel Motion > Min Pixel Motion Threshold
No
Go to Next Frame
Yes
Initialise Long-Term Motion Mask History and Short-Term Motion History
Compare Short-Term Motion Mask History and Motion Mask
Pixel in Motion Mask not exist in New Pixel History Mask
No
Not New Pixel
Yes
Store as New Pixel in New Pixel Mask
Count New Pixel
Evaluate New Pixel Ratio
Check Short-Term Motion Mask History Size
Short-Term Motion History Size < 6
No
Remove Oldest Mask from Short-Term Motion Mask History
Yes
Add Motion Mask into Short-Term Motion Mask History
Check Long-Term Motion Mask History Size
Motion Mask History Size < 30
No
Remove Oldest Mask from Long-Term Motion Mask History
Yes
Add Motion Mask into Long-Term Motion Mask History

Figure 4.2 Motion Detection Workflow

The motion detection process begins with the extraction of the first frame from the video. It is then stored as the "previous frame", serving as a reference for subsequent comparisons. At the same time, the system initialises both the Long-Term Motion Mask History and Short-Term Motion Mask History.

After initialisation, the system goes into a loop and proceeds to extract the next frame. Then, frame difference is calculated using the pixel values of both frames:

$$D(x, y, t) = |I(x, y, t) - I(x, y, t - 1)| \tag{4.1}$$

- $D(x, y, t)$ is the absolute frame difference at pixel (x, y) and time t
- $I(x, y, t)$ is the intensity value of the current greyscale frame
- $I(x, y, t - 1)$ is the intensity of the previous greyscale frame

The result is then used to identify a motion mask, where motion is marked if the pixel difference exceeds 30 since pixel difference that is lower than 30 is believed to be noise in the video, which does not provide any useful information to the particular frame.

$$M_t(x, y) = \begin{cases} 255 \text{ if } D(x, y, t) > 30 \\ 0 \; otherwise \end{cases} \tag{4.2}$$

Then, the motion mask is used to count the number of motion pixels, checking whether the current pixel motion exceeds a minimum threshold of 0.1% of the total pixels in the frame, to filter out motion that is visually insignificant.

$$MotionPixelCount_t = \sum_{x,y} 1(M_t(x, y) = 255) \tag{4.3}$$

This threshold is carefully chosen to balance sensitivity and noise suppression. A value of 0.1% is high enough to eliminate background noise, such as slight lighting changes or video footage noises, but low enough to detect small but important motions like the early flashes and a sudden burst in explosion events. In contrast, using higher thresholds causes the system to overlook these small-scale but important changes, resulting in a decline in the system performance. If the current pixel motion exceeds a minimum threshold, it is considered a valid motion. Otherwise, the system skips the current frame and continues to process the subsequent frame until a valid motion is detected.

If a valid motion is detected, the system performs two parallel operations:

1) It appends the current motion mask to the Long-Term Motion Mask History, enabling temporal analysis such as the update of MHI and MEI. Then, it checks whether the long-term motion mask history size is more than a maximum limit of one second frame size. A maximum buffer size of one second allows the system to retain one second of motion history, which is sufficient for reliably detecting and analysing explosion events. If the size exceeds the limit, the oldest motion mask is removed to maintain a fixed window so only the most recent motion data is retained for use in the next iteration.
2) It evaluates new motion pixels by comparing the current motion mask with the Short-Term Motion Mask History. A new motion pixel refers to a pixel that is marked as motion in the current frame and did not appear in the short-term motion mask history. This helps differentiate between persistent motion (e.g., moving vehicles) and transient motion (e.g., sudden bursts from explosion events). The short-term motion mask history stores the previous 0.2 seconds of valid motion. This 0.2-second window is sufficient to capture the continuous motion mentioned, such as vehicles moving steadily across the frame, allowing the system to differentiate them from sudden explosion events. On the other hand, a shorter duration, like 0.1 seconds, cannot provide enough temporal information to eliminate repetitive patterns, elevating the false alarm rate. Conversely, a longer duration, like 0.3 seconds, can misclassify explosion characteristics, such as small bursts of smoke, as normal continuous motion. A history of 0.2-second frames provides enough temporal motion information to recognise continuous motion, so they are not mistakenly detected as an explosion. At the same time, it allows the system to capture and detect sudden motion patterns from an explosion event. Therefore, 0.2-second frames offer an optimal balance to ensure the system effectively separates normal continuous motion from the sudden motion from explosion events.

The Long-Term Motion Mask History is computed as:

$$H_t^L(x, y) = \bigcup_{i=1}^{\tau_L} M_{t-i}(x, y) \qquad 4.4$$

- τ_L is the number of past frames to consider (1 second)

The Short-Term Motion Mask History is calculated as:

$$H_t^S(x,y) = \bigcup_{i=1}^{\tau_S} M_{t-i}(x,y) \tag{4.5}$$

- τ_S is the number of past frames to consider (0.2 seconds)

From this Short-Term Motion Mask History, the system computes the New Pixel Mask:

$$N_t(x,y) = M_t(x,y) \wedge \neg H_t^S(x,y) \tag{4.6}$$

The number of new motion pixels is then calculated, reflecting how many new motion pixels have appeared in the current frame:

$$NewPixelCount_t = \sum_{x,y} N_t(x,y) \tag{4.7}$$

The New Pixel Ratio is computed by dividing the new pixel count by the total motion pixels, which indicates the ratio of the new pixel relative to the overall motion area:

$$NewPixelRatio_t = \frac{\sum_{x,y} N_t(x,y)}{\sum_{x,y} M_t(x,y)} \tag{4.8}$$

Explosions are often sudden and happen in a short duration. Hence, both of these parameters are useful for differentiating explosions from other large but gradual motion events (moving cars) that may cause false alarms.

Finally, the system checks if the New Pixel Mask History size exceeds the τ_S of 0.2 second. If it does, the system will remove the oldest mask. The system then inserts the current new pixel mask for future comparison.

Figure 4.3 and Figure 4.4 illustrate examples of newly generated pixels.

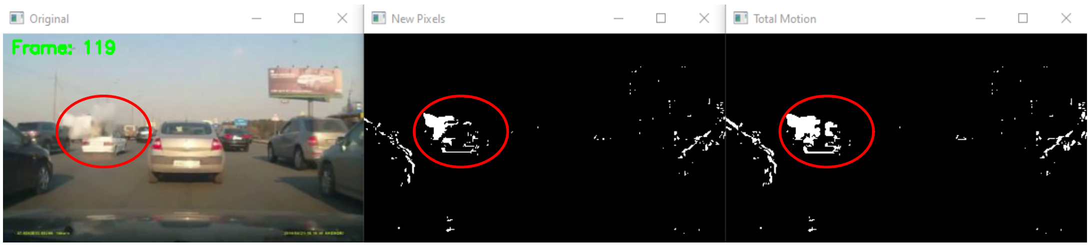


Figure 4.3 New pixel distribution during an explosion event

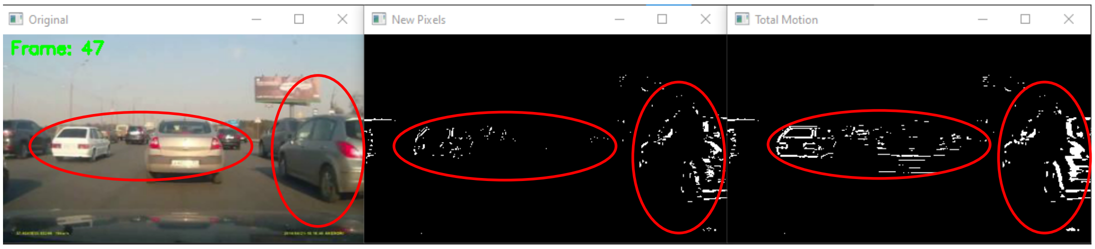


Figure 4.4 New pixel distribution during vehicle motion

Figure 4.3 shows the total motion pixels and newly generated pixels during an explosion event. It can be clearly observed that the motions generated by the explosion are stored as new pixels. In contrast, Figure 4.4 presents the new pixels by vehicle motions. It is noted that many of these motions are not stored as new pixels, as the movements remain consistent over time at the same positions. Such consistent motion is not a characteristic of an explosion event.

4.2.2 Motion History Image (MHI) Generation

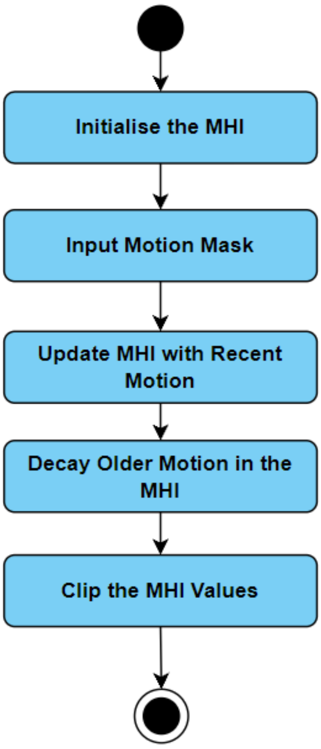


Figure 4.5 MHI Generation

The process begins with the initialisation of the MHI by creating an empty image, typically the same size as the video frames. The image is initialised with all pixel values set to zero, stating no motion at the start. According to Bobick and Davis [11], initialising the MHI is the first step in capturing temporal dynamics in a scene, providing a framework for the subsequent motion analysis. Hence, this step is crucial because it establishes a baseline for recording motion as the video progresses.

After the initialisation of the MHI, the system proceeds to input the motion mask that was generated during the motion detection process. The motion mask is an essential input, as it indicates which pixels in the current frame are associated with detected motion. The system then updates the MHI with recent motion by assigning the relevant pixels to a maximum value, typically known as τ (tau) [13]. This step guarantees that the most recent motion is obviously recorded in the MHI, enabling easy identification of newly detected motion.

MHI Formula:

$$MHI_t\,(x,y,t) = \begin{cases} \tau \text{ if } M_t(x,y) = 255 \\ \max(0, H_t^L(x,y,t-1) - 47)\ otherwise \end{cases} \qquad 4.9$$

- τ: Maximum motion intensity, representing a full-strength motion event
- 47 is the decay value used to decrease the motion intensity over time, minimum value change that can be noticed by eye.

Next, the system continues the process with decaying older motion in the MHI, gradually reducing the intensity of pixels that did not detect motion in the current frame. By subtracting a constant value from these pixel values, the MHI allows the most recent motions to dominate the image. This enables the distinction between new and old motion events. For example, Figure 4.4 shows an obvious distinction between new and old motion events with the intensity of the white colour.

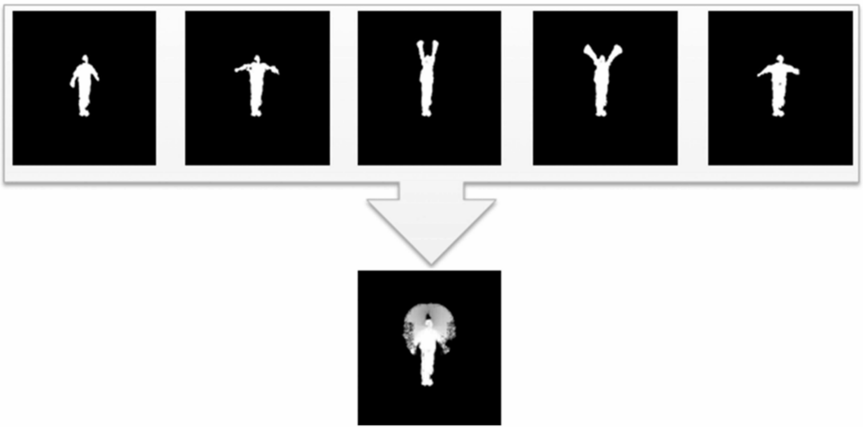


Figure 4.6 MHI of action ‘Wave Arms’ [14]

Finally, the system clips the MHI values to ensure all pixel intensities stay within a valid range, between 0 and τ. This prevents the pixel values from acquiring negative values or beyond the maximum allowable intensity, avoiding any distortion in the representation of motion in the MHI. The process then repeats for each subsequent video frame, providing a dynamic and evolving representation of motion throughout the video. MHI is effective in capturing and analysing temporal motion patterns, causing it to be highly applied in various applications such as human activity recognition, gesture recognition and facial recognition [14, 15, 16].

4.2.3 Motion Energy Image (MEI) Generation

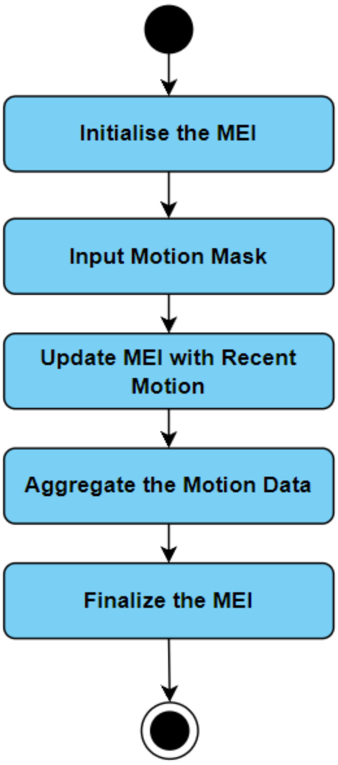


Figure 4.7 MEI Generation

The generation of a MEI initialises by establishing a blank image that will serve as a stored medium for accumulated motion data. This image's pixel values are all configured to zero, as there was no motion at the beginning. Then the motion mask that is derived from the motion detection phase is integrated into the process.

Next, the MEI is updated by incorporating the data from the binary motion mask. For each frame, the corresponding pixels are updated if the binary motion mask shows the value of 1 to reflect the motion. The MEI continues to aggregate motion data during a defined time period, allowing the MEI to provide a thorough overview of all the motion that has occurred within the specified time period.

MEI Formula:

$$MEI_{\tau}(x, y, t, Th) = \bigcup_{i=0}^{\tau-1} M_{t-i}(x, y) \qquad 4.10$$

- τ is the number of past frames to consider (1-second frame size)

Eventually, the MEI is finalised after all the frames within the specified period have been processed. This image is then ready for further analysis by providing valuable insight to determine the characteristics of explosion events, such as the sudden rise in current motion pixels.

4.2.4 Optical Flow Generation

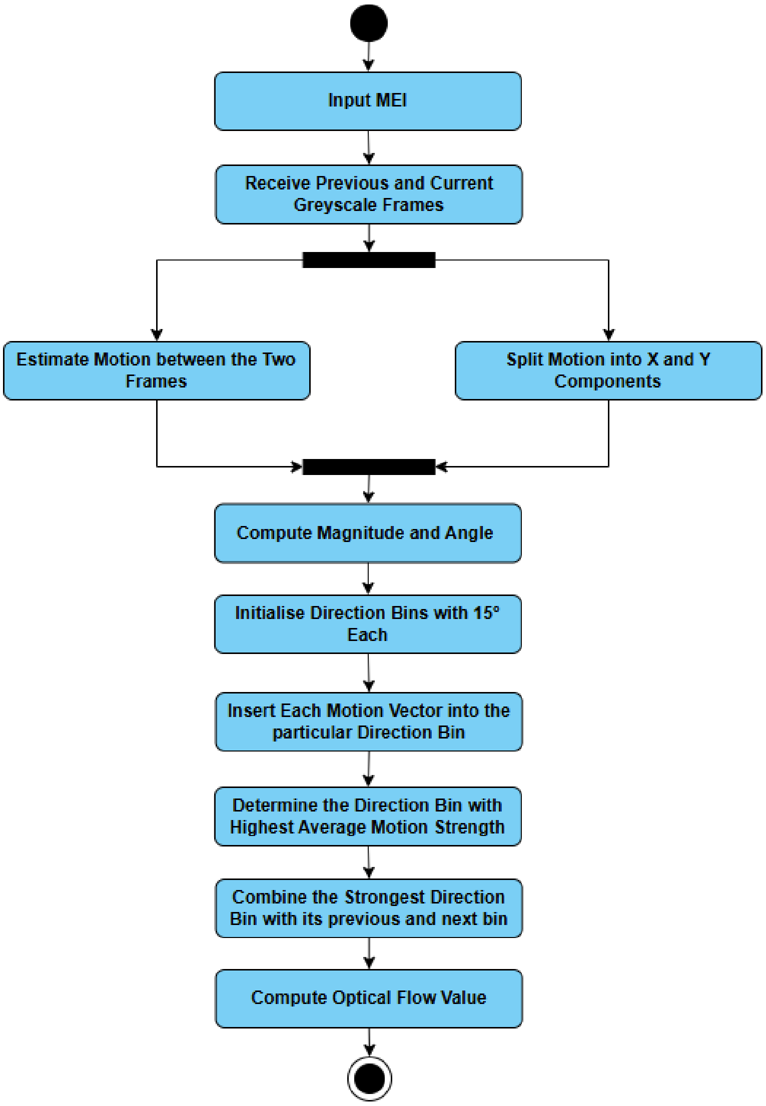

Input MEI
Receive Previous and Current Greyscale Frames
Estimate Motion between the Two Frames
Split Motion into X and Y Components
Compute Magnitude and Angle
Initialise Direction Bins with 15° Each
Insert Each Motion Vector into the particular Direction Bin
Determine the Direction Bin with Highest Average Motion Strength
Combine the Strongest Direction Bin with its previous and next bin
Compute Optical Flow Value

Figure 4.8 Optical Flow Generation

The optical flow generation process begins with the input of MEI, then receive two consecutive greyscale frames from MEI, one representing the previous frame and the other representing the current frame. Greyscale frames are used instead of colour images because motion estimation basically relies on intensity variations. This reduces unnecessary colour channels, easing the required computation power while preserving the necessary motion information that is useful in later processes.

Two greyscale frames are represented as:

$$f(x, y) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} \quad 4.11$$

- $u(x, y)$ and $v(x, y)$ are the horizontal and vertical components of motion at pixel (x, y)

Once the frames are obtained, the system simultaneously estimates the motion between the two frames and splits the detected motion into its respective X and Y components. Motion estimation determines how much pixels move from the previous frame to the current frame. Meanwhile, X and Y components are representing the horizontal and vertical motion directions respectively. These components are crucial as they are needed to compute the angle within the motions.

Afterward, the magnitude and angle of the motion vectors are computed. The magnitude represents the strength of the movement, while the angle indicates the direction in which the motion occurs. In order to categorise the motion, the system initialises direction bins, where each of them covers an angle of 15°. Then, each motion vector derived from the previous step is inserted into its particular direction bin according to its angle. This step groups similar motion directions together within a direction bin so the distribution of motions in the current frame can be told.

Magnitude and angle of motion vectors at pixel (x, y):

$$Magnitude_{(x,y)} = \sqrt{u(x, y)^2 + v(x, y)^2} \quad 4.12$$

$$Angle_{(x,y)} = \tan^{-1} \frac{v(x, y)}{u(x, y)} \; (in\ degrees, from\ 0^o\ to\ 360^o) \quad 4.13$$

Each angle is placed into a bin (24 bins of 15° each):

$$BinIndex_{(x,y)} = \left\lfloor \frac{Angle_{(x,y)}}{\left(\frac{360}{B}\right)} \right\rfloor \quad 4.14$$

- $B =$ numbers of bins

Once all vectors are categorised into their respective direction bins, the system proceeds to evaluate each bin to identify the dominant motion direction. This is achieved by calculating the average motion strength within each bin, representing how strongly motion is concentrated in that particular direction. The bin with the highest average motion strength is then selected as the primary direction of movement within the frames. An explosion event typically generates sudden, intense, and concentrated bursts of motion that spread rapidly outward. The motion is much stronger and directionally non-uniform compared to regular scene movements, such as walking or vehicle motion. Thus, the strongest direction bin in the context of average motion strength is a reliable indicator of an explosion event, capturing the intense and concentrated nature of the motion created by the explosion event.

For each bin i, the average flow magnitude is computed:

$$AverageMagnitude_i = \frac{1}{N_i} \sum_{(x,y) \in bin\ i} Magnitude_{(x,y)} \qquad 4.15$$

- N_i = number of pixels in bin i

Find the bin b^* with the highest average magnitude:

$$b^* = arg\max\left(AverageMagnitude_i\right) \qquad 4.16$$

Next, the system combines the identified strongest direction bin with its immediate neighbours, which are the preceding and succeeding bins. This step ensures the relevant motions are not overlooked due to the minor directional variations that occur from noise or inconsistencies in motion patterns. Finally, the system computes the optical flow value, which quantifies the uniformity of motion across the current frame.

A higher value indicates non-uniform motion, which is a characteristic of explosion events, whereas a lower value suggests a consistent directional motion that is more likely produced by vehicles or pedestrian movement.

The number of motions in the best bin and its two neighbours $(index(b^*) \pm 1, n_{b^*}$:

$$n_{b^*} = \sum_{i-1}^{i+1} size(bin\ i) \qquad 4.17$$

- $i = index(b^*)$

Optical Flow Value:

$$OpticalFlowValue_t = 1 - \frac{n_{b^*}}{MotionPixelCount_t} \qquad 4.18$$

4.2.5 Eroded ROI Extraction

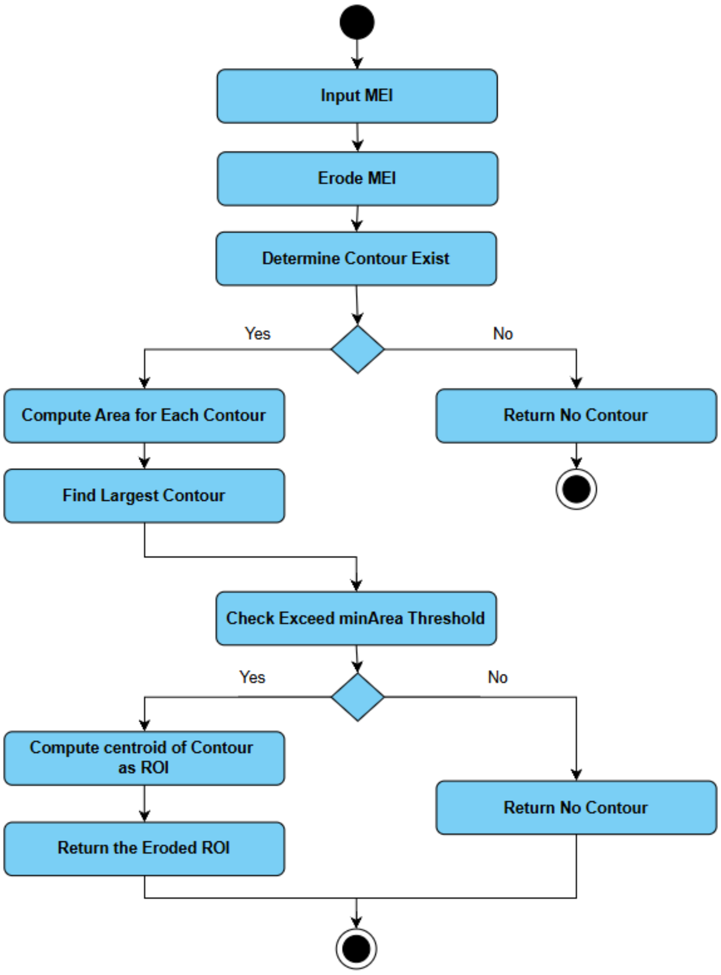


Figure 4.9 Eroded ROI Extraction Workflow

To obtain an eroded region of interest (ROI), the system first receives MEI as input. To remove the small noise pixels and smooth the motion regions, an erosion operation is applied so the system can detect meaningful contours more effectively.

Then, the system checks whether any contours exist within the eroded MEI. If no contours are detected, the process ends here by returning "No Contour". On the contrary, the system proceeds to calculate the area of each detected contour. The contour with the largest area is selected since an explosion event is expected to dominate other contours if any explosion event has occurred.

In this system, the ROI has a maximum size of 2500 pixels, corresponding to a 50 × 50 region, which was found to be sufficient for localised explosion analysis. After the largest contour is identified, the system checks whether its area exceeds a predefined minimum area threshold of 6% of the ROI maximum size (150). This prevents insignificant or noisy contours from being mistakenly treated as a valid ROI. This threshold is chosen because, after the erosion operation, any motion that can still exceed 6% is considered significant enough to be treated as a valid ROI. If the largest contour fails to meet the threshold, the system ends here again by returning "No Contour". On the other hand, if the area is sufficiently large, the system computes the centroid of the largest contour, which serves as the centre of the ROI. Lastly, the system returns the eroded ROI, which will be further processed during later processes.

4.2.6 Non-Eroded ROI Extraction

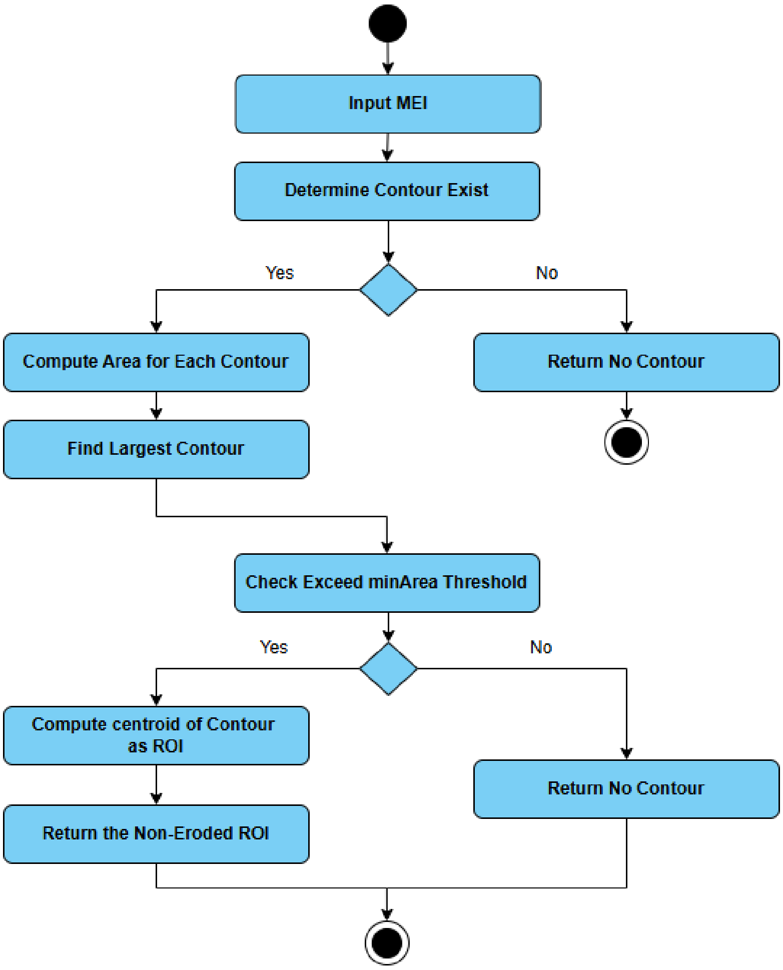

Input MEI
Determine Contour Exist
Yes
No
Compute Area for Each Contour
Return No Contour
Find Largest Contour
Check Exceed minArea Threshold
Yes
No
Compute centroid of Contour as ROI
Return No Contour
Return the Non-Eroded ROI

Figure 4.10 Non-Eroded ROI Extraction Workflow

To obtain a non-eroded region of interest (ROI), the system first receives MEI as input. Unlike the eroded approach, this version does not apply the erosion operation, meaning that the full motion regions are preserved for analysis.

Then, the system checks whether any contours exist within the eroded MEI. If no contours are detected, the process ends here by returning "No Contour". On the contrary, the system proceeds to calculate the area of each detected contour. The contour with the largest area is selected since an explosion event is expected to dominate other contours if any explosion event has occurred.

In this system, the ROI has a maximum size of 2500 pixels, corresponding to a 50 × 50 region, which was found to be sufficient for localised explosion analysis. After the largest contour is identified, the system checks whether its area exceeds a predefined minimum area threshold of 50% of the ROI maximum size (1250). This prevents insignificant or noisy contours from being mistakenly treated as a valid ROI. The threshold is set higher than in the eroded ROI extraction because the frame does not undergo the erosion operation. As a result, larger contours tend to exist since more motion is retained. Hence, only contours larger than 1250 pixels are considered significant enough to represent an explosion in the non-eroded ROI. If the largest contour fails to meet the threshold, the system ends here again by returning "No Contour". On the other hand, if the area is sufficiently large, the system computes the centroid of the largest contour, which serves as the centre of the ROI. Lastly, the system returns the eroded ROI, which will be further processed during later processes.

4.2.7 Global Explosion Detection

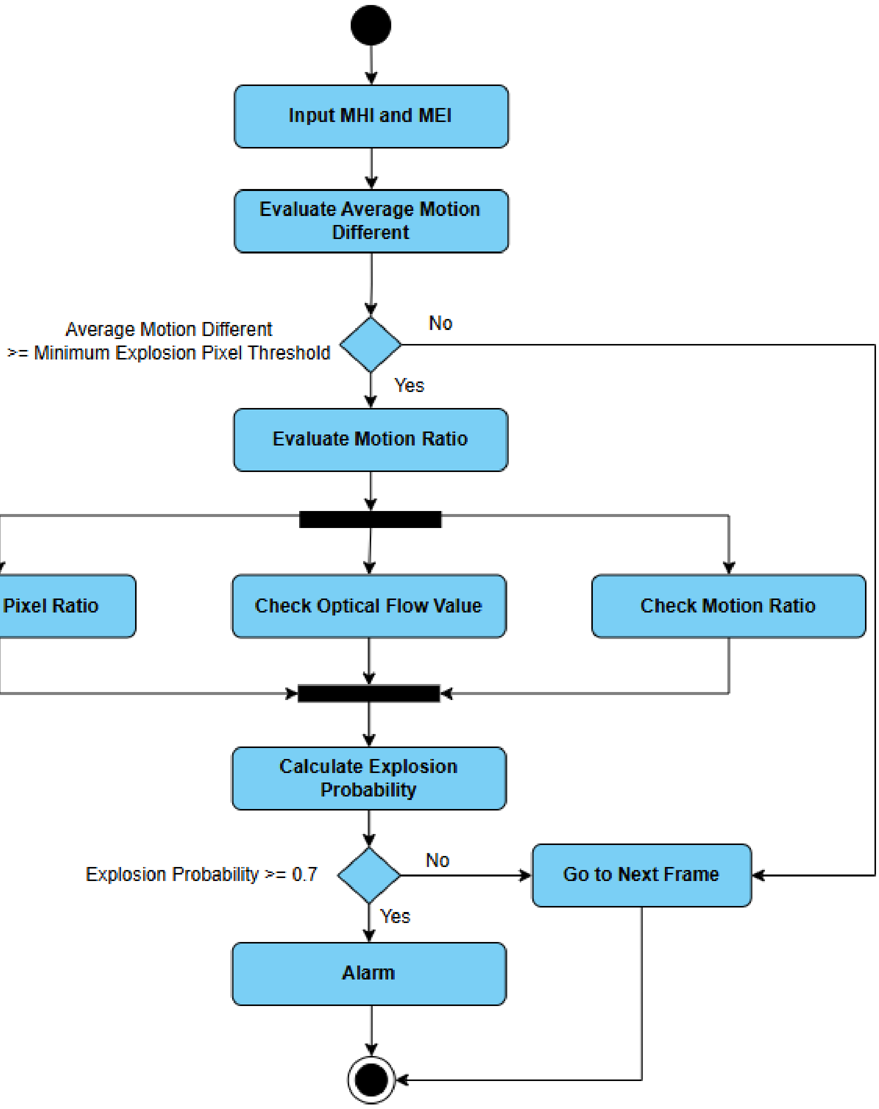

Input MHI and MEI
Evaluate Average Motion Different
Average Motion Different >= Minimum Explosion Pixel Threshold
No
Yes
Evaluate Motion Ratio
Check New Pixel Ratio
Check Optical Flow Value
Check Motion Ratio
Calculate Explosion Probability
Explosion Probability >= 0.7
No
Yes
Go to Next Frame
Alarm

Figure 4.11 Global Explosion Detection

The initial step global explosion involves importing the pre-processed motion data of MEI and MHI. The system continues to evaluate the average motion differences, which measure the difference between the current motion and the average motion over a 0.25 second frames.

Average Motion:

$$AverageMotion = \frac{1}{n}\sum_{i=1}^{n} MotionPixelCount_{t-i} \qquad 4.19$$

- n = number of frames in history buffer

Average Motion Difference:

$$\Delta M_t = |MotionPixelCount_t - AverageMotion| \qquad 4.20$$

This value is subsequently checked against a predefined minimum explosion pixel threshold. The minimum explosion pixel threshold was set to 1000 pixels based on experimental observation, through manual observation and trial across a series of explosion video samples. This threshold often effectively captures strong motion from explosion events while filtering out minor background activity. If the threshold is satisfied, the system proceeds to calculate the motion ratio; else it skips to the next frame.

$$MotionRatio_t = \begin{cases} \dfrac{\Delta M_t}{MotionPixelCount_t}, if\ MotionPixelCount_t > 0 \\ 0, otherwise \end{cases} \qquad 4.21$$

Then, the system uses the features computed earlier, which are the new pixel ratio, optical flow value, and motion ratio. These data are then used in computing explosion probability. If the explosion probability is greater than or equal to 0.7, the system concludes that an explosion is occurring and triggers an alarm. Otherwise, the system will skip to the next frame.

$$\begin{aligned} P_{globalExplosion} &= 0.2 \times MotionRatio_t \\ &+ 0.3 \times NewPixelRatio_t + 0.5 \times OpticalFlowValue_t \end{aligned} \qquad 4.22$$

In the global detection approach, the weight distribution is designed to balance the contributions of the three features when analysing the entire frame. The optical flow value receives the highest weight (0.5) because global explosions usually generate strong and non-uniform motion across large areas of the frame, making it the most reliable global indicator. The new pixel ratio is assigned a moderate weight (0.3) since explosions often introduce a significant number of new motion pixels that were previously inactive. The motion ratio is given the smallest weight (0.2), as it captures the relative change in motion intensity but is less discriminative in full-frame analysis, where many unrelated motions can occur simultaneously.

4.2.8 Eroded ROI Explosion Detection

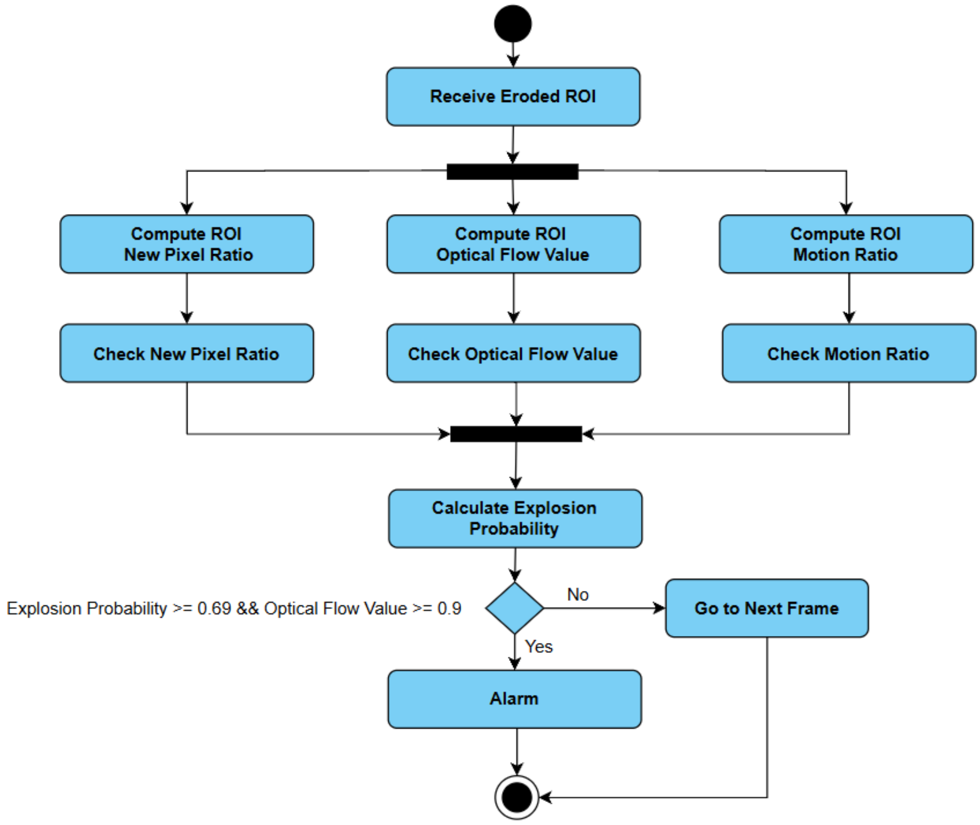

Receive Eroded ROI
Compute ROI New Pixel Ratio
Compute ROI Optical Flow Value
Compute ROI Motion Ratio
Check New Pixel Ratio
Check Optical Flow Value
Check Motion Ratio
Calculate Explosion Probability
Explosion Probability >= 0.69 && Optical Flow Value >= 0.9
No
Go to Next Frame
Yes
Alarm

Figure 4.12 Eroded ROI Explosion Detection

The eroded ROI explosion detection process begins once the eroded ROI is obtained from the generation stage. Within the ROI, the system computes the same three features in the global explosion detection process: the new pixel ratio, the optical flow value, and the motion ratio. The key difference is that these three features are computed only within the localized ROI rather than across the entire frame.

Next, the system proceeds to calculate the explosion probability based on the weighted combination of the three features. Unlike the global detection approach, the eroded ROI detection requires two simultaneous conditions to be satisfied. The explosion probability must be greater than or equal to 0.69, and the optical flow value must be greater than or equal to 0.9. If both conditions are met, the system will then conclude that an explosion is occurring within the eroded ROI and trigger an alarm; else, the system proceeds to the next frame.

$$P_{ErodedRoIExplosion} = 0.3 \times MotionRatio_{ROI} + 0.1 \times NewPixelRatio_{ROI} + 0.6 \times OpticalFlowValue_{ROI} \quad 4.23$$

Compared to the global detection approach, the eroded ROI detection method applies different weight distribution and condition settings because it processes a localised region that has undergone an erosion operation. The extracted eroded ROI contains more refined and reliable motion information. In this case, the optical flow value is assigned to have the highest weight (0.6) since explosions tend to produce concentrated and non-uniform directional bursts of motion that erosion emphasises. The motion ratio is given a moderate weight (0.3) to capture the intensity of motion changes within the eroded ROI, while the new pixel ratio is assigned a smaller weight (0.1) because erosion often reduces scattered pixel activations, lowering its relative significance.

Additionally, the eroded ROI detection requires two simultaneous conditions to be fulfilled, which makes it stricter than the global detection method. This dual-condition approach ensures that only motion regions exhibiting strong non-uniform directional flow and sufficient probability are flagged as explosions, thereby balancing detection sensitivity and false positive reduction.

4.2.9 Non-Eroded ROI Explosion Detection

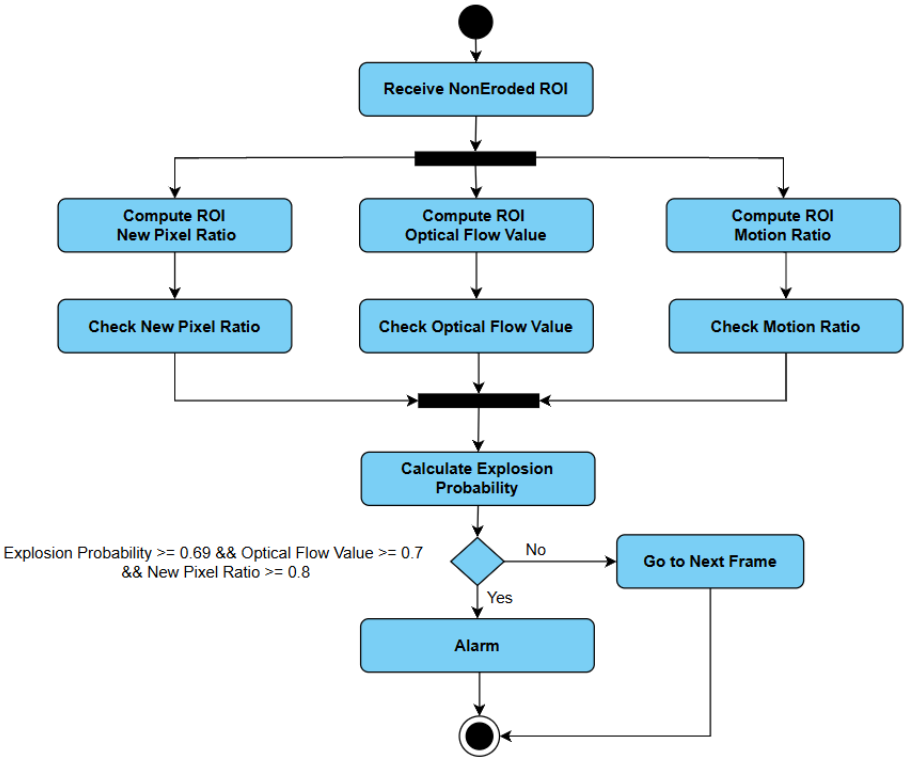

Receive NonEroded ROI
Compute ROI New Pixel Ratio
Compute ROI Optical Flow Value
Compute ROI Motion Ratio
Check New Pixel Ratio
Check Optical Flow Value
Check Motion Ratio
Calculate Explosion Probability
Explosion Probability >= 0.69 && Optical Flow Value >= 0.7 && New Pixel Ratio >= 0.8
No
Go to Next Frame
Yes
Alarm

Figure 4.13 Non-Eroded ROI Explosion Detection

The non-eroded ROI explosion detection process begins once the non-eroded ROI is obtained from the generation stage. Within the ROI, the system computes the same three features in the global explosion detection process: the new pixel ratio, the optical flow value, and the motion ratio. These features are evaluated within the ROI without applying erosion, allowing the system to preserve the original motion regions in their entirety.

Afterward, the system calculates the explosion probability using the same formula with different weight contributions. Unlike the global detection approach and eroded ROI detection methods, the non-eroded ROI detection enforces three simultaneous conditions to be satisfied. The explosion probability must be greater than or equal to 0.69, the optical flow value must be greater than or equal to 0.7, and the new pixel ratio must be greater than or equal to 0.8. If all three conditions are met, the system will then conclude that an explosion is occurring within the non-eroded ROI and trigger an alarm; else, the system proceeds to the next frame.

$$P_{NonErodedRoIExplosion} = 0.1 \times MotionRatio_{ROI} + 0.4 \times NewPixelRatio_{ROI} + 0.5 \times OpticalFlowValue_{ROI} \qquad 4.24$$

Compared to the global and eroded ROI detection approaches, the non-eroded ROI detection method adopts a different weight distribution and stricter conditions because it processes a localised region that retains the full motion information without erosion. In this case, the optical flow value receives the highest weight (0.5) to emphasise the strong directional patterns that are typically generated during explosions, while the new pixel ratio is also highlighted with a weight of 0.4 since sudden bursts of new motion pixels are preserved in the non-eroded ROI. The motion ratio, however, is assigned a lower weight (0.1) because its contribution becomes less significant once the ROI already captures concentrated motion.

Additionally, the non-eroded ROI detection requires three simultaneous conditions to be satisfied, which is stricter than the global and eroded ROI cases. This is done so because the non-eroded ROI preserves more raw motion, potentially leading to the inclusion of noise or unrelated movements within the ROI. By enforcing all three conditions concurrently, the system guarantees that only ROI that is exhibiting strong explosion characteristics will trigger the alarm.

CHAPTER 5

System Implementation

5.1 Hardware Setup

The following table shows the hardware specifications used to develop the proposed system

Description	Specifications
Model	Lenovo Ideapad Gaming 3
Processor	Ryzen 7 5800H
Operating System	Windows 10 Home
Graphic	NVIDIA GeForce RTX 3050
Memory	32.00GB DDR4 RAM
Storage	512GB SSD

Table 5.1 Specifications of Laptop

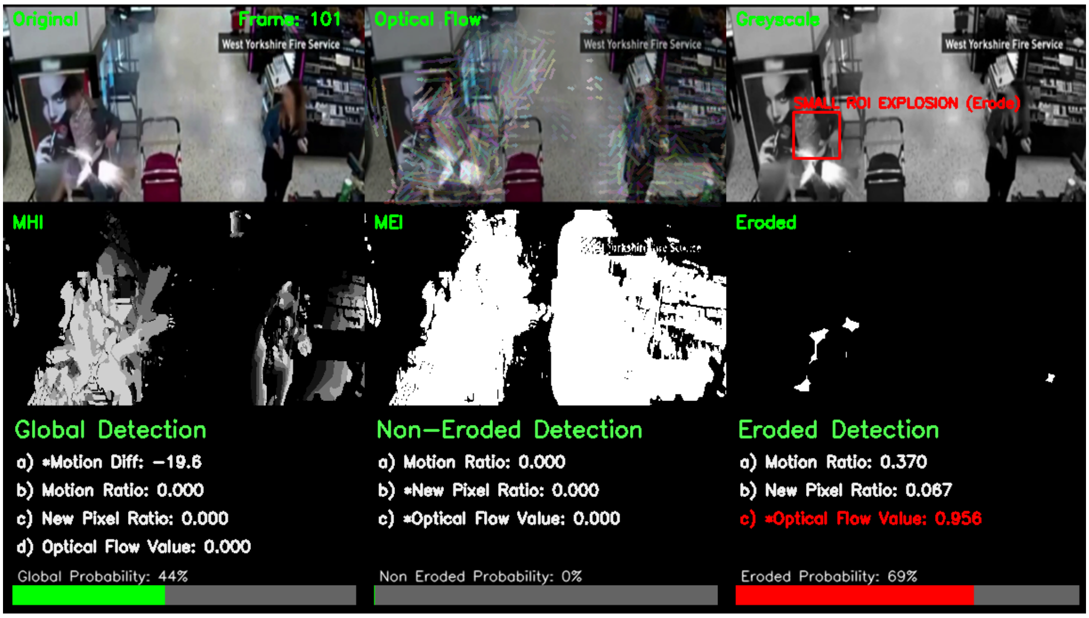
5.2 Software Setup

VISUAL STUDIO 2022	Visual Studio 2022 is an integrated development environment (IDE) that helps developers write, edit, build, and debug code. We use it to program our proposed method in C++ programming language.
OpenCV	OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was developed with the purpose of establishing a unified framework for

	computer vision applications and to accelerate the use of machine perception in commercial products.

Table 5.2 Software Used

5.3 System Operation



Global Detection
a) *Motion Diff: −19.6
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 44%

Non−Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.370
b) New Pixel Ratio: 0.067
c) *Optical Flow Value: 0.956
Eroded Probability: 69%

Figure 5.1 Explosion Video Analytic Interface

After a valid video path is provided, the system begins the process by reading the first frame and presenting the explosion video analytic interface shown in Figure 5.1. The interface consists of a total of nine tiles: six tiles display different video representations, while the remaining three tiles present the current frame's detection metrics. This organised visualisation helps the users to clearly understand the visual transformations of the video frames and the corresponding analytic results.

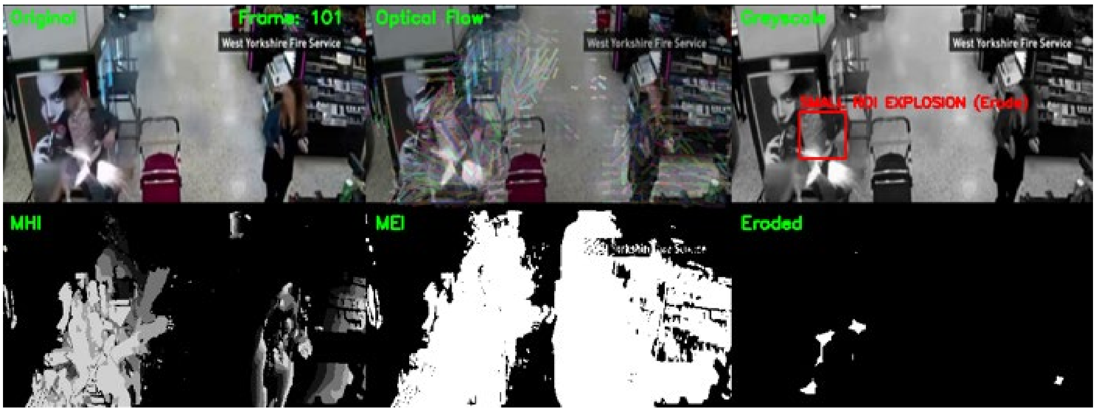
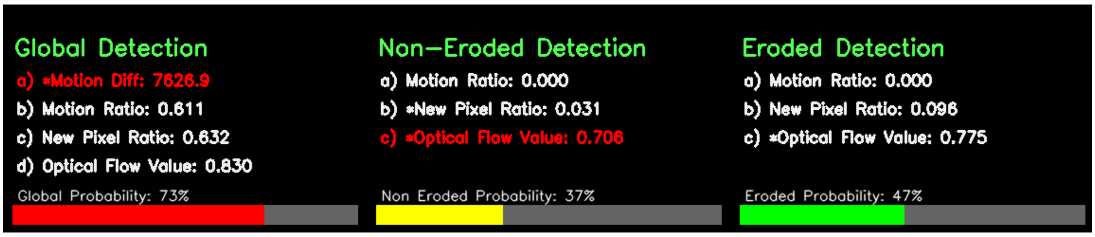

Original
Frame: 101
West Yorkshire Fire Service
Optical Flow
West Yorkshire Fire Service
Greyscale
West Yorkshire Fire Service
SMALL ROI EXPLOSION (Erode)
MHI
MEI
Eroded

Figure 5.2 Video Frame Tiles

As illustrated in Figure 5.2, six video frame tiles are displayed to the user. The upper row consists of the original video, the optical flow visualisation, and the greyscale video, while the lower row contains MHI, MEI, and the eroded video. The optical flow tile provides arrows of the motions that exist within the frame, highlighting the directional patterns of motion within the frame. The greyscale tile allows the user to view the video without disturbing the colour, so the scene can be viewed with greater clarity in terms of structural details. Meanwhile, the MHI and MEI tiles provide temporal motion information that is currently analysed by the system. Users can see the current motions being analysed by the system through these two tiles. Finally, the eroded tile illustrates the video that has undergone the erosion operation discussed earlier.



Global Detection
a) *Motion Diff: 7626.9
b) Motion Ratio: 0.611
c) New Pixel Ratio: 0.632
d) Optical Flow Value: 0.830
Global Probability: 73%

Non—Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.031
c) *Optical Flow Value: 0.706
Non Eroded Probability: 37%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.096
c) *Optical Flow Value: 0.775
Eroded Probability: 47%

Figure 5.3 Frames Details

As shown in Figure 5.3, the system implements three detection methods: global detection, non-eroded detection, and eroded detection. Global detection analyses motion across the entire frame, whereas non-eroded and eroded detection methods concentrate on localised regions of interest. Variables marked with an asterisk (e.g., *Motion Diff in Global Detection tile) indicate that a threshold has been applied. When the measured value exceeds the threshold, the corresponding variable is displayed in red.

In addition, the probability bar provides a visual indicator of explosion likelihood. It remains green and changes to yellow when one or more key variables turn red for the corresponding detection method. The probability bar turns red if any of the detection methods detect an explosion within the video, indicating that the calculated probability exceeds the defined threshold and that the corresponding variables also meet their respective threshold conditions. If any of the detection methods identify an explosion, the frames are automatically paused for users' verification purposes.

Furthermore, when explosions are detected by the non-eroded or eroded detection methods, bounding boxes are drawn on the greyscale tile to highlight the specific region where the explosion is likely occurring shown in Figure 5.2. In contrast, the global detection method does not display bounding boxes since its analysis considers the entire frame rather than just a localised region. This distinction means that global detection is more suitable for identifying large explosions that are clearly visible across the entire frame, whereas localised detection methods are more effective for identifying smaller explosions that may require bounding boxes for better visualisation.

5.4 Implementation Issues and Challenges

The proposed system is inevitably influenced by video quality factors such as low resolution, noise, and blurriness. These factors significantly impact the system's accuracy in detecting explosions. Specifically, noise is the key factor that poses a major challenge for the system, as it will cause motion detection to be hardly detected, increasing the difficulty for the system to accurately distinguish between actual explosions and irrelevant motion. This affects the overall performance of the system, increasing the false detection rate.

Another key challenge is the distance between the explosion and the CCTV camera. When an explosion occurs far from the camera, the system struggles to detect it accurately due to minimal motion change being detected in the footage. Explosions normally generate massive visual changes, but sometimes they appear as mild explosions, which only bring a minor motion in the video, increasing the detection difficulty. Conversely, if there are large objects such as trucks or cars passing close to the camera, a false alarm would be triggered due to excessive motion detected in the scene.

Meanwhile, one of the implementation issues is threshold balancing. Thresholds that are set too low cause large motions, such as passing vehicles, to trigger false alarms, whereas overly strict thresholds lead to missed detection of genuine but mild explosion events.

5.5 Concluding Remarks

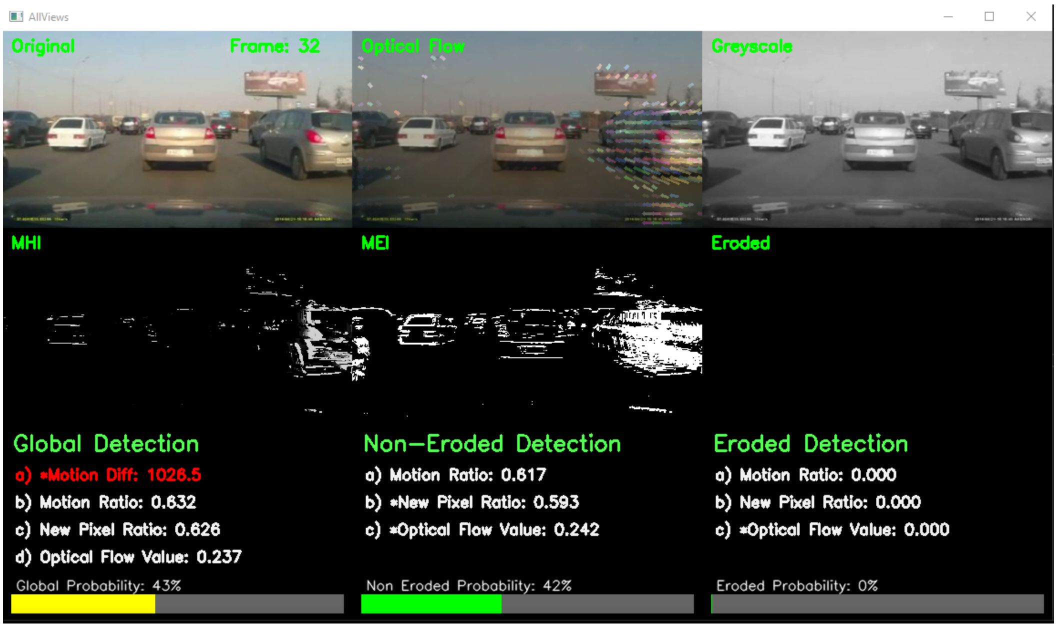
This project successfully implemented an explosion detection system using the C++ programming language and the OpenCV library. Multiple variables, including motion ratio, new pixel ratio and optical flow value, are designed to assist in identifying explosion events. Furthermore, three detection approaches, such as global, non-eroded, and eroded detection, provide a good combination to detect the explosion, improving detection accuracy under different conditions. Lastly, the analytic interface provides a clear and structured visualisation for users to interpret the system's operation and detection results with ease.

CHAPTER 6

System Evaluation and Discussion

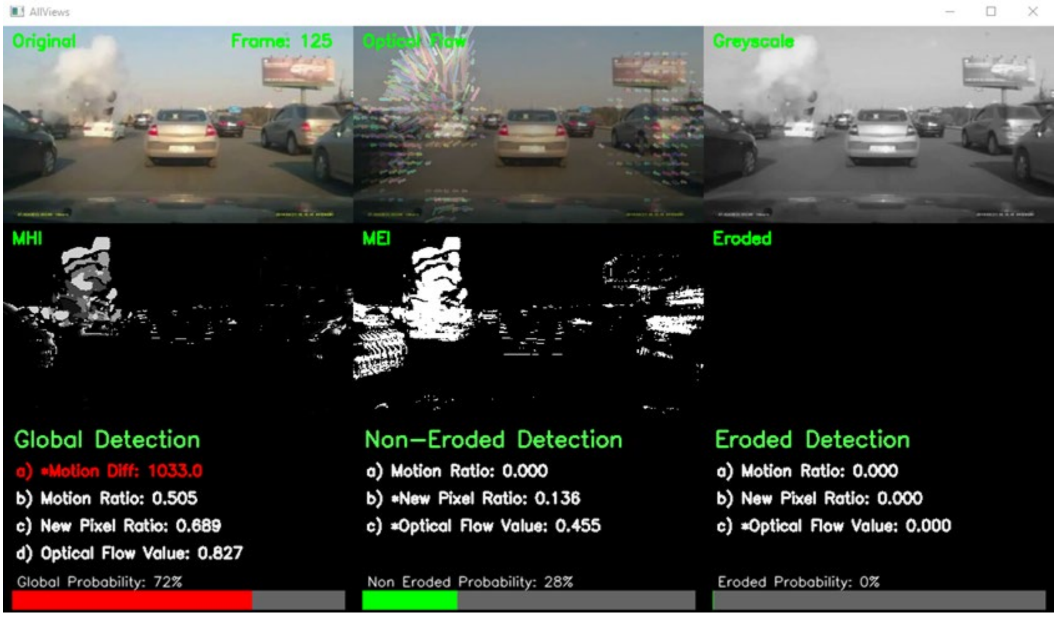
6.1 System Testing

6.1.1 Scenario 1



```
Global Detection
a) *Motion Diff: 1026.5
b) Motion Ratio: 0.632
c) New Pixel Ratio: 0.626
d) Optical Flow Value: 0.237
Global Probability: 43%

Non-Eroded Detection
a) Motion Ratio: 0.617
b) *New Pixel Ratio: 0.593
c) *Optical Flow Value: 0.242
Non Eroded Probability: 42%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%
```

Figure 6.1 Vehicles were moving on the Road



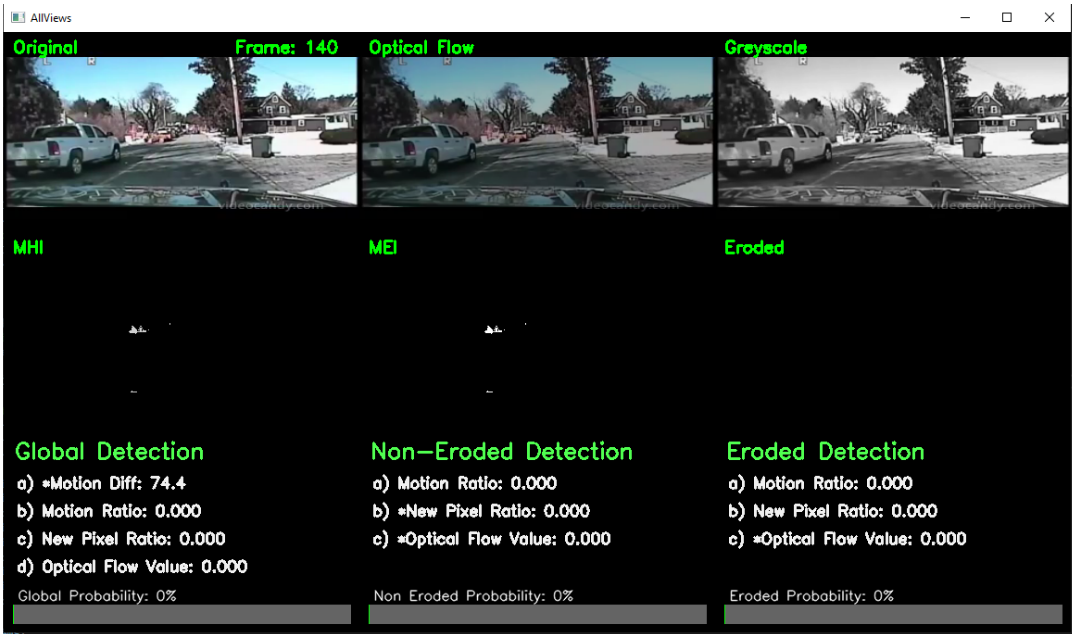
```
Global Detection
a) *Motion Diff: 1033.0
b) Motion Ratio: 0.505
c) New Pixel Ratio: 0.689
d) Optical Flow Value: 0.827
Global Probability: 72%

Non-Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.136
c) *Optical Flow Value: 0.455
Non Eroded Probability: 28%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%
```

Figure 6.2 Explosion happened on a moving vehicle

Initially, there were a few moving vehicles that appeared in the video, as seen in Figure 6.1. During this stage, the system did not detect any explosion events and continued its monitoring process. From the system's perspective, the motion generated by the vehicles was large, exceeding the *Motion Diff threshold in the global detection method. However, because of the consistent forward direction motion by the vehicles, the optical flow value remained low. Thus, false alarms were prevented.

Meanwhile, an explosion suddenly occurred on a white moving vehicle at the top-left corner of the scene, as shown in Figure 6.2. This sudden burst produces a significant motion. Moreover, the burst motion appeared in a region where no previous motion was present, leading to the rise of a new pixel ratio. It was also observed that the motion produced by the explosion event had inconsistent directions, which caused high optical low value. The high values of motion ratio, new pixel ratio and optical flow value generated a probability exceeding 70%, which enabled the system to accurately detect the explosion at an early stage.

6.1.2 Scenario 2

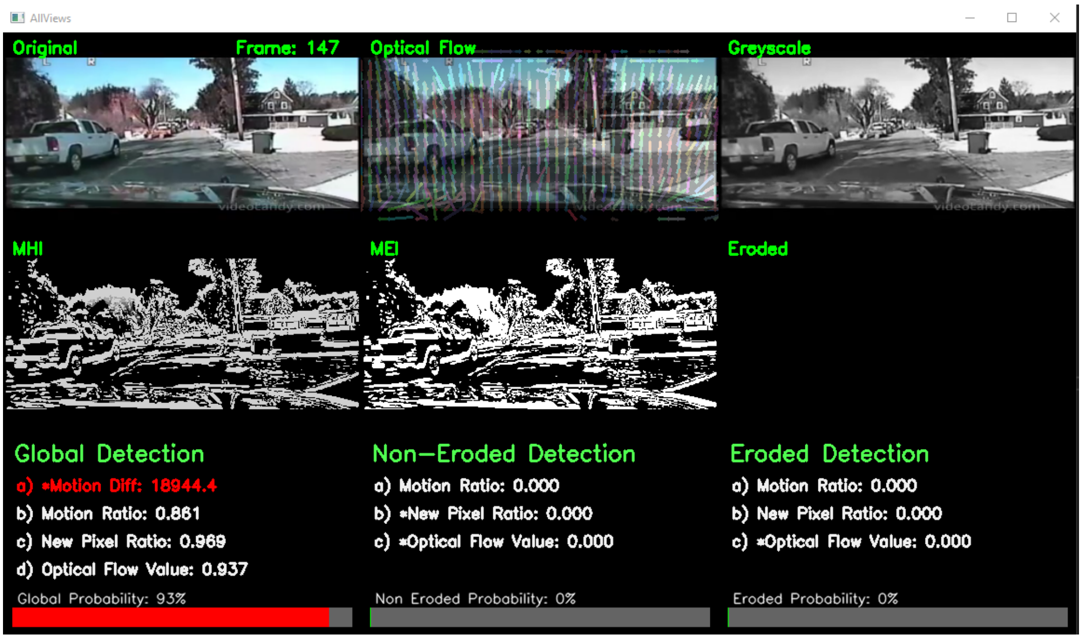


Global Detection
a) *Motion Diff: 74.4
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 0%

Non–Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%

Figure 6.3 Static Scene



Global Detection
a) *Motion Diff: 18944.4
b) Motion Ratio: 0.861
c) New Pixel Ratio: 0.969
d) Optical Flow Value: 0.937
Global Probability: 93%

Non–Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

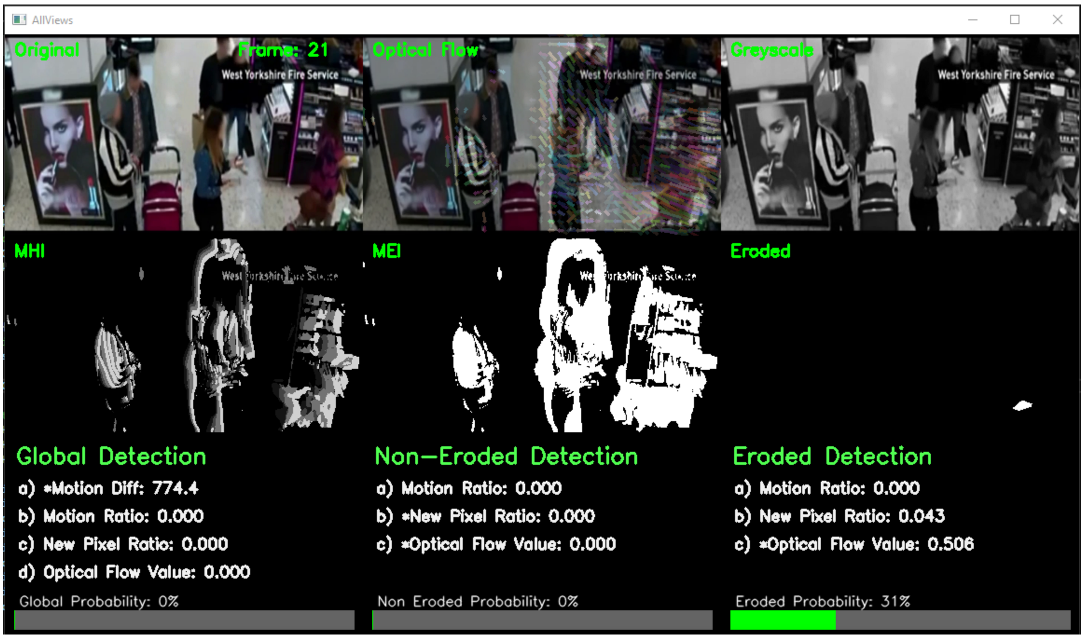
Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%

Figure 6.4 Explosion occurred nearby

Based on Figure 6.3, the scene was static from the beginning, and no visible motion could be observed at Frame 140. At this point, all detection methods showed zero values for motion ratio, new pixel ratio, and optical flow value, which correctly indicated the absence of any significant activity in the scene.

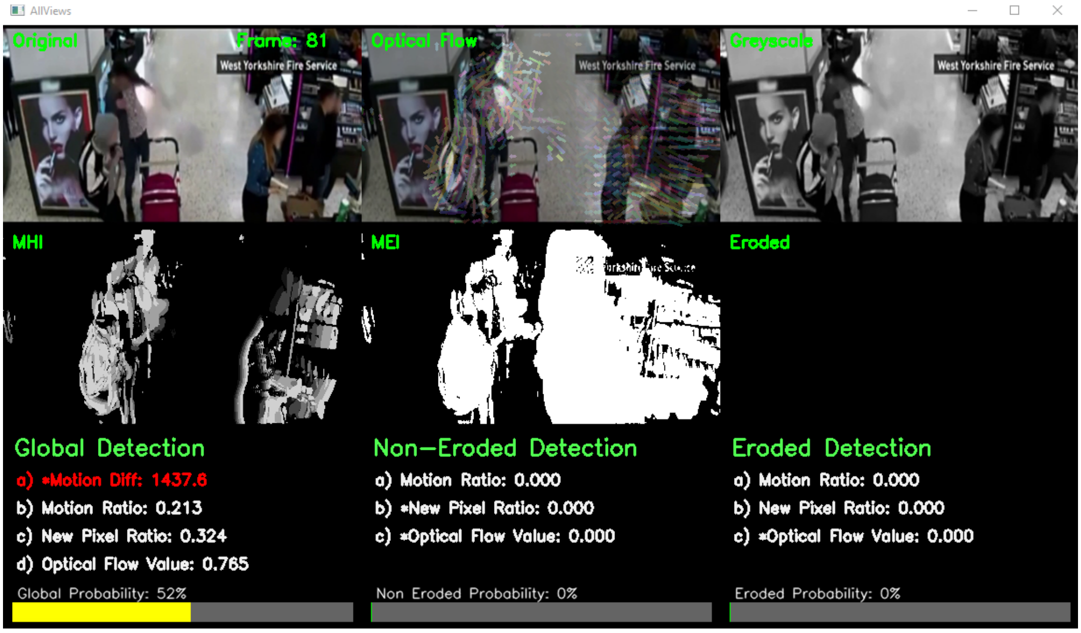
Nonetheless, a massive explosion suddenly occurred nearby at Frame 147. Within this frame, excessive motion was detected by the system. Due to the explosion, the three variables in the global detection method rose to very high values, resulting in a probability of 93%, which successfully triggered the alarm.

6.1.3 Scenario 3



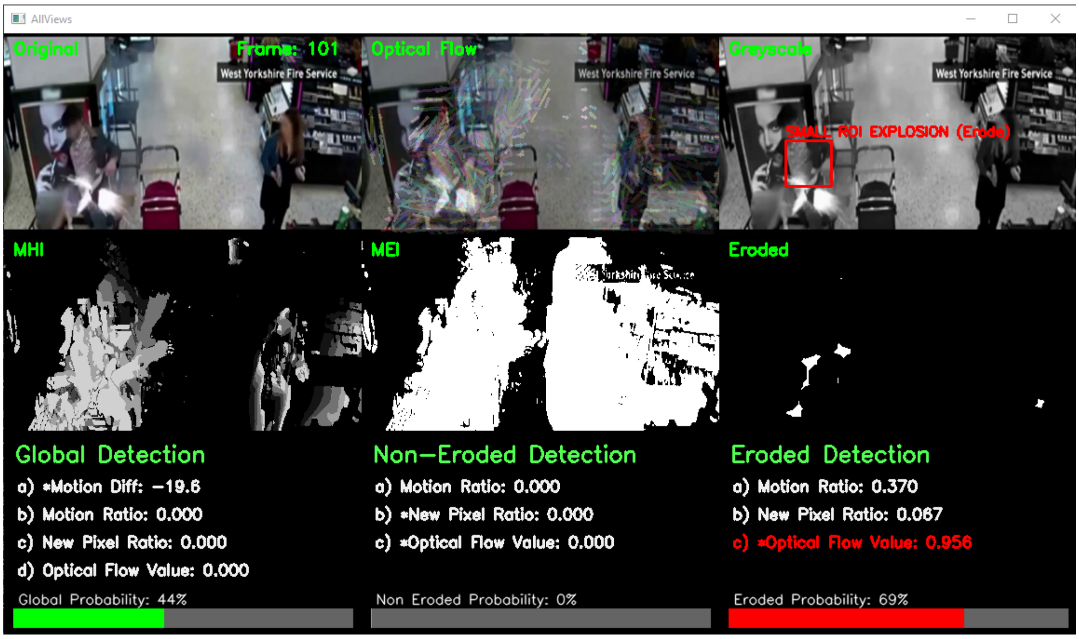
```
Global Detection
a) *Motion Diff: 774.4
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 0%

Non—Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.043
c) *Optical Flow Value: 0.506
Eroded Probability: 31%
```

Figure 6.5 Crowds were moving around



```
Global Detection
a) *Motion Diff: 1437.6
b) Motion Ratio: 0.213
c) New Pixel Ratio: 0.324
d) Optical Flow Value: 0.765
Global Probability: 52%

Non—Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%
```

Figure 6.6 A Sudden Spark Appeared in a Man’s Pocket


AllViews
Original
Frame: 101
West Yorkshire Fire Service
Optical Flow
West Yorkshire Fire Service
Greyscale
West Yorkshire Fire Service
SMALL ROI EXPLOSION (Erode)
MHI
MEI
Eroded

```
Global Detection
a) *Motion Diff: −19.6
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 44%

Non−Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.370
b) New Pixel Ratio: 0.067
c) *Optical Flow Value: 0.956
Eroded Probability: 69%
```

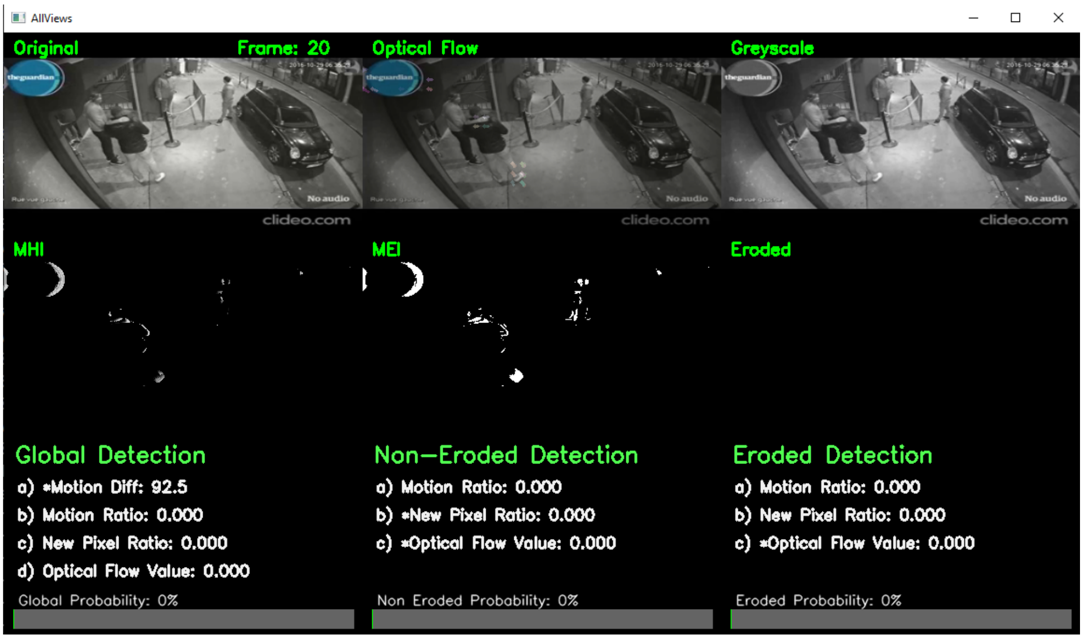
Figure 6.7 The explosion bursts with visible sparks, scattering around

According to Figure 6.5, crowds were moving around in a shopping mall. The activity naturally created a vast amount of motion. Nonetheless, the system was able to suppress this type of motion, as it was consistent and non-sudden because the system kept comparing the motion with the previous 0.25 seconds of motion. There were no false alarms produced in this scene.

Moving forward, a sudden spark was noticed in a man's pocket, as shown in Figure 6.6. However, the system was unable to determine it as an explosion at Frame 81. In Figure 6.7, the crowds started to run away in panic, creating a chaotic scene that made detection more challenging. Despite the chaotic scene, after the frame underwent the erosion operation, localised irregular motion remained visible in the eroded frame. This irregular motion caused a very high value in optical flow value, significantly increasing the probability. As it managed to exceed the optical flow value threshold and reached a probability value of 69%, the eroded detection method successfully detected the explosion and plotted a 50 x 50 bounding box on the greyscale frame.

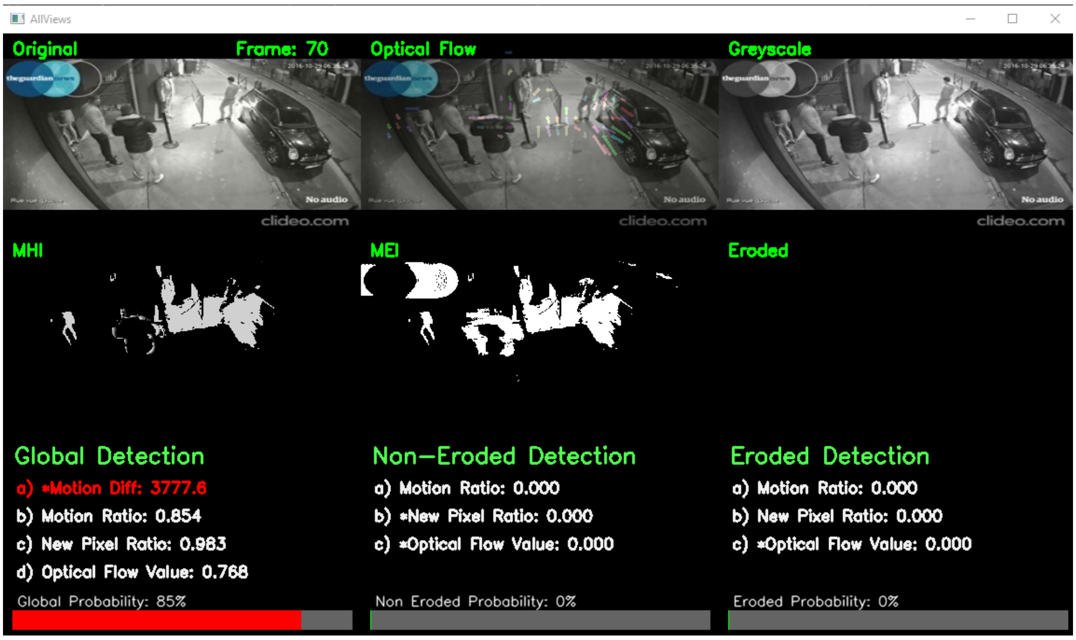
However, it was observed that the bounding box was not perfectly centred on the explosion. As the frame underwent the erosion operation, most of the surrounding motion was suppressed, but the intense light generated by the explosion created multiple contours in the eroded frame. The system selected the largest contour, which did not precisely correspond to the explosion centre. Nevertheless, since the detected motion still originated from the explosion, the detection can be considered genuine, even though the bounding box position was not correct.

6.1.4 Scenario 4



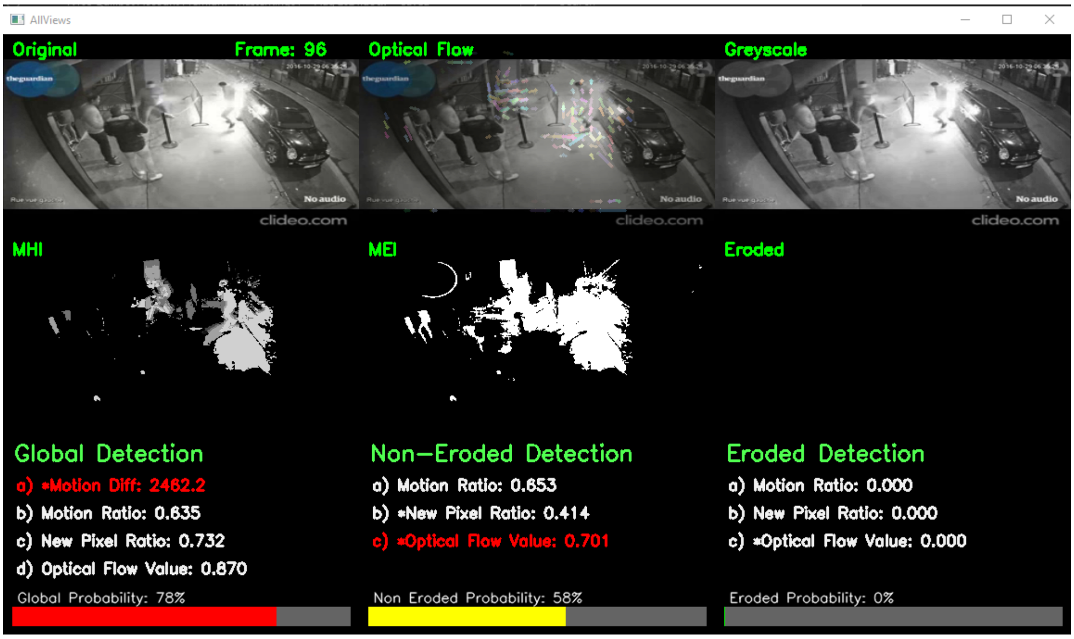
```
Global Detection
a) *Motion Diff: 92.5
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 0%

Non-Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%
```

Figure 6.8 A Few Men Were Chatting on the Street



```
Global Detection
a) *Motion Diff: 3777.6
b) Motion Ratio: 0.854
c) New Pixel Ratio: 0.983
d) Optical Flow Value: 0.768
Global Probability: 85%

Non-Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%
```

Figure 6.9 Light Generated from the Explosion



Global Detection
a) *Motion Diff: 2462.2
b) Motion Ratio: 0.635
c) New Pixel Ratio: 0.732
d) Optical Flow Value: 0.870
Global Probability: 78%

Non—Eroded Detection
a) Motion Ratio: 0.653
b) *New Pixel Ratio: 0.414
c) *Optical Flow Value: 0.701
Non Eroded Probability: 58%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%

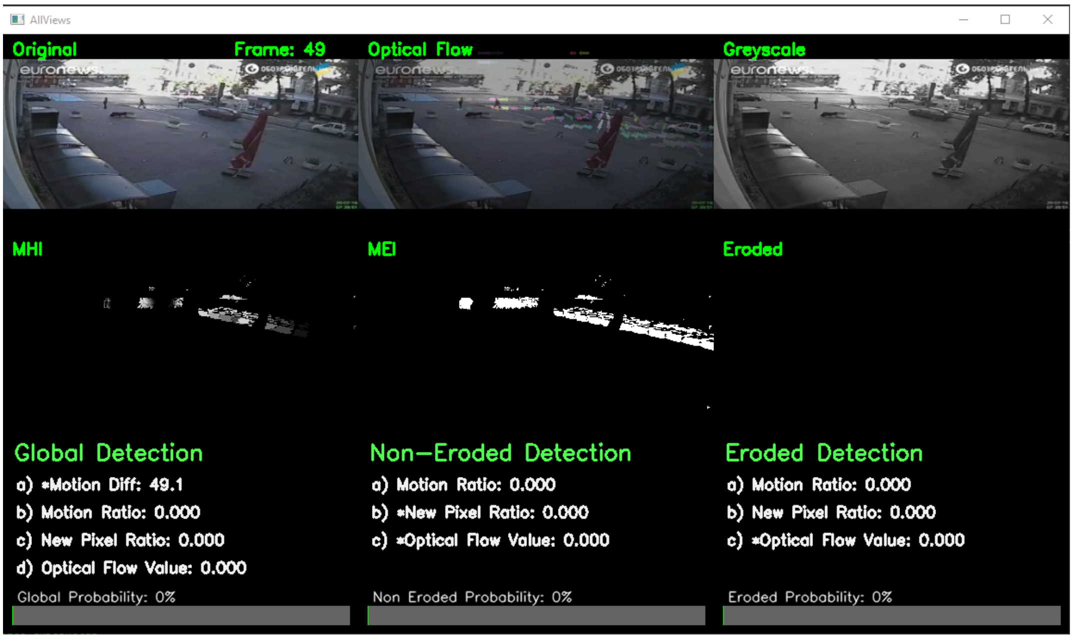
Figure 6.10 Explosion occurred inside a Man’s pocket

In Figure 6.8, a few men were observed chatting on the street. This activity introduced only minor motion, which could be said to be a static scene.

After a while, the system detected an explosion at Frame 70. It was noticed that an intense light was generated near the man standing on the far right based on Figure 6.9. This sudden flash of light caused the surrounding pixels to change drastically, leading to sharp increases in the motion ratio, new pixel ratio, and optical flow value. Consequently, the global detection probability reached 85%, surpassing the 70% threshold and triggering an alarm.

Later, a visible explosion occurred inside the man's pocket at Frame 96, as displayed in Figure 6.10. This event again caused a significant rise in all three variables, particularly in the global detection methods, resulting in another alarm being triggered. From this scenario, it can be concluded that the system is capable of detecting explosions at a very early stage.

6.1.5 Scenario 5

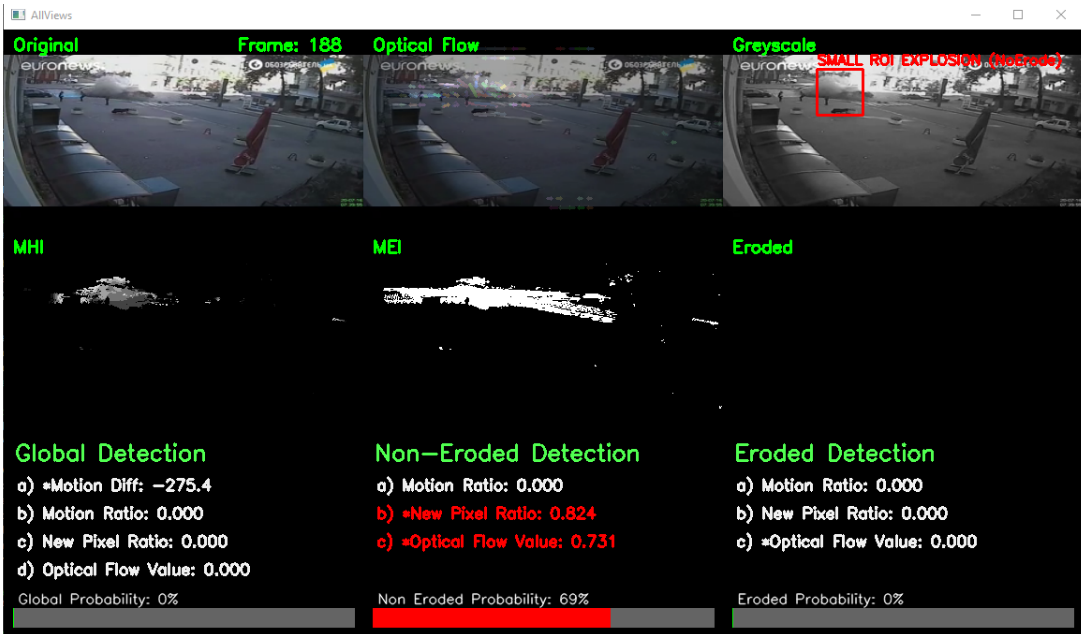


Global Detection
a) *Motion Diff: 49.1
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 0%

Non–Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Non Eroded Probability: 0%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%

Figure 6.11 A vehicle was moving on the road



Global Detection
a) *Motion Diff: –275.4
b) Motion Ratio: 0.000
c) New Pixel Ratio: 0.000
d) Optical Flow Value: 0.000
Global Probability: 0%

Non–Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.824
c) *Optical Flow Value: 0.731
Non Eroded Probability: 69%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%

Figure 6.12 The vehicle exploded

It was observed that a car was moving on the road in Figure 6.11. However, because the vehicle was located far from the camera, the motion generated was relatively small, resulting in low motion values across all detection methods.

Suddenly, Figure 6.12 showed the car exploded on the road. In this scenario, the system successfully detected the explosion through the non-eroded detection method, which focused on the localised motion in the MEI. Since this method concentrates only on a restricted region of interest, a 50 × 50 bounding box was drawn on the greyscale frame and was correctly positioned over the explosion.

It is observed that the motion ratio remained at 0.00 because the mild explosion produced less motion compared to the motion history accumulated over the previous 0.25 seconds. Nonetheless, the rise in the new pixel ratio (0.824) and the optical flow value (0.731) was sufficient to push the probability to 69%, allowing the non-eroded method to trigger the detection accurately.

6.2 System Result

Video Number	Explosion Detected	False Alarm	Explosion Size	Remark
001	Yes	0	Massive	-
002	Yes	0	Massive	-
003	Yes	0	Massive	-
004	Yes	1	Massive	-
005	No	1	Massive	-
006	Yes	0	Mild	-
007	Yes	0	Massive	-
008	Yes	0	Massive	-
009	Yes	0	Massive	-
010	Yes	0	Massive	-
011	No	1	Massive	A person threw an object toward a moving bus, causing a sudden burst of white smoke near the bus. The explosion motion overlapped with the continuous motion of the bus.
012	Yes	0	Massive	-

013	Yes	1	Massive	-
014	No	0	Mild	Explosion occurred too far from the CCTV; the motion was too small to detect.
015	Yes	0	Massive	-
016	No	0	Massive	Explosion happened inside a building, hardly viewable
017	No	0	Mild	A small fire emerged from a motorcycle, but no significant motion was observed to trigger detection.
018	Yes	0	Mild	-
019	No	1	Mild	Small explosion appears in a man pocket; low-intensity explosion and minimal motion registered in the scene.
020	Yes	0	Mild	-
021	Yes	0	Massive	-
022	Yes	0	Massive	-
023	Yes	0	Massive	-
024	Yes	0	Massive	-
025	Yes	0	Massive	-
026	Yes	2	Massive	-
027	No	0	Mild	Small explosion happens on the man hand. The explosion motion is stacked with the man motion, leading to the system fail to detect.
028	Yes	0	Massive	-
029	Yes	0	Massive	-
030	No	0	Mild	Flame visible from a vehicle during petrol refuelling, minimal motion detected.
031	Yes	0	Massive	-
032	Yes	0	Massive	-

033	No	0	Massive	Video quality is poor, system unable to detect any valid motion.
034	Yes	0	Massive	-
035	Yes	0	Massive	-
036	Yes	0	Massive	-
037	Yes	0	Massive	-
038	Yes	0	Massive	-
039	Yes	1	Massive	-
040	Yes	1	Massive	-
041	Yes	1	Massive	-
042	Yes	2	Massive	False alarms occured due to poor video quality, low lighting conditions,
043	Yes	0	Massive	-
044	Yes	0	Massive	-
045	Yes	0	Massive	-
046	Yes	0	Massive	-
047	Yes	0	Massive	-
048	Yes	0	Massive	-

Table 6.1 Results of Testing

Based on table 6.1, out of a total of 48 explosion scenes, 40 of them involved massive explosion events, while the remaining 8 were mild explosions. The system successfully detected 39 cases, resulting in a detection accuracy of 81.25%. This indicates that the designed system is capable of identifying the majority of explosion events.

However, 12 false alarms were recorded. Several of the false alarms were triggered by factors such as poor video quality or low-lighting conditions, which introduced noise into the motion analysis process and confused the detection logic. This caused the system to misclassify some of the normal motion as explosion events.

In terms of missed detections, 9 explosion events were not detected by the system. Among these, 4 were large explosion scenes that generated significant motion and could be clearly observed by the human eye. Nevertheless, certain circumstances made detection more difficult, such as the explosion that occurred inside a building in video 016, where the explosion motion was mostly blocked by the building, and video 033, where poor video quality prevented the system from detecting any valid motion. On the other hand, 5 mild explosion scenes were undetected by the system. This is likely due to their subtle visual cues and limited motion impact within the frame.

In summary, 90% of massive explosion scenes were successfully detected with the system, showing a good capability in detecting these massive explosion scenes. Conversely, it only managed to detect 3 out of 8 of the mild explosion scenes within the dataset. Mild explosions often generate only minimal motion, which is different from common explosion events. As a result, the system lacks the sensitivity needed to consistently detect mild explosion events.

6.3 Comparison of Result

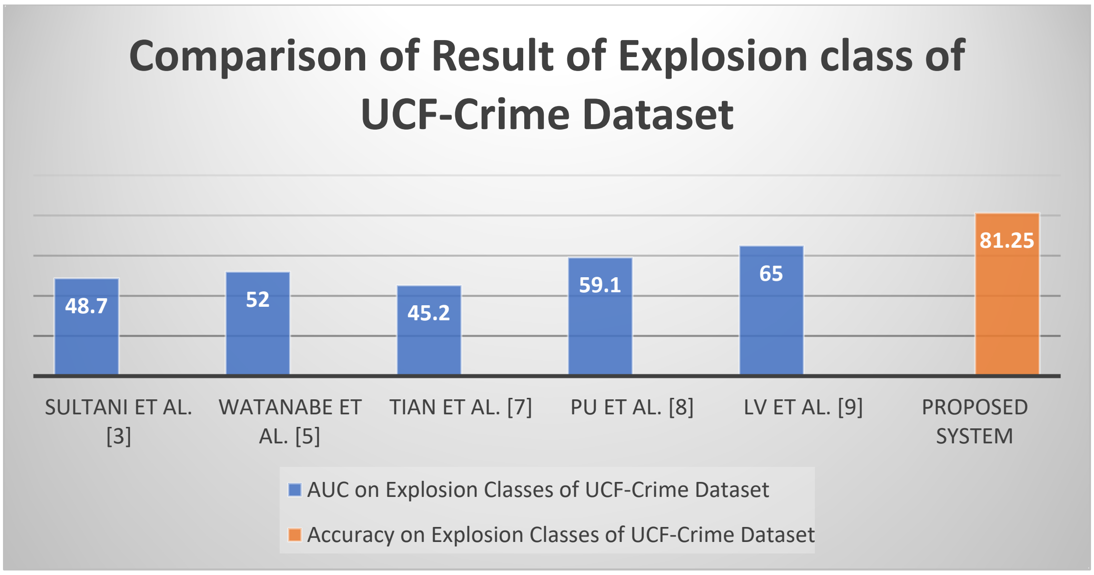


Figure 6.13 Comparison of Explosion Detection Accuracy on UCF-Crime Dataset

Figure 6.13 presents a comparison of explosion detection accuracy on the UCF-Crime dataset between the proposed system and several previous works discussed in chapter 2. Since there are no directly related works focusing solely on explosion detection on the UCF-Crime dataset, the referenced studies mainly focus on anomaly detection, where some of them provide the class-wise results. Among them, Sultani et al. [3], Watanabe et al. [5], and Tian et al. [7] achieved AUC values of 48.7%, 52%, and 45.2%, respectively. More recent approaches, such as Pu et al. [8] and Lv et al. [9], reported higher AUC values of 59.1% and 65%.

On the other side, the proposed system successfully achieved an accuracy of 81.25%. This result demonstrates the effectiveness of combining multiple detection approaches with motion-based variables such as motion ratio, new pixel ratio, and optical flow value.

It should be noted that previous works reported frame-level performance using AUC, while the proposed system reports video-level accuracy. Therefore, although the proposed system shows satisfactory detection accuracy for explosions, a direct one-to-one comparison with existing methods is not applicable. Nonetheless, the results indicate that the proposed system performs reliably for the explosion class.

6.4 Project Challenges

One of the key challenges faced in this project was the lack of directly related previous works. Most existing research primarily focused on anomaly detection rather than specifically on explosion detection. As a result, the proposed system stands as a pioneer attempt to employ a motion-based approach for detecting explosion events in surveillance videos.

In addition, all motion-based variables and detection methods introduced in the proposed system had to be reviewed and developed independently, since no prior references were available to provide guidance. This process involved extensive experimentation and iterative improvements to achieve effective performance.

Another major challenge was the design of detection logics that could work reliably for both massive and mild explosions while minimising false alarms. This required careful balancing of thresholds and pre-set conditions to ensure the system remained sensitive to explosion events without being misled by irrelevant background motion such as moving vehicles.

Finally, this project mainly focused on a motion-based methodology, deep learning techniques were intentionally excluded. While deep learning has become the dominant approach in recent computer vision research, avoiding it caused an additional challenge, as it required the system to achieve competitive performance using handcrafted features and rule-based detection methods instead.

6.5 Objectives Evaluation

I. To develop a real-time explosion detection video surveillance system

The proposed system successfully integrates image processing techniques that can detect explosion events in real time. Therefore, traditional video surveillance systems can be replaced with the proposed system, thereby reducing the need for human monitoring.

II. To detect significant visual changes in the environment

The proposed system is able to detect significant visual changes in the video footage. With these capabilities, explosions can be recognised quickly and effectively with the proposed system. However, it is also noted that some of the significant visual changes in the video footage are not produced by explosions, which could occasionally lead to false alarms.

III. To analyse temporal motion patterns for identifying explosion characteristics

The proposed system successfully integrated MHI and MEI to capture and present motion in the video footage. These motions are further analysed in motion-based variables to detect explosions, including motion ratio, new pixel ratio and optical flow value.

IV. To minimise false explosion event detections

The motion-based variables implemented in the proposed system are able to detect explosion events with minimal false alarms. Specifically, the motion ratio is compared against the previous 0.25 seconds of motion, the new pixel ratio focuses only on newly activated pixels, and the optical flow emphasises irregular motion patterns characteristic of explosions rather than consistent motion generated by vehicles or pedestrians. Meanwhile, the system only generated 12 false alarms in the system testing stage.

6.6 Concluding Remark

The proposed system demonstrated the feasibility of using motion-based methods for real-time explosion detection in video surveillance. The results obtained by the proposed system indicated that the system is capable of identifying explosions, achieving an overall detection accuracy of 81.25%. In particular, its performance in detecting massive explosions was impressive, as it reached a success rate of 90%. Nonetheless, the system capability was limited when it came to detecting mild explosions due to their limited characteristics.

Overall, the project objectives were successfully achieved. The system provides an effective alternative to traditional surveillance by reducing reliance on human monitoring and offering automated detection capabilities.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

This project addresses the challenges of detecting explosion events using the traditional video surveillance systems, which often rely heavily on human operators. Meanwhile, human operators are usually inefficient due to fatigue and oversight, leading to a missed explosion detection and causing disastrous outcomes. To overcome these limitations, this project proposes an intelligent explosion detection system leveraging advanced computer vision and image processing techniques.

The developed system integrates MHI and MEI to effectively analyse motion patterns. Results demonstrated that the system is able to identify valid motion accurately. Despite these motions being associated with regular activities, the proposed system is differentiating effectively between normal events and explosions. Through thorough testing with various scenarios, the system proved its ability by accurately detecting explosion events within the early stage, allowing instant notifications and timely responses.

Other than that, the proposed method integrated three motion-based variables in detecting explosions. Specifically, two of them are novel features: motion ratio and new pixel ratio, which significantly enhanced the accuracy of the detection process. The motion ratio calculates the difference between current motions and the previous 0.25-second motions in ratio, reflecting the chaotic characteristic of the explosions. Then, the new pixel ratio measures the proportion of newly activated motion pixels in the current frame compared to the previous 0.20 seconds, ensuring that the detected motion is sudden rather than constant, such as vehicle motions. Meanwhile, the optical flow value is utilised in our motion-based variables, as explosion events typically generate irregular motion directions. This variable ensures that the biggest motions within the frame will contribute heavily to the probability if it is irregular motion.

In terms of detection methods, there are three detection methods implemented in the proposed system, including global detection, non-eroded detection, and eroded detection. Global detection monitors motion across the entire frame, ensuring that large-scale explosions are captured. Non-eroded detection and eroded detection focus on localised regions within the scene. Non-eroded detection analyses motion regions exist in the MEI, while eroded detection analyses motion regions in the MEI after an erosion operation to remove noise.

By combining these motion-based variables and detection methods, the proposed system is capable of performing its desired function. It successfully achieves a result of 81.25% accuracy, detecting 39 cases out of 48 explosion cases. It is notable that the best previous works reported only 65% AUC on the UCF-Crime dataset, indicating relatively weak performance in detecting explosion events. Since the proposed system does not apply deep learning or machine learning methods, video-level accuracy was adopted as the primary evaluation metric. Nevertheless, the proposed system performed well and produced promising results.

In summary, the proposed intelligent explosion detection system exhibits its capability as a reliable, accurate, and efficient solution for a real-time explosion detection video surveillance system. Its successful performance across diverse scenarios emphasises its ability to provide early alarm to authorities whenever an explosion happens, diminishing the awful damage caused by explosion events. As one of the pioneering motion-based explosion detection systems, this work demonstrates that motion-based techniques can be effectively utilised for detecting explosions.

7.2 Recommendation

For future work, the proposed system can be tested using larger video collections from different datasets and anomaly classes. It should also be tested under diverse conditions, such as varying lighting and environmental settings, to validate performance outside of controlled datasets.

Other than that, hybrid approaches could be explored by combining the proposed motion-based method with deep learning techniques to further enhance detection accuracy and reduce false alarms. Whenever an explosion is detected by a motion-based system, a deep learning model could be applied as a secondary verification stage to confirm whether the detected event is indeed a valid explosion. Finally, it can be integrated with real-world alerting mechanisms, such as SMS or mobile notification, enabling rapid response in real-world surveillance environments.

REFERENCES

[1] S. White, “CCTV Cameras by Countries & Cities (2022 Guide) - Upcoming Security,” *UPCOMING SECURITY*, Mar. 31, 2022. https://upcomingsecurity.co.uk/security-guides/cctv-camera-guides/cctv-by-country/ (accessed Aug. 21, 2024).

[2] “Piper Alpha: How we survived North Sea disaster,” *BBC News*, Jul. 06, 2013. Accessed: Aug. 22, 2024. [Online]. Available: https://www.bbc.com/news/uk-scotland-22840445.

[3] W. Sultani, C. Chen, and M. Shah, “Real-World Anomaly Detection in Surveillance Videos,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6479–6488, 2018.

[4] V. Singh, S. Singh, and P. Gupta, “Real-Time Anomaly Recognition Through CCTV Using Neural Networks,” *Procedia Computer Science*, vol. 173, pp. 254–263, 2020.

[5] Y. Watanabe, M. Okabe, Y. Harada, and N. Kashima, “Real-World Video Anomaly Detection by Extracting Salient Features in Videos,” *IEEE Access*, vol. 10, pp. 125052–125060, 2022.

[6] P. Wu and L. Jing, “Learning Causal Temporal Relation and Feature Discrimination for Anomaly Detection,” *IEEE transactions on image processing*, vol. 30, pp. 3513–3527, Jan. 2021.

[7] Y. Tian, G. Pang, Y. Chen, R. Singh, Johan Verjans, and G. Carneiro, “Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning,” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021.

[8] Y. Pu, X. Wu, L. Yang, and S. Wang, “Learning Prompt-Enhanced Context Features for Weakly-Supervised Video Anomaly Detection,” *IEEE Transactions on Image Processing*, vol. 33, pp. 4923–4936, Jan. 2024.

[9] H. Lv, Z. Yue, Q. Sun, B. Luo, Z. Cui, and H. Zhang, “Unbiased Multiple Instance Learning for Weakly Supervised Video Anomaly Detection,” *Thecvf.com*, pp. 8022–8031, 2023.

[10] S. Abusaleh, A. Mahmood, K. Elleithy, and S. Patel, “A Novel Vision-Based Classification System for Explosion Phenomena,” *Journal of Imaging*, vol. 3, no. 2, p. 14, Apr. 2017.

[11] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[12] A. Abdelbaky and S. Aly, “Human action recognition using short-time motion energy template images and PCANet features,” *Neural Computing and Applications*, vol. 32, no. 16, pp. 12561–12574, Jan. 2020.

[13] M. Vafadar and A. Behrad , “Human Hand Gesture Recognition Using Motion Orientation Histogram for Interaction of Handicapped Persons with Computer ,” in *Image and Signal Processing*, Berlin, Heidelberg: Springer, 2008, pp. 378--385.

[14] F. Murtaza, Muhammad Haroon Yousaf, and S. A. Velastín, “Multi-view Human Action Recognition Using 2D Motion Templates Based on MHIs and Their HOG Description,” *Iet Computer Vision*, vol. 10, no. 7, pp. 758–767, Jun. 2016.

[15] E. Tsironi, P. Barros, C. Weber, and S. Wermter, “An Analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for Gesture Recognition,” *Neurocomputing*, vol. 268, pp. 76–86, Dec. 2017.

[16] X. Fan and T. Tjahjadi, “A Dynamic Framework Based on Local Zernike Moment and Motion History Image for Facial Expression Recognition,” *Pattern Recognition*, vol. 64, pp. 399–406, Apr. 2017.

POSTER

VIDEO SURVEILLANCE: EXPLOSION DETECTION

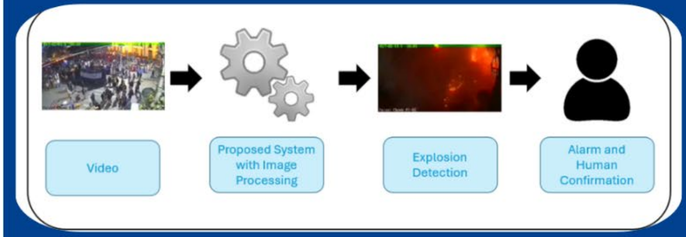
INTRODUCTION

An explosion is a sudden release of energy causing fire, shockwaves, and severe structural damage. Early detection is crucial to reduce harm, enable quick response, and protect lives.

OBJECTIVES

- To develop a real-time explosion detection video surveillance system
- To detect significant visual changes in the environment
- To analyse temporal motion patterns for identifying explosion characteristics
- To minimise false explosion event detections

METHODOLOGY


Video
Proposed System with Image Processing
Explosion Detection
Alarm and Human Confirmation

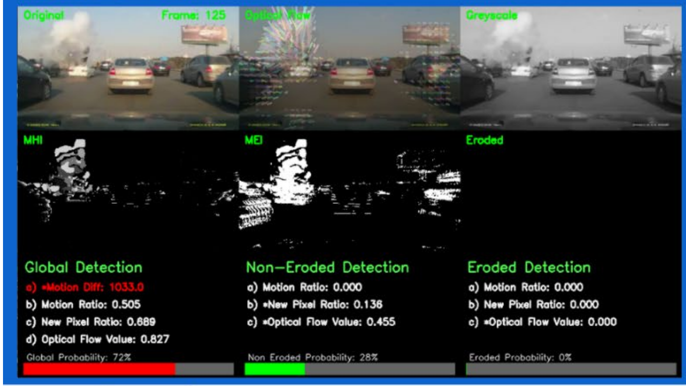
KEY FEATURES

✓ Self-Operating

✓ Real-Time

✓ Accurate

RESULT



```
Global Detection
a) *Motion Diff: 1033.0
b) Motion Ratio: 0.505
c) New Pixel Ratio: 0.689
d) Optical Flow Value: 0.827
Global Probability: 72%

Non-Eroded Detection
a) Motion Ratio: 0.000
b) *New Pixel Ratio: 0.136
c) *Optical Flow Value: 0.455
Non Eroded Probability: 28%

Eroded Detection
a) Motion Ratio: 0.000
b) New Pixel Ratio: 0.000
c) *Optical Flow Value: 0.000
Eroded Probability: 0%
```

CONCLUSION

The proposed system detects explosion events in real-time using image processing techniques. Its autonomous and accurate operations reduces human error, enabling faster response and minimizing explosion damage.

UNIVERSITY TUNKU ABDUL RAHMAN
BACHELOR OF COMPUTER SCIENCE (HONOURS)
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

STUDENT: LEE SHAO YUAN
SUPERVISOR: PROF.LEUNG KAR HANG