Video Surveillance: Anomaly Action Detection at front yard

BY

Lee Yong Jin

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)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Video Surveillance: Anomaly Action Detection at front yard

Academic Session: JAN 2024

I LEE YONG JIN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
22A, Jalan Harmoni 11/5,
Taman Harmoni,
43500, Semenyih, Selangor

Leung Kar Hang
Supervisor's name

Date: 25/04/2024

Date: 25 April 2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 25/04/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***LEE YONG JIN*** (ID No: ***20ACB01393***) has completed this final year project entitled "*Video Surveillance: Anomaly Action Detection at front yard*" under the supervision of Prof Dr. Leung Kar Hang (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology, and Ts Dr Mogana a/p Vadiveloo (Co-Supervisor)* from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*Lee Yong Jin)*

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Video Surveillance: Anomaly Action Detection at front yard**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

lee

Signature : ______________________

Name : __Lee Yong Jin_____________

Date : __25/ 04/ 2024_____________

ACKNOWLEDGEMENTS

I would like to express utmost appreciation to my supervisors, Prof. Dr. Leung Kar Hang who has given me the guidance and suggestion throughout the course of this project. His patience and invaluable experience in computer vision have helped me in the development of this project and kept the project in the right direction.

I would also like to thank my family and friends for their support and love throughout the course. Their encouragement has kept me motivated during my hard times.

ABSTRACT

Surveillance system has become increasingly common as a safety measure to enhance the security of the houses and properties. Anomaly detection plays a vital role in such surveillance system because relying on human supervision will be a waste of time and labor force. Thus, a lot of efforts have been put into this field of study. This project proposed a novel fire detection strategy and implemented it into a workable system. The approach of this project differs from many of the general strategy of anomaly detection which is to use deep learning model to learn the structure and pattern of normal events. However, anomalies do not have a clear definition which is what makes anomaly detection a challenging task. In the context of front yard surveillance, anomalies could be loitering, fighting, explosion, arson, and other suspicious activities. Hence, in order to detect such anomalies more accurately, the focus of this study has been narrowed down to tackle the arson. The system has shown its capability to detect fire within 1 second and with high accuracy by only utilizing the motion information and brightness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MIL	Multiple Instance Learning
CNN	Convolutional Neural Network
YOLO	You Only Look Once
DeepSort	Deep Simple Online Realtime Tracking
GAN	Generative Adversarial Network
sRNN	Stacked Recurrent Neural Network
MEI	Motion Energy Image

Chapter 1

Introduction

In this chapter, we present the problem statement and motivation of our research, the scope of the project as well as the objectives to be achieved. We also outline a brief overview and information on the background of this study. Lastly, we present the contribution of this project to the society.

1.1 Background Information

Anomaly detection is a field of computer vision that has been studied extensively. Many researchers had proposed different approaches in order to tackle this problem. The general definition of anomaly would be anything that deviates from the typical patterns in a provided dataset. In the context of front yard video surveillance, anomaly refers to the unusual or suspicious actions that might suggest a criminal attempt such as vandalism, break in, fighting and loitering.

Although the use of deep learning algorithms had greatly improved the accuracy and robustness of anomaly detection, it is still a challenging task. The difficulty of anomaly detection lies in the variability of anomaly. There is no clear boundary between normal and abnormal events, and it is impossible to define all types of anomalies. Furthermore, the criminals might adapt their behaviors to avoid the system's detection which made anomaly detection an even more difficult task as the anomalies change and evolve over time. Another factor that contributes to the difficulty of anomaly detection would be the lack of anomaly data because anomalous events are rare. Thus, it is more difficult to obtain large amounts of anomaly data.

1.2 Problem Statement and Motivation

Nowadays, surveillance camera is used everywhere in the public areas for security purposes. With the concept of smart city, the number of surveillance cameras will continue to increase. Due to the advancement of technology, front yard surveillance camera has become common in many houses as a security measure to safeguard the house and family members. Manual monitoring of surveillance camera footage would be a waste of time and labour because anomalous events seldom occur. This is where anomaly detection system comes into play. Anomaly detection system that automatically detects unusual activities or potential threats and raise the alarm to the house owner would eliminate the need for human intervention on surveillance cameras. A great design with utilization of computer vison techniques is required in order to build an anomaly detection system that can detect anomalies accurately and in real time.

The aim of this research is to develop a system that is able to detect anomalies, more specifically arson promptly. This is to enhance the safety for properties and residents by detecting fire with a robust system so that action could be taken before the situation worsen. By developing an automated detection system, the need for labour and manual monitoring can be reduced. This ensures a safer neighborhood and makes the surveillance process much more efficient and cost effective.

1.3 Research Objectives

The objectives of the project are:

- **To develop a fully automated surveillance system to detect fire.**
  The model developed will be able to detect unusual pattern which indicates fire in surveillance cameras and raise alarm. The system should run fully automated without user input.
- **To model the changes in MEI.**
  The system will calculate and model the motion in MEI. MEI represents all the motion over a certain period of time. Frame subtraction will be used to calculate the motion or changes. The motion or changes can then be used to identify potential anomaly.
- **To detect the fire within 1 second from the occurrence**
  The system will detect fire and raise the alarm promptly, or within 1 second of the start of fire. The reaction of the system is important as it could minimize the damage caused by the fire.

1.4 Project Scope and Direction

The scope of this project is to develop a model that automatically detects anomalies in surveillance footage and raise the alarm when potential anomalies is detected. The information such as changes across frames and brightness or intensity would be utilized to develop a robust anomaly detection model.

To improve the accuracy and reliability of the system, the scope of this project has been narrowed down to focus on detecting arson. By looking at specific types of anomalies, context information from these anomalies can be useful in extracting relevant features from the data which will improve the accuracy of the detection. For example, arson would cause changes in the intensity of the pixels with a certain pattern. The changes could be measured to detect such anomalies. Other than that, region of fire is always bright which could help to filter out the possible region.

The dataset that is used in this project is UCF_Crime dataset, arson category. The proposed model will be built and evaluated based on this dataset.

1.5 Contributions

This project proposed a novel anomaly detection model that is able to detect fire based on its unique pattern. The system detect fire based only on the changes and motion without relying on the colour information as the colour information are not always reliable due to the lighting condition.

The idea is then being implemented into a system that is able to detect fire automatically given an input video. The system will locate the fire temporally which is the time that the fire occurs. In addition to that, the system is expected to make decision fast, the fire will be detected within 1 second of occurrence.

The proposed model utilizes context information to specifically detect arson. This is different from many of the current approaches in anomaly detection, which is using deep learning model to learn the pattern of normal events and detect the videos that deviates from the learnt pattern as anomalies. With the context of specific types of anomalies, the system can make a more accurate prediction and reduce the rate of false alarm.

1.6 Report Organization

The rest of the report is organized into the following chapters: Chapter 2 Literature Review, Chapter 3 System Methodology, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion, Chapter 7 Conclusion and Recommendations. In Chapter 2, the previous works in related field, mainly anomaly detection are studied and reviewed. Next, the main concept and the overview of the system will be summarized in Chapter 3. The detail system component and modules will be explained in Chapter 4. Chapter 5 will present the implementation details such as software, hardware, and the operation of the actual system. Chapter 6 will be the experimental results of the system. Lastly, the report will be concluded in chapter 7.

Chapter 2
Literature Review

2.1 Previous Works on Anomaly Detection

In the past, anomaly detection had been approached in many different ways. The general principle is that the abnormalities are unexpected behaviours or events that do not conform to the normal behaviour. The common strategy that was used would be finding reconstruction error using autoencoder. The autoencoder will compress the input data and then try to reconstruct it back. The idea is to train the autoencoder with only normal data, when anomalous videos are being input during detection, the reconstruction error will be high as the autoencoder was not trained on it. One of the approaches would be handcrafted features approach where the relevant features of videos are being extracted and input into the machine learning algorithm. However, this requires manual feature extraction which might require domain-specific knowledge and does not generalize well. Hence, deep learning method such as rNN and ConvLSTM has been widely used as it can extract the important features automatically and learn the pattern of the data. Deep learning methods achieved state-of-the-art performance in anomaly detection as it is more robust in handling complex problems and high dimensional data.

2.1.1 Real-world Anomaly Detection in Surveillance Videos. [1]

Sultani et al [1] proposed a deep learning model that predicts anomaly scores which was learned using MIL and ranking loss function. The difficulty in obtaining temporal annotations for anomaly in videos was tackled by using only video level labels. The videos with anomalies were represented as positive bag and normal videos were represented as negative bag. Each of the videos were divided into multiple video segments and each of the segments represented a bag instance. The C3D features of the video segments were extracted using a pre-trained convolution network. These features were then used to train a 3-layer FC neural network. The flow diagram of the proposed model is as shown in Figure 2.1.

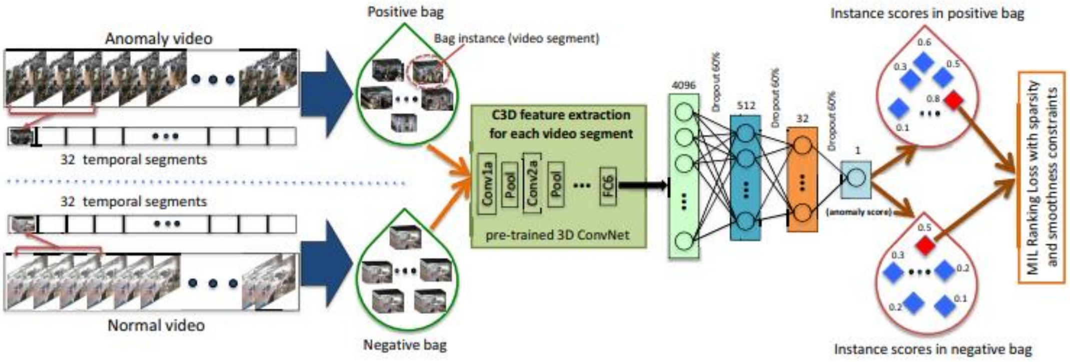


Figure 2.1 Flow diagram of MIL [1]

Anomaly scores were computed for each of the instances in the bags. High anomaly score indicates high possibility of anomalies in the respective instances. The instance with highest anomaly score in positive bag was most likely to be the true positive whereas instance with highest anomaly score in negative bag might be a false positive. Hence, the proposed ranking loss function was used to maximize the anomaly score difference between positive and negative instances. The author incorporated sparsity and smoothness constraints on the instances' anomaly score in order to address the temporal structure of anomalous videos which was disregarded by the loss function.

Method	AUC
Binary classifier	50.0
Hasan *et al.* [2]	50.6
Lu *et al.* [3]	65.51
Proposed w/o constraints	74.44
Proposed w constraints	**75.41**

Table 2.1 AUC comparison between various approaches [1]

The performance of the proposed model was being evaluated using receiver operating characteristic (ROC) curve and area under curve (AUC). The author had benchmarked the proposed model against binary SVM classifier, dictionary based approach proposed by Lu et al [2], and deep auto encoder approach proposed Hassan et al [3]. The proposed model achieved an AUC score of 75.41, which outperformed the above methods. The comparison of AUC scores is shown in Table 2.1. Even though only video level labels are provided, the model was able to predict anomaly score accurately for normal video segments and anomalous video segments as it learned through high iterations.

In addition to that, the use of both normal videos and anomalous videos in training the model had contributed to low false alarm rate. The proposed method achieved a significantly low false alarm rate of 1.9 as compared to 27.2 by Hassan et al and 3.1 by Lu et al. However, the proposed model failed to detect anomaly in a night video due to the darkness. Incorporating video processing techniques such as to increase the brightness might be able to alleviate this issue. The model also generates false alarms in the case of occlusion and sudden gathering of people which showed that it identified normal group activity as anomalous.

2.1.2 Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance. [5]

Nawaratne et al [5] addressed the issue of evolving nature of normal events in anomaly detection by proposing an Incremental Spatiotemporal Learner (ISTL) which evolves over time. The model continuously learns the normal behaviour and refine its understanding on normalities with latest information by utilizing fuzzy aggregation with active learning so that it could differentiate anomalous events from normal events accurately. The model was trained using videos with normal behaviour, then it was validated using testing data that contains both normal and anomalous events.

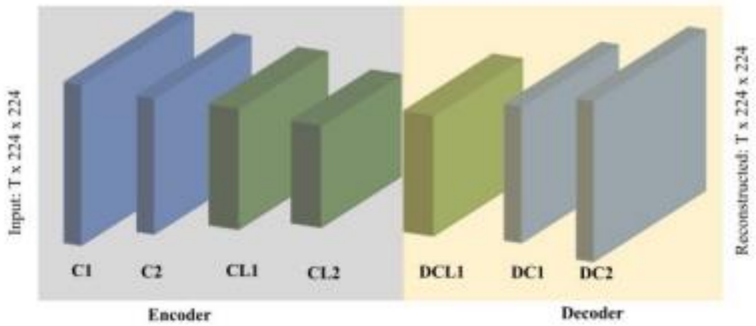


Figure 2.2 Spatiotemporal autoencoder architecture [5]

The ISTL consists of three phases. First is the spatiotemporal learning. The model learnt the normal behaviour's spatiotemporal representation from the training data set using autoencoder. The video frames were being converted into grayscale, resized and normalized before passing into the autoencoder. The autoencoder received temporal cuboid of video frames as input. The structure of autoencoder is shown in Figure 2.2. The autoencoder consists of 2 layers of convolution, 3 layers of ConvLSTM and 2 layers of deconvolution. The CNN (convolution) layers are used to learn the spatial representation whereas the ConvLSTM layers are used to learn the temporal representation.

Next, the ISTL model was used to detect and locate the anomalous event in input videos. Since the model was trained using only videos with normal behaviour, the reconstruction error for any scene that was not learnt during the training phase, which was anomalous event will be high. Hence, a reconstruction error threshold (μ) was used to differentiate the anomalies from the normalities. The value of the threshold will affect the sensitivity of the model in which lower threshold results in higher sensitivity and vice versa. To further reduce false alert due to sudden variations in the video, temporal threshold (λ) was introduced. The alert will be raised only if there are at least λ number of frames with reconstruction error higher than μ. The specific segments and area of the anomalies were localized by calculating the reconstruction error for local cuboid.

Third phase of the ISTL model was to continuously train and update the model so that it learns the new normal behaviours by using active learning with fuzzy aggregation. After the ISTL model was trained with initial normal dataset, it was used for anomaly detection. Any false positives detection from the model that was verified by human observer was used in the continuous learning phase to train the model. In addition to that, a fuzzy measure was given to the video frames based on reconstruction error during the detection phase. A number of video frames with highest fuzzy measure were selected to train the ISTL model to ensure the stability of the model when it learn the updated normal behaviours.

Model	Ped 1 AUC/EER	Ped 2 AUC/EER	Avenue AUC/EER
SF (2009)	67.5/31.0	55.6/42.0	NA
MPCCA (2009)	66.8/40.0	69.3/30.0	NA
MPCCA + SF (2010)	74.2/32.0	61.3/36.0	NA
Conv-AE (2016)	**81.0/27.9**	90.0/21.7	70.2/25.1
S-RBM (2017)	70.3/35.4	86.4/16.5	78.8/27.2
ConvLSTM-AE (2017)	75.5/NA	88.1/NA	77.0/NA
Unmasking (2017)	68.4/NA	82.2/NA	**80.6/NA**
Ours (ISTL)	75.2/29.8	**91.1/8.9**	76.8/29.2

Table 2.2 Comparison of AUC and EER [5]

For the performance of anomaly detection, the proposed model was benchmarked against a few state-of-the-art approaches. Table 2.2 summarized the comparison of performance between the benchmarked models. Three approaches were based on handcrafted feature representation: Social force (SF) model by Mehran et al [5], MACCA model [6], and MPCCA + SF model [7]. The author had selected four deep learning approaches for benchmarking which were Conv-AE [8], S-RBM [9], ConvLSTM-AE [10], and Unmasking-late-fusion [11]. Frame level ROC curves, Area under curve (AUC) and equal error rate (EER) were used as metrics to compare the results between the models. The ISTL model outperformed all three of the handcrafted approaches and on par with the deep learning approaches. The proposed model achieved a fast run time, which was 37ms on average for anomaly detection and localization. This shows that the model is able to handle surveillance video and perform anomaly detection in real time.

2.1.3 Future Frame Prediction for Anomaly Detection – A New Baseline. [12]

Liu et al [12] attempted anomaly detection using a video prediction framework. Different than the commonly used method, which was to detect anomalies based on reconstruction error, the proposed model detected anomalies based on the differences between the predicted and actual frames. The assumption of this approach was that normalities were predictable whereas anomalous events were unpredictable.

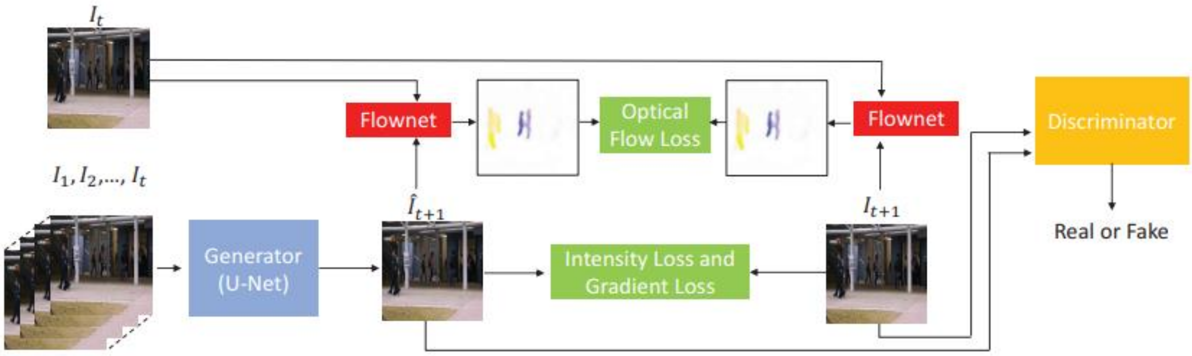


Figure 2.3 Video frame prediction pipeline [12]

Figure 2.3 shows the general flow of the prediction pipeline. The author used U-Net as the predictor for future video frame. The U-Net architecture contained shortcut between high level layer and low level layer with same resolution. This minimized the gradient vanishing and information imbalance problem in frame generation based on encoder and decoder approach. To ensure that the predicted frame was accurate, appearance and motion constraints was imposed. The appearance constraints ensure that the predicted frame was similar to the ground truth in terms of the intensity and gradient. To ensure the consistency in terms of the motion, temporal loss was introduced to minimize the difference between the optical flow of the predicted frame and the ground truth. Optical flow was used as the indicator of motion, and it was estimated using Flownet.

Least Square Generative adversarial network (GAN) was used to further improve the quality of the frame prediction. The U-net predictor was used as the generator network in the Least Square GAN and a patch discriminator was used as the discriminator network. The discriminator network was trained to classify ground truth into class 1 and predicted frame into class 0. Then the generator network was trained to generate frames which were classified as class 1 by the discriminator network. Mean Squared Error (MSE) loss function was imposed in both the training of discriminator and generator networks.

$$S(t) = \frac{PSNR(I_t, \hat{I}_t) - \min_t PSNR(I_t, \hat{I}_t)}{\max_t PSNR(I_t, \hat{I}_t) - \min_t PSNR(I_t, \hat{I}_t)}$$

For anomaly detection, Peak Signal to Noise Ratio (PSNR) was used to measure the quality of the predicted frame. Predicted frames with high PSNR were more likely to be normal. PSNR were calculated for each frame in the testing videos and normalized to range between 0 and 1. Then, a regular score was calculated using the equation as shown above and this score could be threshold to determine the frames with anomaly.

	CUHK Avenue	UCSD Ped1	UCSD Ped2	ShanghaiTech
MPPCA [14]	N/A	59.0%	69.3%	N/A
MPPC+SFA [15]	N/A	66.8%	61.3%	N/A
MDT [15]	N/A	81.8%	82.9%	N/A
Del *et al.* [13]	78.3%	N/A	N/A	N/A
Conv-AE [8]	80.0%	75.0%	85.0%	60.9%
ConvLSTM-AE	77.0%	75.5%	88.1%	N/A
GrowingGas [16]	N/A	93.8%	94.1%	N/A
AbnormalGAN [20]	N/A	**97.4%**	93.5%	N/A
DeepAppearance [21]	84.6%	N/A	N/A	N/A
Hinami *et al.* [17]	N/A	N/A	92.2%	N/A
Unmasking [18]	80.6%	68.4%	82.2%	N/A
Stacked RNN [19]	81.7%	N/A	92.2%	68.0%
Our proposed method	**85.1%**	83.1%	**95.4%**	**72.8%**

Table 2.3 AUC of different methods on Avenue, Ped1, Ped2 and ShanghaiTech dataset [12]

Area Under Curve (AUC) was the metric used to evaluate the performance of the proposed model. Table 2.3 shows the comparison between the proposed model and other approaches. The proposed model outperformed most of the hand-craft features based approaches [13][14][15][16] and deep learning based approaches [8][17][18][19][20][21]. In addition to that, the author compared the future frame predictor used (U-Net) with another video generation network (Beyond-MSE). Proposed method yielded higher AUC and gap between average score of normal and abnormal frames which showed it was better prediction network as compared to Beyond-MSE. The author also demonstrated the significance of motion constraint by showing that the model trained with motion constraint obtained higher AUC than the one without motion constraint.

2.1.4 Anomaly Detection in Video Sequence with Appearance-Motion Correspondence [22]

This research proposed a deep CNN that combined Conv-AE and U-net which shared a same encoder. The first processing stream, which is the Conv-AE was used to learn the common appearance of spatial structure in normal videos whereas the second processing stream, U-net take care of the temporal factor of the videos by associating motions to the common structures and objects.

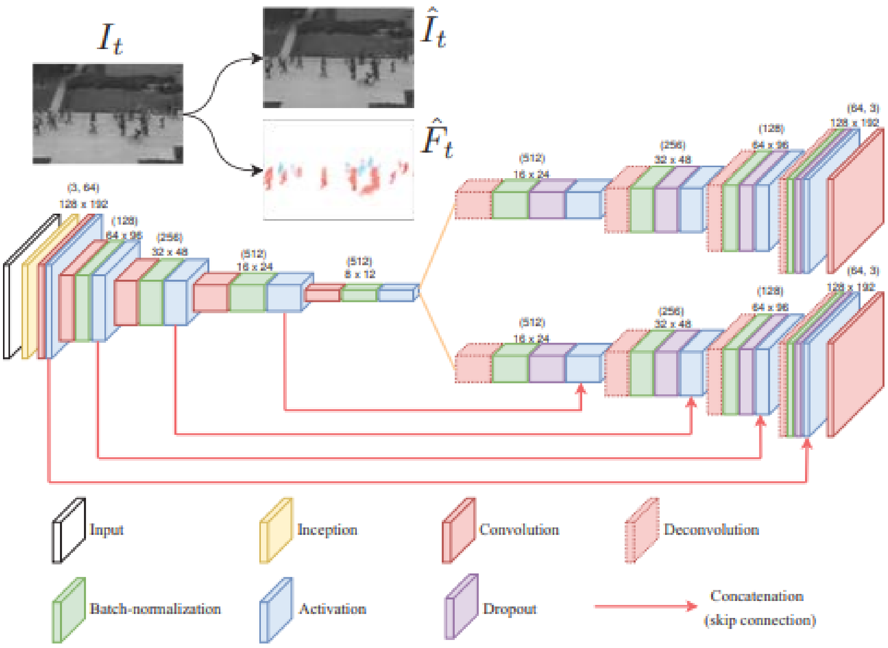


Figure 2.4 CNN that consists of Conv-AE and U-net.

Figure 2.4 shows the CNN architecture. The input of the model is a single video frame. The video frame will go through the layers of encoder, then it will be passed to the two processing streams mentioned earlier. The top right cluster of layers in figure 2.4 is the Conv-AE that will decode the encoded input into a reconstructed frame. The bottom right cluster of layers in figure 2.4 is the U-net that predicts the motions. The inception module that comes after the input layer is used to determine the most suitable convolutional operations for the model in order to avoid the network structure from being too deep until it removes useful features or too shallow that it missed some high-level abstraction.

The Conv-AE learn the common appearance patterns from the normal event by minimizing the l2 loss between the input image and the reconstructed frame. Gradient loss is employed in order to preserve the sharpness of the input image in the reconstructed frame. The U-net learns the motion of common objects and structures by using l1 loss function on the optical flow. l1 loss function is used instead of l2 to decrease the effect of noisy motion measure that is generated when estimating the ground truth optical flow. The author also included GAN loss which is to maximize the efficiency of motion prediction by the U-net. To detect the anomaly, the author proposed an anomaly score estimation that consider a small region of the video frames. This is because estimating the anomaly score with simple average operation on a frame level might cause the anomalies that occur within a small region in the frames to be missed out.

The proposed model had been benchmarked with different datasets which are CUHK Avenue [23], UCSD Ped2 [24], Subway Entrance Gate and Exit Gate [25], Traffic-Belleview and Traffic Train [26]. The proposed model outperformed most of the methods that was benchmarked using CUHK Avenue and UCSD Ped2 datasets. For Subway Entrance Gate and Exit Gate, the model successfully detected most of the anomalies but also generated more false alarm than the other methods. The model did not perform well on the Traffic Train dataset due to the varying light condition and camera jitter. The average precision on Traffic Train dataset improved drastically with the use of Structural Similarity Index (SSIM).

2.1.5 A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework [19]

Luo et al [19] proposed a Temporally-coherent Sparse Encoding (TSC) model. Sparse coding is a common approach in anomaly detection where it learns a dictionary that encodes normal events with small reconstruction loss. The anomalies can be detected when reconstruction loss is high. However, traditional sparse coding might encode similar neighbouring frames into dissimilar sparse code. Hence, the proposed TSC model preserve the locality information by weighing the distance of sparse coefficients between similar frames based on the similarity of the frames. This ensures that neighbouring frames will have similar sparse coefficient.

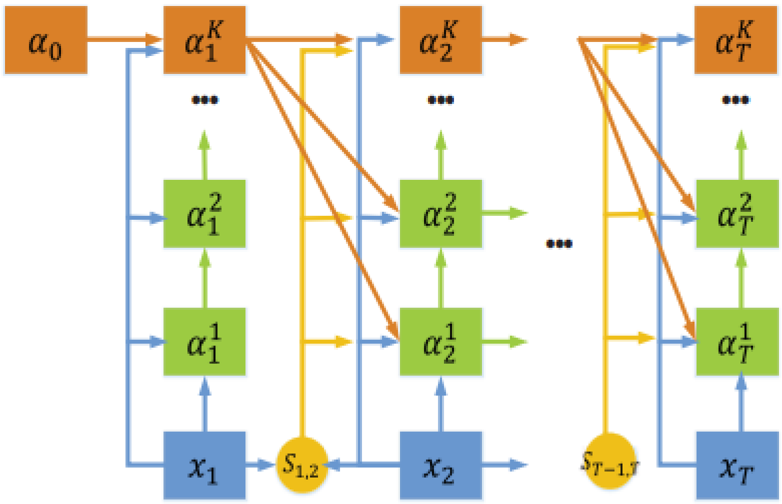


Figure 2.5 sRNN counterpart of TSC

The TSC is mapped to a sRNN model. The proposed sRNN structure is illustrated in figure 2.5. Blue boxes represent the input whereas green and orange boxes are the coding vector. The similarities of neighbouring frames are represented by the yellow circles. Every layers of the proposed sRNN will take in the input x, which is different from standard sRNN where the input is only fed into the first layer. The parameter tuning was done using Autoencoder way. The sRNN has a much shallower architecture which made anomaly detection much faster. Spatial ConvNet that was pretrained on UCF101 dataset was used to extract spatial features from each frame as this has been proven to be able to improve the anomaly detection performance.

2.1.6 Detection of Abnormal Events via Optical Flow Feature Analysis [27]

This research proposed to combine one-class SVM and kernel principal component analysis (KPCA) method to detect anomalies. The system used histogram of optical flow as the feature descriptor for classification. The author computed optical flow using Horn-Schunck method, which formulate it as global energy function for consecutive grey images. The feature used for anomaly detection outline the overall movement information of one frame by combining the histogram of optical flow in sub-frame.

Two detection methods, one-class SVM and kernel principal component analysis were trained on normal data. The aim was to detect test data that was different from the training data as this signifies deviation from normal circumstances. For both of the dataset, Performance Evaluation of Tracking and Surveillance (PETS2009) and University of Minnesota (UMN) dataset that were used in this study, individuals walk towards different direction are considered normal. Anomalies were defined by running or people move towards a same direction.

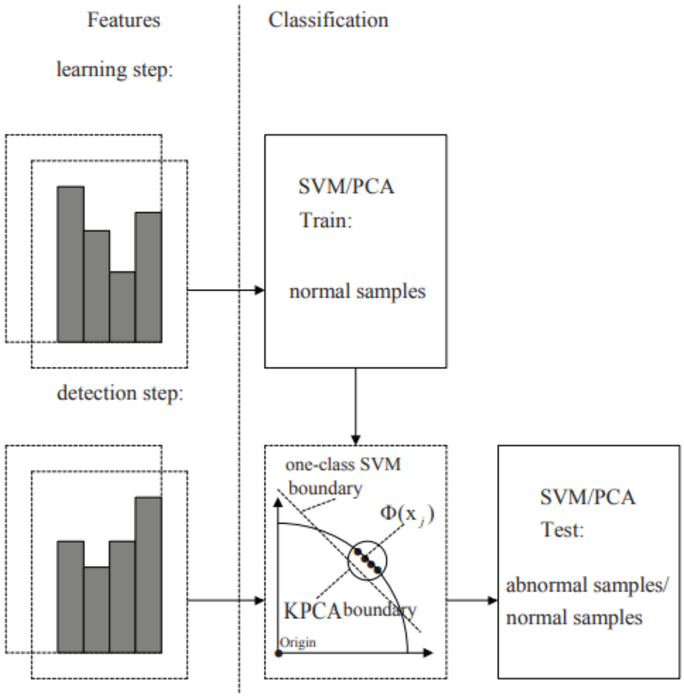


Figure 2.6 SVM and KPCA for anomaly detection

First, the optical flow of each frame is computed, and the histogram of optical flow (HOFO) is calculated. Next, one-class SVM or KPCA is used to classify the features into normal and abnormal. SVM classify the incoming HOFO features based on the support vectors obtained during training step with normal data. Whereas KPCA mapped the features of normal samples into high-dimensional space and the squared distance between the testing samples and principal components is calculated. State transition restriction method was used to filter out short fluctuation clip. The number of frames classified as abnormal must exceed a threshold N in order for the data to be classified as abnormal.

2.2 Limitation of Previous Studies

Most of the previous studies emphasize on learning the normal events by extracting relevant features through various deep learning network. Then, the input frame is reconstructed and the difference between the reconstructed frame and the actual input is measured. Different loss functions had been used to ensure that the reconstructed frame resembles the actual input as close as possible during the training stage. This approach works based on the principle that the model could not reconstruct anomalous event well because it was trained only with normal event. The weakness of such approaches is that they do not consider the contextual information. For example, appropriateness of certain actions depends on the time and place that the actions occur. A person that stays at the public area for a prolonged period of time might not be consider as abnormal, but if a person stays and walk around the front yard of a house, this might hint a break in or criminal attempt.

Chapter 3
System Methodology

3.1 System Approach

The approach of this system is to extract the possible fire regions and then filter those regions. The condition for a region to be potential fire is that the region has to be bright. In addition to that, when the fire is burning, the intensity of the centre of the fire will not have much changes overtime. The surrounding area of the fire will have different brightness and changes overtime. Hence, the potential fire regions can be extracted based on this pattern.

Figure 3.1 Sample data

3.2 System Overview

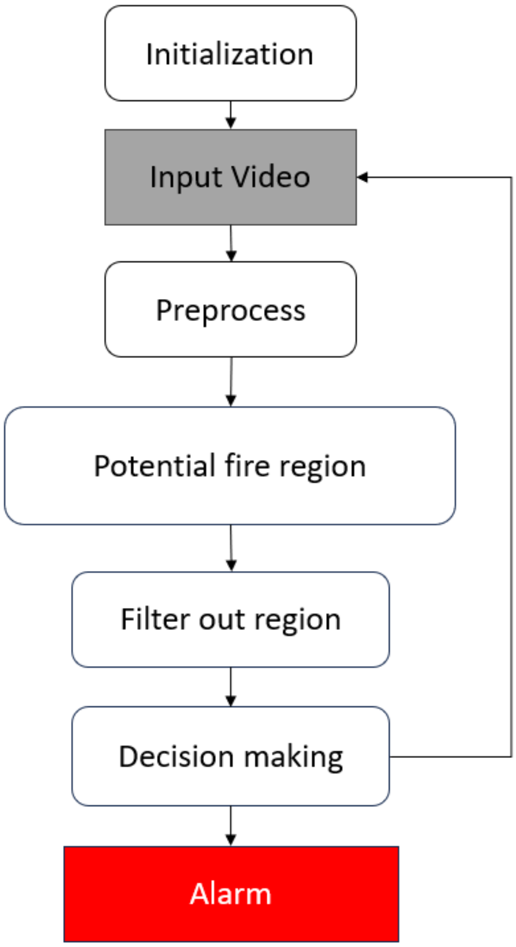


Figure 3.2 System Overview

The general overview of the proposed system is shown in Figure 3.1.

First when the system initializes, it will open the input video files, create output windows to display the output as well as open an output file to record the detection result. The system will then read in the first frame of the input video.

After first frame being read in during initialization, the system proceeds to read in the next frame and pass it through series of different modules. The preprocess modules will carry out necessary setup such as resize and convert the frame to grayscale to prepare the frame for further processing.

Next, the frame will be passed into another module to extract the potential fire regions in the frame. This is be done by calculating the motion and the bright region of the image. The center of the flame will have high intensity and brightness and have little to no changes whereas the area surrounding the center of the flame will have changes overtime as the fire is burning or flickering which can be detected through the motion energy image. Hence, the region with no changes overtime and is inside the bright region will be proposed as the potential fire region.

After that, the region extracted by the previous module will be further processed to filter out the non-flame region. The is done by checking the area surrounding the proposed regions as there will be changes of brightness overtime at the area surrounding the real fire. The regions that satisfy this condition will be kept as fire regions.

Lastly, the decision making module will determine whether fire is detected. To reduce false alarm rate, the system will only raise the alarm if it detects at least 5 frames with fire region in 1 second.

Chapter 4
System Design

4.1 Module Specification

4.1.1 Initialization

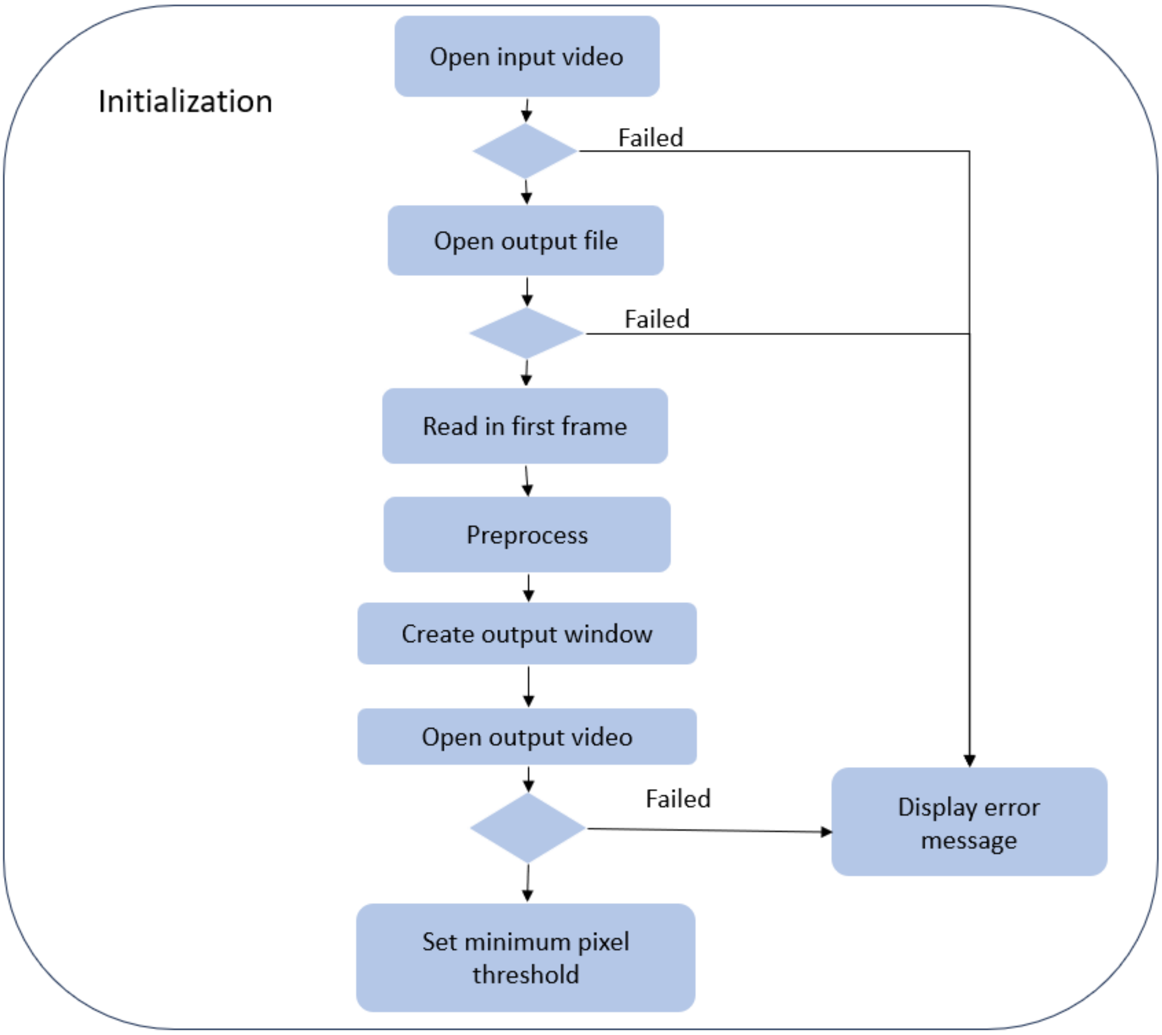


Figure 4.1 Initialization module

When the system first started, it will open the input video. If the input video is not opened, it will display error message and terminate the execution. After that, it will open an output file which is a text file to record down the result of detection. Failure of opening the output file will also results in error message and termination of the execution. The system will read in the first frame of the input video and preprocess it. Then, the system creates an output window which consists of sub-windows to visualize the output from each module. The size for each of the sub-windows is calculated based on the input frame size and number of sub-windows. Then, the system opens an output video to record down the output window. Error message will be shown if output video cannot be opened. The system also set the minimum pixel threshold, which is 0.5% of the input video size. This threshold will be used later in the other module.

4.1.2 Preprocess

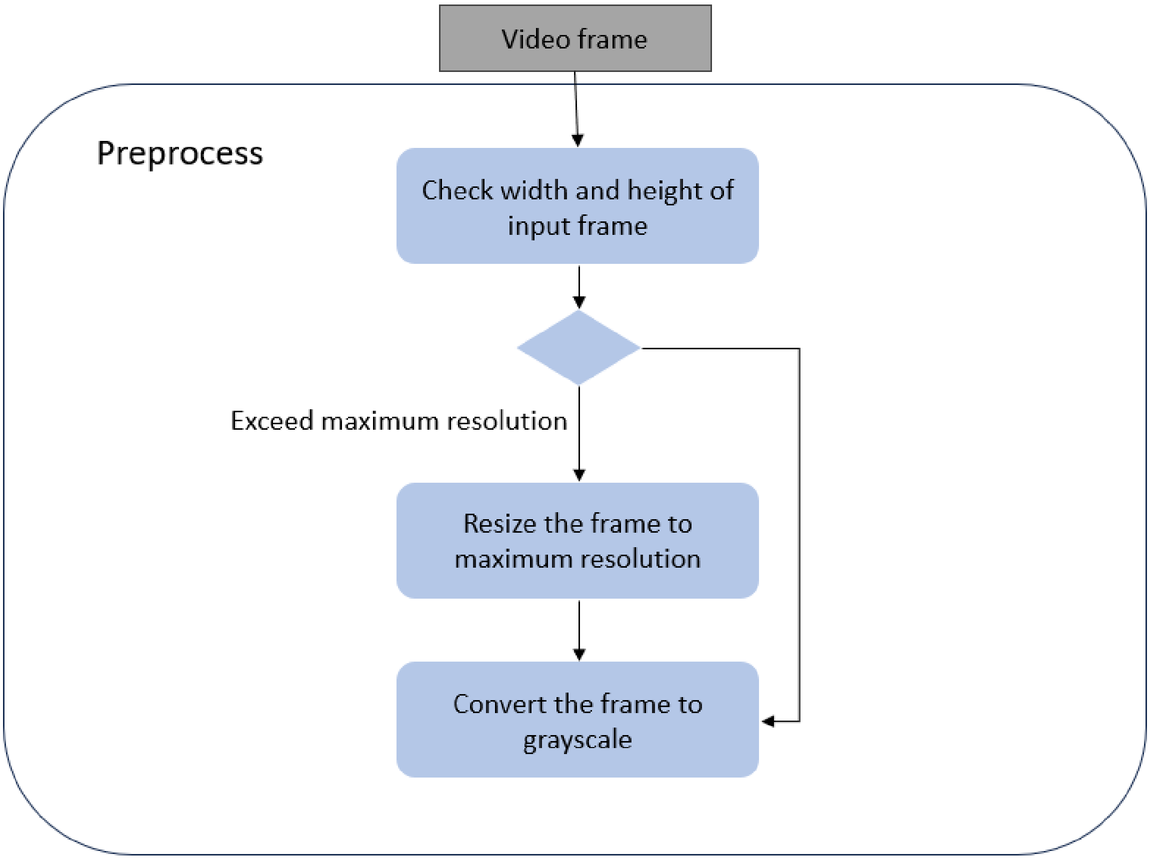


Figure 4.2 Preprocess module

The preprocess module will check if the width or height of the input video frame is greater than the maximum resolution. In this system, the maximum resolution is set to 416. If the input frame exceeds the maximum resolution, the frame will be resized to the maximum resolution while maintaining the aspect ratio. Next, the resized frame will be converted into grayscale.

4.1.3 Extract potential fire region

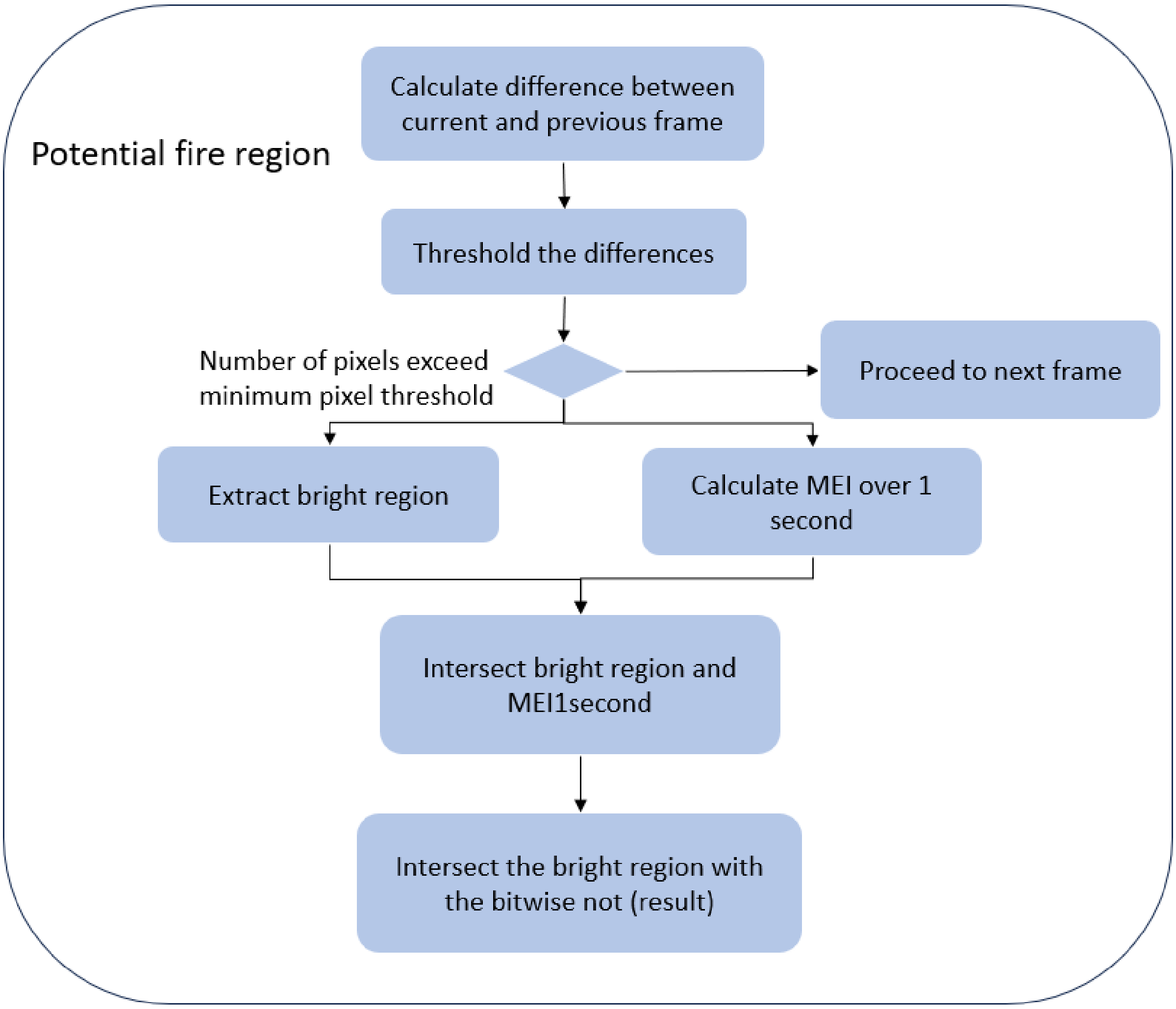


Figure 4.3 Extract potential fire region

This is the main module that detect possible region of fire in the frame. First, it will get the difference by subtracting the previous frame from current frame. Then, the result will be threshold with a value of 20, which means that any changes that are below 20 will be disregarded. The result is a binary image which indicate the motion between current and previous frame. This result will be saved as it is needed to compute the MEI1second for future frames. Then, the system will check if the number of non-zero pixels in the binary image exceed the minimum pixel threshold that was set during the initialization. If the value does not exceed the threshold, the system will not process the current frame any further and will proceed to the next frame as the changes is insignificant.

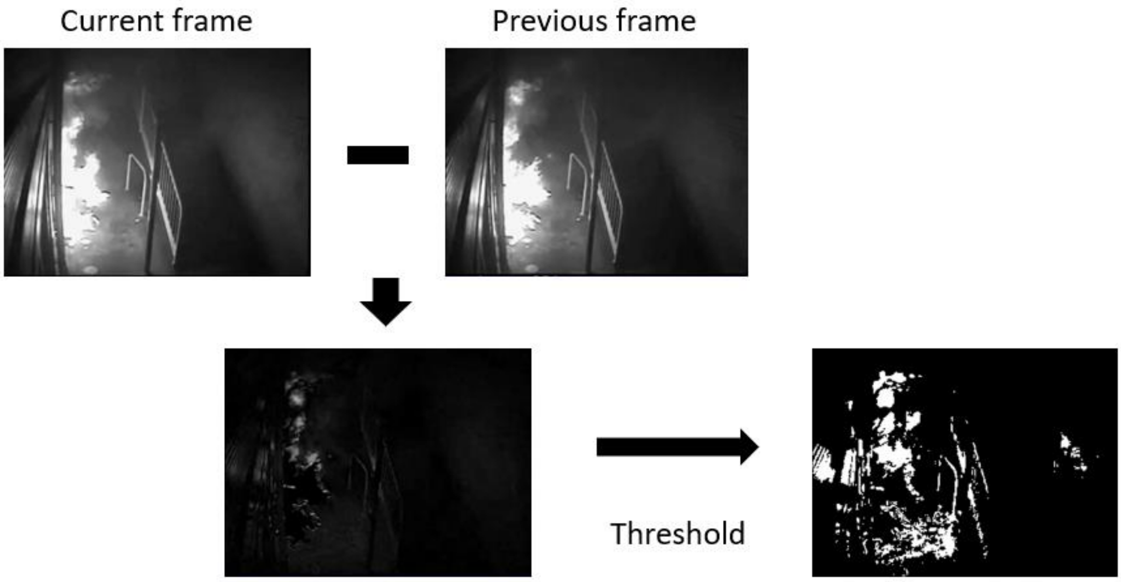


Figure 4.4 Illustration of calculating motion.

Then, the system will proceed to extract the bright region and MEI1second. In this system, the bright region is defined as the region with intensity greater than 128. To calculate the MEI1second, which is all the motion that occurred over past 1 second, the system will combine all the motion images from the past 1 seconds.

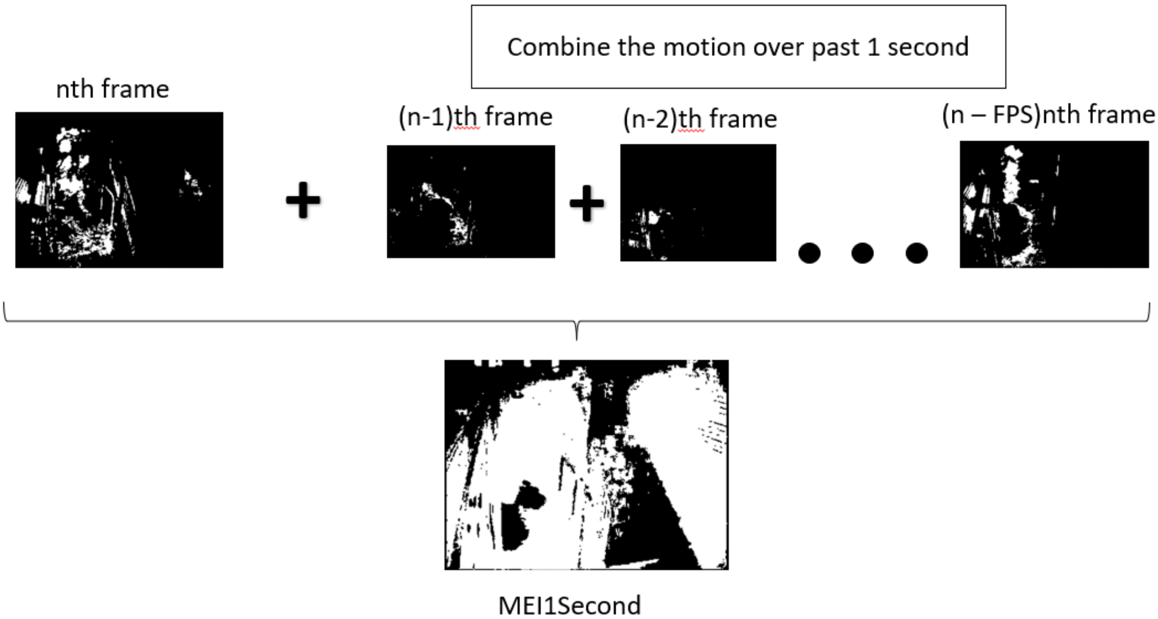


Figure 4.5 Calculate MEI1second

Figure 4.6 illustrate the output of each operation. Intersect bright region (a) and MEI1second (b) will result in a blackhole pattern (c), which is the centre of the fire. To illustrate the hurricane pattern, the result is also intersected with grayscale image (d). The area surrounding the blackhole will have layers of different brightness. To extract the centre of the fire, perform bitwise not on the image and intersect with the bright region. The result is the potential fire region (e).

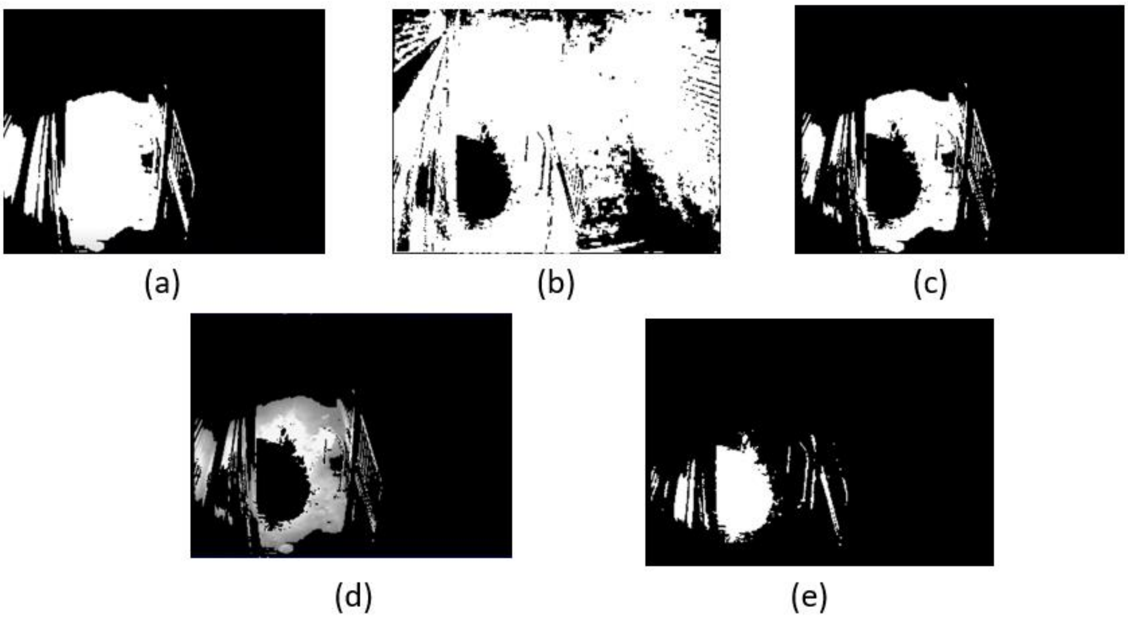


Figure 4.6 Output from each operation

4.1.4 Filter proposed region

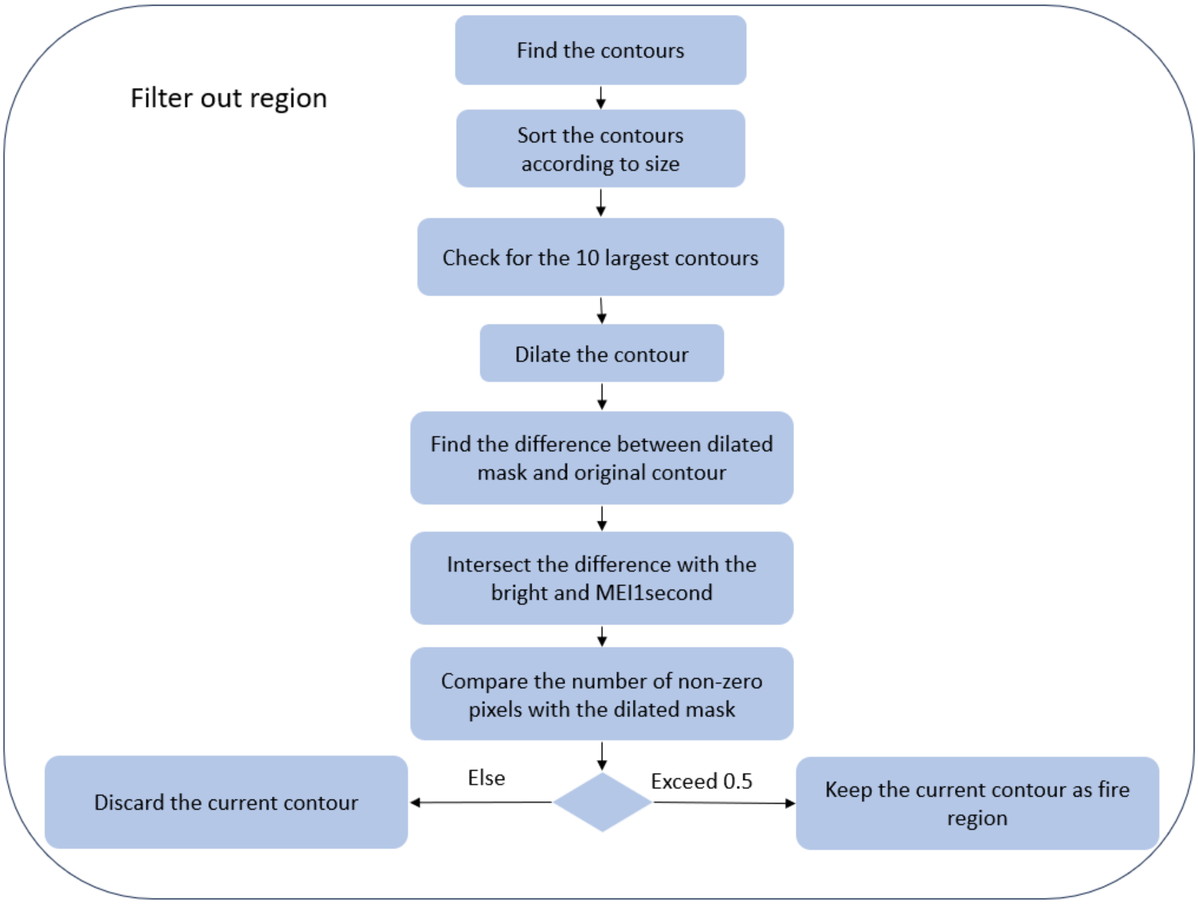


Figure 4.7 Filter proposed fire region

This module will check all the region proposed by the previous module to filter out non-fire region. First, the system will find all the contours and sort it according to the contours' size. Then, it will only check for the 10 largest contours as the smaller contours are most likely noise or isolated pixels. For these 10 largest contours, the system will check if the surrounding area has the hurricane pattern.

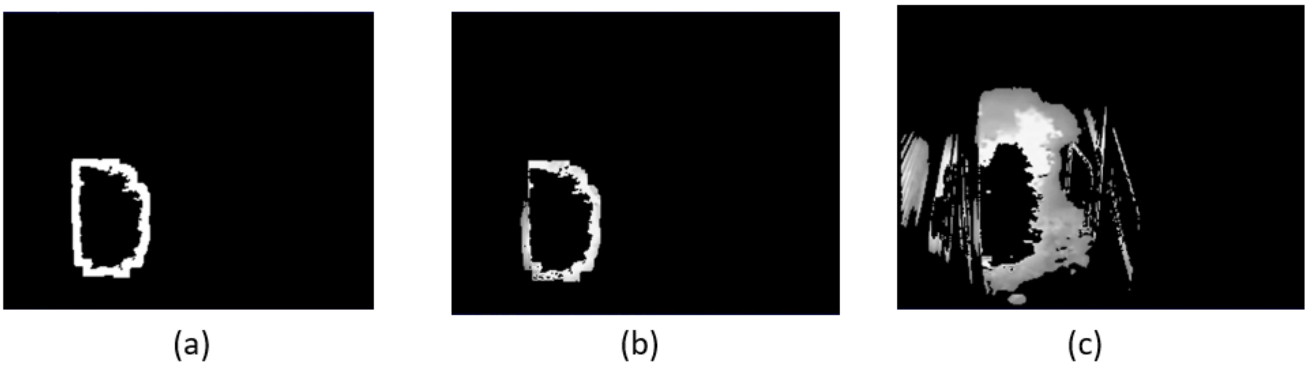


Figure 4.8 Operation to filter proposed fire regions.

The system will dilate the contour and then find the difference between the dilated mask and original contour to get the surrounding area (a). The surrounding area is then intersected with the bright and MEI1second image (c) to get the output image (b). If the number of non-zero pixels of (b) exceeds half/ 0.5 of the number of non-zero pixels of (a), the system classifies this contour as fire region. Else, the contour will be discarded.

4.1.5 Decision making

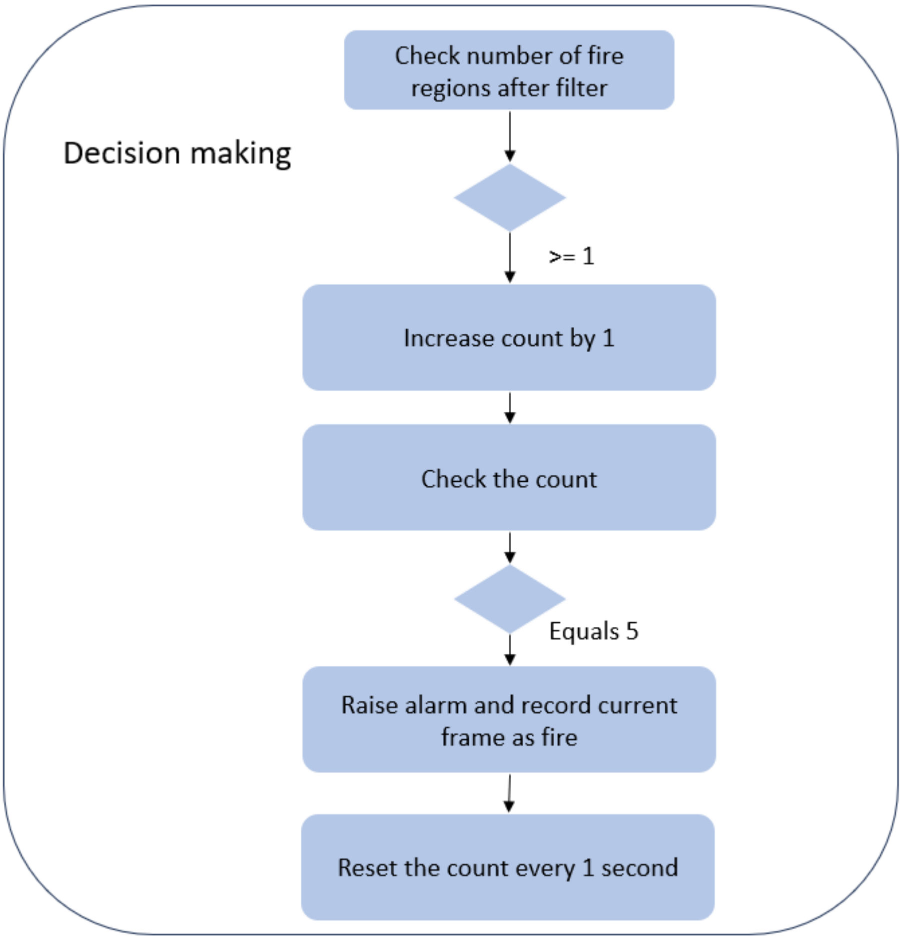


Figure 4.9 Decision making module.

After filtering the proposed regions, the decision making module will check the number of proposed regions left. If there is any region remains and not being filtered out, the count will be increase by 1. Then, when the count hits 5, the system will raise the alarm and indicate current frame as fire. The count will be reset every 1 second. The alarm will only be raised if the system detects 5 frames that contain fire within 1 second. This is to reduce the rate of false alarm.

Chapter 5

System Implementation

5.1 Hardware setup

The hardware involved in this project is computer. The specifications of the computer used in this project is shown in the following table.

Description	Specifications
Model	HP Pavilion Laptop 15-cs3xxx
Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
Operating System	Windows 10
Graphic	Intel(R) UHD Graphics
Memory	12.0GB
Storage	512GB SSD

Table 5.1 Specifications of laptop

5.2 Software setup

The following table shows the software and libraries involved in developing the proposed solution.

Software	Description
Visual Studio 2019	Visual Studio 2019 is the Integrated Development Environment (IDE) from Microsoft. The proposed solution will be built and coded in C++ language using Visual Studio 2019 as the IDE.
OpenCV	OpenCV is an open source library which provides functions for real time computer vision tasks. This library supports C++ language, and will be used for different part in the proposed solution including preprocessing and object detection with YOLO.

Table 5.2 Software used

5.3 System operation

Figure 5.1 Output window

Figure 5.1 shows a sample output window generated from the system. The windows in the first row displays the current frame, bright area and the MEI1second. The first and second windows in the second row are the hurricane pattern and possible fire region that is generated by the system. The third window in second row shows the filtered region. The window in the last row shows the fire region in the actual frame. When fire is detected, the window of the last row will show fire detected in the legend.

output.txt - Notepad

File Edit Format View Help

```
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson002_x264.mp4-Fire detected at frame: 2848
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson005_x264.mp4-Fire detected at frame: 658
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson016_x264.mp4-Fire detected at frame: 49
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson021_x264.mp4-Fire detected at frame: 1109
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson023_x264.mp4-Fire detected at frame: 489
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson024_x264.mp4-Fire detected at frame: 1438
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson029_x264.mp4-Fire detected at frame: 795
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson036_x264.mp4-Fire detected at frame: 88
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson041_x264.mp4-Fire detected at frame: 21
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson042_x264.mp4-Fire detected at frame: 9
C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson044_x264.mp4-Fire detected at frame: 599
```

Figure 5.2 Output file

Other than the output window, the system will also record the detection result in a text file. The name of the video, detection result and the first frame being classified as fire will be recorded.

Chapter 6

System Evaluation and Discussion

6.1 System testing and results

Video name	**Fire detected**	**Ground truth frame**	**Detected frame**	**Delay in detection**
Arson002_x264	Yes	2830	2848	18
Arson003_x264_2	Yes	1	17	16
Arson005_x264	Yes	630	658	28
Arson007_x264	**Yes**	**2187**	**3383**	**1196**
Arson008_x264_2	Yes	22	46	24
Arson009_x264_2	Yes	217	68	-
Arson014_x264	Yes	1037	1042	5
Arson020_x264	Yes	1624	1642	18
Arson021_x264	Yes	1100	1109	9
Arson024_x264	Yes	1427	1438	11
Arson027_x264_2	**Yes**	**868**	**996**	**131**
Arson027_x264_3	Yes	129	146	17
Arson028_x264_2	Yes	612	625	13
Arson029_x264	Yes	780	795	15
Arson034_x264_2	**Yes**	**1**	**33**	**32**
Arson040_x264_2	Yes	640	666	26
Arson041_x264	Yes	3466	21	-
Arson044_x264	**Yes**	**441**	**599**	**158**
Arson048_x264	Yes	185	192	7
Arson049_x264	Yes	97	108	11
Arson050_x264	Yes	875	883	8

Table 6.1 Results of testing

Table 6.1 summarizes the results of testing. The ground truth frame is the frame that the fire started while the detected frame is the detection result. Delay in detection is the difference between ground truth and detected frame. All the above videos have the frame per second (FPS) of 30. The videos in the format of Arson[][][]_x264_[] is the videos that are edited. For example, Arson003_x264_2 is a video clip from Arson003_x264. The editing is done because some of these videos are switching between different scenes. The starting and ending time of the edited videos from original video is listed in the appendix for reference.

Based on the results, the system is able to detect the fire within 1 second or 30 frames for 15 of 21 videos tested. However, for the 4 videos that are in bold (Arson007_x264, Arson027_x264_2, Arson034_x264_2, Arson044_x264), the time taken for the system to detect fire exceeds 1 second. This is because the fire is very small when it started and the system is not able to capture it. The system only detects the fire when it becomes larger later. Figure below shows the fire when it starts burning.

Figure 6.1 Videos that the system failed to detect within 1 second

There are also a few videos highlighted in yellow, where the system raises the alarm before the actual fire begins. This indicates a false positive. To address this issue, the system has been tested using different settings. The threshold value to define bright region has been increased from 128 to 200. The value used in filtering the potential fire regions is also increased from 0.5 to 0.8.

Video name	**Original detected frame**	**New detected frame**	**Ground truth**
Arson009_x264_2	68	224	217
Arson041_x264	21	3499	3466
Arson008_x264_2	46	49	22
Arson049_x264	108	117	97
Arson005_x264	658	Failed	630

Table 6.2 Comparison between new and original settings

By setting the conditions to be stricter, both Arson009_x264_2 and Arson041_x264 can be detected successfully without raising the alarm before the actual fire. To further investigate the effect of such changes, the new settings have been tested with another 3 videos. For 2 of the videos, the detection becomes slower and 1 of it failed to be detected by the system. This shows that a stricter condition can effectively reduce false alarm but also might affect the detection speed and accuracy. Hence, the optimum settings should balance both false positive and accuracy of detection.

6.2 Project challenges

The UCF_crime dataset does not contain the annotation at the temporal level. The videos in the dataset are only being classified as arson. But the actual time or frame that the fire occurs were not given. Hence, the ground truth or the frame that the fire started has to be determined by human. This process is very time consuming and caused a huge challenge for the system evaluation. In addition to that, the ground truth generated by human might also deviate from the actual ground truth. This is because in some cases, the fire in the videos is very small. It is difficult to capture the fire even for human.

Besides, some of the videos in the dataset switches between different scene and places. Hence, some of the videos require editing before it can be used to evaluate the system. This introduces additional time cost to the development of this project.

6.3 Objectives evaluation

- **To develop a fully automated surveillance system to detect fire.**

  The system can operate without any user input. The potential regions are detected by the system without any predefined region.

- **To model the changes in MEI.**

  The motion of the videos is modeled and displayed in the output windows for visualization. The motion is also used in the fire detection.

- **To detect the fire within 1 second from the occurrence**

  The system is able to detect fire within 1 second for most of the cases based on the result of testing. System cannot capture fire that is too small. Hence, there are some cases where the system detection is slower than 1 second.

Chapter 7

Conclusion and Recommendations

7.1 Conclusion

The use of surveillance cameras is becoming more common in the modern world. It acts as the security measure to safeguard the neighborhood from potential danger. Monitoring the surveillance with human is a wastage of manpower and time. Instead, the task should be automated by leveraging the power of technology. A well designed system can perform the task better than human as computer does not loss focus over time as human does.

This project proposed a robust system to detect fire and trigger alarm when potential arson is detected. This takes away most of the human intervention in surveillance monitoring. The system utilizes information from the fire pattern instead of relying on the deep learning models to learn the normal patterns. By narrowing down the scope, the system can achieve better results in identifying more specific anomalies.

Based on the results, the system has shown its capability to capture fire and make detection fast. The system is able to detect fire within 1 second of the occurrence for most of the videos tested. However, the system is not able to capture small fire and is only able to detect when the fire becomes larger and much more noticeable.

7.2 Recommendations

For future work, the system can be tested with more videos of different scenarios. The system uses different threshold value such as to define the bright region, calculating the motion as well as in filtering out the proposed fire regions. The threshold value has a direct impact to the performance of the system. Technique such as histogram of intensity can be used to determine a better threshold value.

Different setting had been experimented and discussed in chapter 6. However, the experiment is not comprehensive enough to generalize to all different scenarios due to time constraints. Hence, future work could focus on finding an optimum setting that could improve the detection rate as well as reduce the false positive rate of the system that could work on various scenarios.

REFERENCES

[1] W. Sultani, C. Chen and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 6479-6488, doi: 10.1109/CVPR.2018.00678.

[2] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In ICCV, 2013.

[3] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In CVPR, June 2016

[4] R. Nawaratne, D. Alahakoon, D. De Silva and X. Yu, "Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance," in IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 393-402, Jan. 2020, doi: 10.1109/TII.2019.2938527.

[5] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, Jun. 2009, pp. 935–942.

[6] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, Jun. 2009, pp. 2921–2928.

[7] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., San Francisco, CA, USA, Jun. 2010, pp. 1975–1981.

[8] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, Jun. 2016, pp. 733–742.

[9] H. Vu, "Deep abnormality detection in video data," in Proc. Int. Joint Conf. Artif. Intell. Org., 2017, pp. 5217–5218.

[10] R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in Proc. IEEE Int. Conf. Comput. Vis., Venice, Italy, Oct. 2017, pp. 2895–2903.

[11] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in Proc. Brit. Mach. Vis. Conf., 2014.

[12] W. Liu, W. Luo, D. Lian and S. Gao, "Future Frame Prediction for Anomaly Detection - A New Baseline," 2018 IEEE/CVF Conference on Computer Vision and Pattern

Recognition, Salt Lake City, UT, USA, 2018, pp. 6536-6545, doi: 10.1109/CVPR.2018.00684.

[13] A. D. Giorno, J. A. Bagnell, and M. Hebert. A discriminative framework for anomaly detection in large videos. In Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V, pages 334–349, 2016.

[14] J. Kim and K. Grauman. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, pages 2921–2928, 2009.

[15] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010, pages 1975–1981, 2010.

[16] Q. Sun, H. Liu, and T. Harada. Online growing neural gas for anomaly detection in changing surveillance scenes. Pattern Recognition, 64:187–201, 2017.

[17] R. Hinami, T. Mei, and S. Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 3639– 3647, 2017.

[18] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu. Unmasking the abnormal events in video. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2914–2922, 2017.

[19] W. Luo, W. Liu, and S. Gao. A revisit of sparse coding based anomaly detection in stacked RNN framework. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 341–349, 2017.

[20] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. S. Regazzoni, and N. Sebe. Abnormal event detection in videos using generative adversarial nets. In 2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017, pages 1577–1581, 2017.

[21] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe. Deep appearance features for abnormal behavior detection in video. In Image Analysis and Processing - ICIAP 2017 - 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part II, pages 779–789, 2017.

[22] T. N. Nguyen and J. Meunier, "Anomaly Detection in Video Sequence With Appearance-Motion Correspondence," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 1273-1283, doi: 10.1109/ICCV.2019.00136.

[23] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In 2013 IEEE International Conference on Computer Vision, pages 2720–2727, Dec 2013

[24] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(1):18–32, Jan 2014

[25] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and David Rei nitz. Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Trans. Pattern Anal. Mach. Intell., 30(3):555–560, 2008.

[26] Andrei Zaharescu and Richard Wildes. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event driven processing. In Kostas Daniilidis, Petros Maragos, and Nikos Pa ragios, editors, Computer Vision – ECCV 2010, pages 563–576, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[27] Wang T., Snoussi H., “Detection of abnormal events via optical flow feature analysis.” Sensors, vol. 15, no. 4, pp. 7156-7171, Mar 2015

APPENDIX

Code:

```
#include <iostream>
#include <fstream>
#include <opencv2/opencv.hpp>
#include <vector>

cv::Mat outputWindow;
cv::VideoCapture inputVideo;
cv::VideoWriter outputVideo;
cv::Mat color[2];
cv::Mat gray[2];
cv::Mat win[9], legend[9];


int prev = 0, cur = 1;
int fps;
int miRecordCount = -1;
int minPixels;
int found_count = 0;
bool flag = false;

struct MIRecord {
   int prev, cur;
   cv::Mat gray;   // gray frame
   cv::Mat MEI1Fr;     // MEI between pre and cur frame
   cv::Mat MEI1Sec;   // MEI in past 1 second
};

void resizeImage(cv::Mat& oriVideo, int maxReso = 416) {
   int oriReso = std::max(oriVideo.rows, oriVideo.cols);
   if (oriReso <= maxReso) return;

   if (oriVideo.cols > oriVideo.rows) {
       cv::resize(oriVideo, oriVideo, cv::Size(maxReso, (int)((double)oriVideo.rows /
oriVideo.cols * maxReso)));
   }
   else {
       cv::resize(oriVideo, oriVideo, cv::Size((int)((double)oriVideo.cols / oriVideo.rows *
maxReso), maxReso));
   }
}

void createOutputWindow(cv::Mat inputVideo, cv::Mat win[], cv::Mat legend[], cv::Mat&
outputWin, int margin = 15, int winPerRow = 3, int winPerCol = 3) {
   int rows = inputVideo.rows;
   int cols = inputVideo.cols;
```

```
    int winCount = 0, legendCount = 0;
    cv::Size outWinSize((cols + margin) * winPerRow - margin, (rows + margin) *
winPerCol);

    outputWin = cv::Mat::ones(outWinSize, inputVideo.type()) * 64;

    for (int i = 0; i < winPerCol; i++) {
        for (int j = 0; j < winPerRow; j++) {
            win[winCount++] = outputWin(cv::Range((rows + margin) * i, (rows + margin) * i +
rows),
                cv::Range((cols + margin) * j, (cols + margin) * j + cols));
        }
    }

    for (int bg = 20, i = 0; i < winPerCol; i++) // create the legend windows
        for (int j = 0; j < winPerRow; j++) {
            legend[legendCount] = outputWin(cv::Range((rows + margin) * i + rows, (rows +
margin) * (i + 1)),
                cv::Range((cols + margin) * j, (cols + margin) * j + cols));
            legend[legendCount] = cv::Scalar(bg, bg, bg); // paint each in different colors
            bg += 30; // such that we can visually see the division from one to other
            if (bg > 80) bg = 20;
            legendCount++;
        }
}

void initiate() {
    // read in the first frame, resize and change it to grayscale
    inputVideo >> color[0];
    resizeImage(color[0]);
    color[0].copyTo(win[0]);
    createOutputWindow(color[0], win, legend, outputWindow);
    minPixels = color[0].cols * color[0].rows * 0.005;
    cv::cvtColor(color[0], gray[0], cv::COLOR_BGR2GRAY);
}

bool compareContourAreas(std::vector<cv::Point> contour1, std::vector<cv::Point>
contour2) {
    double area1 = contourArea(contour1);
    double area2 = contourArea(contour2);
    return area1 > area2; // Sort in descending order (largest to smallest)
}


int main() {
    std::string videoName =
"C:/Users/Hp/Desktop/UCF_Crimes/Videos/Arson/Arson049_x264.mp4";
    std::string outputVideoName =
"C:/Users/Hp/Desktop/UCF_Crimes/Videos/out/Test049_x264.mp4";
```

```
    std::string message = "";
    char str[256];
    inputVideo.open(videoName);
      if (!inputVideo.isOpened()) {
            std::cout << "Error opening video file" << std::endl;
            return 0;
      }

    std::ofstream outputFile;
    outputFile.open("C:/Users/Hp/Desktop/UCF_Crimes/Videos/out/output.txt",
std::ios_base::app);
    if (!outputFile.is_open()) {
        std::cout << "Error opening text file" << std::endl;
        return 0;
    }

    fps = inputVideo.get(cv::CAP_PROP_FPS);

    initiate();

    outputVideo.open(outputVideoName, cv::VideoWriter::fourcc('D', 'I', 'V', '3'),
inputVideo.get(cv::CAP_PROP_FPS),
        cv::Size(outputWindow.cols, outputWindow.rows), true);
    if (!outputVideo.isOpened()) {
        std::cout << "Could not open the output video for writing";
        system("pause");
        return 0;
    }

    cv::Mat diff1Fr, MEI1Fr, bright, bright2, bright3;
    MIRecord miRecord[40];
    miRecord[39].prev = -100;
    miRecord[39].cur = -100;

    cv::Mat MHI, MEI1Sec, temp, preMEI, hsv;
    MEI1Sec = cv::Mat::zeros(color[0].size(), CV_8U);

    while (true) {
        // Read a frame from the video
        inputVideo >> color[1];
        if (color[1].empty()) {
            break; // End of video
        }
        resizeImage(color[1]);

        color[1].copyTo(win[0]);
        sprintf_s(str, "Current frame (color) %d", cur);
        legend[0] = cv::Scalar(64);
        putText(legend[0], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);
```

```
// convert to grayscale
cv::cvtColor(color[1], gray[1], cv::COLOR_BGR2GRAY);

// find MEI of prev and curr frame
cv::absdiff(gray[0], gray[1], diff1Fr);
MEI1Fr = diff1Fr > 20;


if (cv::countNonZero(MEI1Fr) > minPixels){
   // compute MEI1Sec, MHI1Sec
   // circular array
   miRecordCount++;
   miRecordCount %= 40;

   miRecord[miRecordCount].prev = prev;
   miRecord[miRecordCount].cur = cur;
   MEI1Fr.copyTo(miRecord[miRecordCount].MEI1Fr);
   gray[1].copyTo(miRecord[miRecordCount].gray);

   MEI1Fr.copyTo(MEI1Sec);
   MEI1Fr.copyTo(preMEI);
   int valueSub = 0;

   for (int i = miRecordCount; ; ) {
      i--;
      if (i < 0) i = 39;
      if (cur - miRecord[i].prev > fps) break;

      MEI1Sec |= miRecord[i].MEI1Fr;
      temp = MEI1Sec - preMEI;
      MEI1Sec.copyTo(preMEI);
   }
   MEI1Sec.copyTo(miRecord[miRecordCount].MEI1Sec);

   cv::cvtColor(MEI1Sec, win[2], cv::COLOR_GRAY2BGR);
   sprintf_s(str, "MEI 1 second ");
   legend[2] = cv::Scalar(64);
   putText(legend[2], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);

   // find the bright area >128
   //bright = gray[1] > 200;
   bright = gray[1] > 128;
   cv::cvtColor(bright, win[1], cv::COLOR_GRAY2BGR);
   sprintf_s(str, "Bright area ");
   legend[1] = cv::Scalar(64);
   putText(legend[1], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);
```

```
        // intersect bright and MEI1sec
        cv::Mat blackhole;
        temp = bright & MEI1Sec;
        blackhole = temp & gray[1];
        cv::cvtColor(blackhole, win[3], cv::COLOR_GRAY2BGR);
        sprintf_s(str, "Intersect bright1 & MEI1Sec");
        legend[3] = cv::Scalar(64);
        putText(legend[3], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);

        // highlight the blackhole
        cv::Mat center;
        center = ~temp & bright;
        cv::cvtColor(center, win[4], cv::COLOR_GRAY2BGR);
        sprintf_s(str, "Possible center of flame");
        legend[4] = cv::Scalar(64);
        putText(legend[4], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);

        // Get the contours and sort it according to the area
        std::vector<std::vector<cv::Point>> contours;
        std::vector<cv::Vec4i> hierarchy;
        findContours(center, contours, hierarchy, cv::RETR_EXTERNAL,
cv::CHAIN_APPROX_SIMPLE);
        std::sort(contours.begin(), contours.end(), compareContourAreas);


        // loop through the contours (10 largest), check for hurricane pattern
        std::vector<std::vector<cv::Point>> filtered_contours;
        cv::Mat dilated_mask, differenceMask, flame_pattern;
        cv::Mat contour_mask;
        int flame_pixels, mask_pixels;
        cv::Mat kernel = cv::getStructuringElement(cv::MORPH_RECT, cv::Size(5, 5));
        for (int i = 0; (i < contours.size()) && (i < 10); i++) {
            contour_mask = cv::Mat::zeros(center.size(), CV_8UC1);
            drawContours(contour_mask, contours, i, cv::Scalar(255), cv::FILLED);
            dilate(contour_mask, dilated_mask, kernel, cv::Point(-1, -1), 3);
            absdiff(dilated_mask, contour_mask, differenceMask);
            flame_pattern = differenceMask & blackhole;
            flame_pixels = cv::countNonZero(flame_pattern);
            mask_pixels = cv::countNonZero(differenceMask);
            if (flame_pixels >= 0.5 * mask_pixels) {
                filtered_contours.push_back(contours[i]);
            }
        }

        cv::Mat result;
        cv::Mat color_copy;
        color[1].copyTo(color_copy);
        // check if any filtered regions
        if (filtered_contours.size() > 0) {
```

```
            found_count++;

            result = cv::Mat::zeros(center.size(), CV_8UC1);
            drawContours(result, filtered_contours, -1, cv::Scalar(255), cv::FILLED);
            cv::cvtColor(result, win[5], cv::COLOR_GRAY2BGR);
            sprintf_s(str, "Filtered region");
            legend[5] = cv::Scalar(64);
            putText(legend[5], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);

            drawContours(color_copy, filtered_contours, -1, cv::Scalar(0, 0, 255),
cv::FILLED);
         }
         color_copy.copyTo(win[6]);

      }
      if (found_count == 5 && !flag) {
         flag = true;
         message += videoName;
         message += "-";
         message += "Fire detected at frame: ";
         message += std::to_string(cur);
         message += "\n";
         sprintf_s(str, "Fire detected");
         legend[6] = cv::Scalar(64);
         putText(legend[6], str, cv::Point(5, 11), 1, 1, cv::Scalar(250, 250, 250), 1);
      }
      // reset the count every second
      if (cur % fps == 0) {
         found_count = 0;
      }
      // cur frame becomes prev frame
      color[1].copyTo(color[0]);
      gray[1].copyTo(gray[0]);
      cur++;
      prev++;

      // Check for key press to exit
      if (cv::waitKey(30) == 27) { // Press ESC to exit
         break;
      }
      cv::imshow(videoName, outputWindow);
      outputVideo << outputWindow;
   }
   if (found_count < 5 && !flag) {
      message += videoName;
      message += "-";
      message += "No fire detected";
      message += "\n";
   }
```

```
    // Release the VideoCapture and close the window
    inputVideo.release();
    outputVideo.release();
    cv::destroyAllWindows();

    outputFile << message;
    outputFile.close();
      return 0;
}
```

Edited videos detail:

Video name	Starting time	Ending time
Arson003_x264_2	29:07	1:29:03
Arson008_x264_2	10:13	17:20
Arson009_x264_2	00:00	11:01
Arson027_x264_2	00:00	52:16
Arson027_x264_3	52:20	1:30
Arson028_x264_2	31:02	1:20:07
Arson034_x264_2	37:23	1:19:14
Arson040_x264_2	18:07	1:32:11

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

1. WORK DONE **- Working on detecting smoke, not much progress**
2. WORK TO BE DONE **- Study the code given** **- Continue to work on the system and improve the accuracy of detecting flame**
3. PROBLEMS ENCOUNTERED **- Not sure how to define the darker region to detect smoke**
4. SELF EVALUATION OF THE PROGRESS **- Need to put in more work, the progress is likely to be delayed due to holiday**

Supervisor's signature

21 Feb 2024

lee

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

1. WORK DONE **- Not much progress**
2. WORK TO BE DONE **- Continue to work on the system and improve the accuracy of detecting flame**
3. PROBLEMS ENCOUNTERED **- Not sure how to differentiate flames from other motion**
4. SELF EVALUATION OF THE PROGRESS **- Not much progress due to holiday**

Supervisor's signature

21 Feb 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	**Study week no.: 6**
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

1. WORK DONE **- Isolate the brighter region (possible fire)**
2. WORK TO BE DONE **- To detect the hurricane pattern around black hole (intersecting bright region and MEI)** **- Run through some of the videos to generate some demo**
3. PROBLEMS ENCOUNTERED **- No problem for now**
4. SELF EVALUATION OF THE PROGRESS **- Progress is too slow, have to put in more time to speed up**

Lee

Supervisor's signature

7 Mar 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	**Study week no.: 7**
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

1. WORK DONE **- Intersect the bright area with MEI1Sec (to identify the hurricane pattern and blackhole which might indicates the center of flame)** **- Go through few videos to get some output**
2. WORK TO BE DONE **- To filter out the real flames from false positive**
3. PROBLEMS ENCOUNTERED **- Not sure how to differentiate the actual flame, because other bright areas might appear (shown in video)**
4. SELF EVALUATION OF THE PROGRESS **- Progress is too slow, have to put in more time to speed up**

Supervisor's signature

13 Mar 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project I / Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

1. WORK DONE **- Extract the largest region from the intersection of MEI1Second and bright region** **- Draft a block diagram**
2. WORK TO BE DONE **- To filter out the real flames from false positive**
3. PROBLEMS ENCOUNTERED **- Not sure how to differentiate the actual flame, because other bright areas might appear**
4. SELF EVALUATION OF THE PROGRESS **- Progress is too slow, have to put in more time to speed up**

Supervisor's signature

20 Mar 2024

lee

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 10
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

1. WORK DONE **- Block diagram for MEI1Second**
2. WORK TO BE DONE **- Go through more videos and study the pattern** **- Improve the current system so that it can generalize and work for more videos**
3. PROBLEMS ENCOUNTERED **- To filter out the real flames from false positive**
4. SELF EVALUATION OF THE PROGRESS **- Progress is too slow, have to put in more time to speed up**

Supervisor's signature

4 Apr 2024

lee

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 12
Student Name & ID: Lee Yong Jin 20ACB01393	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Video Surveillance: Anomaly Action Detection at front yard	

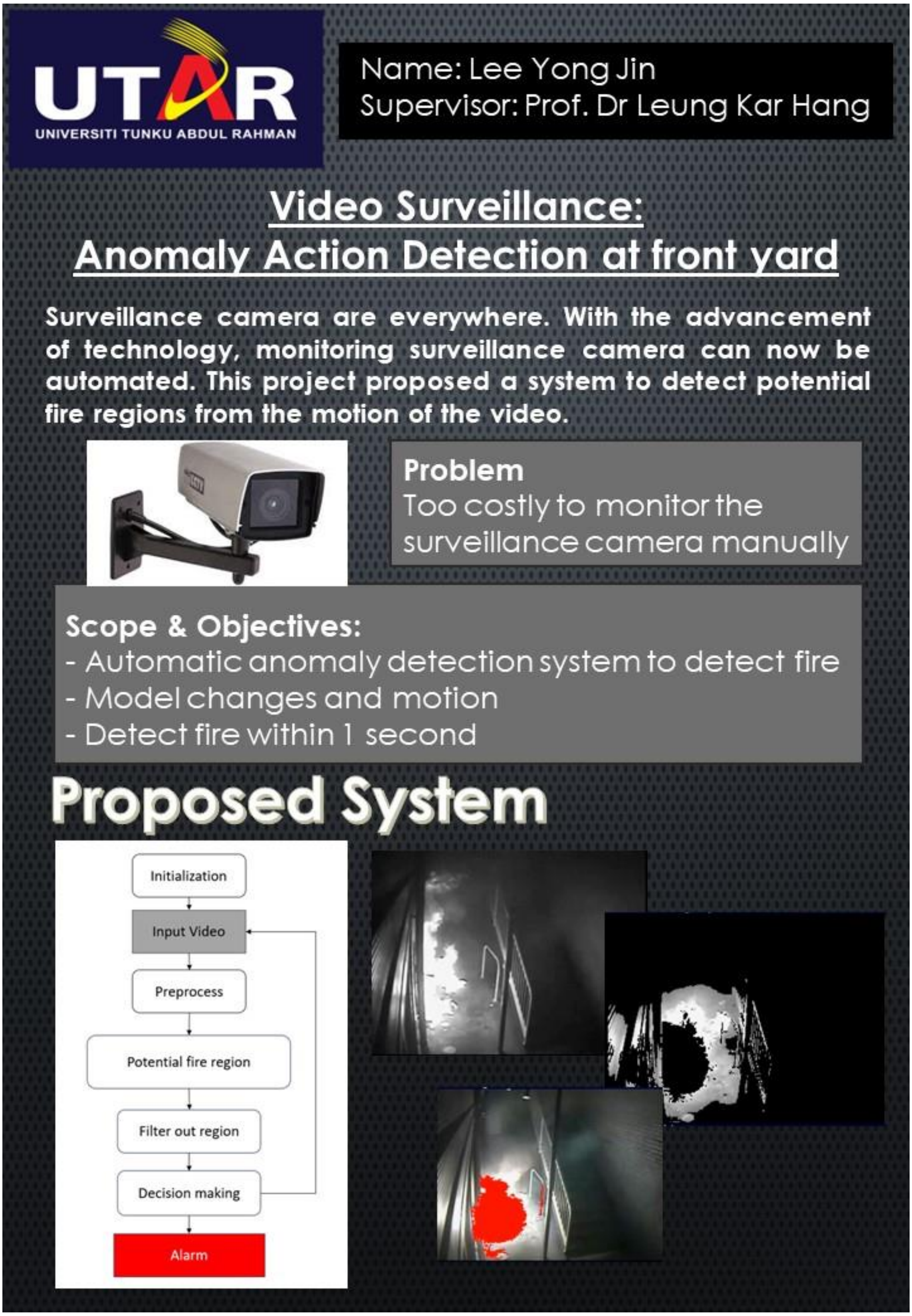
1. WORK DONE
- Started the report
- Try to further improve the system using color feature
2. WORK TO BE DONE
- Complete the report
- Finalize the system
3. PROBLEMS ENCOUNTERED
- Not sure if the color feature is workable
- Need to confirm about the structure of the report
4. SELF EVALUATION OF THE PROGRESS
- Progress is on track

Supervisor's signature

22 Apr 2024

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Name: Lee Yong Jin
Supervisor: Prof. Dr Leung Kar Hang
Video Surveillance:
Anomaly Action Detection at front yard
Surveillance camera are everywhere. With the advancement of technology, monitoring surveillance camera can now be automated. This project proposed a system to detect potential fire regions from the motion of the video.
Problem
Too costly to monitor the surveillance camera manually
Scope & Objectives:
- Automatic anomaly detection system to detect fire
- Model changes and motion
- Detect fire within 1 second
Proposed System
Initialization
Input Video
Preprocess
Potential fire region
Filter out region
Decision making
Alarm

PLAGIARISM CHECK RESULT

20ACB01393_FYP2

ORIGINALITY REPORT

12%	6%	7%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	eprints.utar.edu.my Internet Source	1%
3	Wang, Tian, and Hichem Snoussi. "Detection of Abnormal Events via Optical Flow Feature Analysis", Sensors, 2015. Publication	1%
4	Trong Nguyen Nguyen, Jean Meunier. "Anomaly Detection in Video Sequence With Appearance-Motion Correspondence", 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019 Publication	1%
5	"Intelligent Computing", Springer Science and Business Media LLC, 2022 Publication	1%
6	openaccess.thecvf.com Internet Source	1%

7	Yao Tang, Lin Zhao, Shanshan Zhang, Chen Gong, Guangyu Li, Jian Yang. "Integrating prediction and reconstruction for anomaly detection", Pattern Recognition Letters, 2020 Publication	1 %
8	Weixin Luo, Wen Liu, Dongze Lian, Shenghua Gao. "Future Frame Prediction Network for Video Anomaly Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021 Publication	<1 %
9	Bowen Lu, Zhihan Lv, Songhao Zhu. "Pseudo-3D Residual Networks Based Anomaly Detection in Surveillance Videos", 2019 Chinese Automation Congress (CAC), 2019 Publication	<1 %
10	Rashmika Nawaratne, Damminda Alahakoon, Daswin De Silva, Xinghuo Yu. "Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance", IEEE Transactions on Industrial Informatics, 2020 Publication	<1 %
11	hq-o.com Internet Source	<1 %
12	www.frontiersin.org Internet Source	<1 %
	export.arxiv.org	

13	Internet Source	<1%
14	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
15	www.ijert.org Internet Source	<1%
16	Submitted to University of Warwick Student Paper	<1%
17	Weixin Luo, Wen Liu, Dongze Lian, Shenghua Gao. "Future Frame Prediction Network for Video Anomaly Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022 Publication	<1%
18	repositorium.sdum.uminho.pt Internet Source	<1%
19	www.omicsdi.org Internet Source	<1%
20	repository.nwu.ac.za Internet Source	<1%
21	rrs.mmu.edu.my Internet Source	<1%
22	www.slideshare.net Internet Source	<1%

23	Q. Zhang. "Urban built-up land change detection with road density and spectral information from multi-temporal Landsat TM data", International Journal of Remote Sensing, 8/22/2002 Publication	<1%
24	fict.utar.edu.my Internet Source	<1%
25	www.mdpi.com Internet Source	<1%
26	"Machine Learning and Knowledge Discovery in Databases", Springer Science and Business Media LLC, 2017 Publication	<1%
27	Lin Wang, Xiangjun Wang, Feng Liu, Mingyang Li, Xin Hao, Nianfu Zhao. "Attention-guided MIL weakly supervised visual anomaly detection", Measurement, 2023 Publication	<1%
28	Zhiguang Zhou, Yize Li, Yuna Ni, Weiwen Xu, Guoting Hu, Ying Lai, Peixiong Chen, Weihua Su. "VisCI: A visualization framework for anomaly detection and interactive optimization of composite index", Visual Informatics, 2024 Publication	<1%

29	Congqi Cao, Xin Zhang, Shizhou Zhang, Peng Wang, Yanning Zhang. "Weakly Supervised Video Anomaly Detection Based on Cross-Batch Clustering Guidance", 2023 IEEE International Conference on Multimedia and Expo (ICME), 2023 Publication	<1 %
30	Ning Jiang, Jie Zhang. "A Smart Mat for People Identification Based on BP Neural Network", 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), 2023 Publication	<1 %
31	"Anomalous behaviors detection in moving crowds based on a weighted convolutional autoencoder-long short-term memory network", IEEE Transactions on Cognitive and Developmental Systems, 2018 Publication	<1 %
32	"Computer Networks and Inventive Communication Technologies", Springer Science and Business Media LLC, 2021 Publication	<1 %
33	Thakare Kamalakar Vijay, Nitin Sharma, Debi Prosad Dogra, Heeseung Choi, Ig-Jae Kim. "A multi-stream deep neural network with late fuzzy fusion for real-world anomaly	<1 %

detection", Expert Systems with Applications, 2022
Publication

34	Weixin Luo, Wen Liu, Dongze Lian, Jinhui Tang, Lixin Duan, Xi Peng, Shenghua Gao. "Video Anomaly Detection With Sparse Coding Inspired Deep Neural Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019 Publication	<1%
35	Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, Tao Mei. "To Create What You Tell", Proceedings of the 2017 ACM on Multimedia Conference - MM '17, 2017 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LEE YONG JIN
ID Number(s)	20ACB01393
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Video Surveillance: Anomaly Action Detection at front yard

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 12 **%** **Similarity by source** Internet Sources: 6 % Publications: 7 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Leung Kar Hang	Name:
Date: 25 April 2024	Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB01393
Student Name	Lee Yong Jin
Supervisor Name	Prof. Dr Leung Kar Hang

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

Lee

____

(Signature of Student)

Date: 25/4/2024