Estimating The Last Seen Frame Time of a Static Removed Object in a Recorded Video Feed using Background Subtraction and Feature Extraction.

A Thesis Submitted for the Degree of Master of Computer Science

A.M.C Perera

University of Colombo School of Computing

2024

Declaration

Name of the student: A.M.C Perera
Registration number: 2018/MCS/066
Name of the Degree Program: Master of Computer Science
Project/Thesis title: Estimating the last seen frame time of a static removed object in a recorded video feed using Background Subtraction and Feature Extraction

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.

3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.

4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.

5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.

6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date**
	12-10-2024

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	Supervisor 2	Supervisor 3
Name	Dr. Ajantha Atukorale		
Signature			
Date	12/10/2024		

Dedicated To

My beloved husband Yohan Senevirathne
for his unconditional love, patience,
support, faith, and encouragement

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to all those who have contributed to the successful completion of my Master's thesis in Computer Science, titled "Estimating the last-seen frame time of a static removed object in a recorded video feed using background subtraction." This research project has been a journey of learning, growth, and collaboration, and I am truly thankful for the support and guidance I have received. First and foremost, I am immensely grateful to my supervisor, Dr. Ajantha Atukorale, for his support, insightful guidance, and invaluable mentorship throughout the entire research process. His expertise, constructive feedback, and dedication played a pivotal role in shaping the direction and quality of this work. I am thankful to the faculty and staff of the University of Colombo School of Computing, whose academic environment provided the necessary resources and infrastructure for conducting this research. The friendly and smart environment helped me a lot in my studies and research. Additionally, I would like to express my gratitude to my colleagues who provided support, insightful discussions, and encouragement throughout the course of this project. The exchange of ideas and perspectives greatly enriched the overall research experience. Last but not least, I extend my appreciation to my family for their invaluable support, understanding, and encouragement during the challenging phases of this academic pursuit. Their belief in my capabilities motivated me to persist and achieve success.

ABSTRACT

This research project aims to automate the process of estimating the last seen frame time of a static object removed from a recorded video feed using background subtraction and feature extraction techniques. The motivation behind the study is to enhance the efficiency of video analytics in investigative processes, reducing the need for time-consuming manual reviews of video footage. The system operates using a binary search-inspired algorithm, where the middle frame of the video is analyzed first, and the search is progressively narrowed down to accurately determine the object's disappearance. The system architecture includes five key components: video acquisition, frame preprocessing, background subtraction, feature extraction, and the calculation of the last seen frame time. ORB (Oriented FAST and Rotated BRIEF) is employed for feature extraction, providing a fast and efficient method to detect key object features. The system was tested in a high-performance environment with an Intel i10 processor (2.6 GHz CPU) and 16GB of RAM to ensure fast and efficient processing. OpenCV was used as the primary computer vision library, alongside supporting libraries such as NumPy, Pillow, and Scikit-image. A custom Tkinter GUI was developed for user interaction, and Python was the programming language of choice. While the system performs well in scenarios with simple, low-noise backgrounds, accurately estimating object disappearance times in both short- and long-duration videos, it faces challenges in more complex environments. The system's performance deteriorates in the presence of complex backgrounds, noise, and shadows, leading to significant discrepancies in the estimated disappearance times. To address these limitations, future enhancements include improving the feature extraction algorithm, incorporating shadow detection and removal methods, and developing advanced noise filtering techniques. By integrating these improvements, the system can become more robust and reliable across diverse video conditions, making it highly useful for security and forensic investigations. The proposed approach offers a practical and efficient solution for automated video analysis, with promising applications in fields requiring precise tracking and event identification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Motivation

The motivation behind this research project lies on increasing the efficiency and effectiveness of video analytics in investigative processes. The manual review of entire video footage can be a tedious and resource-intensive task for investigators. By focusing on the automatic estimation of key frame times, especially the last seen frame time of a removed object, the research aims to streamline the video analysis process. This automation allows investigators to pinpoint specific segments of the video, minimizing the need for exhaustive manual reviews and enabling more targeted investigations. The proposed approach not only saves valuable time and resources but also empowers investigators to make informed decisions based on accurate estimations of object disappearances. Ultimately, the research seeks to contribute to the improvement of video surveillance and object tracking systems, making them more efficient tools for law enforcement and investigative agencies.

1.2 Statement of the Problem

"Estimating last seen frame time of a removed static object addresses a gap, within the domain of security and surveillance. The accurate estimation of the last timestamp of a removed object holds immense practical significance."

With the rise of Artificial Intelligence, computer vision applications have employed increasingly to make our life easier. Among these applications, security and surveillance have indeed benefited significantly. AI-powered these systems can analyze vast amounts of video data in real time, and it helps in monitoring suspicious activities, recognizing faces and tracking object of interest. When dealing with suspicious activity recognition, it includes identifying unusual patterns or behaviors, such as unauthorized access, loitering in restricted areas, or erratic movements. Sophisticated AI algorithms can quickly analyze video feeds in surveillance and distinguish between normal and potentially threatening actions, enabling security systems to respond promptly to potential risks. In addition to the aforementioned suspicious activities, there is also the aspect of abandoned object detection, which plays a crucial role in identifying and flagging unattended items in public spaces. Similarly, detection of removed or stolen objects is another significant concern in various settings, requiring advanced security measures and innovative technologies to improve attention and prevent unauthorized removal or theft.

It can be observed that, in the existing literature, a good number of research have been dedicated to enhancing the ability of abandoned object detection over the past few years. However, there has been relatively limited research focused on detecting removed or stolen objects. This study seeks to focus on removed object detection, recognizing the substantial importance of advancing this specific area within the field of security and surveillance. Removed object detection holds significant importance in various areas including museums and art galleries, transportation hubs, retail environments and public events. In museums and art galleries, house valuable artifacts and priceless artworks are vulnerable to theft. Similarly, transport hubs such as airports and train stations, retail environments and public events are frequent targets for theft. There are certain instances where thefts are identified a few hours or a couple of days later. In such scenarios, the delayed detection of thefts often poses challenges for timely intervention. However, identifying the last seen time of the particular object can be crucial in narrowing down the timeframe during which the theft occurred. This information aids security personnel in reviewing surveillance footage more efficiently and pinpointing specific segments of video.

1.3 Research Aims and Objectives

1.3.1 Aim

To implement an application to estimate the last seen frame time of a removed object in a recorded video feed.

1.3.2 Objectives

Possible objectives that could support to build the application,

- Develop an algorithm or technique to enhance the accuracy of the timestamp of when an object was last observed.
- Identify the removed object from the scene.
- Find the last seen frame time for a given object (object annotation).

- Create a user-friendly interface for the application that allows users to interactively select and track objects of interest, visualize the tracking process.
- Develop tutorials, documentation, and user support mechanisms to help users effectively utilize the application and interpret the last seen frame time estimates.

1.4 Scope

The scope of this research project revolves around developing a method for automatically estimating the last seen frame time of a removed object in the video footage. The primary focus is on enhancing the efficiency and effectiveness of video analytics, particularly in investigative processes. The project aims to address the challenge investigators face when manually reviewing extensive video footage by providing an automated solution for identifying the key moment of the disappearance of objects.

In-scope

- Development of an application to estimate the last seen frame time of a removed object in a recorded video feed.
- Development of algorithms and techniques aimed at enhancing the accuracy of the last seen frame time estimation.
- Object annotation – enabling the application to find the last seen frame time for a given object.
- User friendly interface – creating a user-friendly interface that allows users to interactively select and track objects of interest, visualize the tracking process and view the estimated last seen frame time.
- Documentation and user support – developing tutorials, documentation, and user support mechanisms to assist users in effectively utilizing the application and interpreting the last seen frame time estimates.

It's essential to identify what are out of scope areas or aspects that will not be addressed or considered in the course of the research. Here are some potential elements that may be out of scope for the project,

Out-of-scope

- Timestamp of the first appearance of an abandoned object – this application does not involve approximating the first appearance frame time of an abandoned object.
- Real time tracking – the project does not aim to provide real time tracking capabilities for an object in a live video. The focus is on recorded video feed.
- Hardware development – the project does not involve the development of specific hardware components or devices for video recording or processing.

1.5 Structure of the Thesis

The thesis follows a structured format to comprehensively address the research project on estimating the last seen frame time of a static removed object. The introductory section provides a background overview, introducing the significance of estimating the last seen frame time and outlining the research question. The subsequent section, Chapter 2, conducts a thorough review of relevant literature, contextualizing the research within existing knowledge and identifying gaps that this study aims to fill. Chapter 3 describes the research methodology, detailing the approach taken to estimate the last seen frame time, including the chosen algorithms or techniques. Following this, Chapter 4 presents the evaluation by experimenting the developed prototype with manually recorded video files. The conclusion and future work will be the focus of Chapter 5.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

The detection of stolen or removed objects in video footage can be considered a critical area of research due to its applications in surveillance, security, and law enforcement. This literature review explores the advancements and challenges in stolen or removed object detection, focusing on the techniques, methodologies, and technologies employed in research studies. It can be observed, the majority of research related to stolen or removed object detection has been conducted along with abandoned object detection. An abandoned object is an item that has been left behind or discarded by its owner without any apparent intention of returning for it. Identification of an abandoned object on real-time can prevent the terrorists attack through an automated video surveillance system.

2.1.1 Research in Abandoned and Removed/stolen object detection

In this section, a description of several research studies is provided, spanning from most recent to the earlier works.

Qasim [1] introduced an innovative method for detecting abandoned objects that can effectively handle both suspicious and non-suspicious scenes. This approach is structured into two main stages: the Scene Classification Module (SCM) and the Object Detection Module (ODM). The first stage, SCM, is designed to capture temporal features by utilizing a sequential model, which closely monitors the scene to identify any potentially abandoned objects. The sequential model is adept at analyzing the changes over time in the scene, detecting subtle movements or the sudden appearance of stationary objects that may indicate an abandonment. Once such an object is detected, it triggers the next phase of the process, initiating the ODM for further analysis. The second stage, ODM, employs the highly accurate YOLOv8l model to precisely locate and detect the identified objects within the scene. YOLOv8l, a state-of-the-art object detection model, is well-suited for its task due to its speed and accuracy in detecting objects in real-time. By using this model, the method is able to accurately identify and localize abandoned objects, ensuring precise detection. In terms of performance, the proposed method demonstrated outstanding accuracy, achieving rates of 99.20% on the PETS 2006 dataset and 99.70% on the ABODA dataset. These results highlight the method's exceptional ability to localize the target object within diverse environments, confirming its high level of precision. What sets this method apart from others is its seamless integration of scene classification and object detection, which enables it to distinguish between objects that pose varying levels of risk. By incorporating both stages, the approach not only ensures real-time detection of abandoned objects but also provides context for different types of objects that may require different levels of attention or response. This is particularly beneficial for public spaces where security is critical, as it allows for a more nuanced detection system that can differentiate between potentially hazardous objects and those that are less of a concern. Ultimately, the proposed method contributes significantly to the advancement of security measures by offering an enhanced solution for abandoned object detection, improving safety in monitored areas. However, this study lacks operating in estimating the last seen time of a removed static object in the scene.

Park [2] proposed an efficient method to distinguish abandoned objects, stolen objects, and ghost regions in the surveillance video as the existing intelligent video surveillance systems that rely on the foreground analysis generated by the background subtraction have a problem that abandoned objects look like stolen objects and ghost regions. This approach consists of two main strategies: the first one is the dual background model for extracting candidate stationary objects, the second one is object segmentation based on mask regions with CNN features (Mask R-CNN) for providing the object mask information. They have conducted qualitative experiments on their proprietary dataset, specifically addressing discrimination concerns, using the proposed algorithm. These experiments yielded satisfactory outcomes, suggesting its potential for extensive utilization in automatically detecting stolen or abandoned items within open settings like exhibition halls and public parks, where conventional intrusion detection-oriented security systems face deployment challenges. However, this study has a lack of limiting to only recognizing trained objects due to native characteristics of the deep learning model and also not provide neither start timestamp of an abandoned object nor exit timestamp of a removed object.

Jadhav and Momin [3] presents a method for detecting unattended or removed objects in video surveillance data captured by a single static camera. The proposed method consists of three main components: i. foreground blob extraction, ii. object classification and iii. object identification and responsive actions. During the foreground blob extraction phase, dual background modeling approach is employed to distinguish between static regions that have undergone changes and the original background regions. Subsequently, background subtraction is performed to extract foreground objects and applied shadow removal techniques to get accurate foreground objects. Next, attributes of the extracted foreground objects such as height, width, size, color and time are considered in order to classify the objects as moving, stationary and removed. Then a rule-based classifier is employed for object identification. If a static object is recognized as a person, it is categorized as a still person. Then the system holds back from automatically classifying abandoned still persons, as they may be waiting for someone, leaving this decision to manual intervention. If the identified object is a baggage, indicating unattended baggage, the system triggers an alarm. In the case of a removed object, if the object is identified as a person, it signifies a still person leaving the area, and the system dismisses it. If it is identified as a baggage, indicating stolen objects, the system triggers an alarm. This proposed system achieved an accuracy of 84.57% in object detection. However, it encountered challenges in accurately identifying objects in crowded scenes and regions that were occluded. Yet, it does not provide important timestamps of start time of an abandoned object or exit time of a removed or stolen object.

Nam [4] introduced real-time detection methods of abandoned and stolen objects in complex videos containing occlusion, lighting changes and significant perspective distortion etc., considering spatio-temporal relationship between moving people and suspicious drops. To detect abandoned and stolen objects, static regions that have recently changed in the scene are identified through region-based background subtraction and contour-based ghost removal methods by incorporating space first and time first detection methods. The proposed system traces the history of an abandoned object to identify its owner. The activation of the owner retrieval process starts once an owner candidate has been associated with the abandoned object. In cases where owner B1 disappears from the detection zone, the split object B2 is labeled as abandoned. If B1 eventually returns to B2, the alarm is stopped. If B2 is moved by a non-owner B3, B2 is classified as a stolen object. In abandoned object analysis, the process is divided into three primary sections: i. moving scanning, ii. spatial change analysis, and iii. temporal change analysis. In the moving scanning phase, a non-moving object is considered an abandoned object candidate if it remains in the same position for a certain amount of time. First, in spatial change analysis, the Space First Detection (SFD) method measures the distance between a moving object and a non-moving object. If an object is identified as non-pedestrian and has moving objects within a defined radius, it is not considered abandoned. Otherwise, the method calculates how long time the object does not move through temporal change analysis. Finally, if the spending time is longer than a specific amount of time, the object is classified as an abandoned object. On the other hand, the Time-First Detection (TFD) method starts with temporal change analysis followed by spatial change analysis. Once an object is identified as a potential abandoned object, the method continuously monitors its status until the object either shifts from its position or merges with another blob. In stolen object analysis, if a detected blob is an own object or an abandoned object, the object becomes a stolen object candidate while it remains in the same position for a certain amount of time. If there is person P1 in a certain radius around the detected object, the object belongs to P1. Otherwise moving analysis and owner tracking methods determine whether the object is moving with P1 or not. If the candidate object is moved with P1, the object is recognized as a property of P1. Otherwise, the candidate object is classified as a stolen object. This research makes two noteworthy contributions to society. Firstly, it addresses the challenge of detecting abandoned and stolen objects in situations involving occlusion, and secondly, it aims to minimize false alarms and missed detections in complex scenes. However, the proposed system lack of providing either the first appearance time of an abandoned object or the last seen frame time of a removed or stolen object.

Rakumthong [5] proposed a new design and implementation to detect abandoned and removed objects in public places in real time or offline. The detection process consists of four major components: i. video acquisition ii. video processing iii. event detection and iv. result presentation. In video processing module, Gaussian blur is used to reduce white noise caused by random fluctuations and background subtraction is performed to detect moving regions. The event classification module is responsible for classifying type of objects and to identify if that object is unattended or stolen. First, the CascadeClassifier is used. Objects classified by this function are identified as people, otherwise, they are considered static objects. Once the object is classified in the current frame, its corresponding areas in both the background and the first frame are cropped from the frame. Next, the background subtraction method and contour analysis are employed once more on the cropped frames to discover the presence of the object in the first frame. The absence of an object in the first frame indicates the likelihood of an unattended object event. In contrast, for the stolen object events, an object appears in the first frame during processing and subsequently disappears or moves from the current frame. The experiment results demonstrated that the performance of people classification is significantly lower, primarily due to the limitations of the CascadeClassifier class in the object classification process. It fails to detect people when there are changes in their posture or style. The system also does not provide the first appearance time of an abandoned object and last seen time of a removed or stolen object.

Chai [6] presents a method to detect and classify object occlusion and object removal event, enhancing system accuracy and performance. The method involves setting the coordinates of a targeted object, which may be an abandoned object or user defined etc. Events are categorized as normal or abnormal, with abnormal events further divided into object removal and object occlusion. The process consists of two phases: abnormal event detection and abnormal event classification. Object region is introduced to complete the detection. Object region is a region part on the targeted object which helps in detection by revealing texture differences between the background and current image during abnormal events. When an abnormal event happened, the texture in the object region is different for background and current image. But for normal event, the texture will remain the same. The process begins with extracting edges in the images using the Canny detector, which are then utilized to assess texture similarity on the targeted object. Subsequently, the subtraction between the background and current image on the targeted object is performed, determining the total white pixels to classify whether an abnormal or normal event has occurred. If an abnormal event is detected, the abnormal event classification phase distinguishes between occlusion and removal events by comparing the outer region, which is larger than the object and excludes its region part. For occlusion events, differences in texture within the outer region for the current image and background are considered, while removal events exhibit similar textures. The classification extracts the edges on the outer region for the background model and current image, followed by calculating edge similarity on the outer region. In the experiment, the proposed method successfully detected the most of abnormal events. However, there are situations where the method may fail, such as when the targeted object is occluded by one person while another person removes the object. In such cases, the system may incorrectly conclude that the event is an occlusion event. To address this issue, they have proposed a solution by integrating the region-growing method with active contour. However, the system lack of providing the last seen time of the removed or stolen object.

Singh and Agrawal [7] describes an interactive application designed to address the real-time detection of abandoned and removed objects in a video stream using a modular approach. The system divides the complex task into simpler sub-tasks, each handled by an independent module, allowing for the use of various methods for solving specific problems. In each module, several existing methods have been included with the possibility of adding new methods accordingly. This requires minimal reprogramming of the system. The user can observe and compare the performance and accuracy of different methods according to each scenario by switching between these methods. The system consists of seven modules dedicated to preprocessing, background modeling and subtraction, foreground analysis, blob extraction, blob tracking, abandonment analysis and blob filtering. Preprocessing module performs two-man functions: i. contrast enhancement and ii. noise reduction. Contrast enhancement improves the quality of the video. This can be carried out by using several methods such as histogram normalization, image filtering and contrast stretching. Noise reduction decreases the white noise present in the input frame. The system includes three distinct background subtraction algorithms: i. Gaussian Mixture model, ii. Adaptive Median and iii. Running Gaussian Average. These algorithms have been modified to perform object level background updating rather than the conventional pixel level approach. The background subtraction output may have noisy portions containing false foregrounds due to sudden lighting changes and actual foregrounds like shadows. To address this, a separate foreground analysis stage is essential. Subsequently, Blob extraction identifies connected components in the foreground mask, extracting meaningful objects and discarding any blobs smaller than a specified threshold. Next, Blob tracking tracks only the static objects in the scene and this involves comparing each blob in the incoming frame with the existing blobs in the tracking system. For an existing blob to match a new blob, their areas and positions must differ by less than a threshold. In the abandoned analysis module, it prevents false detections of abandoned objects caused by the 'ghost effect' when an object in the background is removed. Finally, this module classifies a static blob into one of four categories: i. abandoned, ii. removed, iii. State change and iv. still person. A blob identified as a state change, or a stationary person is excluded from the tracking system. The application was tested on a number of publicly available and custom-made videos, demonstrating accuracy comparable to contemporary systems while maintaining real-time performance in a typical setting. Notably, the system achieved a very low rate of false positives. The system, being modular, serves as a foundational framework that can be calibrated for optimal performance in diverse scenarios. However, this framework does not provide any methodological approach to find the last seen time of a removed or stolen object.

Miguel and Martínez [8] presents a novel approach to detect abandoned and stolen objects based on fusion of evidence provided by three simple detectors named low gradient detector, high gradient detector and color histogram detector. These detectors rely on the analysis of shape and color information from static foreground regions. In other words, this study primarily focuses on integrating edge and color information to differentiate between unattended and stolen objects, in contrast to other research endeavors that rely solely on either edge information or color information for distinguishing between unattended and stolen objects. Initially, the system identifies the moving regions within the scene. Subsequently, these regions are classified as either static or dynamic objects, as well as human or non-human objects. Finally, objects detected as static and non-human are analyzed with each detector. Information obtained from these detectors is combined to choose the most optimal detection hypotheses. The processing steps for the unattended and stolen object detection module starts with preprocessing the shape extracted from the binary foreground mask to the real object shape. This adjustment is made by active contours, and this has to be done because estimation errors in the object shape can reduce the robustness of the algorithm used. Then the three independent detectors are applied to the candidate object. Each detector computes two evidence values for the hypothesis of an object being unattended and stolen. Finally, a fusion scheme is applied on the evidence obtained from the three detectors. Following the fusion process, two confidence measures are computed to assess the probability of the object being unattended or stolen. Subsequently, the maximum posteriori criterion is utilized to make the decision on whether the object is unattended or stolen. The experimental results demonstrated that this simple proposed scheme is significantly more efficient and stable in comparison to the independent detectors applied on their own. Yet, this research does not provide either the first appearance time of an unattended object or the last seen time of a stolen or removed object.

Bird [9] presented a method for detecting abandoned objects in real world conditions. This approach works in real time, uses color video and benefits from not detecting still people as abandoned objects. This aspect differentiates this work from other researches. The system presented here also operates continuously, so it requires a learned background model that adapts to variations in natural light throughout the day. The system uses standard algorithms to perform low level image processing to detect and track blobs. Using the data generated by these low-level algorithms, a two-tiered high-level system is employed to detect stationary objects and to determine which still objects correspond to actual abandoned objects, and which do not. The application of this logic ensures that a person sitting or sleeping on a bench is not misclassified as an abandoned object, thereby enhancing the accuracy and precision of object detection and classification within the system. In the low-level processing, the background modeling is performed by using the mixture of Gaussians method. Four

Gaussians are used to represent the color at every pixel. The background modeling is restricted to user specified regions of interest in the image. No background learning is performed on the areas outside this region mask. The purpose of this is to block out areas of the image where any background changes detected can safely be considered noise (such as walls), and to remove areas that are too far from the camera for accurate abandoned object identification. This mask is further updated by the long-term logic to prevent abandoned objects from being learned into the background. The noise reduction of the binary foreground mask is performed using a structural noise reduction algorithm. Blob extraction is then performed on the binary foreground mask. This system detects abandoned objects even if they are occluded by moving crowds of people for periods of time. The results show that the method works best in sparsely populated areas where people are regularly detected separately. This system also presents the timestamp corresponding to the time an abandoned object is first detected. However, this research does not engage in detecting removed or stolen objects.

Ferrando [10] presented a novel approach to identify abandoned and stolen objects in a guarded indoor environment. The proposed system consists of three different processing levels: i. low level ii. middle level and iii. high level. Each level is organized in sub-modules, dedicated to specific task. The low level hosts an image processing module which identifies the pixel region, blobs, through the respective bounding boxes coordinates. Background updating module also resides in the same level in order to maintain the object position. This background updating is modifying through the feedback from high level modules. Blobs classified as abandoned objects are excluded from the updating, in order to preserve their position in the following frames, otherwise blobs classified as stolen objects are absorbed immediately in the background. In the middle level, several features are extracted. This level also acts as an interface between the low-level pixel operations and the high-level modules, which operate at the feature level. Feature extraction module and object tracking module are two main modules which reside in the middle level. In object tracking module, position and color features are used. The high-level modules aim at detecting suspicious events as abandoned or stolen objects. Both events involve a division among blobs, one belonging to the person class “human” and the other belonging to object class “non-Human”. Following the "Human" and "non-Human" division, the system categorizes static objects as either abandoned or stolen. This system also has the benefit of detecting multiple abandoned or stolen objects at a time. Yet this research does not provide the start timestamp of a dropped object or the exit timestamp of a stolen or removed object.

D'Orazio [11] proposed a new method to identify abandoned and removed object in video sequences. This approach primarily consists of three stages. The first step is detecting the moving region in the scene by subtracting the current frame from the background model. This is called motion detection. Subsequently, a shadow removal algorithm is applied to extract the true form of identified objects. Finally, moving objects are classified as abandoned or removed by analyzing the boundaries of static regions. The decision between abandoned or removed object has been taken by a new technique. When a static foreground region is detected, the segmented image is considered after shadow removing step relative to current frame. The next step consists in applying an edge algorithm on the foreground region in the segmented image. The same portion is selected in the real image on which the edge algorithm is newly applied. Now the two images containing the edges are matched and a similarity measure is calculated. Finally, if this measure is more than a predefined threshold, then the object is categorized as abandoned. Otherwise, it is determined the object is removed from the background. However, this research is lack of providing the last seen time of a removed object or the first appearance time of an abandoned object.

The following table illustrates the key features, advantages, limitations, and timestamp handling capabilities of the some of the above research works focused on detecting abandoned and stolen objects in video surveillance systems.

Research Work	Proposed Method	Advantages	Limitations	Timestamp Handling (Last Seen/First Seen)
Qasim et al., 2024	Two-stage method: Scene Classification Module (SCM) using sequential model and Object Detection Module (ODM) using YOLOv8l	High accuracy in abandoned object detection (99.20% PETS 2006, 99.70% ABODA) and context-aware detection	Lacks support for estimating the last seen time of a removed static object	Not provided
Park et al., 2020	Dual background model for extracting	Efficient differentiation of abandoned,	Limited to recognizing trained objects; does not	Not provided

	candidate stationary objects and Mask R-CNN for object segmentation	stolen, and ghost regions in complex scenes	provide timestamps of abandoned/removed objects	
Jadhav and Momin, 2016	Dual background modeling, object classification (rule-based classifier), and responsive actions for unattended and removed items	Handles different object types; triggers alarms for unattended baggage and stolen objects	Struggles in crowded/occluded scenes; lacks timestamp information for abandoned/removed objects	Not provided
Nam, 2016	Spatio-temporal relationship analysis, space-first and time-first detection for abandoned/stolen objects	Addresses occlusion, lighting changes, perspective distortion, and false alarms	Does not provide the first appearance time of an abandoned object or the last seen time of a removed object	Not provided
Rakumthong et al., n.d.	Gaussian blur for noise reduction, CascadeClassifier for object classification, background subtraction, and contour analysis	Real-time and offline object detection with background subtraction and event classification	Low people classification accuracy due to limitations in CascadeClassifier; lacks timestamp information	Not provided
Chai et al., 2013	Abnormal event detection using Canny edge detector and abnormal event classification	Successfully distinguishes between object occlusion and removal	May fail when the targeted object is occluded by one person while another removes it; lacks timestamp	Not provided

	based on texture analysis		information	

Table 2.1 Research work comparison

Collectively, these studies demonstrate the potential for effective real-time detection of abandoned and stolen objects in complex scenes but highlight a consistent gap: the lack of focus on providing the precise last seen timestamp of removed or stolen objects. While the methods explored offer considerable promise in increasing accuracy and reducing false alarms, none fully address the temporal aspect of object disappearance, which is crucial for security systems aiming to track stolen or removed items in surveillance footage. Future research should aim to fill this gap by incorporating time-based tracking mechanisms into existing object detection frameworks, ensuring both spatial and temporal accuracy in abandoned and removed object detection systems.

2.1.2 A General Framework for Abandoned or Removed/stolen Object Detection from Video Surveillance

This section presents a comprehensive framework for the detection of abandoned or removed objects from a single static camera. According to the review by Tripathi [12], There are four important stages that should be considered: foreground object extraction, stationary object detection, classification them into human and non-human categories and alarm or alert message generation. Mostly researchers follow up these steps with different algorithms or approaches while exploring innovative techniques for enhancing each stage of the framework to improve the recognition accuracy.

2.1.2.1 Foreground Object Extraction

Foreground objects in the context of video processing refer to elements that stand out from the static or background environment. These objects are characterized by their motion or recent appearance in the video frame. Moving objects, such as people, vehicles, or animals, are considered part of the foreground. Additionally, newly arrived objects, which may initially be dynamic but become static after some time, are also categorized as foreground objects.

Foreground object extraction from the video is the first step in detecting abandoned or removed objects. In order to achieve this prominent task, background subtraction method is employed. Background subtraction is a robust and powerful method for identifying changes in a sequence of frames and to extract objects in the foreground. Later, in the section 2.1.2, background subtraction will be explained more precisely.

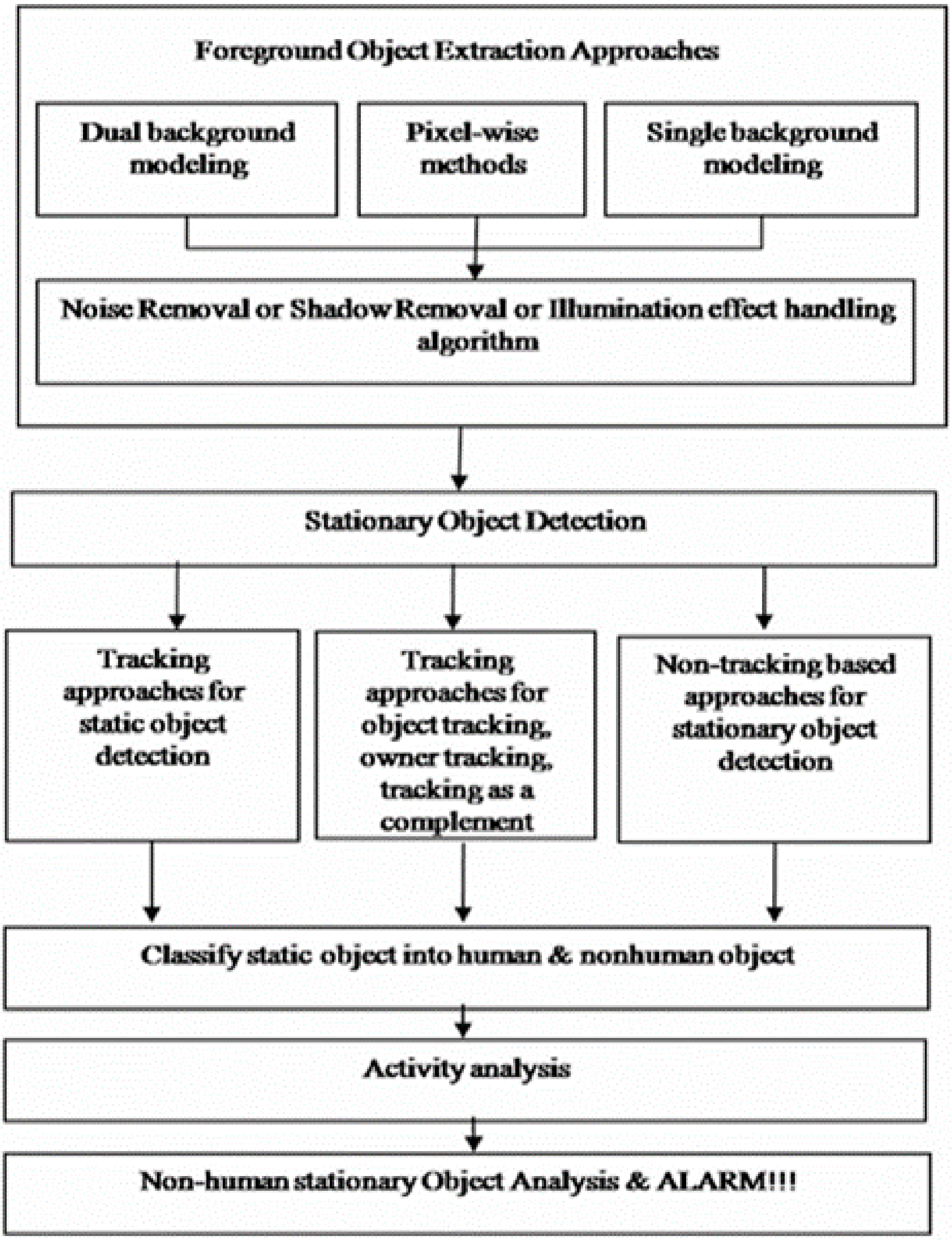

Figure 2.1 A General Framework for Abandoned or Stolen Object Detection

A. Moving Foreground Object Detection

In addition to employing the background subtraction technique, achieving this goal is also feasible through the utilization of change detection-based approach. The change detection methods subtract consecutive frames to detect motion and employ post-processing techniques to reconstruct the entire object. These methods are fast in respect to execution while lacking in accuracy. In contrast, modeling based approaches try to model the background using some temporal and/or spatial cues. A reasonably correct model for the background can help to separate the foreground objects much effectively compared to the previous class of methods. These methods can range from very simple to highly complex in implementation and execution.

B. Stationary Foreground Object Detection

In video surveillance, the identification of moving objects is easily done through the application of various background techniques since these techniques consider only moving objects as a foreground object. Therefore, whenever a new object arrives in the video and becomes static, after a while, it is absorbed in the background. In order to detect the stationary object from surveillance video, basically different approaches have been applied to extract the static object.

Dual background approach : Porikli [13] proposed a method which utilizes dual foreground extraction from dual background modeling which uses two different learning

rates: i. short term and ii. long term to detect temporary stationary foreground objects. This method has been introduced as an alternative to the tracking-based approaches that heavily depend on accurate detection of moving objects, which often fail for crowded scenarios. It can be observed that several researchers have used dual background modeling technique to detect abandoned and removed objects. An issue with this approach is its susceptibility to a high false alarm rate, which is typically cause by imperfect background subtraction resulting from ghost effect in stationary people and crowded scenes. Sometimes, temporarily static objects may also get absorbed by the long-term background model after a given time based on its learning.

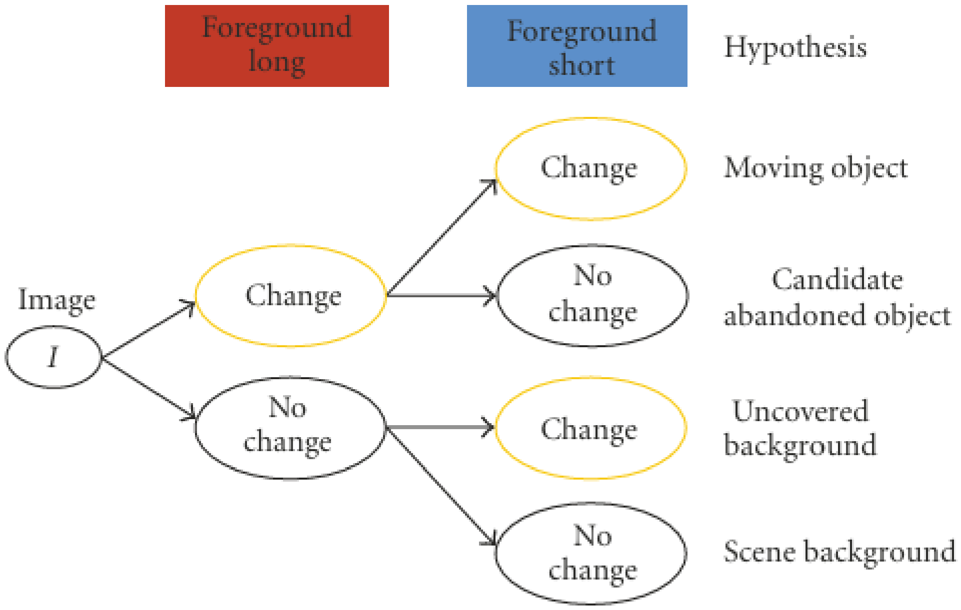

Figure 2.2 Hypothesis on long term and short-term foregrounds

Temporal dual rate background technique : Lin [14] proposed a temporal dual rate foreground integration method for static foreground estimation for single camera video images. This approach involves constructing both short- and long-term background models learned from an input surveillance video online. Subsequently, a simple pixel based finite state machine (PFSM) model was introduced and it uses temporal transition information to identify the static foreground based on the sequence pattern of each object pixel. Due to the utilization

of temporal transition information in the proposed approach, the influence of imperfect foreground extractions in the double-background models can be reduced, thereby improving the accuracy of the constructed static foreground inference.

Mixture of Gaussian models: This approach has been implemented using a variable number, denoted as "n," of Mixture of Gaussian models. Mainly three Gaussian Mixture Models have been used to detect the static foreground object, moving foreground object and removed foreground object in several researches. Tian [15] used three Gaussian Mixtures of Background Model in which first Gaussian distribution models the persistent pixels and represents to the background pixels, static regions are updated to the second Gaussian distribution and third Gaussian distribution represents to the quick changing pixels.

C. Noise Removal, Shadow Removal, and Illumination Handling to Reduce False Detections

Detecting foreground objects without interference from noise, illumination effects, and shadows poses a significant challenge in the field of intelligent video surveillance. The presence of noise in a video input poses a significant challenge in accurately identifying objects. Noise introduces unwanted elements that can interfere with the precise recognition of the intended object. Shadows further complicate the problem by changing the appearance of objects, making it difficult to maintain consistent tracking. Several researchers have utilized different methods to remove the above interferences from the video to minimize the false detections. In researches [16] and [8], morphological operations were used to remove the noise of the foreground frames. Tian [17] utilized texture information and normalized cross-correlation to minimize false positives and shadows. Radial Reach Filter and Gaussian smoothing were employed in another study to reduce false detected foregrounds caused by illumination changes and small holes. Phong Shading Model was utilized in some cases to handle rapid light changes. Different techniques such as 2D-convolution, color normalization, structure noise reduction, Gaussian blur, Gaussian filtering, color correction, and gamma correction have been applied for image enhancement, noise reduction, and handling multimodal backgrounds. Object size considerations, fuzzy color histograms, contour-based approaches, and morphological closing operations were also implemented by various researchers to address specific challenges such as noisy regions, color similarity, ghost removal, and hole filling.

2.1.2.2 Localization of a Static Object Based on Tracking Approaches

Object tracking stands as a crucial and demanding task in the realm of computer vision. It involves generating an object's trajectory over time by tracing its position across consecutive frames. In research, Kalman filtering plays a prominent role in object tracking due to its well-established reputation and widespread use. This method is favored for its ease of implementation and real-time operational capabilities. Kalman filtering operates under the assumption that the tracked object follows a linear dynamic system with Gaussian noise. For cases involving non-linear systems, researchers have proposed alternative methods based on the Kalman filter, such as the Extended Kalman Filter and Unscented Kalman Filter. Moreover, the use of Kalman filtering is evident in various research applications, such as employing a dynamics model with a second-order derivative for specific scenarios. However, when dealing with non-linear and non-Gaussian signals, particle filters, are considered to be more effective than Kalman filtering. Overall, Kalman filtering serves as a versatile and adaptable tool in the realm of research, particularly in scenarios where object movements can be approximated as linear dynamic systems.

2.1.2.3 Localization of a Static Object Based on Non-Tracking Approaches

The tracking information was unable to manage occlusion and low-contrast situations in highly complex video sequences. Due to these limitations, many researchers have shifted to non-tracking-based approaches in intelligent video surveillance. The key focus lies in choosing appropriate features to effectively identify abandoned or removed objects. The objective of feature extraction is identifying the most significant information within the recorded video to make distinction between moving and stationary objects.

2.1.2.4 Classify Static Object into Human & Non-human

The classification method is crucial for reducing false alarms in the detection of abandoned, removed, or stolen objects in Intelligent Video Surveillance. Once static objects are identified in a video, the classification approach needs to be resilient in distinguishing between human and non-human stationary objects. This discrimination is essential for conducting thorough analysis and determining whether the object in question is abandoned, removed, or stolen. For example, a still human and non-human object in public place can be treated as abandoned

object if there is no knowledge of the object features. Classification methods for objects need to demonstrate high sensitivity in order to effectively differentiate between static humans and abandoned objects, as well as between faces and objects based on skin color, and so forth. If the employed classifier is unsuccessful in distinguishing between static humans and non-human objects, the false detection rate increases quickly. In general, classification methods can be categorized into three categories which are based on shape, motion, and feature. Researchers have made significant attempts to extract and integrate features with different classifiers such as SVM, Multi-SVM, k-Nearest Neighbor, Cascade classifier, Neural Network, and HAR to accurately categorize stationary human and non-human objects, aiming to achieve zero false positives. However, their efforts have only managed to reduce false positives to a limited extent.

2.1.2.5 Object Analysis to Recognize Abandoned, Removed or Stolen Objects

Analyzing and making decisions about objects is a crucial and challenging task for an intelligent video surveillance system to accurately identify abandoned, removed, or stolen items. The system must generate real-time alarms to alert security personnel, preventing potential ecological and economic losses and addressing theft cases in public spaces worldwide. To enhance the true positive rate and reduce the false positive rate, numerous prominent researchers have employed diverse analysis approaches. These include Finite State Machines (FSM), the fusion of high gradient, low gradient, and color histogram features, utilization of multiple spatial-temporal and contextual cues for event detection, Bayesian inference framework for event analysis, high-level reasoning to infer abandoned luggage existence, temporal analysis, probabilistic event models, spatial-temporal rules for tracking luggage owners and identifying abandoned items, and region-level analysis.

2.1.3 Background subtraction

Background subtraction is a technique commonly used in computer vision for various applications, including abandoned or removed object detection. The basic idea behind background subtraction is to separate the foreground objects (moving or newly introduced objects) from the background (stationary or relatively constant elements) in a given video sequence or image. In the review of [18], this idea has been articulated as follows:

“The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called the “background image”, or “background model”. As a basic, the background image must be a representation of the scene with no moving objects and must be kept regularly updated.”

2.1.3.1 Background Subtraction Approaches

Many background subtraction methods have been proposed in the past three decades. Each of which has its own strengths and weaknesses in terms of performance and computation requirements. A robust background subtraction algorithm should be able to handle lighting changes, repetitive motions and long term scene changes [19].

A. Running Gaussian Average

The Running Gaussian Average is a simple method offering acceptable accuracy and a high frame rate while having low memory requirements. The method was initially proposed by Wren [20] and later improved by Koller. The method models the background at each pixel location based on a Gaussian probability density function, initializing the background distribution with a running average of the first frame's pixel values. This running average is continuously updated as new frames are processed, providing a computationally efficient way to adapt to changing backgrounds. Foreground-background classification is determined by

comparing the pixel's value to the running average and standard deviation, with a threshold parameter 'k.' A selective background update proposed by Koller et al. addresses issues related to unduly updating the model in the presence of foreground values. The method exhibits advantages such as speed and low memory requirements, although it may not handle redundant movement of objects effectively.

Some other background subtraction methods that can be found in the literature are

B. Temporal Median Filter
C. Mixture of Gaussian
D. Kernel Density Estimation
E. Co-occurrence of Image Variations
F. Eigen-background

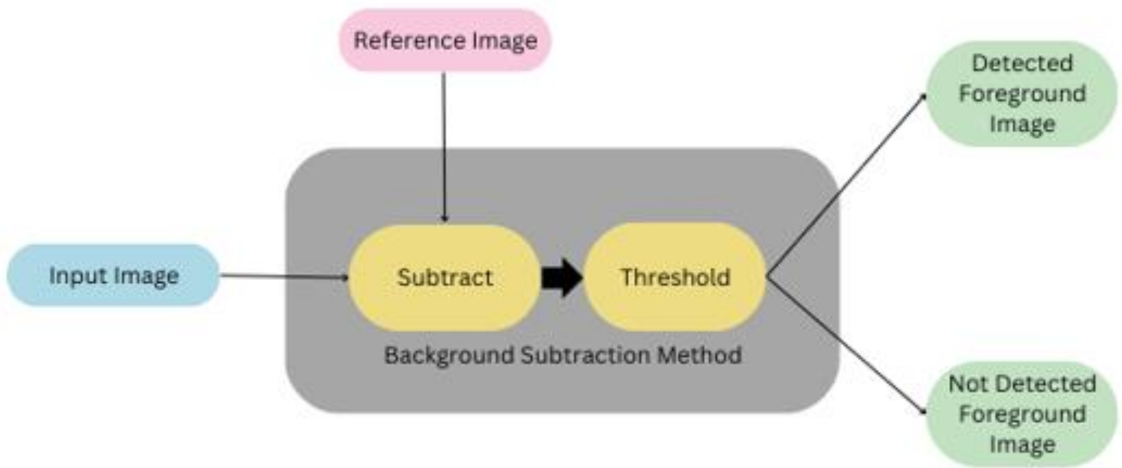

Figure 2.3 Overview of Background Subtraction Process

2.1.3.2 Challenges of Background Subtraction

Background subtraction methods encounter diverse challenges due to the characteristics of video surveillance. In addition to the typical obstacles, numerous background subtraction challenges have been examined in the existing literature. Some of these challenges mentioned in [21] are,

Gradual or Sudden Illumination Changes: Changes in lighting conditions can occur slowly over time (gradual changes) or suddenly (rapid changes). These alterations can be caused by factors like sunrise/sunset, weather conditions, or artificial light adjustments. Background subtraction methods need to be adaptive to varying levels of brightness and darkness. Failure to handle gradual changes may lead to false detections or missed objects. Additionally, sudden changes can introduce temporary anomalies in the background, making it challenging for traditional methods to distinguish between foreground and background during these transitions.

Dynamic Background: Certain parts of a video scene may exhibit movement but should still be considered as part of the background. This movement can be irregular, such as swaying tree branches, or periodic, like waves in a body of water. Traditional background subtraction methods may struggle to differentiate between static and dynamic elements in the background. Methods need to discern whether a moving object is part of the foreground or an element that should be considered as part of the background model. Handling irregular or periodic

movements without misclassifying them as foreground objects is crucial.

Video Noise: Video signals are often impure by various types of noise, including sensor noise or other distortions introduced during acquisition, transmission, or storage. Noise in the video signal can lead to false positives or negatives during background subtraction. Background subtraction methods must be robust enough to distinguish between true changes in the scene and those caused by noise. Techniques such as filtering or preprocessing may be required to enhance the signal-to-noise ratio and improve the accuracy of background modeling.

Camouflage: Camouflage refers to situations where certain objects in the scene poorly differ from the background, making it challenging for background subtraction methods to correctly classify them as part of the background or foreground. In surveillance applications, this can have serious consequences, as the system might fail to detect important objects or may produce false alarms. Adaptive background subtraction methods need to account for variations in appearance and ensure accurate differentiation between background and foreground, even when objects attempt to blend in intentionally or unintentionally.

Shadows: Shadows cast by foreground objects can create complications during background subtraction. These shadows may overlap with the foreground regions, making it difficult to separate and accurate classification of objects in subsequent processing steps. Shadows can introduce challenges in distinguishing between the actual foreground objects and their shadow representations. Ignoring or incorrectly handling shadows may lead to false detections or missed objects. Effective background subtraction methods need to differentiate between the shadows and the true foreground, possibly by incorporating shadow suppression techniques or considering the temporal aspects of shadow movement. Proper handling of shadows is crucial for improving the accuracy of subsequent processing steps, such as object tracking or recognition, in video surveillance system.

CHAPTER 3
METHODOLOGY

3.1 Proposed System Design

Figure 3.1 provides a comprehensive illustration of the system architecture, showcasing the procedural flow of the entire system. This architecture is composed of five primary components: (i). video acquisition, (ii). frame preprocessing, (iii). background subtraction, (iv). feature extraction, and (v). calculate the last seen frame time of the removed object. The proposed algorithm operates in a manner analogous to well-known Binary search algorithm. Initially, it begins by searching for the object in the middle frame of the video sequence. If the object is not detected in the middle frame, the algorithm divides the search interval in half and continues the search process in the appropriate section. This process is repeated iteratively, continuously narrowing down the search area until the specific conditions are met, ensuring the precise identification of the object's last seen.

Furthermore, the initial frame is designated as the background frame and is essential for detecting changes or differences in subsequent frames. The background subtraction technique is employed to highlight these changes, making it a crucial aspect of the overall analysis process.

3.1.1 Video Acquisition

The process begins by acquiring the video file that will serve as the basis for the analysis. This video can come from a variety of sources, such as recorded footage from surveillance cameras, pre-existing video files, or any other medium that captures the scene relevant to the analysis. The system is designed to support multiple video formats, including common ones like mp4, mov, and bmp, among others. These formats are selected for their widespread use and compatibility with different recording devices and media types.

At this stage of the research project, the system is configured to work with only a single video input at a time. This means that, for any given analysis, only one video can be processed, whether it's captured from a single surveillance camera or imported as a standalone video file. The capability of analyzing multiple video streams or files simultaneously is not yet implemented, but future iterations may consider expanding this feature to support a broader range of inputs.

3.1.2 Frame Preprocessing

The frame preprocessing stage is a crucial step in preparing video frames for subsequent analysis, employing various techniques to optimize the frames and make them more suitable for detection and tracking tasks. This stage involves transforming the raw video data into a format that facilitates more efficient and accurate analysis.

A. Gray Scaling

The first step in the frame preprocessing stage is the conversion of each video frame into grayscale. In its original form, a video frame typically contains color information, represented in RGB (Red, Green, Blue) or other color spaces, which significantly increases the complexity of the data. By converting the frame to grayscale, the process simplifies the representation of each pixel by removing the color data and reducing it to a single intensity value. This transformation not only reduces the amount of data to be processed but also retains the essential visual features necessary for object detection and tracking. Grayscale conversion is particularly useful because, in many cases, color is not critical for detecting changes in a scene. The focus is on the intensity variations, which can be analyzed more efficiently without the added complexity of color information.

B. Histogram Equalization

Following the grayscale conversion, histogram equalization is applied to enhance the contrast of the grayscale image. This technique redistributes the intensity values of the pixels in a way that stretches out the most frequently occurring intensity levels, thus improving the overall visibility of details in the image. This step is essential for highlighting the crucial details that may otherwise be missed, allowing the algorithm to detect changes in the scene more accurately.

C. Gaussian Blur

The final step in the frame preprocessing stage is the application of Gaussian Blur, a smoothing technique used to reduce noise and unwanted details in the image. In video frames, minor variations in pixel values often result from noise, which can obscure the detection of relevant objects or events. Gaussian Blur helps to minimize this noise by averaging the pixel values around each point, effectively smoothing out rapid intensity changes. In the context of this research project, where the system aims to track objects and detect their removal, Gaussian Blur is especially useful. By reducing unnecessary details and background noise, the algorithm can focus more clearly on significant changes in the frame, improving the accuracy of object detection and tracking throughout the video sequence.

Together, these preprocessing techniques grayscale conversion, histogram equalization, and Gaussian Blur ensure that the video frames are optimally prepared for the next stages of analysis. They not only simplify the data but also enhance the critical features necessary for effective tracking and detection, making the overall analysis process more streamlined and robust.

3.1.3 Background Subtraction in the middle frame of the video

This section focuses on the critical process of background subtraction, specifically applied to the middle frame of the video sequence. Background subtraction is a widely used technique in video analysis to isolate and detect moving or changed objects within a scene by distinguishing them from the static background. In this research, the first frame of the video is chosen as the reference point for background subtraction. The algorithm compares this reference frame with subsequent frames extracted as the middle frames in each video halves to identify any changes or differences, such as the appearance or removal of objects in the scene. The process involves subtracting the pixel values of the background (first frame) from each subsequent frame, effectively identifying the areas where movement or changes have occurred. This step is essential for determining whether the feature extraction process should be executed next to verify the object presence. Background subtraction in the middle frame serves as the foundation for accurate object tracking and change detection throughout the video analysis.

3.1.4 Feature Extraction in the middle frame of the video

This section addresses the process of feature extraction, applied to the middle frame of the video sequence followed by background subtraction. Feature extraction is a crucial step in video analysis, as it identifies and extracts distinctive patterns or key points from the frame that can be used for object detection, matching, and tracking throughout the video. In this research, the ORB (Oriented FAST and Rotated BRIEF) descriptor is utilized for feature extraction. ORB is an efficient and fast feature extraction technique that combines the speed of the FAST (Features from Accelerated Segment Test) key point detector and the robust binary descriptor of BRIEF (Binary Robust Independent Elementary Features). By using

ORB, the system identifies key features of the removed object in the particular middle frame, such as edges, corners, and other significant patterns, which are invariant to rotation and scale changes. These features act as unique identifiers for objects in the frame, allowing for accurate tracking and recognition of objects across the video. The ORB descriptor's efficiency also makes the process computationally faster, enabling effective analysis while maintaining high accuracy in detecting key objects and their movements.

3.1.5 Calculate the Last Seen of the Removed Object

Once a static object is identified as having been removed from the scene, the system takes the analysis a step further to determine the precise moment when the object disappeared. This important step involves calculating the "last seen frame time," which is the exact point in the video sequence when the object was last observed before being removed. To achieve this, the system relies on two crucial pieces of information: (i) the identified frame number and (ii) the FPS (Frames Per Second) value. The identified frame number refers to the specific frame in the video where the object was last seen as stationary before its removal. This frame serves as the anchor point for determining the disappearance. The FPS value represents the frame rate of the video, or the number of frames displayed per second, which directly influences how time is measured within the video sequence.

By combining these two elements, the system can accurately calculate the estimated time when the object was last present in the scene. Specifically, it takes the identified frame number and divides it by the FPS value, yielding the exact timestamp corresponding to the object's disappearance. This approach provides a highly precise calculation of the moment the object vanished, allowing for further analysis or reporting based on the event's timing. In this way, the system not only detects the removal of objects but also offers a time-based context for when the event occurred, making it highly useful for scenarios such as security monitoring or forensic analysis.

3.1.6 Summary

Chapter 3 outlines the proposed system design and methodology for identifying the last seen frame time of a removed static object in a video feed. The system is composed of five key components: video acquisition, frame preprocessing, background subtraction, feature extraction, and calculating the last seen frame time. The algorithm uses a binary search-like approach to locate the last instance of the object by progressively halving the video frames under consideration. *Video Acquisition:* This step involves gathering video footage from sources like surveillance cameras. The system currently processes one video at a time, supporting formats like mp4, mov, and bmp. *Frame Preprocessing:* The preprocessing phase simplifies the video data for more efficient analysis. This is done through grayscale conversion to reduce color complexity, histogram equalization to enhance image contrast, and Gaussian blur to minimize noise. *Background Subtraction:* Background subtraction is applied to the middle frame of video section to detect object removal or appearance. This isolates the dynamic elements in the video, such as moving or removed objects. *Feature Extraction:* After background subtraction, ORB (Oriented FAST and Rotated BRIEF) is used to extract key features from the particular middle frame. ORB identifies important patterns like edges and corners, which help in detecting the object in the frame. *Calculating the Last Seen Frame Time:* Finally, the system calculates the exact time when the object was last visible using the identified frame number and the video's frames per second (FPS). This allows for precise detection of when the object was removed from the scene, which is crucial for applications like security or forensic analysis.

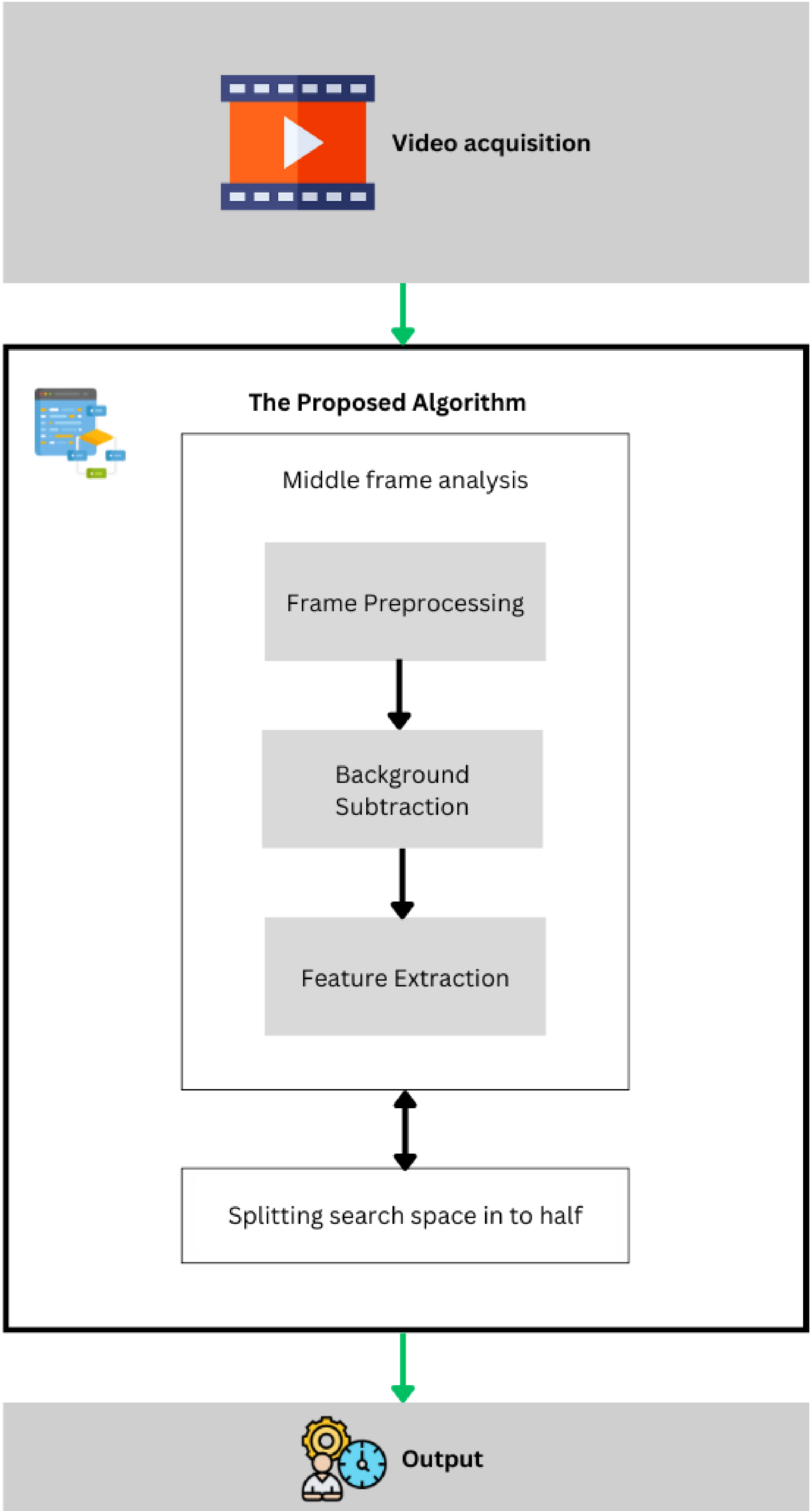

Figure 3.1 Overview of System Architecture

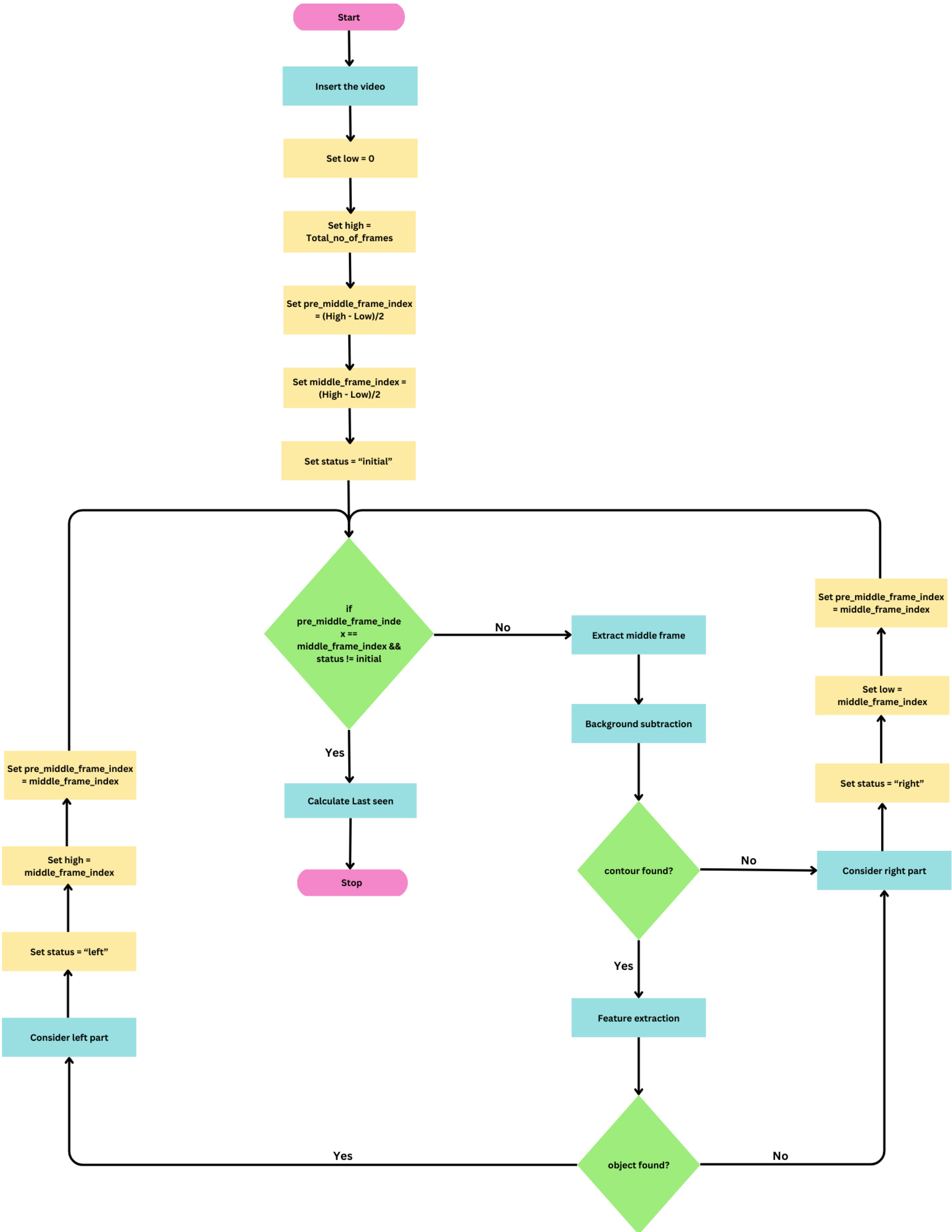

Figure 3.2 Flow Chart

CHAPTER 4
EVALUATION AND RESULTS

4.1 Experimental Setup

In this section, detailed experimental results are presented, obtained from the proposed method. The tests were conducted on a high-performance machine equipped with an Intel i10 processor running at a CPU frequency of 2.6 GHz, supported by 16GB of RAM to ensure efficient processing. The method was implemented using OpenCV, which served as the core computer vision library due to its extensive features and optimization for image and video processing tasks.

Python was chosen as the programming language because of its simplicity, versatility, and compatibility with powerful libraries. Several additional libraries played critical roles in the development of the prototype. For instance, NumPy was utilized for handling numerical operations and efficient array manipulations, which are essential in image processing. Pillow, an image processing library, helped with image manipulation tasks such as format conversions, while scikit-image provided additional image processing utilities to complement OpenCV.

Furthermore, to create an interactive and user-friendly interface, the Graphical User Interface (GUI) was developed using the Custom Tkinter library. Custom Tkinter offers enhanced features over the standard Tkinter library, allowing for better control over the GUI's appearance and functionality. This comprehensive setup, combining powerful hardware and a robust software stack, enabled the successful development and testing of the proposed method, providing a seamless workflow from data input to analysis and output visualization.

Computer Vision library	OpenCV
Other libraries	NumPy, Pillow, Scikit-image
GUI library	Custom Tkinter
Programming language	Python

Table 4.1 Experimental setup

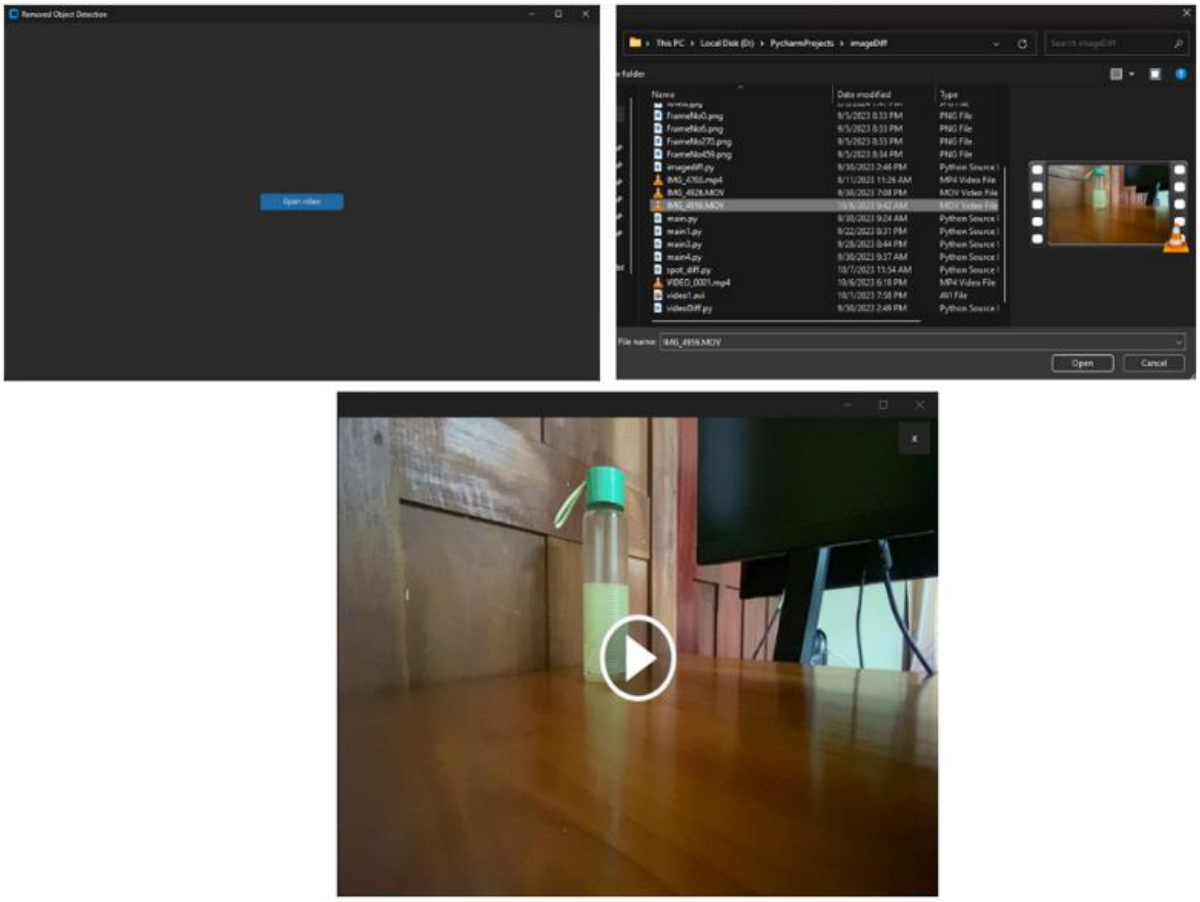

Figure 4.1 Video acquisition

4.2 Evaluation Methodology

The project evaluation follows and experimental approach. This includes testing the system using various video datasets with different characteristics, including variations in objects, video duration, and lighting conditions. These datasets have been specifically chosen to represent a wide range of scenarios that may challenge the system, ensuring a thorough assessment of its capabilities. The evaluation aims to assess several critical aspects of the system's performance, detailed as follows:

- **Accuracy:** The primary focus is on the system's ability to precisely detect and report the last seen frame time of a removed object. This involves analyzing how effectively the system can maintain accuracy under diverse video characteristics, such as varying object sizes, sudden movements, or shifts in perspective.

- **Efficiency:** A crucial aspect of the evaluation process is to measure the computational efficiency of the system, which includes the time taken to process each video and the system's overall resource consumption.

- **Robustness:** The system's resilience in difficult environments is another key factor being tested. This includes its ability to perform consistently under less-than-ideal conditions, such as videos with significant background noise, rapidly changing lighting, and objects that may be partially or fully occluded. By testing in these challenging scenarios, the evaluation will demonstrate the system's capacity to maintain reliable performance despite external disruptions.

4.3 Performance Metrices

To thoroughly evaluate the system, a variety of performance metrics were employed to ensure a comprehensive assessment of its capabilities. Each metric provides valuable insights into different aspects of the system's performance:

- **Detection Accuracy:** This metric refers to the percentage of correct detections made by the system, specifically focusing on its ability to accurately identify the last frame captured before the object was removed from the video. A high detection accuracy is crucial, as it reflects the system's reliability in recognizing when an object has disappeared and ensures that users can trust the results generated by the system.

- **Processing Time:** This metric measures the duration taken by the system to process a given video and compute the last seen frame time. The significance of processing time cannot be overstated, especially for applications that require real-time or near-real-time performance. Efficient processing time ensures that the system can handle video streams swiftly, allowing for timely detection and analysis of object removals.

- **False Positives/Negatives:** This performance metric highlights instances where the system misidentifies an event related to object removal. False positives occur when the system incorrectly identifies an object as being removed when it has not, while false negatives refer to situations where the system fails to detect an actual object removal. Monitoring these instances is essential, as they directly affect the reliability and accuracy of the system's performance. Minimizing both false positives and false negatives is critical for enhancing the overall effectiveness of the detection process.

4.4 Results and Analysis

During the experiments, several recorded videos are individually analyzed. Some of the information extracted from a few of these videos are presented in the following tables.

4.4.1 Analyzing short duration videos without moving objects

Video	Duration (seconds)	Object	Processing Time (seconds)	Effect	Disappeared Time (seconds)	Estimated Time (seconds)
short_bottle.mp4	19s	a bottle	2.26s	• Background with less objects	12s	12s
short_cup.mp4	44s	a cup	2.7s	• Background with less objects	30s	30s
short_penholder.mp4	60s	a pen holder	2.57s	• Background with less objects • Noisy	35s	35s
short_bluePenHolder.mp4	29s	a blue pen holder	2.65s	• Background with more objects	19s	14s
short_shoes.mp4	25s	a pair of shoes	2.59s	• Background with less objects • Shadows	19s	18s

Table 4.2 Experimental Results of short duration videos *without moving objects*

The test results for short-duration videos with no moving objects reveal that the system generally performs well in estimating the disappearance time of static objects. The system accurately identified the disappearance time in four out of the five test videos, showing that it is reliable under controlled conditions, where the background is either simple or contains minimal objects. The processing times are efficient, averaging around 2.54 seconds across all videos, which demonstrates the system's capacity to handle short-duration videos effectively.

For videos with minimal background complexity, such as short_bottle.mp4, short_cup.mp4, and short_penholder.mp4, the estimated disappearance times perfectly matched the actual disappearance times, showing a high level of precision. Even in slightly noisy environments, such as with the short_penholder.mp4 video, the system maintained its accuracy, indicating robustness to mild noise interference.

However, in the case of short_bluePenHolder.mp4, where the background had more objects, the system overestimated the disappearance time by 5 seconds. This suggests that more complex backgrounds can interfere with the accuracy of feature extraction and background subtraction. Similarly, for short_shoes.mp4, where shadows were present, there was a slight 1-second discrepancy between the actual and estimated disappearance times, which could be attributed to difficulties in distinguishing between object shadows and the object itself.

In summary, the system performs accurately and efficiently under simple background conditions, but some improvements are needed to handle more complex scenes with numerous objects or lighting variations such as shadows.

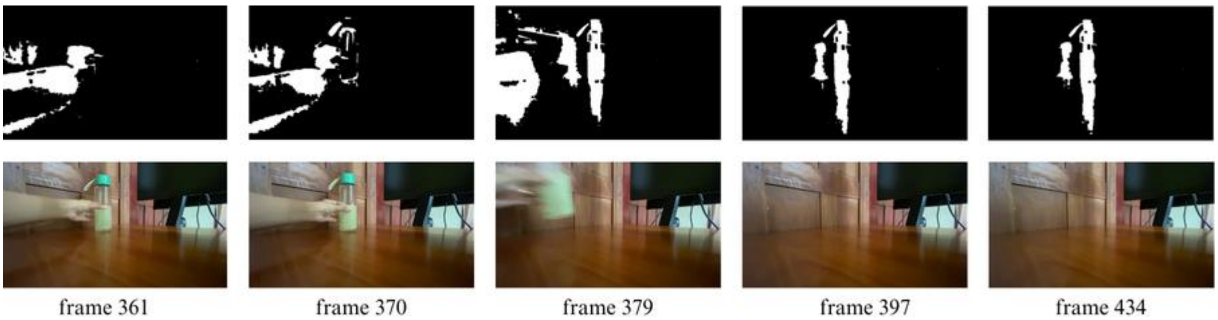

Figure 4.2 Analyzing a short duration video without moving objects

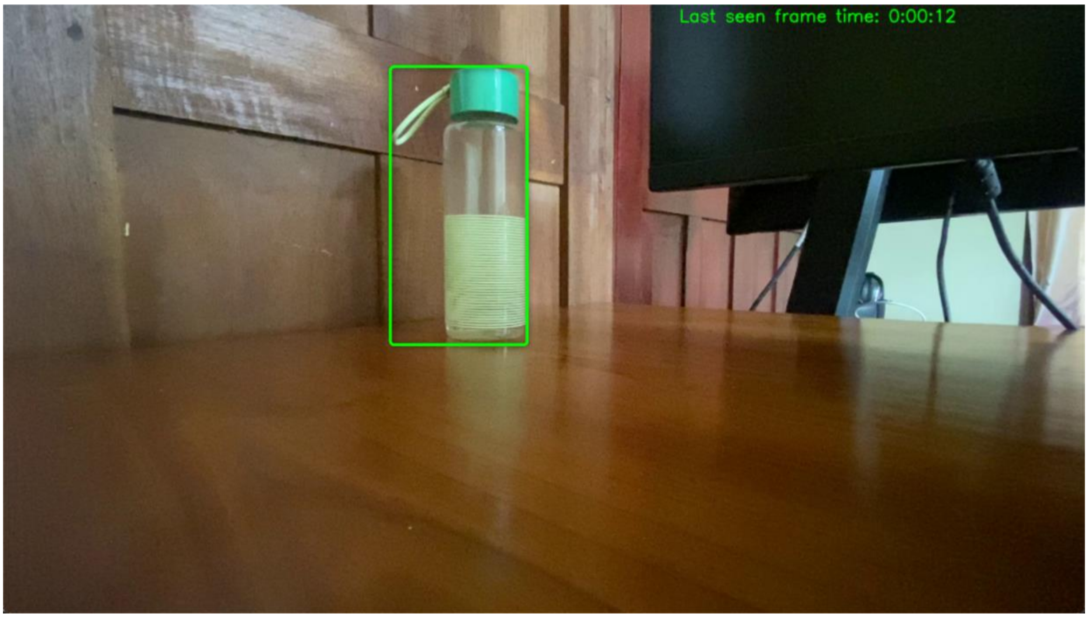

Figure 4.3 Resulting Frame

4.4.2 Analyzing short duration videos with moving objects

Video	Duration (seconds)	Object	Processing Time (seconds)	Effect	Disappeared Time (seconds)	Estimated Time (seconds)
short_m_bottle.mp4	90s	a bottle	3.1s	• Background with less objects	59s	59s
short_m_cup.mp4	31s	a cup	3s	• Background with less objects • Noisy	28s	10s
short_m_penholder.mp4	96s	a pen holder	2.92s	• Background with less objects • Noisy	42s	42s
short_m_bluePenHolder.mp4	90s	a blue pen holder	3s	• Background with more objects • Noisy	78s	21s

short_m_shoes.mp4	43s	a pair of shoes	2.61s	• Background with less objects	29s	29s

Table 4.3 Experimental Results of short duration videos *with moving objects*

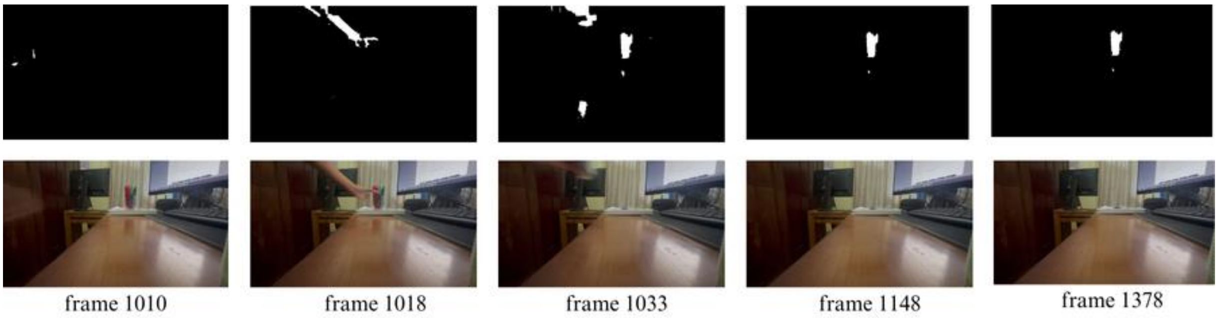

Figure 4.4 Analyzing a short duration video with moving objects

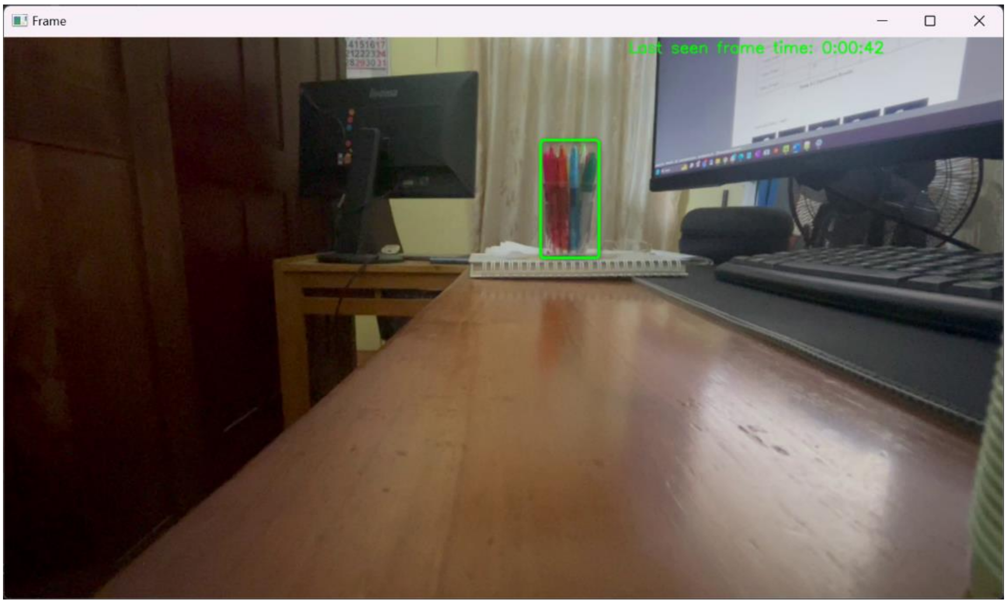

Figure 4.5 Resulting Frame

The test results for short-duration videos with moving objects show a mix of accurate and inaccurate estimations of object disappearance times, depending on background complexity, noise levels, and motion characteristics. The system generally handles videos with less background complexity and minimal noise well, but struggles with noisier and more complex backgrounds.

In cases such as short_m_bottle.mp4, short_m_penholder.mp4, and short_m_shoes.mp4, the system accurately detected the disappearance time of the object, matching the actual time perfectly. These videos had simple backgrounds with fewer objects, suggesting that the system performs reliably under such conditions, even when there is movement involved. The processing times were also relatively quick, averaging around 2.87 seconds, showcasing the system's efficiency for these scenarios.

However, videos with noisy or complex backgrounds posed challenges. For example, in short_m_cup.mp4, where the background was noisy, the estimated disappearance time was 18 seconds earlier than the actual time. This significant discrepancy suggests that noise can interfere with the system’s ability to accurately detect object removal, likely due to the background subtraction and feature extraction being affected by the noise. Similarly, in short_m_bluePenHolder.mp4, the system significantly underestimated the disappearance time by 57 seconds in a video with more background objects and noise. This indicates that when both complexity and noise are present, the system struggles to maintain accuracy.

In summary, the system performs well with moving objects in simple backgrounds but exhibits substantial errors in estimating disappearance times in noisier and more complex environments. Improvements in noise handling and more advanced feature extraction techniques could enhance the system's robustness and accuracy in challenging scenarios

4.4.3 Analyzing long duration videos without moving objects

Video	Duration (h-m-s)	Object	Processing Time (seconds)	Effect	Disappeared Time (h-m-s)	Estimated Time (h-m-s)
long_bottle.mp4	1h28m	a bottle	4.88s	• Background with less objects	1h21m37s	1h21m37s
long_cup.mp4	3h45m	a cup	5.21s	• Background with less objects	55m28s	55m28s
long_penholder.mp4	6h58m	a pen holder	6.43s	• Background with more objects	2h23m45s	35m47s

				• Noisy		
long_mouse.mp4	12h04m	a mouse	9.67s	• Background with less objects	8h59m53s	8h59m53s
long_remote.mp4	24h03m	a remote	13.4s	• Background with less objects • Shadows	14h25m9s	7h57m25s

Table 4.4 Experimental Results of long duration videos *without moving objects*

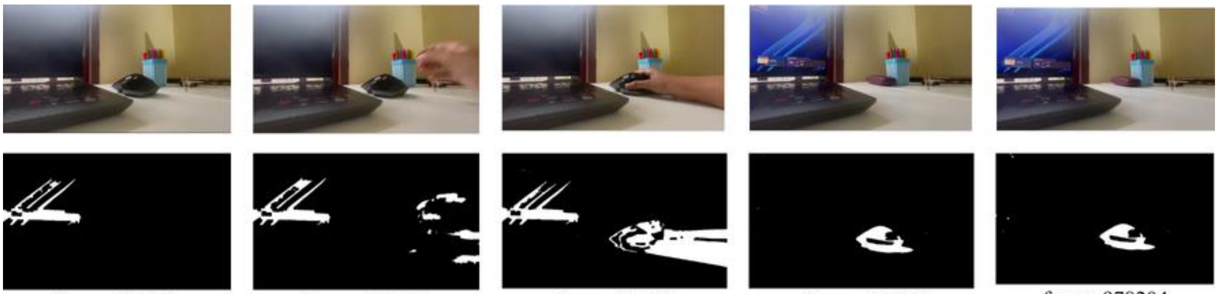

Figure 4.6 Analyzing a long duration video without moving objects

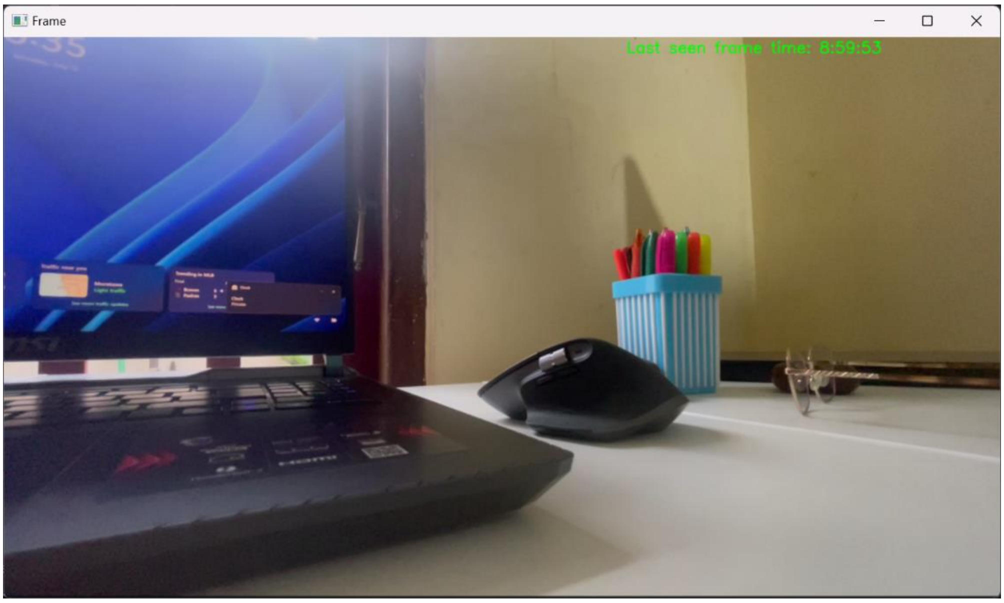

Figure 4.7 Resulting Frame

The evaluation of long-duration videos without moving objects shows mixed performance by the system. While it demonstrates high accuracy for certain videos, there are significant discrepancies in others, particularly in noisy and complex environments.

For videos with simpler backgrounds, such as long_bottle.mp4, long_cup.mp4, and long_mouse.mp4, the system accurately detected the object disappearance times, with no difference between the actual and estimated disappearance times. The processing times for these long videos were impressively low, ranging from 4.88 to 9.67 seconds, indicating that the system can efficiently handle videos of extended durations when the background is less complex.

However, the system struggled with more complex backgrounds and noise. In long_penholder.mp4, where the background contained more objects and noise, there was a substantial error in the estimated disappearance time, with the system estimating the time as 35m47s, compared to the actual disappearance at 2h23m45s. This large discrepancy highlights that the system is sensitive to noise and may not perform well in cluttered environments over extended durations. Similarly, in long_remote.mp4, which contained shadows in the background, the system estimated the disappearance at 7h57m25s, nearly 6.5 hours earlier than the actual time of 14h25m9s. Shadows may have confused the background subtraction algorithm, leading to an inaccurate estimation.

Overall, the system performs exceptionally well in long-duration videos with simple backgrounds, maintaining both high accuracy and low processing time. However, in environments with noise, complex backgrounds, or lighting variations (such as shadows), the system's accuracy decreases significantly. Improvements in handling noisy environments and better techniques for dealing with shadows and background complexity would be necessary to enhance the system’s reliability in more challenging scenarios.

4.4.4 Analyzing long duration videos with moving objects

Video	Duration (h-m-s)	Object	Processing Time (seconds)	Effect	Disappeared Time (h-m-s)	Estimated Time (h-m-s)
long_m_bottle.mp4	1h15m	bottle	4.69s	• Background with less	25m34s	25m34s

				objects		
long_m_cup.mp4	2h45m	a cup	4.91s	• Background with less objects • Noisy	2h05m45s	2h05m45s
long_m_penholder.mp4	5h15m	a pen holder	5.33s	• Background with less objects • Noisy	1h37m44s	1h37m44s
long_m_bluePenHolder.mp4	8h23m	a blue pen holder	7.89s	• Background with more objects • Noisy	4h53m25s	37m55s
long_m_shoes.mp4	24h	a pair of shoes	12.59s	• Background with less objects	2h23m45s	2h23m45s

Table 4.5 Experimental Results of long duration videos *with moving objects*

The test results for long-duration videos with moving objects show promising performance in estimating the disappearance time when the background is less complex and relatively noise-free. However, the system's accuracy drops significantly in videos with noisy or more complex backgrounds.

For videos with simpler backgrounds, such as long_m_bottle.mp4, long_m_cup.mp4, long_m_penholder.mp4, and long_m_remote.mp4, the system performed with excellent accuracy. The estimated disappearance times were exactly aligned with the actual disappearance times, demonstrating the system's strong ability to handle long-duration videos with moving objects. The processing times remained efficient, with an average time of around 5.50 seconds, which is commendable given the long duration of these videos.

However, the system showed considerable inaccuracies when dealing with complex and noisy backgrounds, particularly in long_m_bluePenHolder.mp4. In this case, the system's estimated disappearance time of 37m55s was far earlier than the actual disappearance at 4h53m25s—a discrepancy of over four hours. This result suggests that in more cluttered environments or when faced with noise, the system struggles to accurately track the disappearance of moving objects. This performance issue likely arises due to challenges in background subtraction and feature extraction, particularly when dynamic scenes with noise are involved.

In summary, the system is well-suited for long-duration videos with moving objects as long as the background is simple and less noisy. However, its reliability diminishes in videos with complex or noisy backgrounds, indicating that further enhancements are required to improve accuracy in such challenging environments. Enhancing noise tolerance and improving object detection in dynamic scenes would be key areas to address for more consistent performance across different scenarios.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research project presents a novel approach for estimating the last seen frame time of a static removed object in recorded video footage utilizing background subtraction and feature extraction. The proposed methodology offers a significant advancement in video analytics by automating the identification of potentially removed objects, streamlining investigative processes. By isolating static objects and employing object identification through bounding box tracking, the system pinpoints the specific video segment where the object might have been removed. This methodology leads to significant savings in both time and resources for investigators, allowing them to concentrate on more specific analyses guided by the estimated timestamps. While this research project has made significant progress in automating the estimation of last seen frame time for static removed objects, there is room to improve. Here the limitations, potential improvement and future work are discussed.

5.2 Limitations

From the analysis of the system's performance across various video scenarios, several limitations can be identified.

- The system struggles with videos that feature complex backgrounds or significant noise. In scenarios where multiple objects are present in the background, the accuracy of the estimated disappearance times drops significantly.
- Noise also severely impacts the system's performance, leading to substantial discrepancies between the actual and estimated times.
- Shadows in videos, especially in long-duration ones, introduce confusion for the background subtraction algorithm. This leads to considerable errors in the estimated disappearance times, with some cases showing hours-long discrepancies.
- When both movement and noise are present, the system’s accuracy decreases further, as seen in both short and long-duration videos with moving objects.
- The combination of motion and background complexity presents a challenge for the current feature extraction and background subtraction techniques.

5.3 Potential Improvements

The system demonstrates strong performance in estimating the disappearance times of static and moving objects in short and long-duration videos when the background is simple and less noisy. In such scenarios, the estimated times are highly accurate, with minimal processing delays, indicating that the system is efficient and reliable in controlled environments. However, the system's accuracy significantly diminishes in the presence of complex backgrounds, noise, or lighting variations such as shadows. Short-duration videos with noisy backgrounds or long-duration videos with shadows and multiple background objects show considerable discrepancies between the actual and estimated disappearance times. These limitations highlight the system's sensitivity to environmental changes, background complexity, and noise interference, particularly in more dynamic scenes with moving objects.

To enhance the performance and address the limitations, several improvements could be considered.

- Implementing more sophisticated noise filtering mechanisms could improve the system's ability to handle complex backgrounds.
- Integrating more advanced shadow detection algorithms could significantly reduce the discrepancies caused by shadow interference. Algorithms like Gaussian Mixture Models (GMM) or machine learning techniques for shadow recognition could be explored to mitigate this issue.
- The system could benefit from more robust feature extraction techniques that are better suited for dynamic environments with moving objects. Employing techniques like optical flow, object tracking, or deep learning-based object detection models (e.g., YOLO, Mask R-CNN) could improve the detection and tracking of moving objects in complex scenarios.
- For videos with extended durations, introducing incremental background updates or real-time processing approaches might improve accuracy. Additionally, temporal consistency checks could be implemented to ensure that the background model adapts over time, even in complex environments.

5.4 Future Work

To address the limitations observed in the analysis, several areas of improvement and future enhancements can be explored to increase the system's robustness and accuracy across various video scenarios. One of the main limitations identified was the system's decreased performance when handling complex backgrounds, particularly when multiple objects are present or the environment is noisy. The current feature extraction algorithm, ORB (Oriented FAST and Rotated BRIEF), struggles in such conditions, leading to inaccuracies in estimating disappearance times. Future work should focus on enhancing the feature extraction process by integrating more advanced algorithms. Shadows were found to interfere with the system's ability to accurately estimate disappearance times, particularly in long-duration videos. Future work could focus on implementing shadow detection and removal algorithms to differentiate between objects and their shadows. The presence of noise, especially in videos with complex backgrounds, posed significant challenges for the system. This suggests a need for more advanced noise filtering techniques to be incorporated into the system's background subtraction and feature extraction pipeline.

APPENDICES

REFERENCES

[1] A. M. Qasim, N. Abbas, A. Ali, and B. A. A.-R. Al-Ghamdi, "Abandoned Object Detection and Classification Using Deep Embedded Vision," *IEEE Access*, vol. 12, pp. 35539–35551, 2024, doi: 10.1109/ACCESS.2024.3369233.

[2] H. Park, S. Park, and Y. Joo, "Detection of Abandoned and Stolen Objects Based on Dual Background Model and Mask R-CNN," *IEEE Access*, vol. 8, pp. 80010–80019, 2020, doi: 10.1109/ACCESS.2020.2990618.

[3] L. H. Jadhav and B. F. Momin, "Detection and identification of unattended/removed objects in video surveillance," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India: IEEE, May 2016, pp. 1770–1773. doi: 10.1109/RTEICT.2016.7808138.

[4] Y. Nam, "Real-time abandoned and stolen object detection based on spatio-temporal features in crowded scenes," *Multimed Tools Appl*, vol. 75, no. 12, pp. 7003–7028, Jun. 2016, doi: 10.1007/s11042-015-2625-2.

[5] W. Rakumthong, N. Phetcharaladakun, W. Wealveerakup, and N. Kamnoonwatana, "Unattended and Stolen Object Detection Based On Relocating of Existing Object".

[6] Y. J. Chai, S. W. Khor, and Y. H. Tay, "Object occlusion and object removal detection," presented at the Fifth International Conference on Digital Image Processing, Y. Wang and X. Yi, Eds., Beijing, China, Jul. 2013, p. 88782F. doi: 10.1117/12.2032482.

[7] A. K. Singh and A. Agrawal, "An interactive framework for abandoned and removed object detection in video," in *2013 Annual IEEE India Conference (INDICON)*, Mumbai, India: IEEE, Dec. 2013, pp. 1–6. doi: 10.1109/INDCON.2013.6725905.

[8] J. C. S. Miguel and J. M. Martínez, "Robust Unattended and Stolen Object Detection by Fusing Simple Algorithms," in *2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, Santa Fe, NM, USA: IEEE, Sep. 2008, pp. 18–25. doi: 10.1109/AVSS.2008.16.

[9] N. Bird, S. Atev, N. Caramelli, R. Martin, O. Masoud, and N. Papanikolopoulos, "Real time, online detection of abandoned objects in public areas," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, USA: IEEE, 2006, pp. 3775–3780. doi: 10.1109/ROBOT.2006.1642279.

[10] S. Ferrando, G. Gera, and C. Regazzoni, "Classification of Unattended and Stolen Objects in Video-Surveillance System," in *2006 IEEE International Conference on Video and Signal Based Surveillance*, Sydney, Australia: IEEE, Nov. 2006, pp. 21–21. doi: 10.1109/AVSS.2006.32.

[11] T. D'Orazio, P. Spagnolo, M. Leo, A. Caroppo, and T. Martiriggiano, "AN INNOVATIVE APPROACH FOR ABANDONED OR REMOVED OBJECTS DETECTION".

[12] R. K. Tripathi, A. S. Jalal, and S. C. Agrawal, "Abandoned or removed object detection from visual surveillance: a review," *Multimed Tools Appl*, vol. 78, no. 6, pp. 7585–7620, Mar. 2019, doi: 10.1007/s11042-018-6472-9.

[13] F. Porikli, Y. Ivanov, and T. Haga, "Robust Abandoned Object Detection Using Dual Foregrounds," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, p. 197875, Dec. 2007, doi: 10.1155/2008/197875.

[14] K. Lin, S.-C. Chen, C.-S. Chen, D.-T. Lin, and Y.-P. Hung, "Abandoned Object Detection via Temporal Consistency Modeling and Back-Tracing Verification for Visual Surveillance," *IEEE Trans.Inform.Forensic Secur.*, vol. 10, no. 7, pp. 1359–1370, Jul. 2015, doi: 10.1109/TIFS.2015.2408263.

[15] Y. Tian, A. Senior, and M. Lu, “Robust and efficient foreground analysis in complex surveillance videos,” *Machine Vision and Applications*, vol. 23, no. 5, pp. 967–983, Sep. 2012, doi: 10.1007/s00138-011-0377-1.

[16] M. Bhargava, C.-C. Chen, M. S. Ryoo, and J. K. Aggarwal, “Detection of object abandonment using temporal logic,” *Machine Vision and Applications*, vol. 20, no. 5, pp. 271–281, Jul. 2009, doi: 10.1007/s00138-008-0181-8.

[17] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun, “Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 41, no. 5, pp. 565–576, Sep. 2011, doi: 10.1109/TSMCC.2010.2065803.

[18] M. Piccardi, “Background subtraction techniques: a review,” in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, The Hague, Netherlands: IEEE, 2004, pp. 3099–3104. doi: 10.1109/ICSMC.2004.1400815.

[19] S. Joudaki, M. S. Bin Sunar, and H. Kolivand, “Background subtraction methods in video streams: A review,” in *2015 4th International Conference on Interactive Digital Media (ICIDM)*, Bandung, Indonesia: IEEE, Dec. 2015, pp. 1–6. doi: 10.1109/IDM.2015.7516329.

[20] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 19, no. 7, 1997.

[21] S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *CVPR 2011*, Colorado Springs, CO, USA: IEEE, Jun. 2011, pp. 1937–1944. doi: 10.1109/CVPR.2011.5995508.