Abandoned Object Detection Using Canny Edge Detection and Temporal Frame Analysis for Video Surveillance

Manshi Bhatt
Department Of Computer Science And Engineering
Govind Ballabh Pant Institute Of Engineering And Technology
Uttarakhand, India

ABSTRACT
This paper explores the application of Canny edge detection for abandoned object detection in surveillance videos. Edge-based approaches are investigated by combining Canny edge detection with temporal frame analysis to identify static objects that remain stationary beyond a specified duration threshold. Through experimental evaluation on surveillance footage, both the capabilities and limitations of Canny-based methods are demonstrated. The approach successfully detects abandoned objects under controlled conditions while revealing specific challenges where edge-based techniques struggle. This study provides insights into the practical effectiveness of traditional computer vision methods for object persistence tracking and highlights the need for hybrid approaches in complex surveillance scenarios.

General Terms
Computer Vision, Pattern Recognition, Surveillance, Algorithms.

Keywords
Abandoned Object Detection, Canny Edge Detection, SSD Object Classification, Background Subtraction, Video Surveillance, Static Object Tracking, Temporal Analysis

1. INTRODUCTION
Ensuring public safety in public places such as airports and malls often relies on detecting unattended or abandoned objects. Manual surveillance is prone to human error and fatigue, making automated solutions increasingly valuable. In this work, **the application of Canny edge detection** for abandoned object detection are investigated by analyzing edge consistency across video frames over time. **The study explores** how traditional edge-based approaches perform in surveillance contexts, combining Canny detection with temporal frame analysis to identify objects that remain stationary beyond a defined threshold. To provide broader context, experiments were also conducted with a pre-trained Single Shot Multibox Detector (SSD) [2] for object classification. This research **examines both the capabilities and limitations** of traditional image processing techniques compared to modern deep learning methods, **contributing insights** into the practical effectiveness of edge-based detection in surveillance applications.

2. RELATED WORK
Abandoned object detection has been a key topic in computer vision and surveillance [3]. Several techniques have evolved over time, ranging from traditional background subtraction to deep learning-based object detection models.

In **Cheng et al. (2012)[4]**, a background modeling approach was used to detect stationary objects in surveillance footage. Their system compared frame differences over time to determine if an object remained in the same position beyond a threshold duration.

K. Kim et al. (2005) [5] proposed a method combining background subtraction and foreground tracking to detect

abandoned and removed objects. However, these methods often fail under varying lighting and noisy environments.

To improve edge-based detection, **Canny edge detection [1]** has been widely used for its ability to identify object outlines effectively. **S. Sivaraman and M. M. Trivedi (2013) [6]** discussed how edge-based techniques could be used in real-time surveillance systems, though they sometimes struggle in scenes with cluttered backgrounds.

In recent years, deep learning models have significantly improved detection accuracy. **Liu et al. (2016) [2]** introduced the **Single Shot Detector (SSD)**, which detects multiple objects in a single forward pass, enabling real-time performance. SSD has since been widely applied in surveillance for tasks such as person detection and object classification.

Another well-known deep learning model, You Only Look Once (YOLO)[7], offers fast and accurate object detection by processing the entire image at once. Due to its speed and versatility, YOLO has become a popular choice for real-time monitoring and object tracking in public spaces.

These foundational works inform the methodology, which integrates classical edge-based methods with modern object detection algorithms to achieve reliable and efficient abandoned object detection.

3. PROPOSED METHOD
This section describes the complete methodology used for detecting abandoned objects using a combination of traditional edge-based detection (Canny) and deep learning (SSD). The aim is to ensure accurate detection of static

unattended objects in surveillance footage, which can be scaled for real-time applications.

3.1 Frame Extraction And Preprocessing

To monitor object permanence over time, the system processes video streams by extracting individual frames at a fixed interval [8]. Specifically, frames are sampled at 1 frame per second (FPS), balancing temporal resolution with computational efficiency. Each frame undergoes the following preprocessing steps:

1. Grayscale Conversion: The RGB frame Ft at time t is converted to a grayscale image G_t to simplify processing [9]:

$$Gt = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

2. Noise Reduction: A Gaussian blur[10] is applied to Gt to suppress noise and minor variations:

$$\mathrm{Gt_{blur}} = \mathrm{Gt} * \mathrm{G(x, y, \sigma)}$$

where G(x,y,σ) is a Gaussian kernel with standard deviation σ, and ∗ denotes convolution.

This preprocessing enhances edge detection reliability by reducing false positives due to noise.

3.2 Understanding and Applying Canny Edge Detection

To identify static or abandoned objects across a video sequence, it is essential to detect the consistent presence of object boundaries over time. Canny edge detection is a well-established technique used to highlight strong intensity gradients that typically represent object edges. Its effectiveness in detecting clean and well-localized edges makes it particularly suitable for surveillance-based object tracking. In this approach, after preprocessing each extracted frame, the Canny algorithm is applied to detect edge maps. These edge maps are then compared over sequential frames to determine whether specific objects have remained in a fixed location beyond a threshold duration. The steps of Canny Edge Detection have been studied carefully to make a custom canny edge detection model. The sequential processes underlying the Canny edge detection technique are detailed in the following steps:

3.2.1 Gaussian Blurring

The first step is to apply a Gaussian blur[10] to the input image to reduce noise, which can negatively affect edge detection. A kernel size of 5 and a standard deviation of 1 is used for this purpose. The formula for the Gaussian filter is:

$$G(x, y) = 1/\{2\pi\sigma^2\}e^{\{-\{x^2+y^2\}/\{2\sigma^2\}\}}$$

Where σ is the standard deviation, and x, y are the coordinates of the kernel. This step smooths the image by averaging pixel values within a specified neighborhood, reducing the effect of noise.

3.2.2. Gradient Calculation using Sobel Operator

After blurring, the intensity gradients in both the x and y directions using the Sobel operator[11] are computed. The Sobel kernels for gradient computation are:

$$Kx = \begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}, \quad Ky = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

The gradients Ix and Iy are calculated by applying the Sobel kernels to the image using convolution:

$$\mathrm{Ix} = \mathrm{filter2D(I, Kx)},$$

$$\mathrm{Iy} = \mathrm{filter2D(I, Ky)}$$

The gradient magnitude is then computed as:

$$M = \sqrt{(Ix^2 + Iy^2)}$$

And the gradient direction (or angle) is:

$$\theta = arctan(Ix/Iy)$$

3.2.3. Non-Maximum Suppression

The gradient magnitude provides information about the strength of edges, while the gradient direction helps in determining the orientation of edges. Non-maximum suppression[1] is used to thin the edges by comparing each pixel's gradient magnitude with its neighbors in the direction of the gradient. If the pixel is not a local maximum, it is set to zero.

3.2.4. Double Thresholding

The double thresholding[1] step plays a crucial role in identifying strong, weak, and non-relevant edges. It works by setting two thresholds: a **high threshold (T_H)** and a **low threshold (T_L).** The high threshold helps to identify the strongest edges, while the low threshold allows weak edges that are connected to strong edges to be preserved.

Selecting the high (TH) and low (TL) thresholds in Canny edge detection depends on the image's contrast and noise characteristics. A common practice is to set the high threshold to (10%–20%) of the maximum gradient magnitude. The low threshold is then selected as (40%–60%) of the high threshold.

The values were determined empirically, based on testing across multiple images and selecting the ratio that consistently produced reliable edge detection results. The logic behind this is:

A higher **T_H** makes the algorithm pick only the most prominent edges but may miss finer details.

A lower **T_L** allows weak but potentially valid edges to be kept, provided they are connected to strong edges during hysteresis.

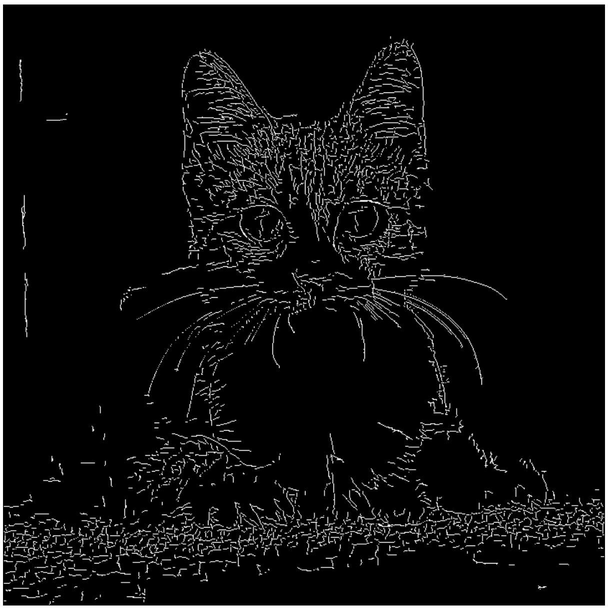
Thus, tuning TH and TL is a trade-off between noise removal and edge completeness. In more advanced systems, adaptive thresholding or dynamic calculation based on image statistics is also used.

3.2.5. Hysteresis

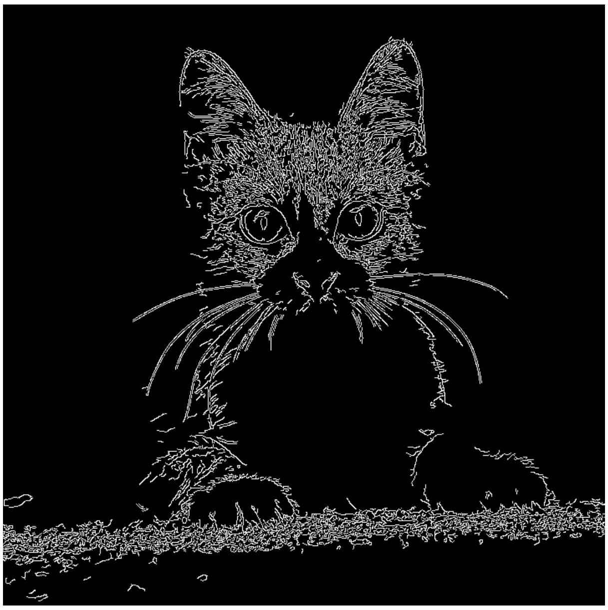
Finally, hysteresis[1] is applied to finalize the edge detection process. During hysteresis, pixels that have a weak gradient magnitude are considered as edges only if they are connected to strong edges. This helps in linking edges that may have been broken due to noise or discontinuities.

3.2.6 Comparison of Canny edge detection methods

To explore the inner workings of the Canny edge detection algorithm, each stage was initially implemented manually, including Gaussian filtering, gradient computation, non-maximum suppression, and thresholding. This manual approach offered greater flexibility and transparency, enabling detailed observation of intermediate outputs and independent adjustment of each step. However, the custom implementation was comparatively slower due to its reliance on high-level Python operations and lacked the performance optimizations present in established libraries. For subsequent experimentation and integration into the abandoned object detection system, the OpenCV implementation (cv2.Canny) was adopted. This version provided significantly faster execution and more refined edge detection results. Although it abstracts the internal processing stages, it proved highly effective for real-time applications, making it a practical and efficient choice for the final detection pipeline.



(a) Edge map generated using the custom Canny algorithm implementation.



***(b)* Edge map generated using OpenCV's built-in cv2.Canny() function.**

Figure 1: Comparison of Canny edge detection methods. The comparison highlights differences in edge sharpness, noise handling, and contour continuity.

3.3 Detecting An Abandoned Object

The proposed pipeline for abandoned object detection is designed to operate on a frame-by-frame basis, integrating spatial and temporal analysis to identify and classify static objects in surveillance video. The methodology is predicated on the assumptions of a fixed camera viewpoint and a static initial scene. The core stages of the pipeline are detailed below.

3.3.1 Background Modeling and Foreground Segmentation

The initial step involves establishing a stable representation of the scene background. The background, B(x,y), is modeled by averaging the first N frames of the video, where N is empirically set to 50 frames to ensure a clean, object-free initial model. This model serves as a static reference for the scene.

For each incoming frame, I(x,y,t), foreground segmentation is performed using frame differencing [12]. A difference image, D(x,y,t), is computed by taking the absolute per-pixel difference between the current frame and the background model:

$$D(x,y,t) = | I(x,y,t) - B(x,y) |$$

This difference image highlights regions of change. To segment these regions, a binary thresholding operation is applied to produce a foreground mask.

The intensity threshold, was set to 30 (on a 0-255 scale). This value was empirically selected through experimentation on a validation dataset to optimize the trade-off between sensitivity to human-scale object interactions and robustness against minor illumination changes and sensor noise.

3.3.2 Morphological Filtering

The raw binary mask M(x,y,t) often contains spurious noise and fragmented regions. To address this, morphological operations [13] are applied. Specifically, a morphological closing operation is performed using a 3×3 square structuring element. This operation consists of two sequential phases: dilation followed by erosion. The dilation phase expands the boundaries of foreground regions, effectively bridging small gaps and connecting nearby object fragments. The subsequent erosion phase contracts these expanded regions back to their approximate original size while preserving the newly established connections. The 3×3 square structuring element was selected after empirical testing, as it provides an optimal balance between noise removal effectiveness and preservation of object detail. This morphological closing operation effectively eliminates small noise-induced holes within object regions, smooths irregular boundaries, and creates more coherent object silhouettes, resulting in a cleaner and more reliable binary mask, Mc(x,y,t).

3.3.3 Static-Object Candidate Extraction

From the cleaned mask Mc(x,y,t), distinct foreground regions are extracted using a connected components analysis algorithm. Each resulting connected component, or blob, represents a potential object candidate. For each blob Ri, its primary geometric features are computed: the bounding box Bi and the centroid Ci=(xc,yc).

To filter out insignificant motion artifacts, a size-based pruning criterion is applied. Blobs with an area smaller than an area threshold, τ_a, are discarded. The parameter τ_a was set to 200 pixels, a value determined to be effective at removing noise while retaining even small objects of interest.

3.3.4 Edge-Based Verification

To further validate that a static region corresponds to a tangible object rather than an artifact (e.g., a shadow), edge verification is employed. The **Canny edge detector** is applied to the grayscale version of the current frame, constrained to the bounding box Bi of each candidate blob.

The parameters of the Canny edge detection algorithm were carefully tuned to balance sensitivity and precision. The high threshold (τ_{high}) was set to 200, a stringent value that ensures only pixels with strong intensity gradients are selected as initial edge points. This setting effectively suppresses noise and prevents the detection of insignificant edges. The low threshold (τ_{low}) was set to 10, enabling edge tracking via hysteresis. This allows the algorithm to link weaker, yet valid, edge segments to stronger ones, ensuring continuity in edge structures without introducing excessive noise. Together, these threshold values were chosen to optimize edge detection performance in the context of static object identification within surveillance footage.

This dual-threshold strategy confirms that the candidate region possesses a coherent edge structure, which is a strong indicator of a physical object's boundary.

3.3.5 Temporal Persistence Analysis

To distinguish between genuinely abandoned objects and temporarily stationary ones, a centroid-based tracker [8] is implemented. This tracker monitors the spatial stability of each candidate blob over a sequence of frames.

For each blob Ri with centroid Ci(t) at frame t, its position is compared to its position in the previous frame, Ci(t−1). If the Euclidean distance is within a small spatial tolerance, τ_{pos}, its stationarity counter, Si, is incremented.

$$if \ \parallel Ci(t) - Ci(t-1) \parallel < \tau pos, \ then\, Si \leftarrow Si + 1;$$

$$else\, Si \leftarrow 0$$

The tolerance τ_{pos} was set to 5 pixels to account for minor detection jitter. If a blob moves beyond this tolerance, its counter is reset to zero, classifying it as a dynamic object.

3.3.6 Static-Object Confirmation and Classification

An object is formally confirmed as "abandoned" if its stationarity counter Si exceeds a predefined temporal persistence threshold, τ_{time}. The threshold τ_{time} was set to 100 frames. Given a typical video processing rate (e.g., 30 frames per second), this corresponds to a dwell time of approximately 3.3 seconds, a duration deemed sufficient to filter out transient events while ensuring timely detection.

Upon confirmation, the image patch defined by the object's bounding box Bi is extracted from the current frame. This patch is then passed to a **Single Shot Detector (SSD)** model [2] for classification. An SSD implementation with a lightweight MobileNet backbone, pre-trained on the COCO dataset, was utilized. The classifier's role is to assign a semantic label (e.g., "suitcase," "bag") to the detected static object. It was noted, however, that the classifier's performance was sensitive to challenging lighting conditions and the presence of object classes not well-represented in its training data.

4. RESULTS

4.1 Evaluation Metrics

To evaluate the performance of the proposed unattended object detection system, five videos from the ABODA dataset were used, and three standard objective metrics were employed[14]: **Correct Object Detection Rate (CODR), False Alarm Rate (FAR),** and **Object Success Rate (OSR).** These metrics are defined as follows:

$$CODR = TP/(TP + FN)$$

$$FAR = FP/(TP + FP)$$

$$OSR = CODR/(CODR + FAR)$$

Where:

TP: True Positives (unattended object correctly detected)

FP: False Positives (non-object detected as unattended object)

FN: False Negatives (missed unattended objects)

4.2 Quantitative Results

The system was evaluated on **five recorded videos** with static cameras and consistent lighting. Each video featured **one abandoned object**, and the system was expected to correctly detect and localize it.

Table 1. Results of unattended object detection in ABODA dataset

Video	TP	FP	FN	CODR	FAR	OSR
V1	1	1	0	1.00	0.50	0.67
V2	1	0	0	1.00	0.00	1.00
V3	1	1	0	1.00	0.00	1.00
V4	0	1	1	0.00	1.00	0.00
V5	0	2	1	0.00	1.00	0.00

Note: Evaluation was done per video (not per frame), as the goal was to detect whether or not the object was eventually identified in the scene.

4.3 Performance Analysis

The results demonstrate a mixed performance profile with clear patterns:

Successful Cases (V1, V2, V3): These videos showed successful object detection with CODR of 1.00, indicating that when conditions were favourable, the Canny-based approach could reliably identify abandoned objects. V2 achieved perfect performance (OSR = 1.00) with no false positives, representing the ideal scenario for this method.

Failed Cases (V4, V5): These videos resulted in failure (CODR = 0.00) with high false alarm rates (FAR = 1.00), indicating fundamental challenges in certain environmental conditions or object characteristics.

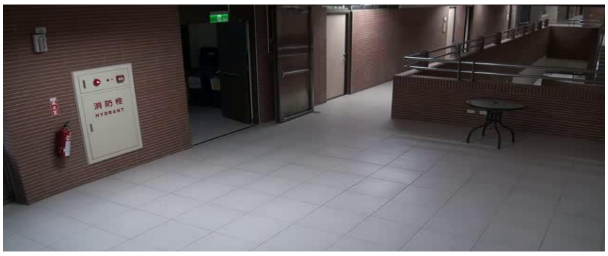
Overall Performance Metrics:

Average CODR: 0.60 (60% detection rate)

Average FAR: 0.50 (50% false alarm rate)

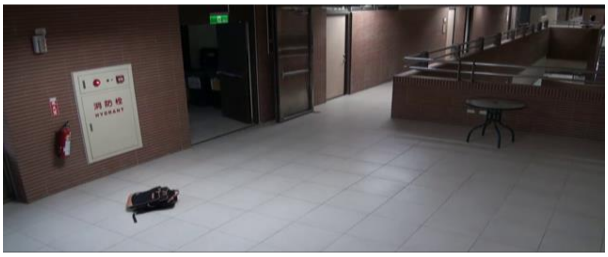
Average OSR: 0.53 (53% overall success rate)

4.4 Qualitative Analysis and Visual Results



(a)Initial Frame

(b) Object Introduced



(c)Object Left Abandoned

(d) Object Detected

Fig.2 Sample output from Video V1 showing correct detection of the abandoned object.

(a)Initial Frame

(b)Object Introduced

(c)Object Left Abandoned

(d)**Object Detected: Successful identification and localization**

Fig.3 Sample output from Video V2 showing correct detection of the abandoned object.

4.5 Project Exploration and SSD Attempt

This project began with an emphasis on classical computer vision techniques, particularly **Canny edge detection[1]**, **frame differencing[12]**, **morphological operations[13]**, and **contour tracking**. These methods were chosen to build a foundational understanding of motion and edge-based detection for unattended objects. The logic was implemented from scratch and successfully tested on 1–2 specific videos.

During exploration, an **SSD-based object detection model** (Single Shot Multibox Detector with MobileNet backbone) was also implemented to evaluate the feasibility of deep learning for the task. However, the SSD model produced reliable results on only **one video** and failed on others. The SSD model's poor performance can be attributed to several key factors. A primary issue was the mismatch between the pre-trained SSD classes and the actual objects present in the surveillance scenes. Mismatch between pre-trained SSD classes and actual objects in the scene, Additionally, the lack of dataset-specific fine-tuning further compromised the model's effectiveness, as it had not been adapted to the particular characteristics and visual features of objects within the ABODA dataset used for evaluation. **SSD was not included in the final evaluation**, but its inclusion demonstrates the scope for future work using deep learning if custom datasets and annotations are available. The results are shown in Fig.4.

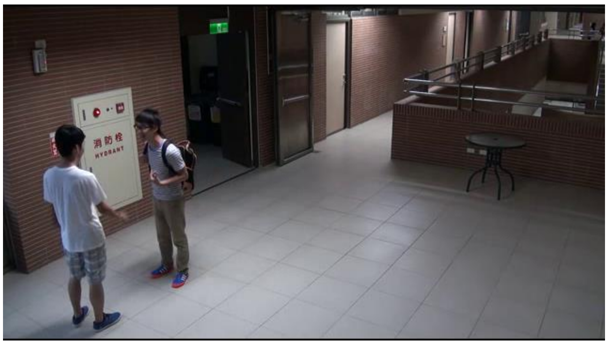


Fig:4(a) Frame where backpack was introduced (~Frame 78): Clear object visibility

Fig:4(b) Object flagged as abandoned (~Frame 1721): Successful tracking

4.6 Detailed Analysis of Success and Failure Factors

The analysis of experimental results revealed several success factors that contributed to the effectiveness of the proposed approach. Notably, the system performed reliably under consistent lighting conditions, indicating that stable illumination plays a critical role in ensuring accurate detection. Similarly, the presence of simple and uncluttered backgrounds enhanced foreground segmentation, while objects with clear and well-defined boundaries were more accurately detected by the Canny-based method. Additionally, larger objects that exceeded the predefined area threshold were consistently identified, demonstrating the system's sensitivity to object scale.

On the other hand, several factors contributed to system failure.

Lighting variations, particularly abrupt changes in illumination, led to a significant drop in performance, underscoring the method's sensitivity to environmental conditions. The presence of complex backgrounds with multiple overlapping edge features often resulted in false positives. Furthermore, small objects or those lacking distinct edge contrast were frequently missed, highlighting limitations in edge-based detection. Lastly, the use of fixed threshold values constrained the system's ability to adapt to different scene characteristics, reducing overall robustness in dynamic surveillance environments.

The experiments, despite exhibiting a 40% failure rate, revealed several strengths and insightful observations regarding the Canny-based detection approach. The method demonstrated reliable performance under consistent lighting conditions, indicating its suitability for controlled environments. Additionally, it performed notably better in scenes with simple, static backgrounds, suggesting its effectiveness in structured surveillance settings. While the use of fixed thresholds presented certain limitations, these outcomes underscored the potential benefits of incorporating more adaptive and context-aware thresholding strategies to enhance robustness.

5. LIMITATIONS AND FUTURE WORK

5.1 Current Limitations

The experimental evaluation of the proposed approach uncovered several noteworthy limitations that impact its robustness and generalizability. A significant challenge observed was the system's **sensitivity to environmental conditions**. While it exhibited satisfactory performance on video sequences captured under consistent lighting and with a stable camera, its effectiveness declined markedly in scenarios characterized by low illumination or background movement. In particular, the method struggled with varying lighting conditions, indicating a lack of adaptability to real-world surveillance environments.

Moreover, the system demonstrated limitations when applied to scenes with **complex or dynamic backgrounds**. The presence of frequent motion or visual clutter compromised the accuracy of detection, suggesting that the current approach is better suited to clean, controlled settings. This restricts its practical utility in crowded or visually intricate surveillance scenarios, such as public transport hubs or open urban spaces. Another critical limitation was the use of **fixed threshold values** for edge detection and object confirmation. The static nature of these parameters reduced the system's flexibility across different video conditions. Scene-specific calibration was often necessary to achieve optimal results, highlighting the need for adaptive or learning-based parameter tuning strategies.

Also, **scalability remains a concern**, as the system relies on fundamental techniques such as edge detection and background differencing. These methods, while computationally efficient, may not perform reliably in large-scale deployments involving high object density, frequent occlusions, or complex spatiotemporal interactions.

5.2 Future Enhancements

Looking ahead, several enhancements can be explored to improve the accuracy, adaptability, and real-world applicability of the system. One promising direction is the integration of deep learning models such as SSD [2] or YOLO [7], trained on more contextually relevant datasets. These architectures have the potential to enhance object detection and support more reliable object classification, particularly in visually complex or cluttered environments. Another important improvement involves the adoption of adaptive thresholding techniques, which would allow the system to adjust its detection parameters dynamically based on scene-specific characteristics and varying lighting conditions. This could increase robustness across diverse surveillance scenarios. In addition, using multi-modal approaches that combine edge-based detection with other features such as texture, color histograms, and motion cues may provide more resilient and context-aware object identification. Moreover, developing comprehensive real-world surveillance datasets with varied illumination, background dynamics, and object categories would enable more rigorous training and evaluation of improved models, supporting broader deployment in practical settings.

6. CONCLUSION

This research developed and evaluated a method for detecting abandoned objects using Canny edge detection combined with temporal frame analysis. The Canny algorithm effectively highlighted object boundaries, while morphological operations filtered noise and isolated meaningful objects that remained unmoved for specified durations.

The experimental evaluation on the ABODA dataset revealed both the potential and limitations of traditional computer vision approaches for surveillance applications. The system achieved successful detection in 60% of test cases, with perfect performance under ideal conditions (controlled lighting and simple backgrounds) but failure in challenging scenarios.

The key contributions of this work include: (1) a detailed implementation and analysis of Canny-based abandoned object detection, (2) systematic evaluation using standard metrics on surveillance footage, (3) identification of specific environmental factors that influence performance, and (4) demonstration of both the capabilities and limitations of edge-based approaches in practical surveillance contexts.

Though lightweight and computationally efficient, the system performed reliably in controlled video settings and can serve as a foundation for real-time surveillance applications. The findings highlight the importance of environmental considerations in surveillance system design and demonstrate the need for adaptive approaches that can handle varying conditions.

Future research directions include integration of deep learning models for improved object classification, development of adaptive thresholding mechanisms, and creation of comprehensive datasets that better represent real-world surveillance challenges. The combination of traditional computer vision techniques with modern deep learning approaches shows promise for creating more robust and reliable abandoned object detection systems suitable for diverse surveillance environments.

The practical implications of this research extend to security applications in airports, malls, and other public spaces where automated surveillance can augment human monitoring capabilities. While the current approach shows limitations in complex scenarios, it provides valuable insights into the fundamental challenges and requirements for effective abandoned object detection in surveillance systems.

7. ACKNOWLEDGEMENT

The authors would like to thank Dr.Bhumika Gupta, Head of Department, Department of Computer Science and Engineering, Govind Ballabh Pant Institute Of Engineering And Technology , for their constant guidance and support throughout this project.

8. REFERENCES

[1] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, no. 6, pp. 679-698, Nov. 1986.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Proc. Eur. Conf. Comput. Vision, 2016, pp. 21-37.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviours," IEEE Trans. Syst., Man, Cybern. C, vol. 34, no. 3, pp. 334-352, Aug. 2004.

[4] F. C. Cheng, S. C. Huang, and S. J. Ruan, "Illumination-sensitive background modeling approach for accurate moving object detection," IEEE Trans. Broadcasting, vol. 58, no. 3, pp. 794-801, Sep. 2012.

[5] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," Real-time Imaging, vol. 11, no. 3, pp. 172-185, Jun. 2005.

[6] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1773-1795, Dec. 2013.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vision Pattern Recognition, 2016, pp. 779-788.

[8] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Comput. Surv., vol. 38, no. 4, pp. 13-es, Dec. 2006.

[9] ITU-R Recommendation BT.601-7, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," International Telecommunication Union, 2011.

[10] I. T. Young and L. J. Van Vliet, "Recursive implementation of the Gaussian filter," Signal Process., vol. 44, no. 2, pp. 139-151, 1995.

[11] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," presented at the Stanford Artificial Intelligence Project, 1968.

[12] M. Piccardi, "Background subtraction techniques: a review," in Proc. IEEE Int. Conf. Syst., Man, Cybern., vol. 4, 2004, pp. 3099-3104.

[13] J. Serra, Image Analysis and Mathematical Morphology. London, U.K.: Academic Press, 1983.

[14] N. Dwivedi, D. K. Singh, and D. S. Kushwaha, "An Approach for Unattended Object Detection through Contour Formation using Background Subtraction," Procedia Computer Science, vol. 171, pp. 1979-1988, 2020.