applied sciences

MDPI

Article

A New, Robust, Adaptive, Versatile, and Scalable Abandoned Object Detection Approach Based on DeepSORT Dynamic Prompts, and Customized LLM for Smart Video Surveillance

Merve Yilmazer [1,*] and Mehmet Karakose [2]

[1] Computer Engineering Department, Munzur University, 62000 Tunceli, Türkiye
[2] Computer Engineering Department, Firat University, 23119 Elazığ, Türkiye; mkarakose@firat.edu.tr
* Correspondence: merveyilmazer@munzur.edu.tr

Abstract: Video cameras are one of the important elements in ensuring security in public areas. Videos inspected by expert personnel using traditional methods may have a high error rate and take a long time to complete. In this study, a new deep learning-based method is proposed for the detection of abandoned objects, such as bags, suitcases, and suitcases left unsupervised in public areas. Transfer learning-based keyframe detection was first performed to remove unnecessary and repetitive frames from the ABODA dataset. Then, human and object classes were detected using the weights of the YOLOv8l model, which has a fast and effective object detection feature. Abandoned object detection is achieved by tracking classes in consecutive frames with the DeepSORT algorithm and measuring the distance between them. In addition, the location information of the human and object classes in the frames was analyzed by a large language model supported by prompt engineering. Thus, an explanation output regarding the location, size, and estimation rate of the object and human classes was created for the authorities. It is observed that the proposed model produces promising results comparable to the state-of-the-art methods for suspicious object detection from videos with success metrics of 97.9% precision, 97.0% recall, and 97.4% f1-score.

Keywords: abandoned object detection; keyframe detection; YOLOv8; DeepSORT; LLM

Academic Editor: Andrea Prati

Received: 1 February 2025
Revised: 27 February 2025
Accepted: 28 February 2025
Published: 4 March 2025

Citation: Yilmazer, M.; Karakose, M. A New, Robust, Adaptive, Versatile, and Scalable Abandoned Object Detection Approach Based on DeepSORT Dynamic Prompts, and Customized LLM for Smart Video Surveillance. *Appl. Sci.* **2025**, *15*, 2774. https://doi.org/10.3390/app15052774

1. Introduction

Closed circuit television (CCTVs) systems consist of multiple security cameras connected to one or more video monitors [1]. Such security systems are used in many public areas, such as schools, hospitals, and airports, to detect various abnormal behaviors, suspicious objects, and movements. Video surveillance systems (VSSs) that record 24 h a day, 365 days a year are widely used today. In such systems, video inspections are performed by human operators. It is not possible to examine multiple video frames simultaneously with a high attention level. This inevitably leads to the problem of a decreasing attention level as operator fatigue increases [2,3]. Visual surveillance involves the practice of identifying, classifying, and detecting normal and abnormal human activities. Walking, running, waving, or talking are considered normal human movements. Explosive attacks, fights, running crowds, theft, and abandoned objects (bags, suitcases, etc.) are considered abnormal human movements. Especially, in order to prevent explosive attacks, the detection of abandoned/suspicious objects is of great importance for human security [4]. Various methods are suggested for the fastest and most accurate anomaly detection [5]. Nowadays, with the spread of security and monitoring systems in many areas, video data have also

grown rapidly. The increasing success rates of intelligent systems and models, such as machine learning and deep learning, on visual data have enabled image analysis and video analysis to be performed with automatic systems. In this way, video and image analyses that require a long time and effort can be performed in a shorter time and with higher accuracy [6]. In a video analysis, each frame of the video is treated as an independent image. Thus, each frame is processed separately to detect objects, behaviors, and situations from the video [7]. Applications such as object detection and suspect detection in video analysis include the classification of objects or samples within a frame and the determination of their locations. Object tracking is the continuous tracking of the detected object or sample by assigning a number to it in consecutive frames [8]. Detection, classification, segmentation, and tracking of various situations, events, and people found in video data are carried out using various image and video processing techniques. However, deep learning-based techniques are mostly used in the field of video and image processing because they have a high success rate and a self-learning structure [9,10]. Deep learning consists of modeling artificial neural networks to work like the human brain. It is defined as a system that can automatically extract features from large amounts of data and learns on its own with minimal human effort. There are various deep learning networks, such as the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Deep Generative Networks. The areas using deep learning are basically as follows [11,12].

- Autonomous vehicles;
- Voice and video processing;
- Natural language processing;
- Handwritten character recognition;
- Medical image processing;
- Signature verification;
- Big data.

Suspicious/abandoned object detection is important for ensuring public safety, preventing potential dangers and taking precautions against security breaches. The dual background model [13] is used to extract static foregrounds [14], and deep learning-based [15,16] methods are generally used to detect abandoned objects. The traditionally used binary background model for abandoned/suspicious object detection and background model for extracting static foregrounds have disadvantages such as staticity, misleading results against illumination change, shadow area problems, and the difficulty of accurate detection in complex backgrounds. Deep learning-based methods are often preferred because they provide a good performance on image data, although they require preprocessing, such as data labeling, data augmentation when necessary, and the selection of the appropriate deep learning model [17]. Although deep learning-based methods require a large amount of data for model training, the presence of almost identical data together causes overfitting in the model. Overfitting causes the model to memorize the results and produce successful results only on those data [18]. Many studies have been conducted on abandoned object detection. However, due to the neglect of environmental differences and the limited real-time applications, it has been determined that false negatives/positives occur in real-life applications. In this study, modern artificial intelligence techniques, such as YOLOv8, DeepSORT, Generative Artificial Intelligence (GenAI), and keyframe detection, were used to overcome these difficulties. The use of these techniques allowed a faster and more accurate detection of abandoned objects. In general, by detecting keyframes used in this study, efficiency was increased by processing only important frames. YOLOv8, which meets the need for fast and secure intervention, was used to detect object and human classes. By tracking object and human classes with DeepSORT, the problem of inadequacy in the feature extraction was solved and tracking uncertainties in consecutive frames was prevented.

Distance measurement and explanation outputs between human and object classes were created via GenAI with prompt engineering techniques. The main contribution point that distinguishes the proposed method in this study from other studies is the integrated use of artificial intelligence-based systems in all steps. In addition, the applicability of the model is increased with LLM-based explanations and the explanations are made understandable. The advantages of the method proposed within the scope of this study over the existing methods are presented below.

- This study focuses on providing a new and versatile method for abandoned object detection.
- Real-time, fast, and high-accuracy abandoned object detection is achieved with the integrated use of deep learning-based models.
- The use of YOLOv8, DeepSORT, and GenAI makes abandoned object detection more accurate and faster, while keyframe detection increases the efficiency by ensuring that only important frames are processed.
- The applicability of the model is supported by the result explanation module, which is suitable for improvement with user feedback based on the large language model.
- It is a pioneering approach to perform abandoned object detection in public areas in real time and with a low hardware cost.
- The created system will be used in subsequent studies as a module of the Video Analytics Artificial Intelligence library.

The remaining sections of the paper are structured as follows: Section 2 includes a comparative review of the proposed works in the literature on abandoned object detection. Section 3 provides an overview of the proposed method. Section 4 includes the experimental results of the proposed method and its comparison with state-of-the-art methods. Section 5 includes a general evaluation of the paper.

2. Related Work

The detection of suspicious/abandoned objects in public areas plays an important role in public safety. For this reason, intelligent systems, such as deep learning and machine learning, have recently been used to automatically and quickly detect suspicious situations in places constantly monitored by CCTV cameras. This section includes a review of the state-of-the-art methods proposed for the detection of abandoned objects. Dwivedi et al. [19] proposed a new method for removing suspicious objects in the foreground based on background subtraction. They categorized whether the foreground objects were static or moving by comparing their center of gravity coordinates. If the foreground object is stationary for a predefined period of time and its size is also within the predefined range, it is classified as an unattended/abandoned object. Testing their proposed method on the ABODA dataset, the researchers measured the Correct Object Detection Rate (CODR), Object Success Rate (OSR), and False Alarm Rate (FAR) as 70.83%, 67.25%, and 35.41%, respectively. Lwin, Tun [20] proposed a new alternative method for suspicious object detection systems that have become mandatory for public safety. In the system they proposed, they detected six different types of objects, namely a person, backpack, handbag, book, umbrella, and suitcase, using the YOLOv4 algorithm. They tracked the detected objects using the Kalman Filter (KF). Ahammed et al. [21] proposed a suspicious object detection system consisting of four stages. They defined the method steps as object detection, object tracking, owner analysis, and abandoned object analysis. They detected two classes, human and object, using the YOLOv3 neural network. They used time and distance information to determine the owner of the detected objects. They defined objects that were over a certain distance and moved away for more than 10 s as abandoned objects. They showed that they could detect abandoned objects and their owners with 65.66% accuracy.

Jeong et al. [22] proposed a three-stage method, the block diagram of which is presented in Figure 1, which integrates advanced learning algorithms and background masks to improve the accuracy of automatic abandoned object detection in public areas. In the first stage, data preprocessing was performed to remove noise from the images. In the second stage, abandoned object recognition (AOR) was performed to distinguish between dynamic and static entities. In the third and final stage, feature correlation analysis was performed to detect abandoned objects.

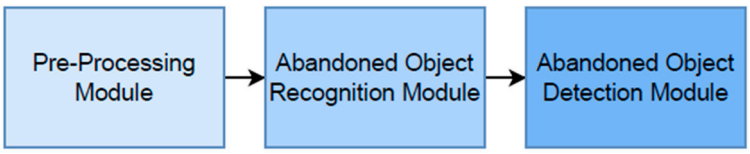


Figure 1. Abandoned object detection based on background subtraction [22].

Preetha et al. [23] proposed a new method for detecting abandoned objects in public areas and identifying people who left the object at the scene. Firstly, they improved the image quality of videos by using residual dense networks. Abandoned objects were identified by measuring the distance between them and people. In order to reduce the False Alarm Rate, a fuzzy rule-based threat assessment module was used in the method. As an alternative to the abandoned object type, Melegrito et al. [24] proposed a YOLOv3-based method for detecting abandoned grocery carts in parking lots after shopping. They showed that their proposed method could detect abandoned grocery carts in parking lots with a mAP value of 93.00%. Park et al. [25] proposed a suspicious object detection method using PETS2006, ABODA and their own datasets with a binary background model. They improved their proposed method according to different situations, such as sudden illumination changes, long-term abandonment, and the owner retaking the object. The distance between the fixed foreground objects and their owners is measured, and if it is outside the default distance, an abandoned object alarm is sounded. For the detection of abandoned objects, it is possible to augment the methods based on background subtraction [26] and deep learning and machine learning [27–30] in the literature. A comparison of various studies proposed in the literature to perform abandoned object detection in public areas is presented in Table 1. When the abandoned object detection studies in the literature are examined, it is seen that they focus on the separation of static and dynamic objects with the background subtraction method. This method may fail to detect abandoned objects due to the constant change in the background in images containing complex conditions (different lighting conditions, windy weather, crowded environment, etc.). When the deep learning-based methods, which are another method used for abandoned object detection in the literature, are examined, it is determined that the use of similar frames in consecutive frames in images obtained from CCTV cameras that record 24 h a day, 7 days a week may affect the method performances in training. Similar images in consecutive frames may cause overfitting in the model and increase the computational cost. Therefore, in this study, keyframe detection is primarily performed to prevent overfitting and reduce the hardware computational cost. Then, a robust method that can perform real-time abandoned object detection, even in many complex cases, is proposed by using YOLOv8 and DeepSORT in an integrated manner.

Table 1. Abandoned object detection literature comparison.

References	Dataset	Method
[19]	ABODA Dataset	Background subtraction method
[20]	Private Dataset	YOLOv4
[22]	KISA Dataset. ABODA Dataset	Background Matting with Dense ASPP
[25]	PETS2006, ABODA, Private Dataset	Dual background model
[31]	PETS2006 and AVSS2007, ABODA Dataset	Short- and long-term background models
[32]	IITP20 Dataset, PETS06 Dataset, ABODA Dataset, CAVIAR Dataset	Convolutional Neural Network (CNN)
[33]	iLIDS and ISLab Dataset	SVM classifier

3. Proposed Approach

In the proposed study, the open source The ABandone Object DAtaset (ABODA) was used for abandoned object detection. the ABODA dataset is an open source dataset that includes 11 real-world scenes created for abandoned object detection. The dataset includes both crowded and less-crowded scenes. Detailed environment information of the videos is presented in Table 2. The 11 videos were first separated into frames. After the frames were passed through the keyframe stage, they were labeled in the YOLO format. For YOLOv8, 70% of the dataset was used for training and 30% for testing. Within the scope of the proposed method, all training and tests were performed on Google Colab with NVIDIA Tesla T4 GPU 16 GB GDDR6 Memory, Intel(R) Xeon(R) Central Processing Unit (CPU) @ 2.00GHz hardware infrastructure. NVIDIA is headquartered in Santa Clara, California, USA, and Google is headquartered in Mountain View, California, USA. The ABandoned Object DAtaset (ABODA) dataset consists of 11 videos created in various indoor and outdoor locations for the purpose of detecting abandoned objects. Each of the videos, which is approximately 2–3 min long, has a size of 720 × 480.

Table 2. ABODA dataset description.

Video	Recorded Location	Lighting
1	Indoor	Normal indoor illumination
2	Outdoor	Daylight
3	Outdoor	Daylight
4	Outdoor	Daylight
5	Outdoor	Night
6	Outdoor	Night illumination
7	Indoor	Normal indoor illumination
8	Indoor	Contrast illumination
9	Indoor	Normal indoor illumination
10	Indoor	Normal indoor illumination
11	Indoor (Crowded)	Normal indoor illumination

The implementation steps of the proposed method, the block diagram of which is presented in Figure 2, are shown below and in Algorithm 1.

- Keyframes based on deep transfer learning were detected in the dataset.
- The YOLOv8 deep neural network was used in different sub-architectures to detect people and objects, such as bags, suitcases, etc., in the videos.
- The weights of the YOLOv8l model with the highest performance were recorded to be used as the input in the DeepSort algorithm.
- The object and human classes detected using the Deepsort object tracking algorithm were tracked. The ID and coordinate information of the human and object classes were recorded in a *.txt file for each frame.

- To detect abandoned objects, the duration of the object staying in the same place in consecutive frames was determined. Then, the distance between the human and the object class was measured using the coordinate information in the *.txt file.

Algorithm 1: Steps to implement the proposed approach

```
Step 1: Keyframe Detection
keyframes = detect_keyframes(dataset)
Step 2: Object and Human Detection with YOLOv8
for each frame in video:
   Object and human detection is done with YOLOv8
   detections = YOLOv8.detect(frame)
   Classes of objects: People, bags, suitcases etc.
   people = filter_classes(detections, "person")
   objects = filter_classes(detections, ["bag", "suitcase", "object"])
   Deepsort tracks objects
   tracked_objects = Deepsort.track(people + objects)
   The ID and coordinate information of the tracked objects are saved
   save_tracking_info_to_txt(tracked_objects)
Step 3: Abandoned Object Detection
abandoned_objects = []
   for each frame in video:
   Get coordinate and ID information from *.txt file
   tracking_data = load_tracking_data_from_txt(frame)
The time the object stays in the same place is calculated
   if is_object_still(tracking_data, frame):
   time_stayed = calculate_time_stayed(tracking_data)
   If the object is stationary for a certain period of time, it may be abandoned
   if time_stayed > threshold_time:
The distance is checked
   distance = calculate_distance(tracking_data["person"], tracking_data["object"])
   If the distance is greater than 50 cm, it is considered abandoned
   if distance > 50:
   abandoned_objects.append(tracking_data["object"])
Step 4: Generating Descriptive Output with GPT-3.5-Turbo
for each abandoned_object in abandoned_objects:
   Send the object and human coordinate information to the GPT-3.5-Turbo model
   descriptive_output = generate_descriptive_output_with_GPT3(tracking_data)
                                        print(descriptive_output)
```

In addition, the coordinate information in the *.txt file was provided as a prompt to the GPT-3.5-Turbo large language model, allowing the descriptive output to be generated for the relevant frame. * represents the file name.

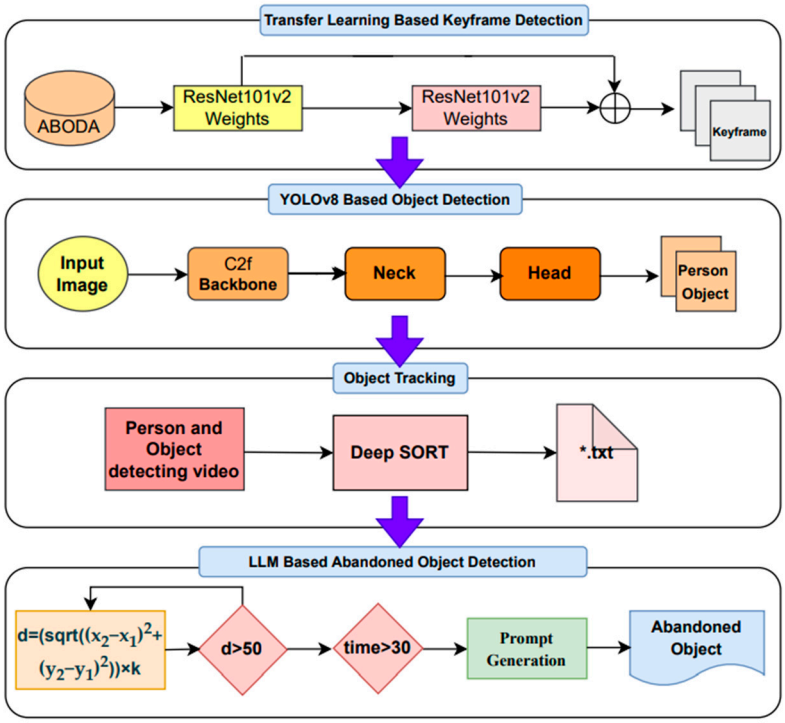


Figure 2. Block diagram of the proposed approach.

3.1. Keyframe Detection with Transfer Learning

Keyframe detection is defined as the extraction of unique representative frames while preserving the key activities of the original video. It facilitates video analysis and processing [34]. In this study, deep transfer learning was used on the ResNet101v2 backbone to detect keyframes among 37653 frames in the ABODA dataset consisting of 11 videos in total. The block diagram of the ResNet101v2-based keyframe detection is presented in Figure 3. First, all the frames were divided into two classes as "keyframe" and "nonkeyframe". Initial keyframe selections were made based on expert knowledge during the data preprocessing stage. If there was no human or object class on the image, it was determined as non-keyframe. Images containing human and/or object class were determined as keyframe. Binary classification was performed with the dataset, and the weights obtained as a result of training were recorded. Later, keyframes in the videos will be extracted using these weights. Model training parameters are presented in Table 3.

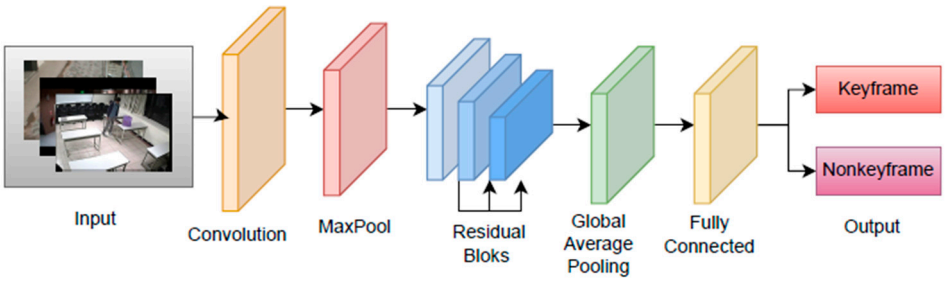


Figure 3. Block diagram of keyframe detection based on ResNet101v2.

Table 3. Transfer learning training parameter details.

Dataset	Model	Epoch	Optimizer	Image Size
ABODA	ResNet101v2	10	Adam	720 × 480

ResNet101v2 is an architecture trained on the ImageNet dataset. Although it is deeper than VGG networks, it has lower complexity. ResNet101 architecture was created by creating 101-layer blocks using more 3-layer blocks. Thanks to the increased layers, the accuracy gains of the model were significantly increased [35]. Due to these advantages, ResNet101v2 deep neural network architecture was used for keyframe detection in this study.

As a result of the training, a 98.9% train accuracy was obtained. The keyframe detection method was tested using the images allocated for testing. Classification predictions are generally analyzed in 4 categories. These are: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). Using these classification estimates, the accuracy, precision, recall, and f1-score evaluation metrics were calculated. Thus, we determined how successfully the model classified the keyframe and non-keyframe classes. The model evaluation results are presented in the Experimental Results section.

Accuracy is calculated by dividing the correct classifications (TP, TN) by the total (TP, TN, FP, FN). It shows the overall accuracy of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Precision is used to determine success in a positively predicted state. A good classifier is desired to have a precision value of 1.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall is used to determine how many of the classes that should be predicted as positive are predicted as positive. A good classifier wants the recall value to be 1.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1-score is the harmonic mean of precision and recall values. A high F1-score value indicates a balance between the correct and incorrect identification of classes.

$$\text{f1score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{4}$$

Keyframe extraction was performed from the videos provided as the input using the recorded model weights after training. The performance of detecting keyframes in the proposed keyframe detection method was measured by calculating the compression ratio (CR), computation time, kayframe_precision, keyframe_recall, and keyframe_f1score. The explanations of the metrics used in the Equations are presented in Table 4.

Table 4. Keyframe detection metric details.

Metric	Meaning
CR	Compression ratio
N_k	Total number of extracted keyframes
N_f	Total number of frames in the original video
N_m	Number of missed keyframes
N_a	Number of correctly extracted keyframes

Compression ratio shows the rate of compression based on the number of extracted keyframes and the number of original keyframes. It is calculated using Equation (5).

$$\text{CR} = \left(1 - \frac{N_k}{N_f}\right) \times 100 \tag{5}$$

Keyframe_Precision is the ratio of the total number of keyframes extracted correctly to the total number of keyframes extracted by the technique from the video sequence. It measures the accuracy of the keyframe extraction technique. It is calculated using Equation (6).

$$\text{Keyframe\_Precision} = \frac{N_a}{N_k} \times 100 \quad (6)$$

Keyframe_Recall is obtained by dividing the number of correctly extracted keyframes by the sum of the missed and correctly extracted keyframes. It is calculated using Equation (7).

$$\text{Keyframe\_Recall} = \frac{N_a}{N_a + N_m} \times 100 \quad (7)$$

Keyframe_F1-Score is a method of evaluating the performance of an algorithm by combining both precision and recall to obtain a metric using the harmonic mean. It is calculated using Equation (8).

$$\text{Keyframe\_F1} - \text{Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

Calculation time is the time it takes for the keyframe extraction technique to extract keyframes (measured in seconds) [36].

3.2. YOLOv8-Based Person and Object Detection

YOLOv8 is designed by Ultralytics to perform object detection, classification, and regression tasks. YOLOv8 basically uses a similar backbone to YOLOv5. However, due to the changes made in CSPLayer, this module is called the C2f module in YOLOv8. The C2f module increases the detection accuracy by combining high-level features with contextual information [37]. The block diagram of the YOLOv8-based person and object detection is presented in Figure 4. After the videos in the ABODA dataset are passed through the keyframe detection stage, the person and object classes in the obtained keyframes are labeled in the YOLO format. A total of 70% of the labeled data is reserved for training and 30% for testing. Training was performed for 15 epochs for each of the YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l sub-architectures.

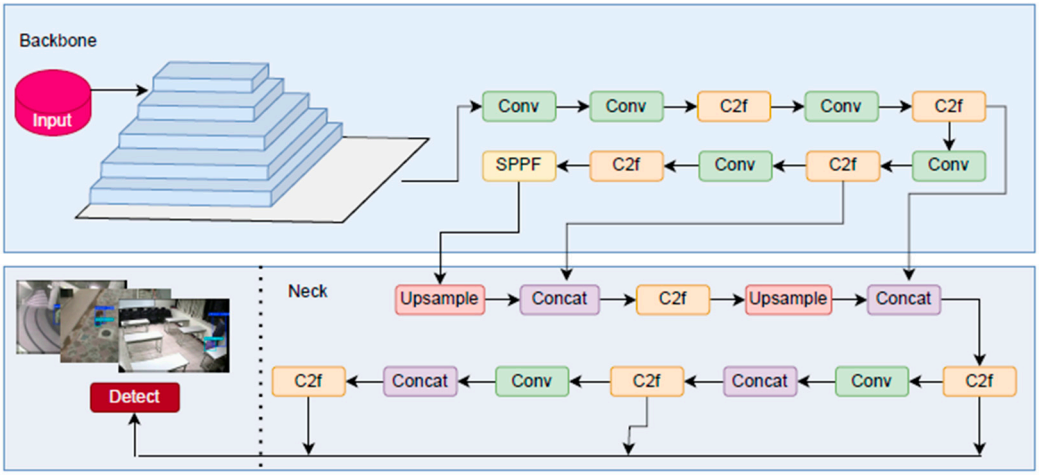


Figure 4. Block diagram of YOLOv8-based person and object detection.

The reasons for using the YOLOv8 model compared to other object detection methods are as follows:

- It has a single stage and high detection speed;
- It provides high accuracy (mAP) on visual data;

- Its architecture is not complex;
- It does not require high-standard hardware.

The performances of the models were measured using precision and recall Equations (2) and (3) and mAP Equation (9). The obtained results are presented in the experimental results section.

$$\text{mAP} = \frac{\sum_{d=1}^{D} \text{AP}(d)}{D} \tag{9}$$

3.3. Object Tracking with DeepSORT

Object tracking is a video processing task that allows the tracking of objects or instances detected within consecutive frames in videos. It is used to develop many real-world applications, such as autonomous driving, video processing, robotics, and human–computer interaction. Probabilistic, feature extraction, and deep learning-based models have been developed for object tracking purposes. In the last layer of the CNN (Convolutional Neural Network), classification and object detection are performed according to a vector representing the object. In the DeepSORT algorithm, each detected object is passed through the neural network to obtain a vector, and two objects are associated using these vectors. An ID number is assigned to each object tracked on a frame [38,39].

In this study, objects detected with YOLOv8 were tracked with the DeepSort algorithm. Thus, unique identification numbers were assigned to objects in consecutive frames. Due to the efficiency, flexibility, ease of use, and customizability advantages of YOLO for object detection, YOLOv8 + DeepSORT algorithms were preferred in this study. The reasons why DeepSORT is preferred over other object tracking methods are presented below:

- High tracking accuracy due to using deep features;
- Not having a high computational cost;
- Providing image tracking, even in densely crowded environments.

3.4. Abandoned Object Detection

Large language models have started to be used in many areas with the development of transformer architecture. In order to obtain the desired outputs from large language models, correct inputs must be provided. This situation has led to the development of prompt engineering techniques, such as zero-shot learning, one-shot learning, few-shot learning, instruction-based prompting, and chain of thought. These techniques are used to produce customized prompts, such as learning without examples, learning with a single example, learning from more than two examples, learning through instructions, and solving by thinking step by step. In this study, the instruction-based prompting technique was used to teach the language model how to perform certain instructions [40–42]. In order to determine whether the detected object was suspicious or abandoned, the duration of the object being in the same location and the Euclidean distance were taken into account. Then, LLM outputs strengthened with prompt engineering were created for expert personnel information. For this purpose, first of all, the location information of the detected human and object classes was recorded in a *.txt file. Since the videos were recorded at 15 FPS, approximately 15 frames represent 1 s. In order to determine the frame that remained motionless in the same position for 30 s, the coordinates of the object should not change in the 450 consecutive frames. For this, it was first checked whether the object remained stationary for 450 frames, as mathematically shown in Equation (10).

$$C_t = \begin{cases} C_{t-1} + 1, \text{ if } |x_t - x_{t-1}| = 0 \text{ and } |y_t - y_{t-1}| = 0 \\ 0, \text{ other situations} \end{cases} \tag{10}$$

The distance between the human classes labeled with 1 and the object classes labeled with 0 for each frame was calculated using the Euclidean distance formula presented in Equation (11). k (cm/pixel) is the scale factor. It is used to convert the calculated distance in pixels to the real-world distance.

$$d = \left(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\right) \times k \quad (11)$$

Objects whose distance was measured greater than the specified threshold value (50 cm) and remained motionless in the same position for 30 s were marked as abandoned objects. The coordinates of the human and object classes in each 450-frame result were added to the task section of the prompt created with the instruction-based prompting technique to create the explanation. Thus, LLM-based explanation outputs were created for expert personnel. The pseudo-code of the Python library that produces the abandoned object detection output by taking the output video of the DeepSORT algorithm and the txt file containing the class coordinate information in the frames as the input is presented in Algorithm 2. The abandoned object detection flowchart is given in Figure 5.

Algorithm 2: Abandoned Object Detection

Input: DeepSort Output Video
Class Coordinate Information txt folder
Output: Abandoned Object
while f < 450
Time calculation
if |x(t) − x(t − 1)| = 0 & |y(t) − y(t − 1)| = 0
f = f + 1
else f = 0
while i < d
Measure distance between person and object
d = (sqrt((x2 − x1)^2 + (y2 − y1)^2)) × k
if distance (d) > 50
then Object = Abandoned Object
else uptade new frame
i = i + 1
end

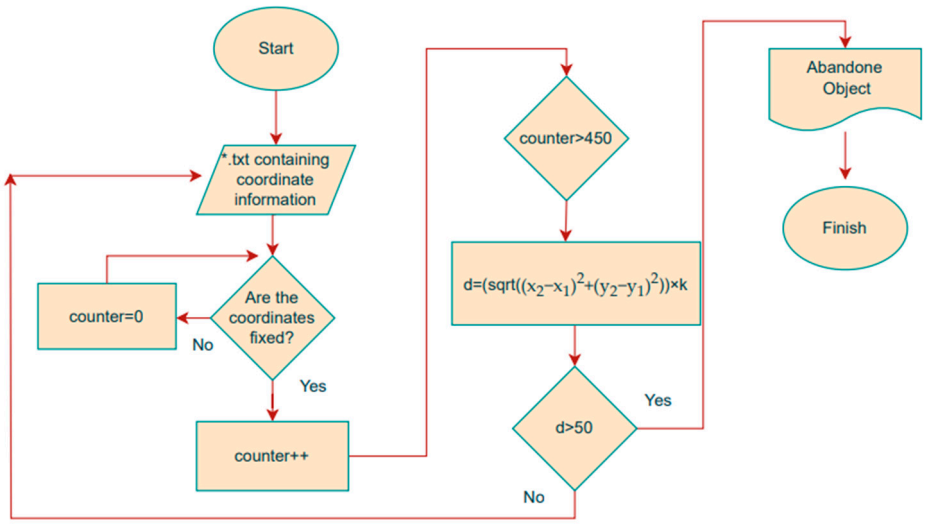


Figure 5. Abandoned object detection flowchart.

4. Experimental Results

A deep learning-based new method is proposed for the detection of abandoned objects that may endanger public safety in public areas. An adaptive, robust, and versatile abandoned object detection method consisting of four stages, namely keyframe detection, object detection, object tracking, and distance measurement, was created. The results of this innovative model are presented in this section.

4.1. Keyframe Detection Results

Keyframe detection was performed using the ResNet101v2 transfer learning algorithm. In order to determine the classification success of the dataset divided into two as "keyframe" and "non-keyframe", the accuracy, precision, recall, and f1-score were calculated. The results of the calculated model evaluation criteria are presented in Table 5.

Table 5. Keyframe detection model evaluation results.

Accuracy	Precision	Recall	F1-Score
70%	80%	70%	74%

Keyframes were extracted from the videos using the weight file obtained as a result of transfer learning. The number of original frames in each video (Nf), the number of frames extracted after the model (Nk), the number of missed keyframes (Nm), the number of correctly extracted keyframes (Na), video compression ratio (CR), computation time, and the evaluation results of the keyframe extraction method are presented in Table 6.

Table 6. Keyframe detection model evaluation for the ABODA dataset.

Video	N_f	N_k	N_a	N_m	CR	Calculation Time (s)	Keyframe Precision	Keyframe Recall	Keyframe F1-Score
V_1	2189	1058	1000	60	51.7%	337	94.5%	94.3%	94.4%
V_2	2818	692	680	23	75.4%	393	98.3%	96.7%	97.5%
V_3	2608	924	900	40	64.8%	359	97.4%	95.7%	96.5%
V_4	1169	588	570	55	49.7%	164	96.9%	91.2%	93.9%
V_5	3297	945	930	73	71.3%	495	98.4%	92.7%	95.5%
V_6	6744	3004	2955	100	55.5%	1043	98.4%	96.7%	97.5%
V_7	4801	1482	1346	84	69.1%	681	90.8%	94.1%	92.4%
V_8	4781	1547	1507	98	67.6%	675	97.4%	93.9%	95.6%
V_9	2485	611	600	65	75.4%	322	98.2%	90.2%	94.0%
V_10	2260	1339	1298	39	40.8%	307	96.9%	97.1%	97.0%
V_11	4501	3915	2500	77	13.0%	657	63.9%	97.0%	77.0%

4.2. YOLOv8-Based Person and Object Detection Results

ABODA, an open source dataset frequently used to detect abandoned objects, was passed through the keyframe detection phase and then labeled in the YOLO format for human and object detection. Using the labeled data, the YOLOv8 deep neural network was trained with nano-, small, medium, and large sub-architectures to detect humans and objects, such as bags, suitcases, etc., on video frames. Precision and recall mAP changes of the models are presented in the graph in Figure 6. Graphs showing the changes in the Mean Average Precision (mAP50, mAP50-95) metrics of the YOLOv8l model with the best performance on the test data throughout the training and the precision/recall curve are presented in Figure 7. The model outputs presented in Figure 8 show that YOLOv8 architecture produces successful results for object detection in various environments and lighting conditions. In the literature, most studies on abandoned object detection have focused only on object detection in datasets. In the method developed in this study, both

object (bag, suitcase, etc.) and human detection were performed. Real-time, versatile, abandoned object detection was performed by tracking human and object classes in consecutive frames. When the graphs are examined, it is seen that the human class performs at a value of 0.984 mAP, while the object class performs at a value of 0.966 mAP. The fact that the PR curve is above 0.90 shows that the model has a very low rate of producing false positives.

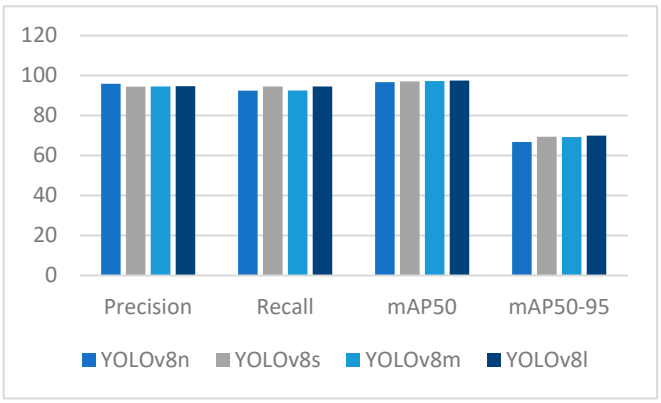


Figure 6. Performance comparison of YOLOv8 deep neural network sub-architectures.

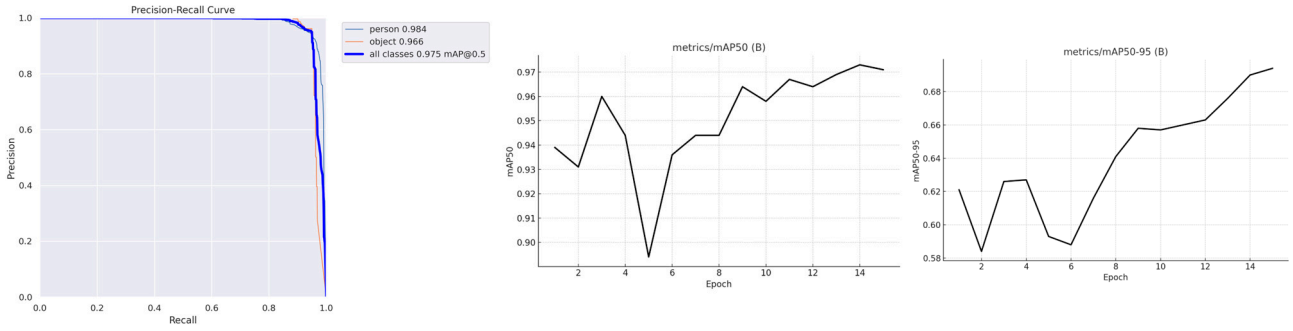


Figure 7. Mean average precision change curve—precision/recall curve.

Indoor/outdoor environments, crowds, different lighting conditions, and partial images pose challenges for object detection in video analysis. With deep learning models, dynamic and static objects can be detected accurately without any preprocessing, such as background subtraction. Due to its features, such as high accuracy, speed, and multiple object detection, the YOLOv8 deep neural network is used for person and object (bag, suitcase, etc.) detection in this study. The model output in which the human and object classes are detected from the sample image provided as the input to the YOLOv8 model is presented in Figure 9.

(**a**) Input image. (**b**) Model output image.

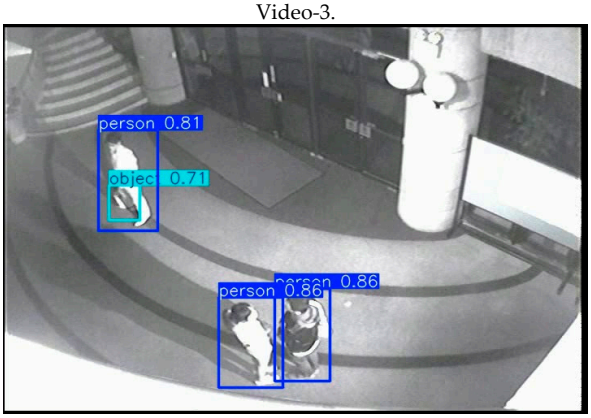
Figure 8. Person—object detection model input and output.

Video-1.

Video-2.

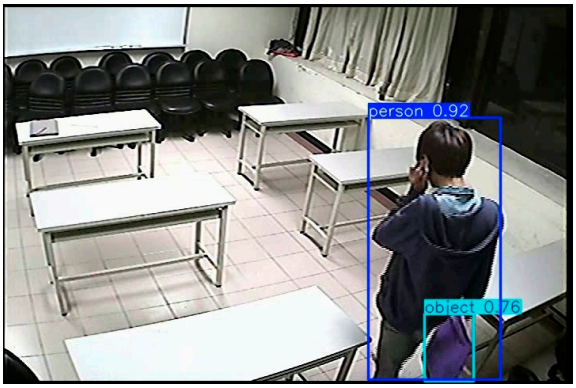
Video-3.

Video-4.

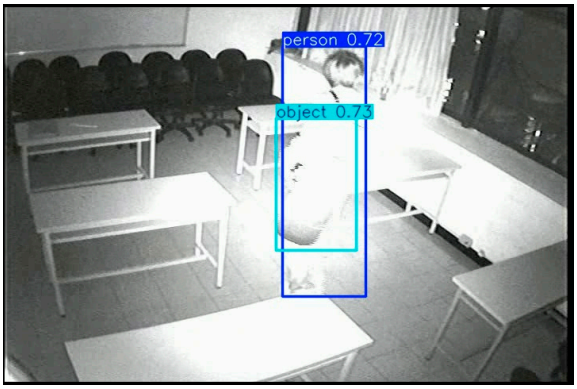


Video-5.

Video-6.

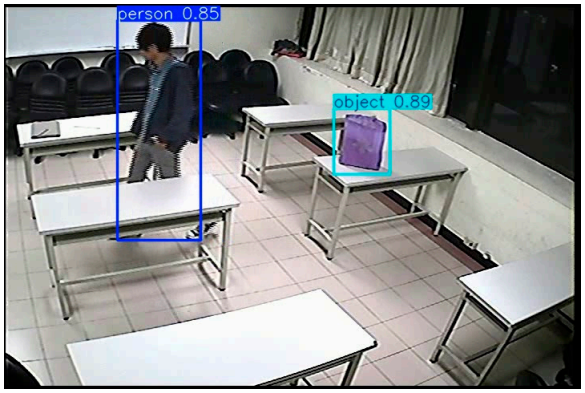


Video-7.



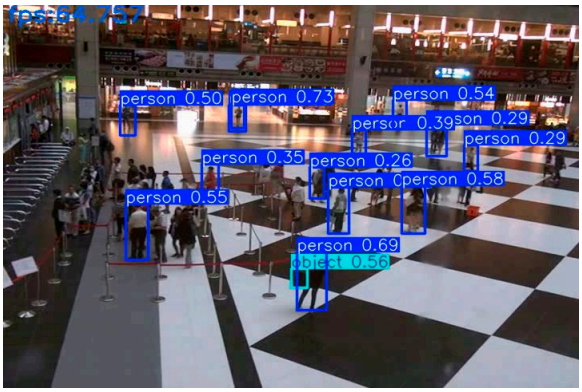
Video-8.

Figure 9. *Cont*.



Video-9.

Video-10.



Video-11.

Figure 9. YOLOv8-based person and object detection model outputs.

4.3. Object Tracking and Abandoned Object Detection Result

The detection of abandoned objects that may be dangerous for public safety in public areas is an important aspect in video surveillance. Therefore, in this study, a new method based on YOLOv8 + DeepSORT is proposed to detect abandoned/suspicious objects. The human and object classes detected with YOLOv8 are tracked with the Deep SORT algorithm. An identification number is assigned to each class tracked in consecutive frames. The location information of the detected human and object classes was saved in a *.txt file, as seen in Figure 10b. Then, objects that are in the same location for 450 consecutive frames and the distance between the human and object class is greater than 50 cm are labeled as abandoned objects. The height of the people in the images was determined as 175 cm to provide a real-world calibration. The 720 × 480 image pixels were compared to the pixels covered by the human class to create a scale factor, k, in cm/pixel. The measured distances were multiplied by the scale factor to obtain the real-world distance.

The total suspicious object detection success criteria from the videos were measured as 97.8% precision, 96.8% recall, and 97.4% f1-score. The success rates of the model for each video are presented in Table 7. With the development of the transformer architecture, large language models have shown great success in the fields of text generation, understanding, and interpretation [40,41]. In this study, generative artificial intelligence (GenAI)-based explanations were created for frames in order to improve user experiences. For this purpose, the prompt seen in Figure 11 was provided as an input to the GPT-3.5-Turbo large language model. As the output, information about the human and object classes on the frame was created. In addition, information about whether the object was abandoned was also generated. The distribution of TP, FP, and FN values obtained from the model according to the number of frames in each video is presented in the graph in Figure 12. The success of the description text produced by LLM was measured by calculating the commonly used BLEU,

ROUGE, and METEOR metrics. The metrics explained below were automatically calculated using the nltk, rouge-score, and sacrebleu libraries in Python (https://www.python.org/).

BLEU: Calculated to determine how similar the text produced by the model is to human-generated texts.

ROUGE: Calculated to determine how similar the text produced with the reference text in the field of text generation is in terms of meaning.

METEOR: Calculated to evaluate the success of the model in language generation by considering synonyms and word order.

In this study, successful explanations were generated with 78% BLEU, 83% ROUGE, and 89% METEOR scores from LLM enhanced with the instruction-based prompt engineering technique. By creating an LLM-based explanation, the usability of the system was made clear and understandable.

(**a**) Sample object and human detection image.

```
0 0.584028 0.446875 0.0625 0.322917 0.819299
1 0.247222 0.725 0.1 0.108333 0.856338
0 0.522917 0.51875 0.0986111 0.3875 0.871586
```

(**b**) Sample txt file.

Figure 10. Object and human detection image/sample txt file.

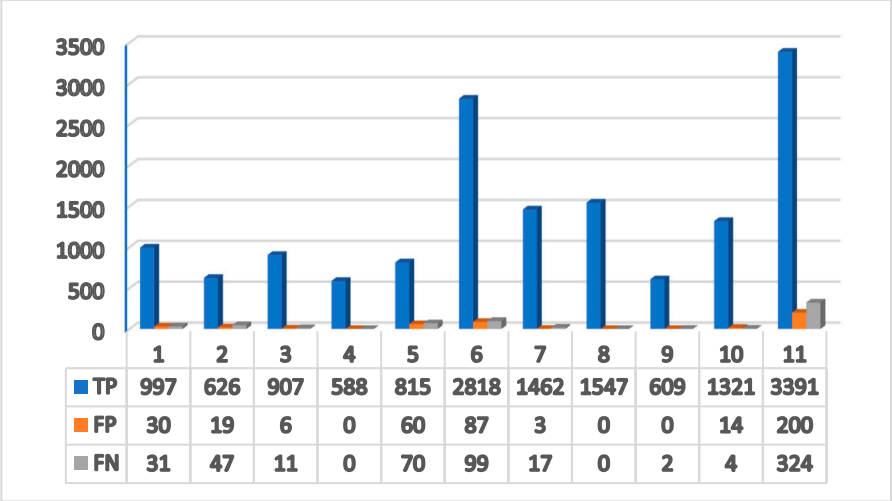
Table 7. Abandoned object detection model evaluation results.

Video	Precision	Recall	F1-Score
1	97.0%	97.0%	97.0%
2	97.0%	93.0%	95.0%
3	99.3%	98.8%	99.0%
4	100%	100%	100%
5	93.1%	92.1%	92.6%
6	97.0%	96.6%	96.8%
7	99.8%	98.9%	99.3%
8	100%	100%	100%
9	100%	99.7%	99.8%
10	98.9%	99.7%	99.3%
11	94.4%	91.2%	92.7%

Role: Description will be created for abandoned object detection.
Context: For the data given below,
0 0.338889 0.384375 0.086111 0.293750 (Human)
1 0.384028 0.404167 0.090278 0.308333 (Object)
Here '0' represents the human and '1' represents the object class.
Other values represent the location of the human and object classes.
The location of the object in the frame will be specified (for example, the upper right corner).
The distance between the object and the human will be calculated using the Euclidean distance.
The formula used to calculate the distance:
D = (sqrt((x2 - x1)^2 + (y2 - y1)^2)) x k
Here, (x1, y1) are the coordinates of the human and (x2, y2) are the coordinates of the object.
k=175 / (480 x person height)
If the distance between each human class and the object class is more than 50 cm, the object is considered an abandoned object. Check all measurements after calculations and only report the object as abandoned if all calculations are greater than 50 cm.
Task: 0 0.584028 0.446875 0.0625 0.322917 0.8199299
1 0.247222 0.725 0.1 0.108333 0.856338
0 0.522917 0.51875 0.0986111 0.3875 0.871586
Explain the given object detection output.
Determine whether the object is abandoned by calculating the distance.

Output:
The given object detection output consists of three detections.
1. The first detection is of an human located at coordinates (0.584028, 0.446875) with a width of 0.0625 and a height of 0.322917. The confidence score for this detection is 0.8199299.
2. The second detection is of a object located at coordinates (0.247222, 0.725) with a width of 0.1 and a height of 0.108333. The confidence score for this detection is 0.856338.
3. The third detection is of an human located at coordinates (0.522917, 0.51875) with a width of 0.0986111 and a height of 0.3875. The confidence score for this detection is 0.871586.
1. Calculate the distance between the human at (0.584028, 0.446875) and the object at (0.247222, 0.725).
D = sqrt{(0.584028 - 0.247222)^2 + (0.446875 - 0.725)^2} = 0.436 x 1.13 = 0.493
2. Calculate the distance between the human at (0.522917, 0.51875) and the object at (0.247222, 0.725).
D = sqrt{(0.522917 - 0.247222)^2 + (0.51875 - 0.725)^2} = 0.344 x 1.13 = 0.389
Since all calculated distances are less than 50 cm, the object is not considered abandoned in this scenario.

Figure 11. Prompt and sample output generated for LLM.



	1	2	3	4	5	6	7	8	9	10	11
TP	997	626	907	588	815	2818	1462	1547	609	1321	3391
FP	30	19	6	0	60	87	3	0	0	14	200
FN	31	47	11	0	70	99	17	0	2	4	324

Figure 12. Abandoned object detection TP, FP, and FN value comparison.

The features of the method proposed in this study for the detection of abandoned objects, its contributions to the relevant field in the literature, and its contribution rates are presented in Table 8.

The main components that directly affect the model's performance are keyframe detection, object detection with YOLOv8, and object tracking with DeepSORT. Therefore, ablation experiments were performed to find the component that affected the model's performance the most among these three components. The experimental results are presented in the graph in Figure 13. Accordingly, it was seen that object detection with YOLOv8 increased the model accuracy the most, followed by object tracking with DeepSORT.

Table 8. Contributions of the proposed method to the literature.

Feature	Contribution	Contribution Rate
New	New in terms of integrating advanced deep learning technologies for object detection and tracking.	The method supported by new transformer-based models is an innovative study in the field with a confidence rate of over 97%.
Robust	In cases where objects are partially visible in consecutive frames or disappear and become visible again, DeepSORT can be used to track them when they become visible again. Robust due to its ability to continue its function even in such unexpected situations.	Our model can detect abandoned objects with a 97.4% f1-score confidence rate.
Adaptive	Adaptive due to providing high detection rates on images obtained under different lighting and climate conditions.	Our model can adapt to 15% data changes at a rate of 90%.
Versatile	Versatile due to its capacity to perform multiple phased functions, such as keyframe detection, detection of human/object classes, tracking of human/object classes, and abandoned object detection.	Our model is versatile with a minimum f1-score of 92% in indoor, outdoor, and different lighting conditions.
Scalable	Scalable due to its applicability on datasets of different sizes than those used experimentally in the study.	Although the calculation time increases as the video size increases, the keyframe detection rate remains limited to 20% at most.

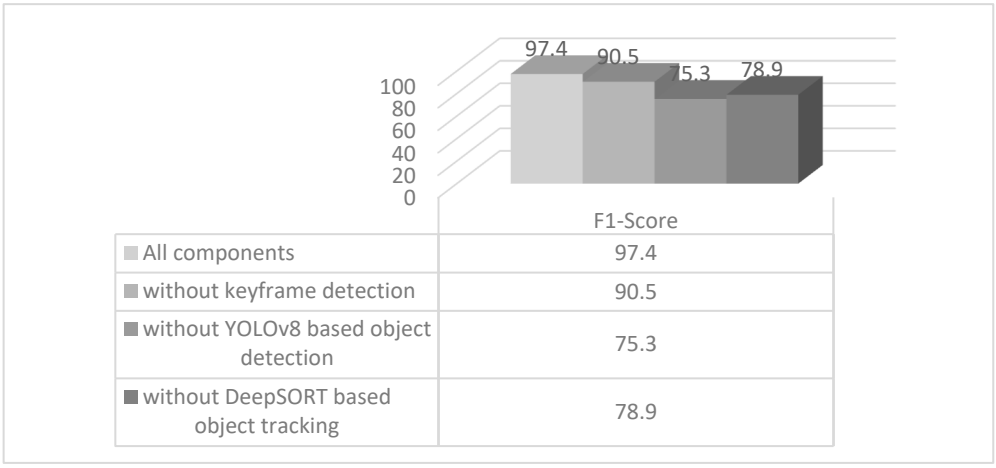


Figure 13. Ablation experiments results.

The output images of the proposed model are presented in Figure 14. Other methods proposed in the relevant field in the literature were examined and the comparison of the method proposed in this study with other methods is presented in Table 9 and Figure 15. It was seen that there were diversities such as different lighting conditions, causing outdoor and indoor images in the datasets used for abandoned object detection to be similar to real-world scenes. While using different types of data contributes to the learning scope of the model in deep learning algorithms, it has been observed that it can cause incorrect definitions in methods based only on background and foreground extractions. In this article, transfer learning-based keyframe detection was implemented to make the dataset most suitable for training and to prevent duplicate data. Then, YOLOv8-based human and object detection with a fast and effective object detection potential was realized. The tracking of objects in all consecutive frames was performed with the Deep SORT algorithm and the determination of abandoned objects was achieved. With these advantages, a more effective abandoned object detection method was presented in terms of software and hardware efficiency compared to similar methods in the literature.

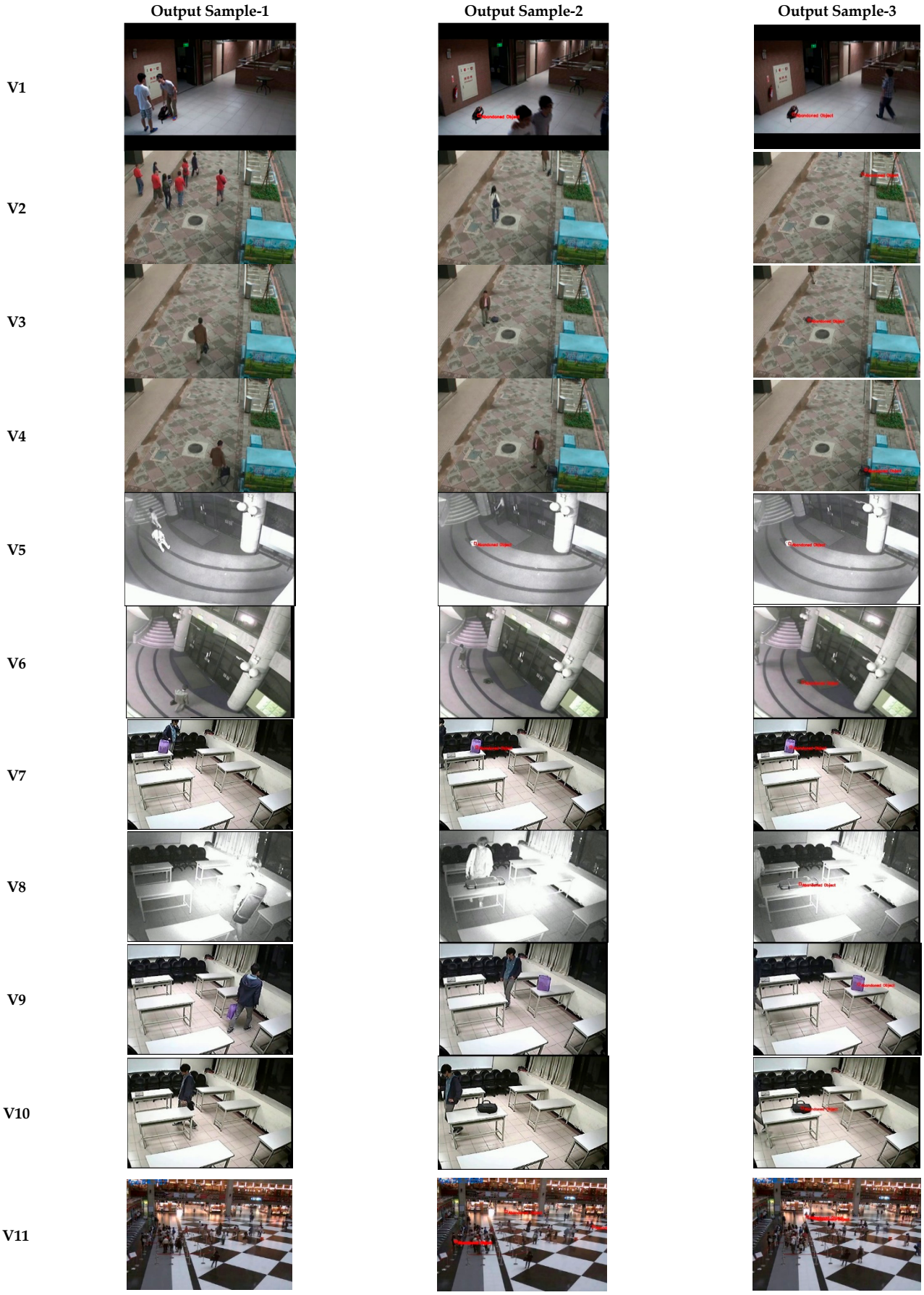


Figure 14. Proposed method outputs.

Table 9. Literature method comparison.

Ref.	Dataset	Method	*p* (%)	*r* (%)	*f*1 (%)
[15]	ABODA	Dual background model YOLO-NAS	75.00	75.00	75.00
[18]	ABODA	Background subtraction	35.41	70.83	-
[19]	Self-collected dataset	YOLOv4 + Kalman Filter (KF)	79.00	86.00	83.00
[20]	i-Lids AVSS-AB data	YOLOv3	95.50	56.40	-
Our Method	ABODA	Keyframe detection, YOLOv8l + Deep SORT	97.90	97.00	97.40

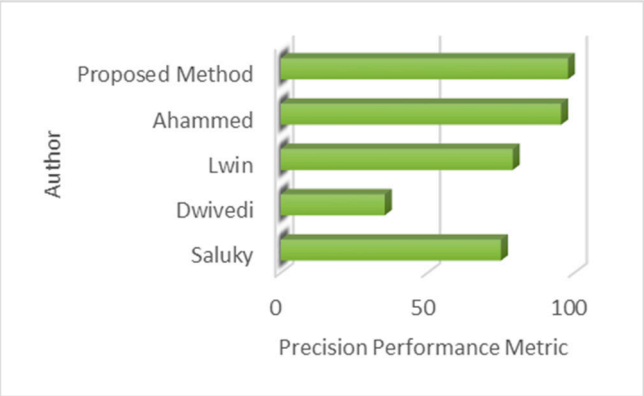


Figure 15. Proposed method precision comparison.

In addition, the proposed work is compared with various studies in terms of accuracy, speed, and computational efficiency. The USD (Unknown Sensitive Detector) [43] model can detect unknown objects with high accuracy, especially with SAM integration. However, it requires high GPU usage and a long processing time in large visual models. Practical Abandoned Object Detection [22] reduces the false positive rate with background blurring and a dense ASSP, but causes heavy computational delays in real-time applications. The Real-Time Deep Learning Method [44] is a fast and real-time method due to the use of a CNN. However, accuracy losses are experienced compared to new state-of-the-art techniques. In this study, keyframe detection increases speed and efficiency by preventing unnecessary frames from being processed. YOLOv8 provides fast and high-accuracy object detection. DeepSORT reduces false positives by providing real-time object tracking. The LLM-based explanation provides meaningful explanations of detected objects. Since algorithms such as YOLOv8 and DeepSORT perform an image-based analysis, they can increase computational power and cause an inefficient use of hardware. Instead of processing all the frames, processing only frames that will contribute to the learning of the model reduces the load on the CPU and GPU. It also reduces RAM consumption since unnecessary data are not stored. In this study, 50% processing power/hardware efficiency was achieved by performing keyframe detection before object detection.

5. Conclusions

In recent years, video surveillance systems have been used in various public areas, such as hospitals, airports, bus terminals, and university campuses, to ensure public safety. Closed circuit television (CCTV) systems allow the recorded images to be transferred to consecutive monitors and monitored by expert personnel. It takes a long time to detect abnormal situations by monitoring videos from multiple monitors at the same time, and situations that need to be detected may escape the attention of the staff. For this reason, smart systems have begun to be used to automate video inspections. In this study, a fast, reliable, and accurate abandoned object detection method for video surveillance is

presented using YOLOv8 and DeepSort. The same scenes can be found consecutively in continuously recorded videos. Using repeated frames in model training may result in memorization in the model and also increases the hardware cost. Therefore, in this study, deep transfer learning-based keyframe detection was first performed to remove repeated frames in the videos. Human and object classes in videos trained with the YOLOv8 deep neural network were detected using the extracted keyframes. Detected human and object classes were tracked in consecutive frames with the DeepSort algorithm and the distance between them was measured to detect abandoned objects. When compared with other methods proposed in the relevant field, it has been observed that the method proposed in this study can perform real-time detection with fast and high accuracy rates. A new method has been developed for the detection of abandoned objects, created with state-of-the-art technologies, robust even in unexpected situations, adaptive under different conditions, versatile enough to perform different functions, and scalable for datasets of different sizes. Future work will focus on the hybrid use of different deep learning architectures on larger datasets, especially those containing crowded images, such as sports competitions.

Author Contributions: Software, M.Y.; Validation, M.K.; Writing—original draft, M.Y.; Writing—review & editing, M.Y. and M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the TUBITAK (The Scientific and Technological Research Council of Turkey) under Grant No: 5220154. This study was supported by the Firat University Scientific Research Projects Support Program (FUBAP) under Grant No: MF.24.18.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: Thanks to Fatih Mert for his contributions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wonghabut, P.; Kumphong, J.; Satiennam, T.; Ung-Arunyawee, R.; Leelapatra, W. Automatic helmet-wearing detection for law enforcement using CCTV cameras. *IOP Conf. Ser. Earth Environ. Sci.* **2018**, *143*, 012063. [CrossRef]
2. Berardini, D.; Migliorelli, L.; Galdelli, A.; Frontoni, E.; Mancini, A.; Moccia, S. A deep-learning framework running on edge devices for handgun and knife detection from indoor video-surveillance cameras. *Multimed. Tools Appl.* **2024**, *83*, 19109–19127. [CrossRef]
3. Cohen, N.; Gattuso, J.; MacLennan-Brown, K. *CCTV Operational Requirements Manual 2009*; Home Office Scientific Development Branch: St. Albans, UK, 2009.
4. Tripathi, R.K.; Jalal, A.S.; Agrawal, S.C. Suspicious human activity recognition: A review. *Artif. Intell. Rev.* **2018**, *50*, 283–339. [CrossRef]
5. Croitoru, F.A.; Ristea, N.C.; Dăscălescu, D.; Ionescu, R.T.; Khan, F.S.; Shah, M. Lightning fast video anomaly detection via multi-scale adversarial distillation. *Comput. Vis. Image Underst.* **2024**, *247*, 104074. [CrossRef]
6. Wang, H.; Wang, W.; Liu, J. Temporal memory attention for video semantic segmentation. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2254–2258. [CrossRef]
7. Jha, S.; Seo, C.; Yang, E.; Joshi, G.P. Real time object detection and trackingsystem for video surveillance system. *Multimed. Tools Appl.* **2021**, *80*, 3981–3996. [CrossRef]
8. Khan, S.; AlSuwaidan, L. Agricultural monitoring system in video surveillance object detection using feature extraction and classification by deep learning techniques. *Comput. Electr. Eng.* **2022**, *102*, 108201. [CrossRef]
9. Yilmazer, M.; Karakose, M.; Tanberk, S.; Arslan, S. A New Approach to Detecting Free Parking Spaces Based on YOLOv6 and Keyframe Detection with Video Analytics. In Proceedings of the 2024 28th International Conference on Information Technology (IT), Zabljak, Montenegro, 21–24 February 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–4.

10. Yilmazer, M.; Karakose, M. Mask R-CNN architecture based railway fastener fault detection approach. In Proceedings of the 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 23–25 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1363–1366. [CrossRef]
11. Yapıcı, M.M.; Tekerek, A.; Topaloğlu, N. Literature review of deep learning research areas. *Gazi J. Eng. Sci.* **2019**, *5*, 188–215. [CrossRef]
12. Ucar, A.; Karakose, M.; Kırımça, N. Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends. *Appl. Sci.* **2024**, *14*, 898. [CrossRef]
13. Park, H.; Park, S.; Joo, Y. Detection of abandoned and stolen objects based on dual background model and mask R-CNN. *IEEE Access* **2020**, *8*, 80010–80019. [CrossRef]
14. Ferariu, L.; Chile, C.F. Fusing Faster R-CNN and Background Subtraction Based on the Mixture of Gaussians Model. In Proceedings of the 2020 24th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 8–10 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 367–372. [CrossRef]
15. Girshick, R. Fast r-cnn. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015. [CrossRef]
16. Saluky, S.; Nugraha, G.B.; Supangkat, S.H. Enhancing Abandoned Object Detection with Dual Background Models and Yolo-NAS. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 547–554.
17. Qasim, A.M.; Abbas, N.; Ali, A.; Al-Ghamdi, B.A.A.R. Abandoned Object Detection and Classification Using Deep Embedded Vision. *IEEE Access* **2024**, *12*, 35539–35551. [CrossRef]
18. Li, G.; Ji, J.; Qin, M.; Niu, W.; Ren, B.; Afghah, F.; Ma, X. Towards high-quality and efficient video super-resolution via spatial-temporal data overfitting. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 10259–10269. [CrossRef]
19. Dwivedi, N.; Singh, D.K.; Kushwaha, D.S. An approach for unattended object detection through contour formation using background subtraction. *Procedia Comput. Sci.* **2020**, *171*, 1979–1988. [CrossRef]
20. Lwin, S.P.; Tun, M.T. Deep convonlutional neural network for abandoned object detection. *Int. Res. J. Mod. Eng. Technol. Sci.* **2022**, *4*, 1549–1553.
21. Ahammed, M.T.; Ghosh, S.; Ashik, M.A.R. Human and Object Detection using Machine Learning Algorithm. In Proceedings of the 2022 Trends in Electrical, Electronics, Computer Engineering Conference (TEECCON), Bengaluru, India, 26–27 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 39–44. [CrossRef]
22. Jeong, M.; Kim, D.; Paik, J. Practical Abandoned Object Detection in Real-World Scenarios: Enhancements Using Background Matting with Dense ASPP. *IEEE Access* **2024**, *12*, 60808–60825. [CrossRef]
23. Preetha, K.G. A fuzzy rule-based abandoned object detection using image fusion for intelligent video surveillance systems. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 3694–3702. [CrossRef]
24. Melegrito, M.P.; Alon, A.S.; Militante, S.V.; Austria, Y.D.; Polinar, M.J.; Mirabueno, M.C.A. Abandoned-cart-vision: Abandoned cart detection using a deep object detection approach in a shopping parking space. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 13–15 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5. [CrossRef]
25. Park, H.; Park, S.; Joo, Y. Robust detection of abandoned object for smart video surveillance in illumination changes. *Sensors* **2019**, *19*, 5114. [CrossRef]
26. Russel, N.S.; Selvaraj, A. Ownership of abandoned object detection by integrating carried object recognition and context sensing. *Vis. Comput.* **2023**, *40*, 4401–4426. [CrossRef]
27. Teja, Y.D. Static object detection for video surveillance. *Multimed. Tools Appl.* **2023**, *82*, 21627–21639. [CrossRef]
28. Dubey, P.; Mittan, R.K. A Critical Study on Suspicious Object Detection with Images and Videos Using Machine Learning Techniques. *SN Comput. Sci.* **2024**, *5*, 505. [CrossRef]
29. Dhevanandhini, G.; Yamuna, G. An optimal intelligent video surveillance system in object detection using hybrid deep learning techniques. *Multimed. Tools Appl.* **2024**, *83*, 44299–44332. [CrossRef]
30. Nalgirkar, S.; Sharma, D.K.; Chandere, S.L.; Sasar, R.K.; Kasurde, G.N. Next-Gen Security Monitoring: Advanced Machine Learning for Intelligent Object Detection and Assessment in Surveillance. *Int. J. Multidiscip. Innov. Res. Methodol.* **2023**, *2*, 6–13.
31. Lin, K.; Chen, S.C.; Chen, C.S.; Lin, D.T.; Hung, Y.P. Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1359–1370. [CrossRef]
32. Amin, F.; Mondal, A.; Mathew, J. A large dataset with a new framework for abandoned object detection in complex scenarios. *IEEE Multimed.* **2021**, *28*, 75–87. [CrossRef]
33. Wahyono; Pulungan, R.; Jo, K.H. Stationary object detection for vision-based smart monitoring system. In *Intelligent Information and Database Systems, Proceedings of the 10th Asian Conference, ACIIDS 2018, Dong Hoi, Vietnam, 19–21 March 2018*; Proceedings, Part II 10; Springer: Cham, Switzerland, 2018; pp. 583–593. [CrossRef]

34. Abdulhussain, S.H.; Ramli, A.R.; Saripan, M.I.; Mahmmod, B.M.; Al-Haddad, S.A.R.; Jassim, W.A. Methods and challenges in shot boundary detection: A review. *Entropy* **2018**, *20*, 214. [CrossRef] [PubMed]
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
36. Sadiq, B.O.; Muhammad, B.; Abdullahi, M.N.; Onuh, G.; Muhammed, A.A.; Babatunde, A.E. Keyframe extraction techniques: A review. *ELEKTRIKA-J. Electr. Eng.* **2020**, *19*, 54–60.
37. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
38. Vats, A.; Anastasiu, D.C. Enhancing retail checkout through video inpainting, yolov8 detection, and deepsort tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5530–5537. [CrossRef]
39. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3645–3649. [CrossRef]
40. Vaswani, A. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
41. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [CrossRef]
42. Adu, G. Artificial Intelligence in Software Testing: Test Scenario and Case Generation with an AI Model (GPT-3.5-Turbo) Using Prompt Engineering, Fine-Tuning and Retrieval Augmented Generation Techniques. Master's Thesis, University of Eastern Finland, Kuopio, Finland, 2024.
43. He, Y.; Chen, W.; Tan, Y.; Wang, S. Usd: Unknown sensitive detector empowered by decoupled objectness and segment anything model. *arXiv* **2023**, arXiv:2306.02275.
44. Smeureanu, S.; Ionescu, R.T. Real-Time Deep Learning Method for Abandoned Luggage Detection in Video. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018.