Loitering Based Human Crime Detection in Video Surveillance using Beluga Whale Adam Dingo Optimizer and Deep Convolutional Neural Network

Nischita Waddenkery
Department of Computer Science and Engineering,
VIT University, India
nischita.waddenkery@gmail.com

Shridevi Soma
Department of Computer Science and Engineering,
VIT University, India
shridevisoma@pdaengg.com

Abstract: *Video Surveillance (VS) systems play a crucial role in maintaining security in public spaces, commercial establishments and residential areas. Detecting and preventing human-related crimes within the footage captured by these systems is a challenging task. Traditionally, VS systems rely on basic motion detection, which often leads to false alarms and inefficient use of resources. Loitering, a behavior frequently associated with criminal activities, requires more nuanced detection to reduce false positives and improve response times. Accurate tracking of individuals, especially in crowded environments, is another challenge. The chief objective of this research is to address these challenges by introducing an innovative Loitering-based Human Crime Detection (LHCD) module in VS. This module combines enhanced euclidean based Deep Simple Online Real-time Tracking (DSORT) with the Segmentation Quality Assessment (SQA) algorithm to accurately assess human travel distances. Also, this research integrates the Beluga Whale Adam Dingo Optimizer (BWADO) and a Deep Convolutional Neural Network (DCNN) to boost the precision and efficiency of Human Crime Detection (HCD) within loitering areas. The introduced approach demonstrates the effectiveness of introduced module, which reduces false alarms and enhances response times in VS. Outcomes demonstrate that the introduced approach outperforms existing approaches in various performance measures like accuracy (99.76%), F1-score (99.89%), recall (98.59%), precision (98.9%) and processing time (1.78s) demonstrating its superior effectiveness and potential for advancements in the field.*

Keywords: *Video surveillance, loitering behavior, human crime detection, enhanced euclidean distance, deep learning.*

Received October 18, 2023; accepted March 14, 2024
https://doi.org/10.34028/iajit/21/3/1

1. Introduction

Video Surveillance (VS) has become an essential part of modern society, playing an essential part in ensuring public safety, securing critical infrastructure, and deterring criminal activities. It serves as a vigilant eye that tirelessly monitors public spaces, residential areas, transportation hubs, and commercial establishments [13]. In an era where the proliferation of cameras is pervasive, the potential for VS to aid in crime detection is immense. However, amidst the vast streams of visual data captured by surveillance cameras, the efficient and accurate identification of criminal activities remains a formidable challenge.

Human Crime Detection (HCD) in VS is a vital application of modern technology that contributes significantly to public safety and refuge. It involves the use of surveillance cameras and advanced computer vision techniques to monitor and analyze video feeds in real-time or post-event to identify criminal activities and individuals involved in unlawful actions. This technology has revolutionized law enforcement, providing a vigilant and unblinking eye in various environments, including public spaces, transportation hubs, residential areas, and commercial establishments [10].

Moreover, one promising avenue of research and development in VS is Loitering-based Human Crime Detection (LHCD). LHCD focuses on identifying individuals or objects that linger or remain in a specific area for a protracted period without a clear purpose. This behavior is often associated with pre-criminal activities, like scouting a target or preparing for a theft. By targeting loitering, it becomes possible to detect and prevent crimes before they escalate, making public spaces safer and more secure [9, 11, 29, 35].

Recognizing the significance of enhancing crime detection in VS, researchers and technologists have been tirelessly working to develop novel approaches and algorithms to address the challenges posed by modern surveillance systems. For instance, the integration of artificial intelligence [26] and Deep Learning (DL) [19] techniques, like Convolutional Neural Networks (CNNs) [31] and Recurrent Neural Networks (RNNs) [15, 18], has significantly improved crime detection accuracy.

However, these DL models often demand substantial computational resources and extensive labeled datasets for training, making them computationally expensive and resource-intensive. Moreover, the decision-making processes lack transparency, which hinders the ability to interpret why certain predictions are made. Another promising approach, which employs object detection and tracking algorithms like You Only Look Once (YOLO) [14] and faster region-based CNN [30], excels at identifying specific objects or individuals across video frames.

Nevertheless, these algorithms face challenges when tracking objects in complex environments with occlusions, rapid movements, or crowded scenes, potentially leading to real-time performance issues and delays in crime detection. Moreover, crowd-sourced video analysis platforms like Citizen and Next-door, which involve the public in crime detection, have ethical and privacy concerns. Relying on unverified user-generated content introduces inaccuracies and biases in reporting, potentially resulting in false allegations or unwarranted panic within communities [25].

Despite many advantages of LHCD in VS, challenges like data overload, false alarms, privacy concerns, and adaptive criminals necessitate ongoing research and innovation to enhance the accuracy and efficiency of these systems. Lastly, criminals are becoming increasingly practical in circumventing traditional surveillance techniques, necessitating the development of more sophisticated and adaptive solutions to keep pace with the evolving tactics. Addressing these challenges is paramount to harnessing the full potential of VS for crime detection and ensuring the security and privacy of individuals and communities [1, 4]. This research aims to introduce a novel technique for LHCD in VS. This technique leverages advanced computer vision and DL methods to analyze video data and identify instances of suspicious loitering behavior. By combining the power of modern technology with the insights gained from analyzing human behavior, the effectiveness of VS is enhanced in introduced crime detection. The major contributions of this research work are listed below.

- **Contributions**
  - The research introduces a novel LHCD module in VS, which is a pioneering effort in the field. It combines Deep Simple Online Real-time Tracking (DSORT) algorithm with the Segmentation Quality Assessment (SQA) algorithm and employs the Enhanced Euclidean Distance (EED) for precise calculation of human travel distances. Additionally, this module distinguishes between loitering and non-loitering behavior by setting individualized thresholds, enhancing the system's ability to rapidly respond to potential security threats.
  - The loitering behaviour detection event is done with the innovative Beluga Whale Adam Dingo Optimizer (BWADO) with Deep Convolutional Neural Network (DCNN). This unique integration of modules represents a significant contribution to the area of VS, allowing for the focused identification of criminal activity within only the loitering behavior area. It's vital to note that this work is the first to combine these two modules, showcasing its pioneering nature.
  - Beluga Whale Optimization Algorithm (BWOA)'s global exploration capabilities helps avoid getting stuck in local optima, while Adam Dingo Optimization Algorithm (ADOA)'s local optimization fine-tunes the solutions. By combining them, the strengths of both algorithms are harnessed to potentially achieve better convergence and solution quality.
- **Organization**

Section 2 elaborates the review on traditional approaches. Section 3 explains the introduced topology. Section 4 showcases the experimental outcomes and assessment. Section 5 provides a conclusion.

2. Review

Extensive literature has been explored in the past, focusing on crime detection within VS systems, employing various approaches and techniques. This section provides a comprehensive review of the findings and results from recent research efforts in this field.

2.1. Crime Detection

A multiscale information aggregation forecast model based on VS was designed by Li *et al*. [20] to predict and analyze short-term time series with limited data, like criminal activity. Primarily, a multi-scale human traffic insight and quantification model were recognized to attain human traffic time series data, which was essential for strengthening the chief attributes of succeeding short-term time series analysis. Then, the Anti Saturation Gate Control Model (ASGCM) for prediction was suggested, which included entry-level approaches and incorporated anti-saturation conversion systems. These modifications made ASGCM more complex, reduced its reliance on long-term features and addressed issues related to gradient disappearance and exploration problems. Nevertheless, it should be noted that this technique has a limitation in that it concentrates solely on tracking the overall crime trend without delving deeper into individual cases. Qasim and Verdu [28] employed a combination of a DCNN and a Convolutional Gate Recurrent Unit (ConvGRU) to construct an automated system designed for identifying anomalies in videos. The architecture utilized ResNet to extract top-level attribute representations from incoming video frames, while the Gate Recurrent Unit (GRU) was responsible for capturing temporal attributes. The GRU, known for its sensitive recurrence and efficient parallelization, had enhanced the accuracy of the video irregularity detection model. However, it was noted that the system had not achieved precise anomaly detection in all cases. An intelligent VS system that optimized memory usage for effective surveillance was developed by Biswas *et al*. [7]. This system recorded high-resolution video during suspicious movements or instances of violence and switched to low-resolution when activity was normal. Suspicious movements in consecutive frames were detected, important frames were saved in high resolution and less crucial ones were discarded. Additionally, Contrast Limited Adaptive Histogram Equalization (CLAHE) with respect to the Color Channel was utilizes to enhance image contrast and improve object visibility in suspicious frames. The enhanced frame quality was evaluated using two no-reference methods: No-reference Image Quality Metric for Contrast distortion (NIQMC) and Blind Image Quality Measure of Enhanced images (BIQME). However, it should be noted that this method was not effective in motion-based object detection. A 7-layered Javeria DCNN with specific hyperparameters denoted as J.DCNN was employed by Amin *et al*. [6] to analyze abnormal behavior within video segments. Additionally, a model was developed by them, combining Javeria Quantum and CNN (J.QCNN), to conduct an inclusive analysis of irregular video frames. This approach featured a 4-qubit quantum network with five layers and an optimized loss function known as J.QCNN. Notably, the suggested J.QCNN approach possessed unique characteristics not present in conventional DL architectures. Both of these systems were trained from scratch to detect anomalies in few of the most challenging publicly available VS datasets. Waddenkery and Soma [34] developed a hybrid optimized DL approach for detecting theft crimes in video footage obtained from surveillance cameras. Initially, the videos were concised by the video summarization approach. Afterwards, the summarized video data was input into a deep maxout network. The weights of this network were updated using the ADOA to identify theft crime events as well as normal events. If the video summarization process omits important information, it potentially leads to reduced accuracy in identifying theft crime events or normal events. Bogus accident video frames were created from usual traffic footage by Zahid *et al*. [37] maintaining scene consistency. Pre-trained DCNNs were fine-tuned on these fake frames to detect real accidents. Four models, including alexnet, googlenet, squeezenet and ResNet-50, were employed on both regular and irregular traffic frames. However, a drawback was identified in the sole reliance on spatial data. Kamoona *et al*. [16] developed a weakly supervised Deep Temporal Encoding-Decoding (DTED) approach for detecting anomalies in VS by Multiple Instance Learning (MIL). This approach incorporated both irregular and normal video clips at the training stage, organized within the MIL approach. This method employed a DTED network to capture the spatio-temporal evolution of video instances over time. Additionally, a novel loss function was suggested to maximize the mean distance among predictions for usual and irregular instances.

2.2. Behavior Detection

A novel approach was developed by Nazir *et al*. [24] to prevent shoplifting through the detection of suspicious behavior. CNN were employed to extract spatial features from pixel values. In contrast, this approach involved the adoption of object detection with You Only Live Once Version5 (YOLOV5) and DSORT to track individuals in video footage, utilizing resulting bounding box coordinates as temporal features. However, challenges were identified in accurately tracking individuals using bounding box coordinates in crowded or complex environments, potentially leading to tracking errors or false alarms in detecting suspicious behavior. An innovative approach called dynamic frame-skipping was developed by Mumtaz *et al*. [23] to create meaningful temporal systems for model learning. Additionally, a new DL model, based on the Inflated 3D-convnets (I3D) model, was developed to arrest both spatial and time-based data from video frames. Pouyan *et al*. [27] developed three detection sections: head shelter detection, crowd detection and loitering behaviour detection, with the goal of facilitating appropriate actions and preventing burglary. The first two sections underwent retraining of the YOLOV5 model using manually annotated datasets. Furthermore, loitering detection was done based on the DSORT algorithm. A fuzzy inference machine was employed to incorporate expert knowledge in the form of rules, assisting in making the final decisions regarding predicted robbery potential. However, this method encountered limitations in tracking low-resolution human video images. Ahmed and Yousaf [5] addressed the challenge of detecting suspicious activities in surveillance videos using the CNN with autoencoder. The attributes were extracted using a 3-Dimensional Convolutional neural network (C3D) and fed them into the suggested autoencoder framework. This framework identified activity localization by analyzing high reconstruction loss. Lower reconstruction loss was observed for normal video clips, while video clips with suspicious actions exhibited higher reconstruction loss. The suspicious clips were also extracted from lengthy tailing videos and used them to categorize diverse suspicious activities with the suggested Generative Adversarial Network (GAN). However, implementing this technique in real-time offered difficulties. Anomaly Detection assisted by Graph neural network (AD-Graph) based framework for video anomaly detection, was developed by Ullah *et al*. [33]. It aimed to capture temporal information from frames by extracting 3D visual and motion features, organizing them into a knowledge graph format. The framework employed robust clustering to group similar graph neighborhoods and applied spectral filters using spectral graph theory for anomaly detection. However, this model faced challenges in scenarios with minimum-resolution images, minimum illumination, fast motion and crowded scenes. Table 1 depicts a summary of discussed conventional approaches.

Table 1. Comparison of traditional detection models.

Ref. no	Techniques used	Commonalities	Differences	Limitations
[20]	ASGCM	Short-term time series prediction	Concentrates on overall crime trend, not individual cases	Limited depth in individual case analysis
[28]	DCNN, ConvGRU, ResNet	Anomaly detection in videos	Utilizes ResNet and GRU for spatial and temporal attributes	Challenges in precise anomaly detection in all cases
[7]	CLAHE	Optimized memory usage for surveillance	High-low resolution switching, CLAHE for image enhancement	Ineffective in motion-based object detection
[6]	J.DCNN, J.QCNN	Analyzing abnormal behavior in video segments	Javeria QCNN combines quantum and CNN for inclusive analysis	Effectiveness on challenging datasets not specified
[34]	Hybrid DL, ADOA	Theft crime detection in surveillance footage	ADOA for weight updates, Deep maxout Network, Video summarization	Potential accuracy reduction if video summarization omits important information
[37]	DCNN	Real accident detection from normal traffic footage	Fine-tuning pre-trained DCNNs (alex net, google net, squeeze net, ResNet-50) on fake frames	Sole reliance on spatial data identified as a drawback
[16]	DTED, MIL	Anomaly detection in VS using MIL	DTED network, Novel loss function	Challenges in scenarios with minimum-resolution images, minimum illumination, fast motion, and crowded scenes
[24]	YOLOV5, DSORT	Preventing shoplifting through suspicious behavior	NN for spatial features, YOLOV5 and DSORT for object detection and tracking	Challenges in tracking individuals accurately in crowded or complex environments
[23]	I3D	Temporal systems for model learning	Dynamic frame-skipping, DL model based on I3D for spatial and temporal data from video frames	Method for tracking low-resolution human video images not specified
[27]	YOLOV5, DSORT, Fuzzy Inference Machine	Head cover, crowd, and loitering behavior detection	Sections retrained using YOLOV5, loitering detection based on DSORT, Fuzzy Inference Machine for decision-making	Limitations in tracking low-resolution human video images
[5]	CNN Autoencoder, C3D, GAN	Detecting suspicious activities in surveillance videos	CNN autoencoder approach, C3D for feature extraction, GAN for categorizing suspicious activities	Difficulties in implementing the technique in real-time
[33]	AD-Graph	Video anomaly detection	Captures temporal information with 3D visual and motion features, Knowledge graph, Spectral graph theory	Challenges in scenarios with minimum-resolution images, minimum illumination, fast motion, and crowded scenes

2.3. Problem Statement

The research gap in the context of LHCD lies in the need for more accurate, scalable and adaptable methods that efficiently detects criminal activities in diverse surveillance environments while ensuring real-time performance. Current approaches often struggle with limitations in accuracy, scalability and adaptability to varying conditions, posing challenges for seamless integration into existing systems and benchmarking for consistent evaluation. Bridging this research gap involves developing innovative techniques that address these shortcomings and strike a balance between effective crime detection and privacy considerations, ultimately advancing the field of VS based crime prevention.

3. Introduced Topology

The introduced method comprises two main phases: loitering behaviour detection and HCD modules. Figure 1 explains the introduced flow diagram.

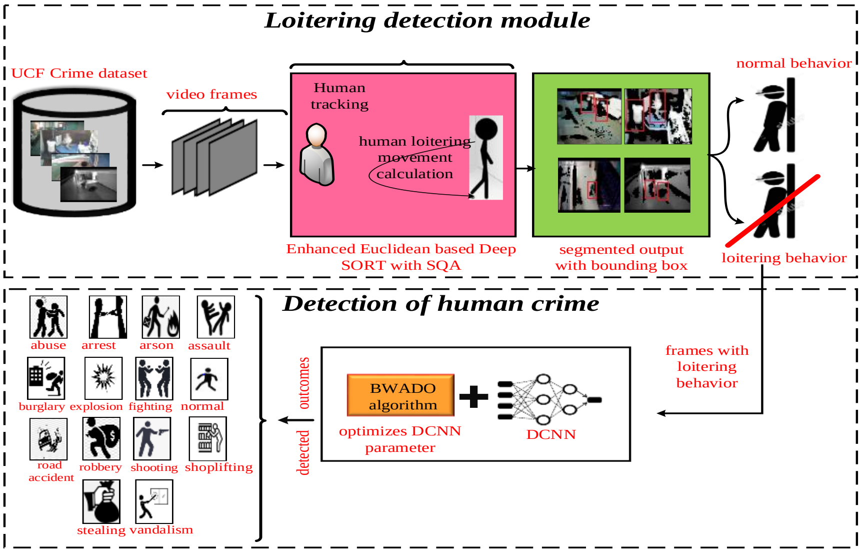


Figure 1. Flow diagram of introduced model.

- **Loitering Detection Module**

In this module, the objective is to swiftly identify potential security threats, thereby preventing crimes from occurring or escalating. This is achieved by tracking individuals within the video frame using the DSORT algorithm combined with SQA. To measure the distance traveled by humans, an EED calculation is employed. Subsequently, each person is categorized as exhibiting either loitering or non-loitering behavior based on predefined individual thresholds.

- **Human Crime Detection Module**

In the HCD module, once a loitering event is identified, the system concentrates exclusively on the region associated with loitering behavior within the video feed. This focused approach aims to capture additional evidence related to criminal activities. To facilitate HCD, the BWADO is introduced in combination with a DCNN. At last, data based on crime detection is sent to control room.

3.1. Data Acquisition

The input data for this research is gathered from the University of Central Florida (UCF) crime dataset [32]. This comprises images derived from the UCF Crime Dataset videos, emphasizing criminal and normal activities. Extracted from every 10th frame of each full-length video, the dataset includes a total of 14 classes like arson, assault, burglary, explosion, robbery, vandalism, fighting, abuse, stealing, shoplifting, road accidents, arrest, shooting and normal videos, encompassing diverse behaviors such as abuse, arson, and normal scenarios. All images are uniformly resized to 64x64 pixels and stored in the .png format, ensuring consistent processing. The dataset is divided into training and testing subsets, with the training subset containing 1,266,345 images and the test subset consisting of 111,308 images which is used for tasks like activity recognition and video analysis, training DL models to discern criminal activities from routine occurrences.

Assume the database *D* represented by Equation (1) stores the videos and denoted as a set containing multiple video elements *V*. *n* signifies the total number of videos present in the dataset *D*. In other words, it's the cardinality of the set *D*.

$$D = \{V_1, V_2, \dots, V_n\} \tag{1}$$

where, V_1 and V_2 represents the first and second videos of the dataset. Similarly, V_n illustrates the overall number of videos present in D^{th} repository.

3.2. Preprocessing

To alleviate the computational complexity, a preprocessing was employed to handle videos within the tabular dataset that exceeded 15,000 rows. A decision was made to partition these films into multiple clips, each comprising a maximum of 15,000 rows. As a result of this process, a total of 544 distinct clips were extracted from the initial set of 317 videos. This preprocessing streamlined the computational burden and facilitated more efficient processing and analysis of the dataset, ensuring optimal performance and resource utilization.

3.3. Detection of Loitering Behaviors

In a wide range of situations or settings, individuals engage in activities that varies significantly. People often explore their environment or have specific goals and targets in mind. Generally, people tend to move towards their intended destinations or objectives within the given contexts. This include completing a purchase at a store, reaching a specific location, or achieving a goal associated with their activities. However, in certain situations, individuals with harmful or malicious intentions displays unique behavioral patterns. These individuals act in a way that raises suspicion or poses a potential threat. One notable behavior that is exhibited by individuals with malicious intent is heightened loitering behavior. This involves lingering in a certain area for an extended period. Surveillance cameras are commonly used to monitor various areas. They often focus on critical or high-traffic areas like cash counters, entrances, or accident scenes. These areas are considered important points of observation because they are more likely to capture relevant activities and behavior. To measure the loitering degree, every person in the preprocessed dataset is tracked using the DSORT with SQA algorithm. The total distance traveled by every person is then calculated by EED every 10 frames within a 500-frame snippet. Loitering is determined by aggregating these distances. Notably, changes in people's positions become noticeable after about 10 frames. Therefore, tracking the distance traveled begins after this point, with updates occurring at every one-step interval within the 500-frame window. It's important to mention that the cameras used for data collection are uncalibrated, so the algorithm provides relative distances. Additionally, due to limited movement within each 10-frame interval, distance and displacement are treated as equal.

- **SQA with deep SORT for Tracking Humans**

In the phase of bounding box segmentation based human tracking, the input *D* is subjected, where the significant regions from the video are easily segmented using SQA.

The process begins by dividing the image into multiple regions in the primary step, employing a predefined set of segmentation parameters. Subsequently, the introduced SQA network as depicted in Figure 2 is applied in the second step to evaluate the quality of these segmentation outcomes, assigning each result a corresponding score. In the next step, the segmentation outcomes are organized into an ordered array based on their quality scores. The top-ranking result, representing the highest quality, is chosen as the ultimate segmentation outcome [21].

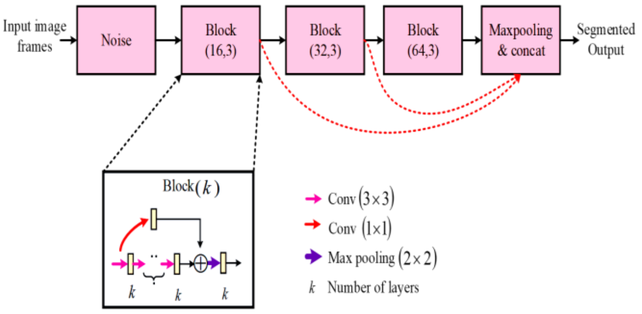


Figure 2. SQA architecture.

When provided with an image *V* containing a bounding box *b*, the initial step is to segment a collection of segmentation results using Equation (2).

$$S = seg(V, b, \phi) \tag{2}$$

where *s*={s$_1$, …, s$_m$} and *S* represents the final segmented outcomes and the bounding box-based segmentation. ϕ={ϕ_1,…, ϕ_m} represents the sets of parameter settings.

The segmentation process initiates by dissecting the image into multiple distinct regions, a crucial step achieved through the application of specific segmentation parameters. This approach undergoes multiple shifts to create a series of bounding boxes, with the shifting distance relative to the bounding box’s dimensions specifically, its height *h* and width *w* scaled by a factor α. This α value is chosen from the set {0, 0.01, 0.02, 0.05} randomly, resulting in ten segmentation outcomes ultimately. These parameters enable the algorithm to identify areas of similarity within the image, whether it be in terms of color, texture, or other visual attributes. By segmenting the image in this manner, the goal is to create meaningful partitions that correspond to different objects, surfaces, or elements present within the scene. The SQA network analyzes various aspects of each segmented region, such as its boundary coherence, internal homogeneity, and contextual relevance within the broader image context. Based on these criteria, it assigns a quality score to each segmented region, quantifying the degree to which it accurately represents a distinct object or feature in the scene.

To further enhance the quality assessment process and provide deeper insights into human activity within the scene, the integration of DSORT is done. DSORT is a tracking algorithm that monitors the movement and behavior of individuals within the image. By tracking the trajectories and interactions of people over time, DSORT enables the assessment of loitering degree, which refers to the extent of individuals lingering or spending prolonged periods within specific regions of interest. This integration enriches the quality assessment process by incorporating human-centric metrics, allowing for an additional comprehensive understanding of the scene dynamics beyond mere visual segmentation.

It primarily relies on the frame-by-frame information association technique and Kalman filtering. This approach is employed to evaluate the ongoing tracks within the present video frames. This typically involves tracking velocities (X’, Y’, H’, γ’) for every individual organize of the detected bounding box and positions (U, V, H, γ) of the bounding box.

As a consequence, the location of every individual current track in the present frame is estimated for the subsequent frame. This estimation relies on the spatial data of the bounding box. Additionally, to capture the presence characteristics of all detection and track, an attribute withdrawal process is conducted by an appearance descriptor. With the data taken out from the appearance descriptor, the original detection outcomes are linked to the current tracking outcomes in the subsequent frame. To accomplish this, a detection threshold is well-defined to filter out low-confidence detections. In the subsequent frame, every detection outcome is related by this threshold. The DSORT algorithm [3] utilizes a cost matrix to signify the appearance and spatial resemblances among the original detections and existing tracks. This is achieved through two distinct distance values by Equation (3).

$$D_{(i,j)} = (D_j - Y_i)^t s_i \; (D_j - Y_i) \tag{3}$$

where Y_i and s_i refers to the i^{th} projection track in measurement space and D_j is utilized for the j^{th} original detection.

In each new video frame, detections and tracks are linked using the described cost functions. In the video sequence, if a new detection effectively links with an existing track in the next frame, tracking proceeds. If not, the track is set to zero, signifying a failure in the new detections. In such cases, failed detections in frame *f* become tentative tracks. The DSORT algorithm continually verifies and associates them with original detections in subsequent (*f*+1), (*f*+2), …, (*f*+*t*) frames. Successful associations confirm and update the track, while failures result in immediate deletion. The role of the DSORT with SQA algorithm in human tracking involves assessing the quality of segmented regions containing individuals or groups of people within an image. The combined approach of integrating DSORT with SQA enables a holistic evaluation of both visual segmentation quality and human activity patterns within the scene.

- **Enhanced Euclidean Based Distance Calculation**

In the context of tracking people’s moving positions or trajectories, the EED is used to calculate the dissimilarity between two trajectories represented as probability distributions [36]. Below is a simplified equation for calculating the EED:

Figure 3 is the snippet allocation and loitering calculation process. Clearly, when a video of length *l* is available, it is divided into overlapping snippets denoted as *S*, each consisting of 10 frames. If an individual is observed loitering, their calculated travel distance must exceed a specific threshold value denoted as θ_n. Depending on the extent of their movement, every individual person is assigned a loitering score ranging from 0 to 100.

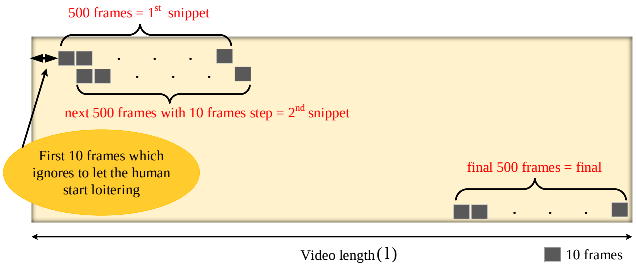

500 frames = 1st snippet
next 500 frames with 10 frames step = 2nd snippet
First 10 frames which ignores to let the human start loitering
final 500 frames = final
Video length (l)
10 frames

Figure 3. Division of videos into snippets: every 500 frames form a snippet and these snippets advance by one step within the video frames.

Calculate displacement D_j for an individual in one step (10 frames) by Equation (4):

$$D_j = \sqrt{(x_{i+10} - x_i)^2 - (y_{i+10} - y_i)^2} \tag{4}$$

where (x_0, y_0) is the initial position of the person in the first frame of the step, (x_n, y_n) is the final position of the person in the last frame of the step and i=10 j, j=0:(n-1).

To facilitate the calculation of displacement for individuals in a video divided into snippets, the Euclidean distance in equation is normalized by isolating it by the lowest value of the square root of the total squares of dual spatial vectors. However, it becomes incredible to measure an effective outcome when the present sampling value on one side is 0. To address this issue and improve the Euclidean distance calculation, a constant stability parameter α is added, as shown in Equation (5).

$$D_j = \frac{\sqrt{(|x_{i+10} - x_i| - |y_{i+10} - y_i|)}}{\min((x_{i+10} - x_i)^2, (y_{i+10} - y_i)^2) + \alpha} \tag{5}$$

Aggregate displacements for a snippet $D_{snippet}$ using Equation (6).

$$D_{snippet} = \sum_{i=M}^{50+M} D_j \quad M = 0 : (j - 50) \tag{6}$$

The above equation represents the sum of displacements for all individuals within the snippet.

The variable D_j represents the EED movement of an individual from the i^{th} frame to the $i+10^{th}$ frame within one snippet. Additionally variable t signifies the number of steps within a single snippet, which is set at 50 because 50 multiplied by 10 equals 500 frames. Moreover, x, y denotes the position of the head portion for every human and their movement over 10 frames are indicated by 'i' and '$i+10$'. $D_{snippet}$ represents the aggregation of these distances for each snippet, which consists of 500 frames (50 multiplied by 10). Once the calculation of $D_{snippet}$ for one snippet is complete, the video progresses with one step and the aggregation process is repeated. Repeat this process for each snippet to calculate the travelled distance during each snippet. This procedure is aimed at taking into account the variations in human behavior throughout the video.

A simple example is provided to illustrate Euclidean distance calculation is provided below.

Suppose there exists a video capturing a person walking down a hallway, and the interest lies in analyzing their movement.

- Frame 1(i=1): the person is at coordinates (x1, y1).
- Frame 11(i+10=11): the person is at coordinates (x11, y11).

The Euclidean distance D_j between these two points is calculated using Equation (5). This distance indicates the distance covered by the person from frame 1 to frame 11 within the snippet.

Now, consider a scenario where there are 50 frames in each snippet, and the video comprises a total of 500 frames. The video is divided into 50 snippets, each with 10 frames. Upon calculating the Euclidean distance for each snippet, the distances are gathered to provide a total measure of the person's movement throughout the entire video. This process is repeated for each snippet, with the snippet window sliding one step at a time through the video. This facilitates tracking changes in the person's movement pattern over time and accounting for variations in their behavior throughout the video. The units or dimensions of these displacement calculations are in pixels.

The rationale behind selecting a 500-frame snippet and the significance of every 10*th* frame lie in balancing the need for temporal granularity, practical considerations, and capturing behavioral variability. The 500-frame snippet allows for a suitable time window to observe meaningful motion patterns within individuals' trajectories while managing computational resources efficiently. Sampling every 10th frame within this snippet reduces redundancy and computational load while still capturing essential movement information at regular intervals. This approach enables the analysis to account for variations in human behavior over short time spans, providing a comprehensive understanding of movement patterns throughout the video. Overall, the choice of these parameters reflects a thoughtful balance between capturing sufficient temporal detail and practical constraints, ultimately enhancing the effectiveness of trajectory analysis.

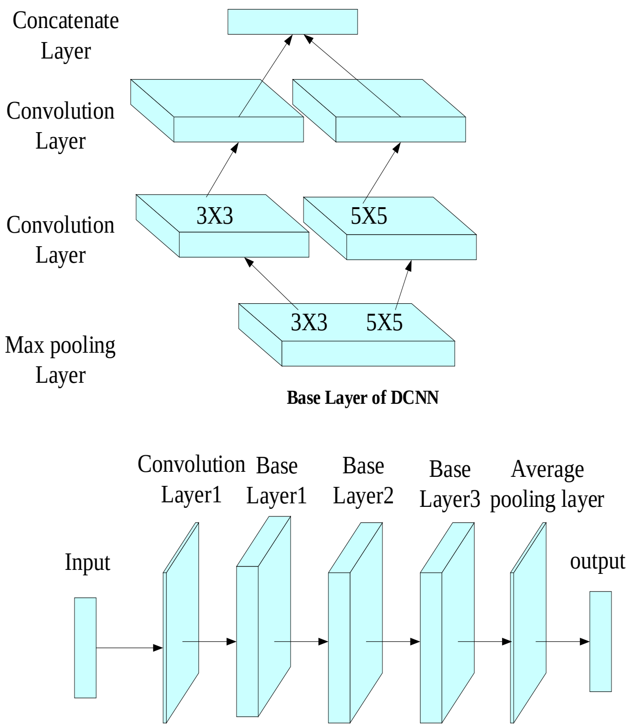
The combination of SQA with DSORT algorithm effectively discerns and categorizes human behavior as either loitering or non-loitering in VS applications. Furthermore, this approach specifically targets and flags instances of loitering, which are crucial for detecting potential criminal activities. By incorporating another DL technique focused on identifying human crimes in the context of loitering behavior, a comprehensive system is established to enhance security measures and aid in crime prevention.

3.4. Detection of Human Crime

In the realm of HCD, the pursuit begins by harnessing the potential of the BWADO integrated with a DCNN. This powerful combination aims to enhance the ability to identify and address criminal activities within frames previously labeled for loitering detection.

- **Deep Convolutional Neural Networks**

Figure 4 illustrates the structure of the DCNN network, which commences its initial convolutional layer with a specific 7×7 convolution kernel followed by a subsequent max-pooling layer. Following this, two convolution layers are employed utilizing both single and mixed 3×3 and 5×5 convolution kernels, succeeded by another max-pooling layer, resulting in a configuration comprising six stacked convolution layers and three max-pooling layers. The mixed convolution kernel aids in feature extraction across various sizes and reduces connection parameters between neurons. An examination of all feature maps from the third convolutional layer and the second max-pooling layer demonstrates the influence of the convolution kernel size and movement stride on the output size of the feature maps. In this study, a stride of 2 is implemented for the initial convolution layer and first max-pooling layer, while subsequent convolution and pooling layers maintain a stride of 1. To ensure consistency in output feature maps for convolution layers with 5×5 kernel, padding of size 2×2 is incorporated. Additionally, pooling operations are utilized to address the impact of feature map resolution and precise location, thus mitigating overfitting and maintaining the network's recognition efficacy.



Base Layer of DCNN

Figure 4. DCNN architecture.

DCNNs [8] organize detection concepts into two main layers pooling (sub-sampling) and convolution layers. The k^{th} feature map f_m, denoted as $f_{m}{}^{k}_{ij}$ is created using the $\tanh$ function with connection weights $w^k\,\lambda$ and biases $b^k\,\lambda$, as specified in Equation (7).

$$f_{m}{}^{k}_{ij} = \tanh((w^k \cdot x)_{ij} + w^k) \tag{7}$$

The sub-sampling layer creates spatial invariance by decreasing the f_m's resolution. Every f_m in this layer resembles to the one in the previous layer. Equation (8) outlines the method used to describe the pooling process.

$$\lambda_j = \tanh\left(W \sum_{M\cdot M} \lambda_i^{m\cdot m} + \omega\right) \tag{8}$$

where $\lambda_i^{m\cdot m}$ represents the inputs, W and ω are trainable scalar and bias, respectively. These parameters are needed to be tuned for enhancing the accuracy rates.

Table 2. Hyperparameter setting.

Parameters	Setting
Batch size	8
Learning rate	0.001
Epochs	100
Dropout	0.1

Table 2 represents the hyperparameters parameters setting of proposed network. Cross-validation techniques are used to assess the performance of different hyperparameter configurations on validation data, helping in the identification of optimal settings.

The Table 2 summarizes a set of hyperparameter settings for a machine learning model. It specifies a batch size of 8, a learning rate of 0.001, 100 training epochs, and a dropout rate of 0.1. These settings control the training process and influence the approach's performance and complexity.

- **Presentation of BWADO**

The BWADO is a hybrid algorithm introduced which combines the strengths of dual distinct optimization approaches: the BWOA and the ADOA for tuning the DCNN parameters. The BWOA [39] is part of a broader category of nature- inspired metaheuristic algorithm that mimics the social behaviors and foraging patterns of beluga whales. This algorithm is known for its global exploration capabilities, which aid in avoiding local optima. The ADOA [34] is a population-based algorithm inspired by dingo hunting behavior. It incorporates the Adam optimizer for efficient optimization, excelling in real-time problem-solving but faces challenges with weight decay and multi-objective issues. It excels in local exploitation to fine-tune solutions. By integrating these two algorithms, the BWADO aims to harness their complementary features, potentially achieving superior convergence and solution quality. This hybridization strategy is designed to benefit optimization tasks by balancing global exploration and local exploitation for enhanced performance and to optimize solutions effectively.

- **Initialization**

Initialize the DCNN's weight and bias parameters by equation for the populations for BWOA and ADOA, typically with random solutions for the optimization of weights and biases.

- **Solution Update Equation of BWOA**

In BWOA, each solution or individual in the population is represented as a vector. Let $x_i^{(t)}$ represent present solution of the i^{th} individual in the population at generation *t*. The solution update equation in BWOA involves two main components: the movement of individuals towards the leader (a dominant solution) and a random exploration factor which is demonstrated by below Equation (9).

$$x_i^{(t+1)} = x_i^{(t)} + \Delta x_i^{(t)} \tag{9}$$

where $x_i^{(t+1)}$ represents the updated solution or individual at generation *t+1*, $\Delta x_i^{(t)}$ denotes the change or movement of the individual towards the leader and incorporates both exploitation and exploration.

- **Solution Update Equation of ADOA**

The updated equation of ADOA is demonstrated by Equation (10).

$$y_i^{(t+1)} = y_i^{(t)} + \eta_i^{(t)} \cdot \left(\frac{m_i^{(t)}}{\sqrt{v_i^{(t)}}} + Er_i^{(t)} - Et_i^{(t)} \right) \tag{10}$$

where $y_i^{(t+1)}$ is the updated solution or individual at generation *t+1*, $y_i^{(t)}$ is the present solution at generation *t*, $\eta_i^{(t)}$ is a learning rate term that scales the update, typically adjusted dynamically during optimization. $m_i^{(t)}$ and $v_i^{(t)}$ represents the momentum and learning rate terms computed using Adam's mechanism. $Er_i^{(t)}$ and $Et_i^{(t)}$ are the exploration and exploitation terms.

- **Objective Function**

In the hybridization strategy, the tuning of weights and biases during the training of the DCNN involves a synergistic combination of the exploration capabilities of the BWOA and the local exploitation strengths of the ADOA. Throughout the training process, BWOA is employed to iteratively explore the weight space, leveraging its global exploration capabilities to navigate a wide solution space and adjust the weights accordingly. Simultaneously, ADOA is utilized to perform fine-tuned adjustments to the biases, focusing on local exploitation and precise tuning of bias parameters. The integration of these updates aims to capitalize on the complementary features of BWOA and ADOA by mitigating complexity, enhancing the adaptability of the DCNN by striking a balance between global exploration and local exploitation. This iterative hybridization process optimizes both the weight and bias parameters, contributing to an improved and more robust performance of the DCNN across diverse optimization landscapes.

The weights and biases of DCNN are characterized by $w^k\lambda$ and $b^k\lambda$, the updated weights and biases are denoted by $\Delta w^k\lambda$ and $\Delta b^k\lambda$ respectively. The objective function of weight and bias updation are determined by Equations (11) and (12).

- **Weight Update Using BWOA**

$$(w^{\mathrm{k}}\lambda)^{(t+1)} = (w^{\mathrm{k}}\lambda)^{(t)} + \Delta(w^{\mathrm{k}}\lambda)^{(t)} \tag{11}$$

where $\Delta(w^k\lambda)^{(t)}$ is computed using BWOA's update rules for weight optimization.

- **Bias Update Using ADOA**

$$(b^{\mathrm{k}}\lambda)^{(t+1)} = (b^{\mathrm{k}}\lambda)^{(t)} + \Delta(b^{\mathrm{k}}\lambda)^{(t)} + \eta_i^{(t)} \cdot \left(\frac{m_i^{(t)}}{\sqrt{v_i^{(t)}}} + Er_i^{(t)} - Et_i^{(t)} \right) \tag{12}$$

where $\Delta(b^k\lambda)^{(t)}$ is computed using BWOA's update rules for weight optimization.

- **Hybridization**

The updated weights and biases obtained from BWOA and ADOA are combined to produce new weights and biases by Equations (13) and (14).

$$(w^k\lambda)^{(t+1)} = \alpha \cdot (w^k\lambda^{(t+1,BWOA)} + (1-\alpha) \cdot (w^k\lambda)^{(t+1,ADOA)} \tag{13}$$

where $(w^k\lambda)^{(t+1,BWOA)}$ and $(w^k\lambda)^{(t+1,ADOA)}$ are the weight solutions obtained from *BWOA* and *ADOA*, respectively.

$$(b^k\lambda)^{(t+1)} = \beta \cdot (b^k\lambda^{(t+1,BWOA)} + (1-\beta) \cdot (b^k\lambda)^{(t+1,ADOA)} \tag{14}$$

where $(b^k\lambda)^{(t+1,BWOA)}$ and $(b^k\lambda)^{(t+1,ADOA)}$ are the bias solutions obtained from *BWOA* and *ADOA*, respectively. The α and β parameters are adjusted between 0 and 1 to control the balance between *BWOA* and *ADOA* for weights and biases, respectively. Thus, the DCNN's performance is evaluated with new weights and biases.

Hybridizing the BWADO with DCNNs encounter limitations including increased burden, heightened training, and integration challenges. To mitigate these issues, balanced exploitation, automated hyperparameter tuning, exploration of hybrid architectures is employed. Tuning the weights and biases in a DCNN refines the model, adapting it to specific training patterns for improved accuracy and precision. This optimization minimizes the error, enhancing the DCNN's architecture and learning capabilities. Fine-tuning ensures adaptability across diverse datasets, prevents overfitting, and facilitates efficient generalization to new data. Moreover, this adaptive process reduces training time, enabling the DCNN to navigate varying task complexities for a more robust and versatile neural network. The algorithm parameters requirements for implementation is denoted by Table 3.

Table 3. Algorithmic parameters.

Parameters	Values
Size of population	50
Total iterations	1000
Times of replication	30
Probability factor	[0.1, 0.05]

- **Computational Complexity**

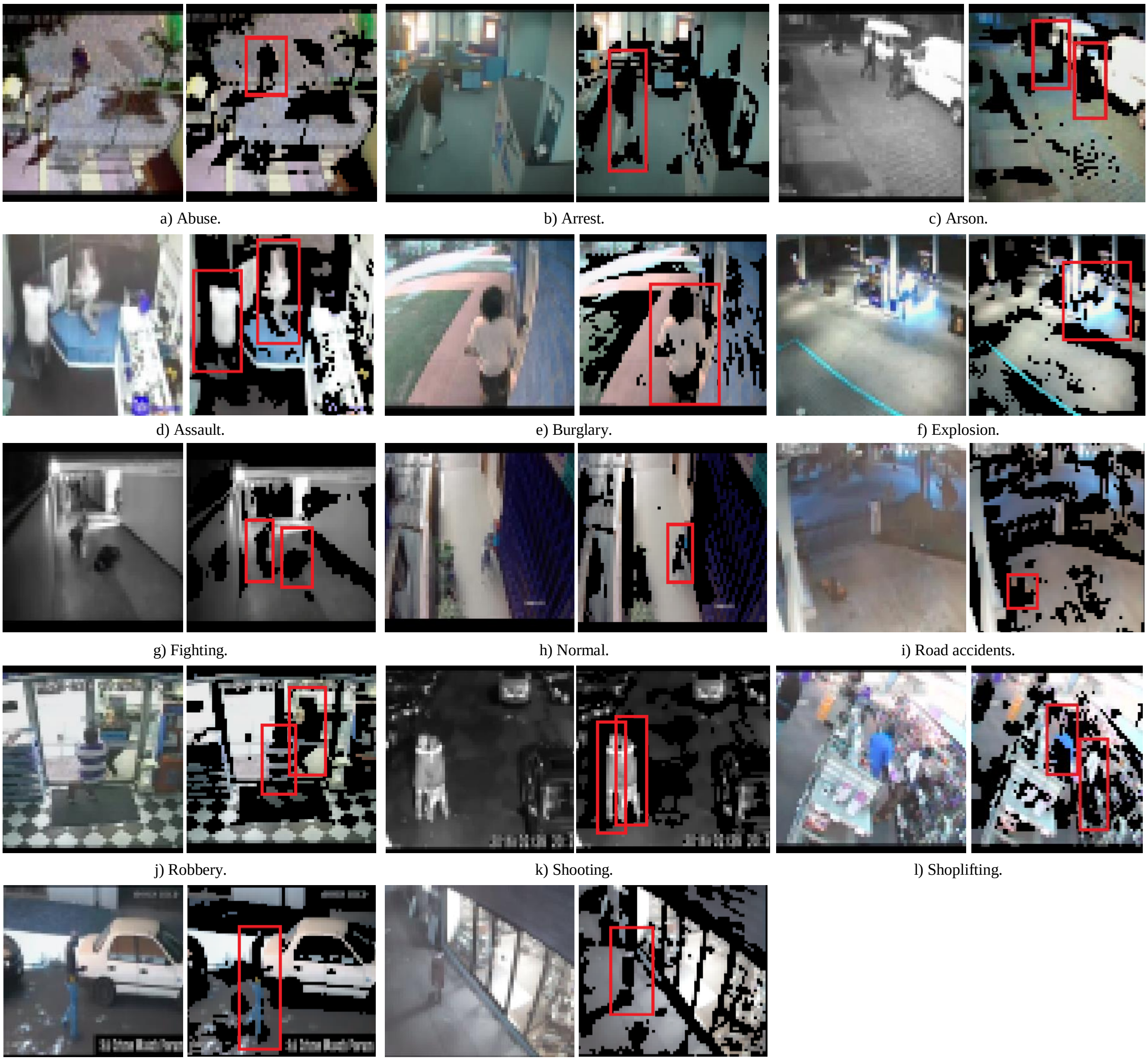
The approximate complexity of the hybrid BWADO is illustrated using Equation (15).

$$C\left(n\times\left(0.1\times t_{\max-BWOA}\right)+\left(f\left(t_{\max-ADOA}\right)\right)\right) \quad (15)$$

where the overall beluga whales are represented by n. During the whale drop phase, the complexity is inclined by the likelihood of whale drop, denoted as w^k and the balance factors, denoted as α and β. This complexity is estimated as $n\times0.1\times t_{max\text{-}BWOA}$. $t_{max\text{-}BWOA}$ and $t_{max\text{-}ADOA}$ is the maximum iteration of *BWOA* and *ADOA*. $C(n\times(1+1.1\times t_{max\text{-}BWOA}$, $C(f(t_{max\text{-}ADOA})$ and $C(n\times(1+1.1\times t_{max\text{-}BWOA})+(f(t_{max\text{-}ADOA})))$ is the computational complexity of *BWOA*, *ADOA* and introduced hybrid *BWADO*.

4. Implementation Outcomes

In this section, the experimental outcomes of the introduced model and performance comparisons are analyzed. The implementation is done in python. The performance measures used for comparison are accuracy, F1-score, recall, precision, processing time, loitering potential, humanness loss, classification loss and Receiver Operating Characteristic (ROC) curve.



a) Abuse.

b) Arrest.

c) Arson.

d) Assault.

e) Burglary.

f) Explosion.

g) Fighting.

h) Normal.

i) Road accidents.

j) Robbery.

k) Shooting.

l) Shoplifting.

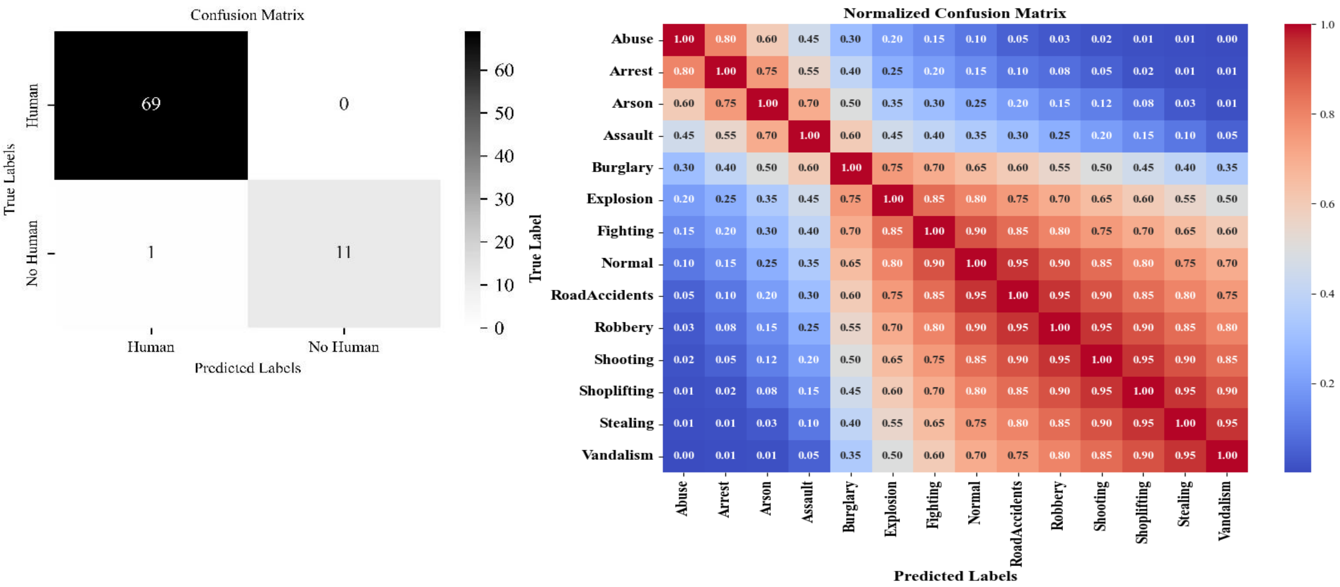
m) Stealing.

n) Vandalism.

Figure 5. Original and the segmented (bounding box) images.

From Figure 5, it is noted that the first column images show the original image of the video frames from the dataset. The combination of the SQA system with the DSORT algorithm provides a powerful capability to accurately differentiate and classify human behavior into two distinct categories: loitering and non-loitering. By combining these two approaches, the system gains the ability to not only track individuals but also to assess their behavior accurately. Specifically, it focuses on distinguishing between loitering behavior, where individuals linger in a certain area for a prolonged period and non-loitering behavior, where individuals move through the surveillance field without prolonged stops. The segmented outcomes with bounding box are plotted in second columns.

The confusion matrix curves of human tracking done using SQA system with the DSORT is portrayed in Figure 6-a) and LHCD done using BWADO-DCNN is portrayed in Figure 6-b). It displays the model's predictions and their correspondence with actual outcomes, categorizing results into True Positives (TP: correct positive predictions), True Negatives (TN: correct negative predictions), False Positives (FP: incorrect positive predictions) and False Negatives (FN: incorrect negative predictions).



	Human	No Human
Human	69	0
No Human	1	11

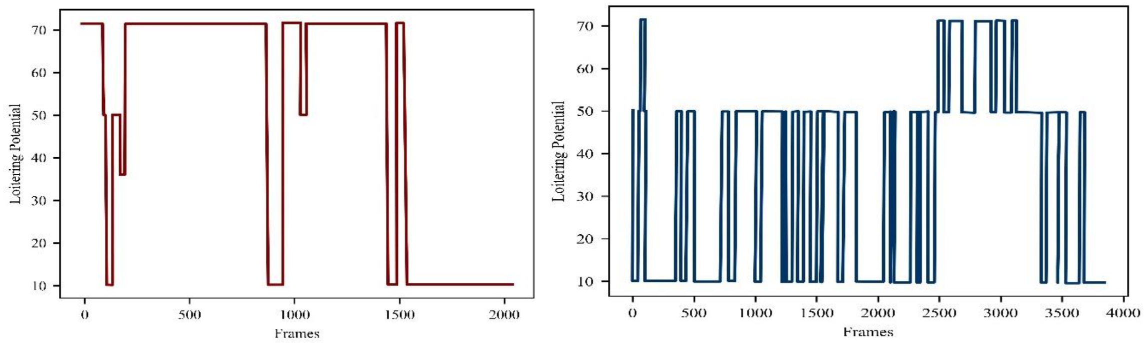
	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	Normal	RoadAccidents	Robbery	Shooting	Shoplifting	Stealing	Vandalism
Abuse	1.00	0.80	0.60	0.45	0.30	0.20	0.15	0.10	0.05	0.03	0.02	0.01	0.01	0.00
Arrest	0.80	1.00	0.75	0.55	0.40	0.25	0.20	0.15	0.10	0.08	0.05	0.02	0.01	0.01
Arson	0.60	0.75	1.00	0.70	0.50	0.35	0.30	0.25	0.20	0.15	0.12	0.08	0.03	0.01
Assault	0.45	0.55	0.70	1.00	0.60	0.45	0.40	0.35	0.30	0.25	0.20	0.15	0.10	0.05
Burglary	0.30	0.40	0.50	0.60	1.00	0.75	0.70	0.65	0.60	0.55	0.50	0.45	0.40	0.35
Explosion	0.20	0.25	0.35	0.45	0.75	1.00	0.85	0.80	0.75	0.70	0.65	0.60	0.55	0.50
Fighting	0.15	0.20	0.30	0.40	0.70	0.85	1.00	0.90	0.85	0.80	0.75	0.70	0.65	0.60
Normal	0.10	0.15	0.25	0.35	0.65	0.80	0.90	1.00	0.95	0.90	0.85	0.80	0.75	0.70
RoadAccidents	0.05	0.10	0.20	0.30	0.60	0.75	0.85	0.95	1.00	0.95	0.90	0.85	0.80	0.75
Robbery	0.03	0.08	0.15	0.25	0.55	0.70	0.80	0.90	0.95	1.00	0.95	0.90	0.85	0.80
Shooting	0.02	0.05	0.12	0.20	0.50	0.65	0.75	0.85	0.90	0.95	1.00	0.95	0.90	0.85
Shoplifting	0.01	0.02	0.08	0.15	0.45	0.60	0.70	0.80	0.85	0.90	0.95	1.00	0.95	0.90
Stealing	0.01	0.01	0.03	0.10	0.40	0.55	0.65	0.75	0.80	0.85	0.90	0.95	1.00	0.95
Vandalism	0.00	0.01	0.01	0.05	0.35	0.50	0.60	0.70	0.75	0.80	0.85	0.90	0.95	1.00

a) Human tracking by UCF crime dataset.

b) LHCD by UCF crime dataset.

Figure 6. Confusion matrix curve.

The loitering potential in the TP and FN video frames are depicted in Figures 7-a) and (b). The figure illustrates a curve representing the computed loitering potential for each snippet throughout the entire video. The fluctuations in this curve correspond to instances when the system identifies potential loitering behavior, as it analyzes patterns in individuals' movements to identify optimal times for crime activities. Consequently, the calculated loitering potential increases, mirroring the perceived risk. However, it's worth noting that in situations with low-resolution video footage, tracking algorithms occasionally falter, resulting in the loss of human subjects, which affects the accuracy of the system's assessments.



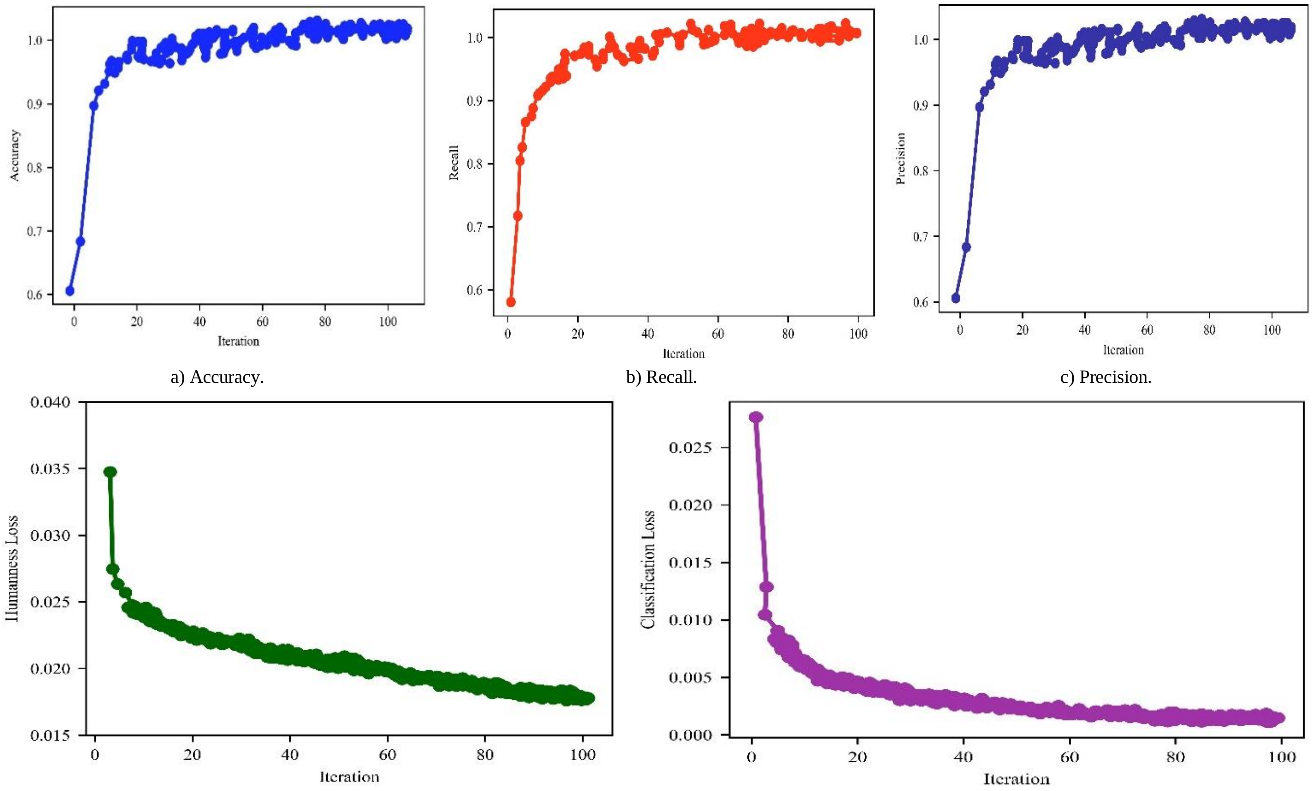
a) True positive video frames.

b) False negative video frames.

Figure 7. Loitering potential curve for each 10 frames.

Figure 8 presents two distinct types of losses, as well as precision, recall and accuracy curves that collectively assess the effectiveness of the introduced approach. The "humanness loss" curve in Figure 8-d) signifies the model's loss in accurately detecting humans by predicted bounding boxes, while the "classification loss" curve in

Figure 8-e) represents the loss incurred in correctly predicting human and crime classes, both for training and validation datasets. Notably, all these losses exhibit a gradual decrease as accuracy increases, underscoring the efficacy of the introduced approach. Additionally, precision in Figure 8-c) is a measure of the ratio of appropriately predicted positive samples to the overall positive predictions, while recall in Figure 8-b) quantifies the quantity of properly predicted positive classes out of the overall positive classes. The accuracy curves in the Figure 8-a) graphically demonstrate the progressive improvement in accuracy rates as the number of training epochs increases.



a) Accuracy.

b) Recall.

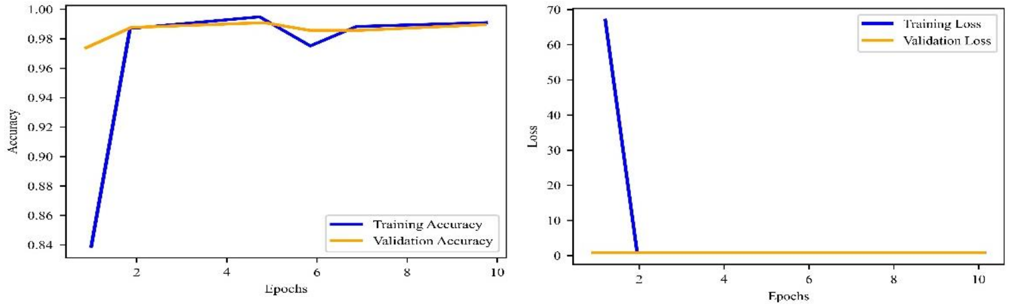
c) Precision.

d) Humanness loss.

e) Classification loss.

Figure 8. Performance analysis in terms of iterations.

The accuracy and loss curves in Figures 9-a) and (b) serve as compelling evidence of the introduced detection approach's effectiveness. As the number of iterations increases, the loss consistently decreases, signifying the approach's improved capability in minimizing errors and enhancing its precision in classifying instances. Conversely, the accuracy curves reveal a continuous upward trend, highlighting the approach's growing proficiency in making accurate predictions. Notably, the validation accuracy and loss exhibit smoother trends compared to their training counterparts, underscoring the approach's robustness and stability in generalization across different datasets. These converging trends collectively validate the approach's efficacy in detection tasks, showcasing its capacity to learn and generalize effectively

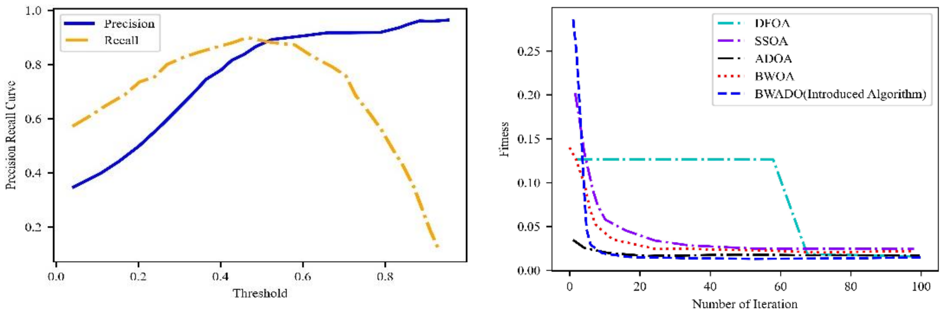


a) Accuracy curves.

b) Loss curves.

Figure 9. Evaluation of training and validation.

The precision-recall curve against various threshold values is presented for predictions made on the original test data. In Figure 10-a), it is noted that the precision and recall depend on the threshold value. Users have the flexibility to select a threshold value that aligns with the specific requirements of their use case. It's important to note that precision tends to increase as the threshold value is raised. In Figure 10-a), a threshold value of 0.8 yields a precision of 0.95 and a recall rate of 0.56. This figure demonstrates the trade-off between precision and recall and allow users to make informed decisions based on their specific needs. In Figure 10-b), the convergence curve analysis for objective function optimization is presented. It reveals that as the number of iterations rises, the fitness function steadily decreases, signifying effective minimization of the objective function and approaching an optimal solution. To assess the performance of the introduced BWADO, a comparison is made with the convergence curves of other algorithms: BWOA, ADOA, Sparrow Search Optimization Algorithm (SSOA) [22] and Direction and the Frequency Of Arrival (DFOA) [17]. The convergence of an optimization algorithm is influenced by several factors, including its complexity, computational demands, memory requirements and the number of parameters that require adjustment. Notably, the observations demonstrate that BWADO achieves a significantly faster rate of convergence in comparison to both BWOA and ADOA. This accelerated convergence implies high efficiency in reaching optimal solutions within a shorter computational timeframe, highlighting its superiority in optimization tasks.

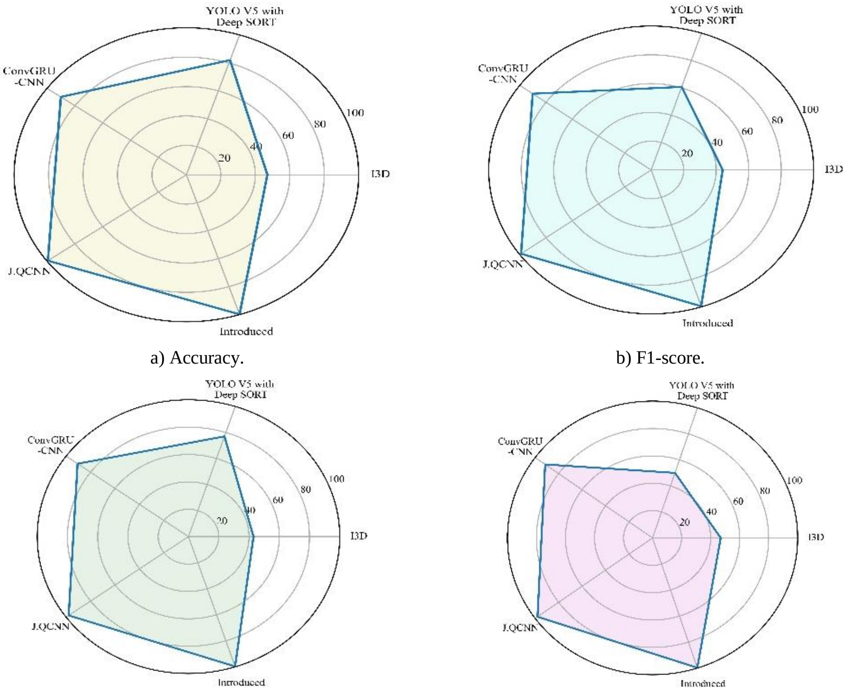


a) Precision-recall curve.

b) Fitness curve.

Figure 10. Performance evaluation.

The provided performance comparison Figure 11 and Table 4 offers a comprehensive evaluation of various video analysis techniques, shedding light on critical measures like (a) accuracy, (c) precision, (d) recall and (b) F1-score. Each technique is meticulously examined, drawing upon the strengths and weaknesses identified earlier to provide a nuanced interpretation of the results. ConvGRU-CNN [28] emerges as a robust contender, displaying a commendable accuracy of 90.09%. The balanced precision (90.62%) and recall (91.43%) contribute to a competitive F1-score of 90.17%, indicating reliable performance in anomaly detection. However, its strengths should be viewed in the context of its potential challenges, as achieving exceptionally high accuracy and precision poses difficulties compared to certain techniques.



a) Accuracy.

b) F1-score.

c) Precision.

d) Recall.

Figure 11. Comparison with existing approaches.

Table 4. Performance comparison using UCF crime dataset.

Techniques	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
ConvGRU-CNN [28]	90.09	90.62	91.43	90.17
J.QCNN [6]	99.41	97.9	98	99
I3D [23]	47	43	47	44
YOLO V5 with DSORT [24]	69.37	77	50	60.7
Introduced	99.76	98.9	98.59	99.89

J.QCNN [6] shines with a good accuracy of 99.41%, showcasing exceptional overall classification performance. The high precision (97.9%) and recall (98%) contribute to a F1-score of 99, positioning it extremely fit option for accurate classification tasks. However, this approach is not effective in handling large amounts of data because it does not utilize any optimizations. On the contrary, I3D [23] faces challenges with a lower accuracy of 47%, coupled with relatively low precision (43%) and recall (47%), resulting in a subpar F1-score of 44%. These metrics highlight limitations in achieving accurate and comprehensive classification compared to methods with higher precision and recall.

YOLO V5 with DSORT [24] achieves a moderate accuracy of 69.37%, making it a viable option for specific applications. However, challenges in precision (50%) and recall (60.7%) indicate the need for enhancements in accurately classifying instances. YOLO V5 with DSORT underscore its versatility but also emphasize potential limitations in achieving high precision and recall. The introduced technique stands out prominently in the performance comparison, boasting exceptional accuracy (99.76%), high precision (98.9%), recall (98.59%) and an outstanding F1-score of 99.89%. This impressive performance is attributed to the innovative combination of BWOA with DCNN in the proposed BWADO. This unique integration allows for focused identification of criminal activity within loitering behavior areas, showcasing the pioneering nature of this work. The incorporation of BWOA's global exploration capabilities, preventing local optima entrapment and ADOA's local optimization for fine-tuning solutions, contributes to potentially achieving better convergence and solution quality, further enhancing the efficacy of the introduced method.

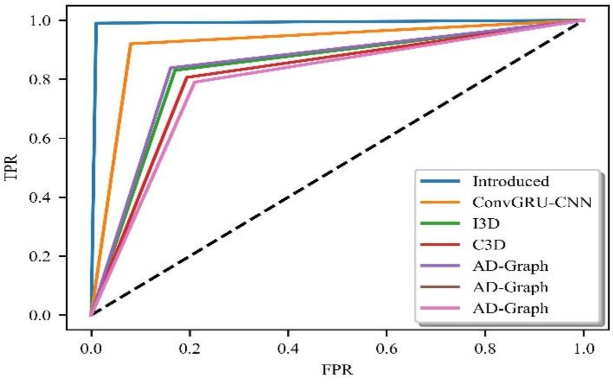


Figure 12. ROC curve.

Constructed by plotting TP rates on the y-axis against FP rates on the x-axis, the ROC curve serves as a powerful tool to evaluate the neural network's discriminatory capabilities. This visual representation is particularly crucial in assessing the network's effectiveness in properly detecting positive cases while diminishing the misclassification of negative cases. In Figure 12, the ROC plots for various techniques reveal insightful Area Under the Curve (AUC) values that signify the discriminatory prowess of each method. ConvGRU-CNN [28] exhibits an impressive AUC value of 0.92, representing its substantial ability to differentiate between positive and negative cases. However, when compared to the introduced approach, which boasts a remarkable AUC value of 0.99, ConvGRU-CNN's performance is surpassed. Similarly, I3D [23] and C3D [5] showcase AUC values of 0.83 and 0.86, respectively, reflecting their proficiency in discrimination. AD-Graph [33] and DTED [16] exhibit slightly lower AUC values of 0.8385 and 0.79, respectively. Notably, all these AUC values are notably lower than the exceptional AUC value achieved using the introduced approach. The higher AUC value achieved by the introduced approach signifies its superior discriminatory performance compared to the other evaluated techniques, emphasizing its efficacy in correctly identifying positive cases while minimizing false positives.

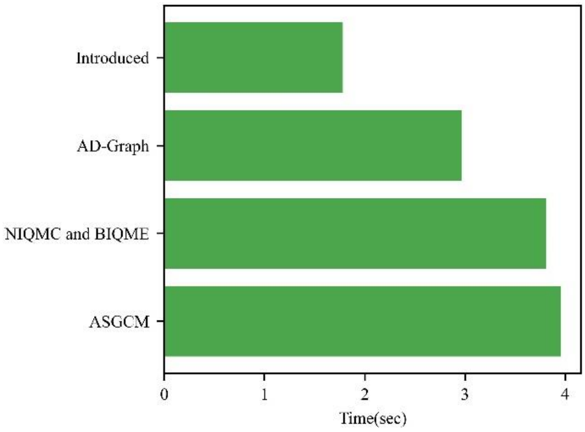


Figure 13. Processing time comparison.

Table 5. Processing time comparison.

Techniques	Time (s)
ASGCM [20]	3.96
NIQMC and BIQME [7]	3.81
AD-Graph [33]	2.97
Introduced	1.78

The assessment of processing times for various techniques, as depicted in Figure 13 and Table 5, sheds light on the efficiency disparities among these methods. Notably, the introduced technique emerges as the most time-efficient, completing the task in a mere 1.78 seconds. Comparatively, ASGCM, NIQMC and BIQME exhibit processing times of 3.96 seconds, 3.81 seconds and 3.81 seconds, respectively. While these methods showcase reasonable processing times, the introduced technique significantly outperforms them, emphasizing its prowess in expeditious data analysis. Additionally, AD-Graph stands out with a relatively shorter execution time of 2.97 seconds. However, it is crucial to note that despite its commendable performance in terms of execution time, AD-Graph has previously been identified with certain weaknesses, including challenges in scenarios with minimum-resolution images, minimum illumination, fast motion and crowded scenes. The highlighted weaknesses of existing methods, especially those associated with AD-Graph, underscore the significance of the introduced technique's efficiency. Its superior processing speed not only positions it as a top-performing option but also addresses potential limitations identified in other methods.

Table 6. Comparison of human tracking models.

Segmentation techniques	Accuracy (%)
NFC [2]	91.8
LGDC [12]	95.37
YOLOV7 with Deep SORT [38]	95.37
Introduced tracking model	99.76

The Table 6 presents a comparison of segmentation techniques in terms of their accuracy percentages. The Neuro Fuzzy Classifier (NFC) [2] achieves an accuracy of 91.8%, showcasing its proficiency in object or pattern recognition with some degree of precision which exhibits a weakness in handling complex and dynamic scenario. The Local Geometric Descriptor Classifier (LGDC) [12] and YOLOV7 with Deep SORT [38] both demonstrate a higher accuracy of 95.37%, indicating enhanced capabilities in accurately classifying and segmenting objects based on local geometric features. But, local geometric descriptors struggle when confronted with complex scenes where objects undergo significant transformations. Similarly, YOLOv7 with Deep SORT in detection tasks is sensitive to occlusions and crowded scenes. Notably, the Introduced tracking model surpasses all other techniques with an impressive accuracy of 99.76%, suggesting the introduction of an innovative or customized tracking model that excels in precise object segmentation and tracking. The higher accuracy percentages across these techniques signify their effectiveness in various segmentation tasks, with the introduced model standing out as particularly noteworthy for its superior accuracy.

Table 7 presents the model’s performance metrics, which are assessed using K-fold cross-validation. In this scenario, K varies from 1 to 10, indicating the number of folds utilized for model evaluation. The “Mean Value” row in Table 4 compiles the average of individual metric across entire 10 cross-validation epochs, including metrics like (a) accuracy (b) F1-score (c) precision and (d) Recall providing a comprehensive summary of the introduced approach’s overall performance. Additionally, the mean value across these iterations is calculated to offer a summarized assessment.

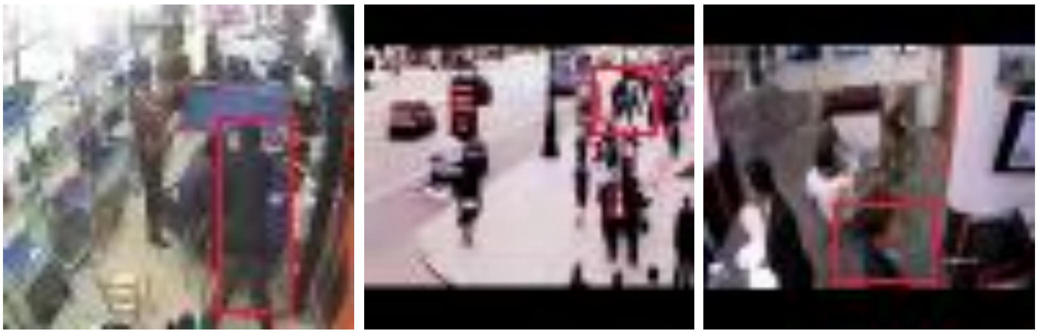
Table 7. Introduced approach comparison in (a) accuracy (b) F1-score (c) precision and (d) Recall for 10-fold validation.

K-fold	Accuracy (%)	F1-score (%)	Precision (%)	Recall (%)
1-Fold	99.78	99.2	99.25	99.34
2-Fold	99.67	99.03	98.56	99.3
3-Fold	98.8	98	97.86	98.79
4-Fold	99.8	99.79	99.93	98.69
5-Fold	97.67	98.09	98.01	98.07
6-Fold	98.65	98.04	97.7	98.36
7-Fold	97.5	98.13	98.47	98.27
8-Fold	99.56	99.75	98.36	99.83
9-Fold	99.19	99.67	98.85	98.64
10-Fold	99.34	99.05	98.8	99.08
Mean Value	99.032	99.931	98.261	98.762

- **Case Study on Crowd-Sourced Video Analysis**

To rigorously assess the efficacy of the introduced approach, a thorough investigation is conducted through a case study focused on crowd-sourced video analysis. This entails a meticulous process wherein a subset of image frames depicting crowded scenarios is meticulously curated from the dataset. These frames serve as the foundation for training the novel model introduced within the research. The chief aim of this model is to discern and flag instances wherein individuals engage in loitering behavior.

With the approach duly trained on the collected dataset, it is then systematically applied to the entire repository of crowd-sourced video footage. During this phase, the model diligently scrutinizes each frame, meticulously analyzing the behavior of individuals within the crowded scenes. Through its sophisticated segmentation-based detection mechanism, the model endeavors to accurately identify and classify instances of loitering behavior among the subjects captured in the video frames.



a) Sample 1. b) Sample 2. c) Sample 3.

Figure 14. Segmentation results of crowd-sourced frames.

Following the comprehensive analysis of the crowd-sourced video footage, the findings are meticulously documented and presented for analysis. This includes the visual representation of the segmentation (bounding box) results obtained from the analyzed frames. Figure 14-a), (b) and (c) serve as tangible manifestations of the model's performance, providing insightful glimpses into its efficacy in identifying and delineating instances of loitering behavior amidst crowded environments.

In essence, this meticulous case study serves as a robust validation mechanism, offering a thorough exploration of the introduced approach's capabilities in the context of crowd-sourced video analysis. Through careful curation, rigorous training and systematic application, the research endeavors to shed light on the model's effectiveness in detecting and characterizing loitering behavior, thereby contributing to the advancement of surveillance technology and public safety measures.

5. Conclusions

This research work represented a significant advancement in the field of VS technology, as it successfully addressed the challenges associated with detecting and preventing human-related crimes. The introduced LHCD module, powered by Enhanced Euclidean-based DSORT and the SQA algorithm, provided a nuanced and highly accurate approach to assess human travel distances, resulting in a substantial reduction in false negatives and processing times (1.78s). Furthermore, the integration of the BWADO and DCNN elevated the system's capabilities, allowing for precise identification of criminal activities within loitering areas. The research outcomes demonstrate the effectiveness of the introduced LHCD module in significantly reducing false alarms and improving response times in VS systems. The integration of BWADO and DCNN further enhances the system's capabilities, providing a focused identification of criminal activity within loitering behavior areas. This pioneering combination of algorithms harnesses the strengths of global exploration and local optimization to achieve better convergence and solution quality. The research contributes to the advancement of VS technology, offering a more precise and effective approach to LHCD, thus bolstering security in various domains.

The future scope of this research entails real-time optimization of the LHCD module for swift response in live surveillance scenarios and further advancements in DL techniques to boost system accuracy and robustness. Additionally, integration with Internet of Things (IoT) devices and privacy-preserving measures will provide a more comprehensive and privacy-conscious security solution for a wide range of applications.

Data Availability Statements

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no potential conflict of interest.

References

[1] Abdulghafoor N. and Abdullah H., "A Novel Real-Time Multiple Objects Detection and Tracking Framework for Different Challenges," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9637-9647, 2022. https://doi.org/10.1016/j.aej.2022.02.068

[2] Abdullah F. and Jalal A., "Semantic Segmentation Based Crowd Tracking and Anomaly Detection Via Neuro-Fuzzy Classifier in Smart Surveillance System," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 2173-2190, 2023. https://doi.org/10.1007/s13369-022-07092-x

[3] Ahmed I., Ahmad M., Ahmad A., and Jeon G., "Top View Multiple People Tracking by Detection Using Deep Sort and YOLOv3 with Transfer Learning: Within 5G Infrastructure," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3053-3067, 2021. https://doi.org/10.1007/s13042-020-01220-5

[4] Ahmed S., Bhatti M., Khan M., Lövström B., and Shahid M., "Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos," *Applied Sciences*, vol. 12, no. 12, pp. 1-21, 2022. https://doi.org/10.3390/app12125772

[5] Ahmed W. and Yousaf M., "A Deep Autoencoder-Based Approach for Suspicious Action Recognition in Surveillance Videos," *Arabian Journal for Science and Engineering*, vol. 49, no. 3, pp. 3517-3532, 2024. https://doi.org/10.1007/s13369-023-08038-7

[6] Amin J., Anjum M., Ibrar K., Sharif M., Kadry S., and Crespo R., "Detection of Anomaly in Surveillance Videos Using Quantum Convolutional Neural Networks," *Image and Vision Computing*, vol. 135, pp. 104710, 2023. https://doi.org/10.1016/j.imavis.2023.104710

[7] Biswas T., Bhattacharya D., and Mandal G., "Dynamic Strategy to Use Optimum Memory Space in Real-Time Video Surveillance," *Journal of Ambient Intelligence and Humanized*

Computing, vol. 14, no. 3, pp. 2771-2784, 2023. https://doi.org/10.1007/s12652-023-04521-z

[8] Cai C., Gou B., Khishe M., Mohammadi M., Rashidi S., Moradpour R., and Mirjalili S., "Improved Deep Convolutional Neural Networks Using Chimp Optimization Algorithm for COVID19 Diagnosis from the X-Ray Images," *Expert Systems with Applications*, vol. 213, pp. 119206, 2023. https://doi.org/10.1016/j.eswa.2022.119206

[9] Fathy C. and Saleh S., "Integrating Deep Learning-Based IOT and Fog Computing with Software-Defined Networking for Detecting Weapons in Video Surveillance Systems," *Sensors*, vol. 22, no. 14, pp. 1-22, 2022. https://doi.org/10.3390/s22145075

[10] Gandapur M., "E2E-VSDL: End-to-End Video Surveillance-Based Deep Learning Model to Detect and Prevent Criminal Activities," *Image and Vision Computing*, vol. 123, pp. 104467, 2022. https://doi.org/10.1016/j.imavis.2022.104467

[11] Georgiou A., Masters P., Johnson S., and Feetham L., "Uav-Assisted Real-time Evidence Detection in Outdoor Crime Scene Investigations," *Journal of Forensic Sciences*, vol. 67, no. 3, pp. 1221-1232, 2022. https://doi.org/10.1111/1556-4029.15009

[12] Gómez J., Aycard O., and Baber J., "Efficient Detection and Tracking of Human Using 3D LiDAR Sensor," *Sensors*, vol. 23, no. 10, pp. 1-12, 2023. https://doi.org/10.3390/s23104720

[13] Huszár V., Adhikarla V., Négyesi I., and Krasznay C., "Toward Fast and Accurate Violence Detection for Automated Video Surveillance Applications," *IEEE Access*, vol. 11, pp. 18772-18793, 2023. DOI:10.1109/ACCESS.2023.3245521

[14] Ingle P. and Kim Y., "Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities," *Sensors*, vol. 22, no. 10, pp. 1-21, 2022. https://doi.org/10.3390/s22103862

[15] Jaouedi N., Boujnah N., and Bouhlel M., "A Novel Recurrent Neural Networks Architecture for Behavior Analysis," *The International Arab Journal of Information Technology*, vol. 18, no. 2, pp. 133-139, 2021. https://doi.org/10.34028/iajit/18/2/1

[16] Kamoona A., Gostar A., Bab-Hadiashar A., and Hoseinnezhad R., "Multiple Instance-Based Video Anomaly Detection Using Deep Temporal Encoding-Decoding," *Expert Systems with Applications*, vol. 214, pp. 119079, 2023. https://doi.org/10.1016/j.eswa.2022.119079

[17] Khairuddin A., Ali N., Alwee R., Haron H., and Zain A., "Parameter Optimization of Gradient Tree Boosting Using Dragonfly Algorithm in Crime Forecasting and Analysis," *Journal of Computer Science*, vol. 15, no. 8, pp. 1085-1096, 2019. https://doi.org/10.3844/jcssp.2019.1085.1096

[18] Kotkar V. and Sucharita V., "Fast Anomaly Detection in Video Surveillance System Using Robust Spatiotemporal and Deep Learning Methods," *Multimedia Tools and Applications*, vol. 82, no. 22, pp. 34259-86, 2023. https://doi.org/10.1007/s11042-023-14840-0

[19] Kumar K. and Reddy H., "Crime Activities Prediction System in Video Surveillance by an Optimized Deep Learning Framework," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, 2022. https://doi.org/10.1002/cpe.6852

[20] Li Z., Zhang X., Xu F., Jing X., and Zhang T., "A Multi-Scale Video Surveillance Based Information Aggregation Model for Crime Prediction," *Alexandria Engineering Journal*, vol. 73, pp. 695-707, 2023. https://doi.org/10.1016/j.aej.2023.04.045

[21] Meng F., Guo L., Wu Q., and Li H., "A New Deep Segmentation Quality Assessment Network for Refining Bounding Box Based Segmentation," *IEEE Access*, vol. 7, pp. 59514-59523, 2019. https://doi.org/10.1109/access.2019.2915121

[22] Mithoo P. and Kumar M., "Social Network Analysis for Crime Rate Detection Using Spizella Swarm Optimization Based BiLSTM Classifier," *Knowledge-Based Systems*, vol. 269, pp. 110450, 2023. https://doi.org/10.1016/j.knosys.2023.110450

[23] Mumtaz A., Sargano A., and Habib Z., "Robust Learning for Real-World Anomalies in Surveillance Videos," *Multimedia Tools and Applications*, vol. 82, no. 13, pp. 20303-20322, 2023. https://doi.org/10.1007/s11042-023-14425-x

[24] Nazir A., Mitra R., Sulieman H., and Kamalov F., "Suspicious Behavior Detection with Temporal Feature Extraction and Time-Series Classification for Shoplifting Crime Prevention," *Sensors*, vol. 23, no. 13, pp. 1-19, 2023. https://doi.org/10.3390/s23135811

[25] Patrikar D. and Parate M., "Anomaly Detection Using Edge Computing in Video Surveillance System: Review," *International Journal of Multimedia Information Retrieval*, vol. 11, no. 2, pp. 85-110, 2022. https://doi.org/10.1007/s13735-022-00227-8

[26] Pazho A., Neff C., Noghre G., Ardabili B., Yao S., Baharani M., and Tabkhi H., "Ancilia: Scalable Intelligent Video Surveillance for the Artificial Intelligence of Things," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 14940-14951, 2023. https://doi.org/10.1109/jiot.2023.3263725

[27] Pouyan S., Charmi M., Azarpeyvand A., and Hassanpoor H., "Propounding First Artificial Intelligence Approach for Predicting Robbery Behavior Potential in an Indoor Security Camera," *IEEE Access*, vol. 11, pp. 60471-60489, 2023. https://doi.org/10.1109/access.2023.3284472

[28] Qasim M. and Verdu E., "Video Anomaly Detection System Using Deep Convolutional and

Recurrent Models," *Results in Engineering*, vol. 18, pp. 101026, 2023. https://doi.org/10.1016/j.rineng.2023.101026

[29] Sahay K., Balachander B., Jagadeesh B., Kumar G., Kumar R., and Parvathy L., "A Real Time Crime Scene Intelligent Video Surveillance Systems in Violence Detection Framework Using Deep Learning Techniques," *Computers and Electrical Engineering*, vol. 103, pp. 108319, 2022. https://doi.org/10.1016/j.compeleceng.2022.108319

[30] Shoitan R., Moussa M., and Nemr H., "Attribute Based Spatio-Temporal Person Retrieval in Video Surveillance," *Alexandria Engineering Journal* vol. 63, pp. 441-454, 2023. https://doi.org/10.1016/j.aej.2022.07.053

[31] Singla N., Nagpal S., and Singh J., "Frame Duplication Detection Using CNN-Based Features with PCA and Agglomerative Clustering," *in Proceedings of the International Conference on Communication and Intelligent Systems*, New Delhi, pp. 383-91, 2022. https://doi.org/10.1007/978-981-19-2130-8_31

[32] UCF Crime Dataset, https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset, n.d, Last Visited, 2024.

[33] Ullah W., Hussain T., Min Ullah F., Khan M., Hassaballah M., Rodrigues J., Baik S., and Albuquerque V., "Ad-Graph: Weakly Supervised Anomaly Detection Graph Neural Network," *International Journal of Intelligent Systems*, vol. 2023, pp. 1-12, 2023. https://doi.org/10.1155/2023/7868415

[34] Waddenkery N. and Soma S., "Adam-Dingo Optimized Deep Maxout Network-Based Video Surveillance System for Stealing Crime Detection," *Measurement: Sensors*, vol. 29, pp. 1-13, 2023. https://doi.org/10.1016/j.measen.2023.100885

[35] William P., Shrivastava A., Karpagam N., Mohanaprakash T., Tongkachok K., and Kumar K., "Crime Analysis Using Computer Vision Approach with Machine Learning," *in Proceedings of the 3rd Mobile Radio Communications and 5G Networks*, Kurukshetra, pp. 297-315, 2023. https://doi.org/10.1007/978-981-19-7982-8_25

[36] Yang Z., Liao W., Wang H., Bak C., and Chen Z., "Improved Euclidean Distance Based Pilot Protection for Lines with Renewable Energy Sources," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8551-8562, 2022. https://doi.org/10.1109/tii.2022.3148318

[37] Zahid A., Qasim T., Bhatti N., and Zia M., "A Data-Driven Approach for Road Accident Detection in Surveillance Videos," *Multimedia Tools and Applications*, vol. 83, pp. 17217-17231 2023. https://doi.org/10.1007/s11042-023-16193-0

[38] Zi X., Chaturvedi K., Braytee A., Li J., and Prasad M., "Detecting Human Falls in Poor Lighting: Object Detection and Tracking Approach for Indoor Safety," *Electronics*, vol. 12, no. 5, pp. 1-12, 2023. https://doi.org/10.3390/electronics12051259

[39] Zhong C., Li G., and Meng Z., "Beluga Whale Optimization: A Novel Nature-Inspired Metaheuristic Algorithm," *Knowledge-Based Systems*, vol. 251, pp. 109215, 2022. https://doi.org/10.1016/j.knosys.2022.109215

Nischita Waddenkery is a research scholar in Poojya Doddappa Appa (PDA) College of Engineering, Kalaburagi, Karnataka, India. She is pursuing full time Ph.D in department of Computer Science and Engineering with Specialization in Image processing, Artificial Intelligence, Deep Learning. Author has published more than 5 papers in Journals and Conference. The areas of interest are Artificial Intelligence, Cloud Computing, Image Processing, Networking, Data Structure and programming languages are Python, .Net, Java.

Shridevi Soma is working presently as Professor in Department of Computer Science and Engineering, Poojya Doddappa Appa College of Engineering, Kalaburagi Karnataka, India. She has 20 years of Teaching and 10 years of Research Experience, and completed her B.E, M.Tech. and Ph.D. in Computer Science and Engineering. Her Research Area includes Digital Image Processing and Pattern Recognition, Cloud Computing, Internet of Things, Big Data Analytics. She published more than 30 Research papers in above mentioned areas, also Guiding Research Students.