Heliyon 10 (2024) e34922

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Real-time object detection, tracking, and monitoring framework for security surveillance systems

Sani Abba [a,*], Ali Mohammed Bizi [a], Jeong-A Lee [b,**], Souley Bakouri [a], Maria Liz Crespo [c]

[a] Department of Computer Science, Faculty of Science, Gubi Campus, Abubakar Tafawa Balewa University, Along Ningi/Kano Road, P.M.B. 0248, Bauchi, Nigeria
[b] Computer Systems Laboratory, Department of Computer Engineering, Chosun University, Dongku SeoSukDong 375, Gwangju, 501-759, South Korea
[c] Multi-disciplinary Laboratory (MLab), Abdussalam International Centre for Theoretical Physics (ICTP), Via Beirut 31, 34014, Trieste, Italy

ARTICLE INFO

Keywords:
Object detection and tracking
Background subtraction
Component object labeling
Approximate median filtering
Video surveillance system
Deep learning

ABSTRACT

The concept of security is becoming a global challenge, and governments, stakeholders, corporate societies, and individuals must urgently create a reasonable protection mechanism for good. Therefore, a real-time surveillance system is essential for detection, tracking, and monitoring. Many studies have attempted to provide better solutions but more research and better approaches are essential. This study presents a real-time framework for object detection and tracking for security surveillance systems. The system has been designed based on approximate median filtering, component labeling, background subtraction, and deep learning approaches. The new algorithms for object detection, tracking, and recognition have been implemented using Python and integrated with C# programming languages for ease of use. A software application framework is designed, implemented, and evaluated. The experimental results based on MOT-Challenge performance metrics show that the proposed algorithms have much better performance in terms of accuracy and precision on the MOT15, MOT16, and MOT17 datasets compared to state-of-the-art approaches. This framework also provides an accurate and effective means of monitoring and recognizing moving objects. The software development, including the design of the framework user interfaces, is coded in the C# programming language and integrated with Python using Microsoft Visual Studio (2019 edition). The integration is performed to provide a convenient user interface and to enable the execution of the framework as a standard and standalone software application. Future studies will consider the dynamic scalability of the framework to accommodate different surveillance application areas in overcrowded scenarios. Multiple data sources are integrated to enhance the performance for different scene times, locations, and weather conditions. Furthermore, other object-detection techniques such as You Only Look Once (YOLO) and its variants shall be considered in future studies. These techniques allow the framework to adapt to complex situations in which security surveillance is challenging.

* Corresponding author.
** Corresponding author.
E-mail addresses: sabba@atbu.edu.ng (S. Abba), bizi@fedpodam.edu.ng (A.M. Bizi), jalee@chosun.ac.kr (J.-A. Lee), sboukari@atbu.edu.ng (S. Bakouri), mcrespo@ictp.it (M.L. Crespo).

https://doi.org/10.1016/j.heliyon.2024.e34922
Received 17 March 2024; Received in revised form 18 July 2024; Accepted 18 July 2024
Available online 20 July 2024

1. Introduction

Security is a globally understood concept that allows stakeholders in governments, corporate organizations, security agencies, and individuals to safeguard against threats. Multiple layers of protective mechanisms are required to detect, track, and monitor people in a surveillance system. Therefore, there is a critical need for a real-time surveillance system for the detection, tracking, and surveillance of humans in overcrowded environments [1].

Currently, there is a considerable increase in visual systems and their applications in different fields, such as the detection of video anomalies, object recognition, and object tracking. These application areas are clearly explained based on predefined goals and objectives using datasets and well-defined performance-evaluation techniques. However, Perimeter Intrusion Detection (PID), a visual monitoring technique, must be properly studied and clearly defined. The primary purpose of this technique is to promptly detect the existence of an unauthorized object in a potentially insecure environment [2]. Facial detection in images is widely used in various applications. These include social networks to help Internet users identify people, security surveillance systems to identify suspects, monitoring population growth, and avoiding pedestrian crashes on a highway. Therefore, an approach that offers rapid and reliable object detection based on scene formation to identify people in indoor environments is a promising solution [3]. The proposed approach is based on the You Only Look Once (YOLOv4) network, which combines the classification of the area of interest with a faster Re-current Convolutional Neural Network (R–CNN) technique. This allows the technique to detect individuals in an indoor environment with greater precision and a reasonable inference time [3].

In computer vision, object tracking is an interesting and challenging area of research. The primary task of tracking objects is the complexity of the camera-axis orientation and object occlusion. The issue of variations in remote scene environments is another complexity associated with object tracking. These complexities make the technique of tracking objects demanding calculations and requiring considerable computational time. Consequently, a stochastic gradient optimization technique coupled with a particulate filter for object tracking will provide a solution. This approach uses the Maximum Average Correlation Height (MACH) filter to detect the object to be tracked. The object to be tracked is detected according to the existence of a method for measuring the peak correlation and mean similarity, and the object detection results are sent to the tracking routine for processing. In this approach, the gradient descent technique is used to track the object and optimize particle filters. This technique accelerates particle convergence and reduces the time required to track an object [4].

Adaptive Optical Flow Segmentation (OFATS) was introduced as a framework for automatic change detection. This approach uses optical flow data as well as an objective function. The procedure involves motion detection according to the optical flow estimation using the deep learning technique and modified area segmentation, which uses the adaptive threshold selection technique [5]. Another approach to a deep learning framework for vehicle detection and tracking from unmanned aerial vehicle videos for monitoring track flow in multifaceted road networks was presented [6]. The proposed approach can scale variations and orientations in track videos. This procedure involves using the You Only Look Once (YOLOVv3) object detection technique and custom-labeled track datasets. To track vehicles, a detection and tracking procedure is adopted, and the deep appearance features of the object are used for the identification of the vehicle. In addition, Kalman filtering is used to estimate vehicle movement.

An object-tracking framework using a virtual simulation environment with deep Q-learning algorithms was proposed [7]. The approach uses the network to evaluate the environment using the deep reinforcement learning model to control the occurrences in the virtual simulation environment and uses sequential pictures originating from the virtual city environs as input to the model. Then, a pre-training of the model is performed with the help of several sets of sequenced training images, and the procedure is refined to ensure the adaptability of execution during the tracking process.

Unmanned Aerial Vehicles (UAVs) are used in various applications including civilian, military, and mission-critical applications. They are capable of flying in a well-organized and coordinated model called swarms. To coordinate the movement of aerial vehicles as swarms, visual cameras are used to visually monitor each member of the swarm. Therefore, the major complexity is the development of robust and flexible solutions for detecting and tracking moving aerial vehicles using frame sequencing.

A framework using deep learning has been presented. This approach uses a combination of the You Only Look Once (YOLO) object detection technique and machine vision algorithms for object navigation [8]. An alternative approach is the two-phase UAV object tracking framework. This approach combines the two stages of (1) target detection using multi-featured discrimination and (2) bounding box estimation using the instance-aware attention networking stage. In the first stage, a small target feature representation scheme is used, which combines handcrafted, low-level deep, and high-level deep features. This combination enables the correlation filter to accurately predict the location of the object in question. In the second stage, the two approaches are combined. The instance-aware over union and instance-aware attention networks approaches are combined to evaluate the target size using the bounding-box methodology [9].

A framework for high-performance algorithms using a fast Fourier transformation to search, detect, and track underwater moving objects in acoustic wavefront signaling, surveillance, and monitoring has been proposed [10]. Its main focus is to model and estimate the range and speed of targets which are deep underwater dynamic objects. This approach introduces the use of Kronecker product signal algebra and the Kuck algorithm-based programming technique for parallel programming paradigms.

Another useful contribution is a model for detecting and classifying small objects based on deep learning which is used for waste management. Its primary objective is to detect and classify small pieces of waste for smart waste management and control. This approach is based on a combination of detection and classification techniques. In the first technique, an Arithmetic Optimization Algorithm (AOA) is used in conjunction with enhanced RefineDet (IRD) models; the AOA algorithm allows for the selection of IRD hyperparameters. In the second technique, the Functional Link Neural Network (FLNN) technique is employed to classify waste objects into different categories [11]. A measure for evaluating vehicle detection software based on video data used in the automobile industry

has provided a solution for object classification [12]. This approach uses sub-measures to combine rectangles, shapes, and distances. The combination of these sub-measures aims to reduce the problem of the poor adaptability of the Jaccard index in object recognition. In addition, a detection quality technique is introduced in the sequence of video images for the late detection of objects.

Despite the aforementioned approaches and contributions, this study aims to provide a framework for detecting, tracking, and monitoring objects for security surveillance systems. It demonstrates how machine learning techniques may be employed to overcome security challenges through video surveillance. These research findings will be useful to security personnel (military) for the detection, tracking, and monitoring of dynamic objects. It can also be used for security surveillance in public places by most organizations. In addition, the research findings can help in the traffic monitoring of objects in surveillance systems. Therefore, the overall goal of this study is to provide flexibility in terms of programmability and dynamic comprehensibility of security surveillance through the detection, tracking, and monitoring of objects in surveillance systems.

The primary contributions of this study are as follows.

1. A framework for the dynamic detection, tracking, and monitoring of objects in security surveillance systems is established.
2. A software application is designed and implemented using Python and then integrated with the C# programming language for ease of use. The developed software application provides flexibility in terms of programmability and dynamic comprehensibility for security surveillance through the detection, tracking, and monitoring of objects in a surveillance system.
3. Machine-learning techniques are developed to overcome security challenges via video surveillance.
4. The high performance of the proposed framework under the multiple-object tracking (MOT) challenge metrics and datasets indicates that the framework can be deployed in actual situations to aid security and surveillance systems.

The remainder of this article is organized as follows: Section 2 discusses the related studies. Section 3 discusses the design and implementation of the proposed framework. Section 4 presents the experiments and a discussion of the results. Section 5 provides the concluding remarks.

2. Related works

The literature review provides some research related to object detection and tracking. For instance, a recent and comprehensive review of multiple object tracking was presented by the authors Wenhan et al. The review investigates the present and future advances in multiple object tracking proposes a solution to various problems, and provides future direction. The review provided contributions in fourfold including MOT system formulation, categorization, and evaluation. In addition, a comprehensive performance evaluation based on different datasets was provided [13].

Ciaparrone et al. [14] presented a comprehensive survey that focused on deep learning models that provide solutions to Multiple Object Tracking (MOT) based on videos from a single camera source. The survey provided a detailed survey of how deep learning is used in the four stages of MOT algorithms. In addition, the authors presented an experimental comparison of published research based on three MOT challenge datasets. The survey provided information regarding several similarities among the best performers and suggested areas of research to focus on soon.

A study of the development of convolutional neuronal networks was conducted by Li et al. [15]. The survey provided the basic structure of Convolutional Neural Networks (CNN) and demonstrated the differences between artificial neural networks based on how they work. In addition, the survey analyzed the structural framework of a CNN consisting of convolutional, subsampling, and fully connected layers. Finally, the survey highlights the merits of CNN in areas such as image processing and speech analysis.

Uddagiri and Das [16] presented an approach for comparing background verification techniques. The compared techniques included frame differencing, Gaussian methods, and probabilistic models. Their findings demonstrated that simple tasks such as traffic analysis and adaptive median filtering yield better accuracy and reasonably low processing time. The goal of their study was mainly focused on methods that can tackle the effects of noise, variations in climate conditions, and the problem of segmentation in moving objects.

In computer vision, object tracking is a challenging area that researchers need to focus on. To overcome this problem, Yilmaz et al. [17] reviewed state-of-the-art tracking methods and new trends associated with them. The survey categorized the tracking methods according to the objects and how the object's motion is represented. It described the representative methods in each category and determined the merits and demerits of each method. In addition, the survey presented essential features associated with object tracking, including the selection of appropriate image features, motion models, and object detection.

Another approach for object motion segmentation in videos that adds frame-level object detection and object tracking was presented by Breyer and Brock [18]. This process removes the temporally consistent object tube that uses an off-the-shelf detector, combining motion cues to produce the final segmentation. This approach mitigates typical problems associated with the weakening of supervised and unsupervised video segmentation; for instance, object scenes without motion, persistent camera motion, and objects moving as an individual entity. This method provides better accuracy and temporally consistent segmentation for individual objects. The results were presented on four video segmentation datasets: YouTube object, SecTracv2, egoMotion, and FBMS.

Cheung and Kamath [19] compared numerous background subtraction algorithms for detecting moving vehicles against pedestrians in urban traffic video sequences. The approach considers different techniques ranging from simple, for instance, frame differencing and adaptive median filtering, to more complex probabilistic modeling techniques. Complex technique times have been previously proven to indicate excellent performance results. However, the authors demonstrated that simple techniques, such as adaptive median filtering, can provide better results with minimal computation.

Rao and Satyanarayana [20] presented an approach to single-object tracking from sequences of frames using a live camera or saved videos. In the proposed approach, the median approximation technique is used to detect moving objects frame-by-frame with a high level of accuracy and efficiency. As the object is detected, Kalman filtering and a template matching algorithm are used to accurately track it. To guarantee any changes that may hinder the tracking process, a template is generated dynamically. The benefits of this approach include cost-effectiveness, videoconferencing, and surveillance. The experimental results prove the correctness and high performance of the proposed approach.

Another proposal for object detection and tracking using background subtraction was presented by Malik et al. [21]. The approach uses a background subtraction technique that uses original images to apply color and brightness distortions. This allows the object to be subtracted and tracked using the connected-component labeling method. This procedure enables the image shadow to be removed, yielding 79 % accuracy.

A performance comparison of different background subtraction algorithms was performed by Alawi et al. [22]. The study investigated techniques of frame differencing and approximate median filtering. In addition, a more complex probabilistic modeling method was investigated. The authors demonstrated that a common technology, such as an approximation median filtering method, can yield better results with minimal computational requirements.

A robust and efficient system for moving-object detection and tracking in traffic monitoring and surveillance applications was proposed by Cucchiara et al. [23]. The system incorporates a statistical and knowledge-based background update method and HSV color information to suppress image shadowing. To track objects, a module that employs a symbolic reasoning method is used. This allows the system to be used in various application domains.

Hussain and Al Balushi [24] presented an approach to real-time face emotion classification and recognition employing a deep learning model. This approach solves the problem of unreliable detection by gathering candidates from the outputs of both the detection and tracking processes. To select meaningful numbers of candidates in real time, a scoring function using a fully convolutional neural network was used to perform the computation on the entire image. In addition, the approach uses deep learning appearance representation based on trained large-scale person re-identification datasets to enhance the identification capabilities of the proposed tracker. The experimental results demonstrated the benefits of their findings.

Another interesting approach to simple online and real-time object tracking was proposed by Bewley et al. [25]. Its main goal is to efficiently join objects for use in online and real-time applications. The tracking performance is influenced by the detection quality; changing the detector can yield a tracking performance of 18.9 %. The proposed tracker has an update rate of 260 Hz making it achieve a 20x performance speed and accuracy compared with state-of-the-art trackers in the literature.

Kim et al. [26] revisited Multiple Hypothesis Tracking (MHT) algorithms. This study demonstrated that classical MHT algorithms can provide excellent performance compared with state-of-the-art approaches on standard datasets. A method for training online appearance models for individual track hypotheses was presented. The approach demonstrated that appearance models can be better learned efficiently using a regularized least-squares framework, requiring fewer operations for an individual hypothesis. Results based on state-of-the-art tracking by detection datasets, namely PETS and MOT, were provided.

Fang et al. [27] proposed an approach to the temporal generative modeling framework to describe the appearance and dynamic motion of multiple objects within a specified time duration. This approach employs the concept of recurrent autoregressive networks, comprising both internal and external memory modules. External memory is used to store the short-term inputs of individual trajectories in a slated timeframe, while internal memory is used to learn and highlight the long-term tracking history and join them together. Experimental results based on the MOT15 and MOT16 dataset benchmarks were provided, demonstrating the benefits of the proposed approach.

Chu et al. [28] proposed an instance-aware tracker that combines single object tracking (SOT) with multiple object tracking (MOT) using the encoding awareness technique to track and target objects within and between models. The approach constructs individual target models by fusing information to determine target objects arising from the background and other tracking instances. To preserve the oneness of the entire target model, the proposed instance-aware tracker uses response maps from the entire target model and associates the special location to enhance the accuracy of the entire object. In addition, the proposed method demonstrates a dynamic model refreshing technique learned using CNN. This allows the elimination of initialization noise and the possibility of adapting to changes in target size and appearance. Experimental results based on the MOT15 and MOT16 datasets were provided and demonstrated the best performance compared with previous studies.

Bae and Yoon [29] proposed an online multi-object tracking method using confidence-based data association and discriminative deep appearance learning techniques. The procedure includes the definition of the tracklet confidence by employing the features of detectability and continuity of the tracklet, and subdivision of the multi-object tracking problem into smaller units as per the tracklet confidence value. The experimental results demonstrated the usefulness of the approach compared with state-of-the-art batch and online techniques.

Amir et al. [30] presented another approach to online tracking. This approach demonstrated an online technique that encoded long-term temporal dependencies within several cues. The complexity of the tracking method lies in its ability to accurately track occluded object targets or objects that possess the same properties as those of nearby objects. To mitigate this problem, the approach presented a scheme of Recurrent Neural Networks (RNN) that combines reasoning based on several cues over a temporal window. This approach provides the means to correct data-associated errors and the ability to obtain observations from occluded states. The authors demonstrated the robustness of their approach by tracking multiple targets based on features, such as motion, appearance, and interaction. The results demonstrated the performance of the proposed approach compared to previous approaches based on the MOT challenge benchmark.

A solution to overcome the problem of unreliable detection and real-time multiple people tracking with deeply learned candidate

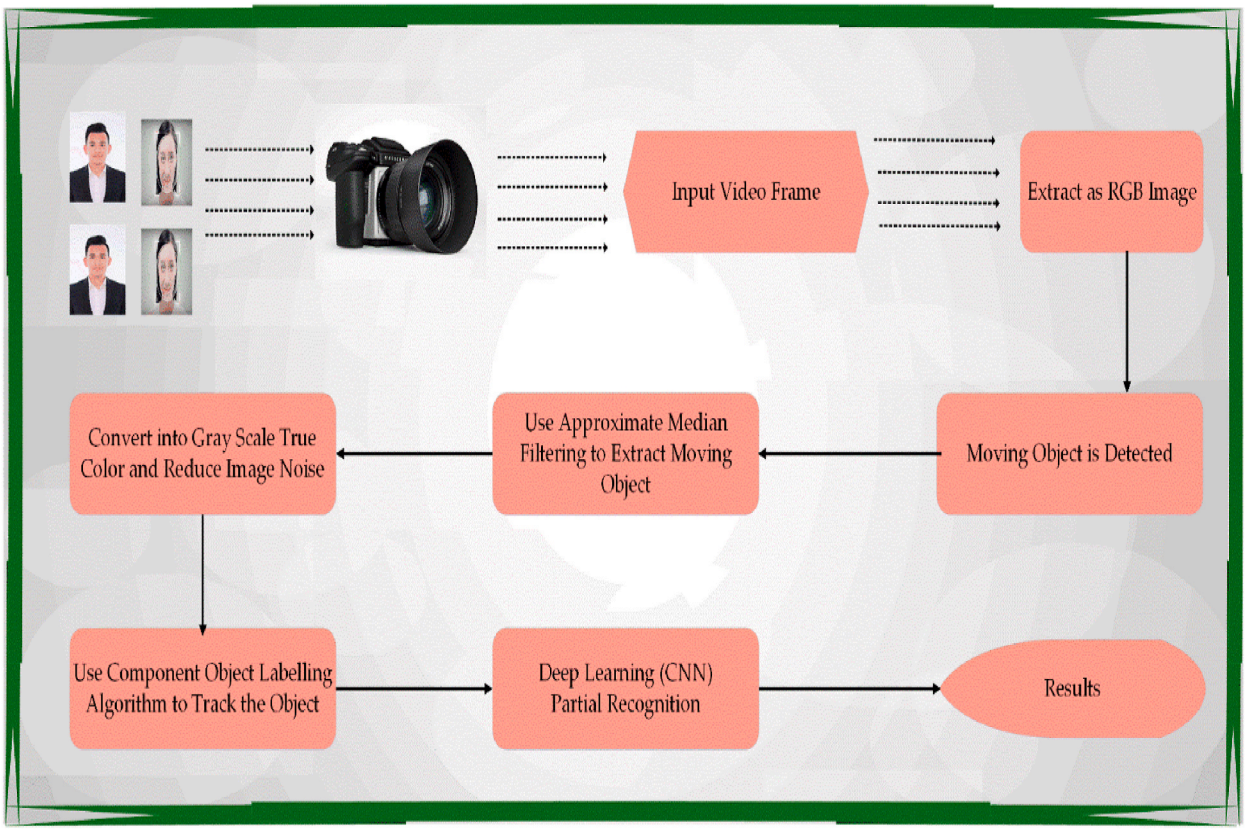


Fig. 1. Block diagram of the proposed framework.

selection and person re-identification was proposed by Chen et al. [31]. The approach employs optimal selection techniques in real time to select candidates based on the scoring function, and CNN to balance the computation on the whole images. In addition, a deeply learned appearance representation based on trained large-scale person re-identification datasets is used to enhance the capabilities of the proposed tracker. Extensive experimental results based on people's identification datasets demonstrated the real performance of their approach.

Sheng et al. [32] presented an approach to heterogeneous association graph fusion for target association in multiple object tracking. This approach uses a heterogeneous association graph to fuse high-level detection and low-level evidence to associate the target of the object. The procedure involves the use of a fused association graph to construct track trees and provide solutions using the multiple hypothesis tracking framework for its well-known tree pruning capability. Moreover, an adaptive weight assignment technique was presented to provide the benefits of object motion and appearance. The approach was evaluated based on the MOT challenge benchmarks and produced better results using the MOT17 dataset.

Kim et al. [33] presented a similarity mapping method with an improved Siamese network for multi-object tracking systems. The idea is based on reducing system complications and the number of hyperparameters required to be turned for a given environment. The system uses both the object's appearance and geometric information and provides end-to-end training capability. Results based on the MOT16 and KITTI benchmarks were compared with state-of-the-art approaches.

Another approach to multiple-object tracking via the Feature Pyramid Siamese Network (FPSN) was provided by Lee and Kim [34]. The main goal was to address the problems associated with conventional MOT metrics. A modified MOT scheme was presented to mitigate the problem of conventional MOT metrics. The idea is based on the FPSN to solve structural simplicity and multiple-level discriminative features. In addition, a spatiotemporal motion was provided to mitigate the effect of the lack of motion information and to improve the MOT metrics performance. A performance evaluation was performed to compare the proposed FPSN-MOT with the conventional MOT challenge metrics.

Another approach for online multiple human tracking using deep discriminative correlation matching was presented by Fu et al. [35]. The approach employs a matching scheme that takes advantage of the CNN and a discriminative correlation filter (DCF) serving as a target classifier to target the discriminative required target from the background and other surrounding targets. The extracted features are the outputs of the final convolutional layers learned using DCFs. The convolutional layers provide the means to encode the target appearance that yields better discrimination and strength to changes in appearance. In addition, a likelihood function is used to fuse the spatiotemporal relationship and provide a correlation-matching score to enhance the association levels. Results based on the MOT17 challenge datasets in comparison with state-of-the-art approaches were presented and demonstrated the benefits of the

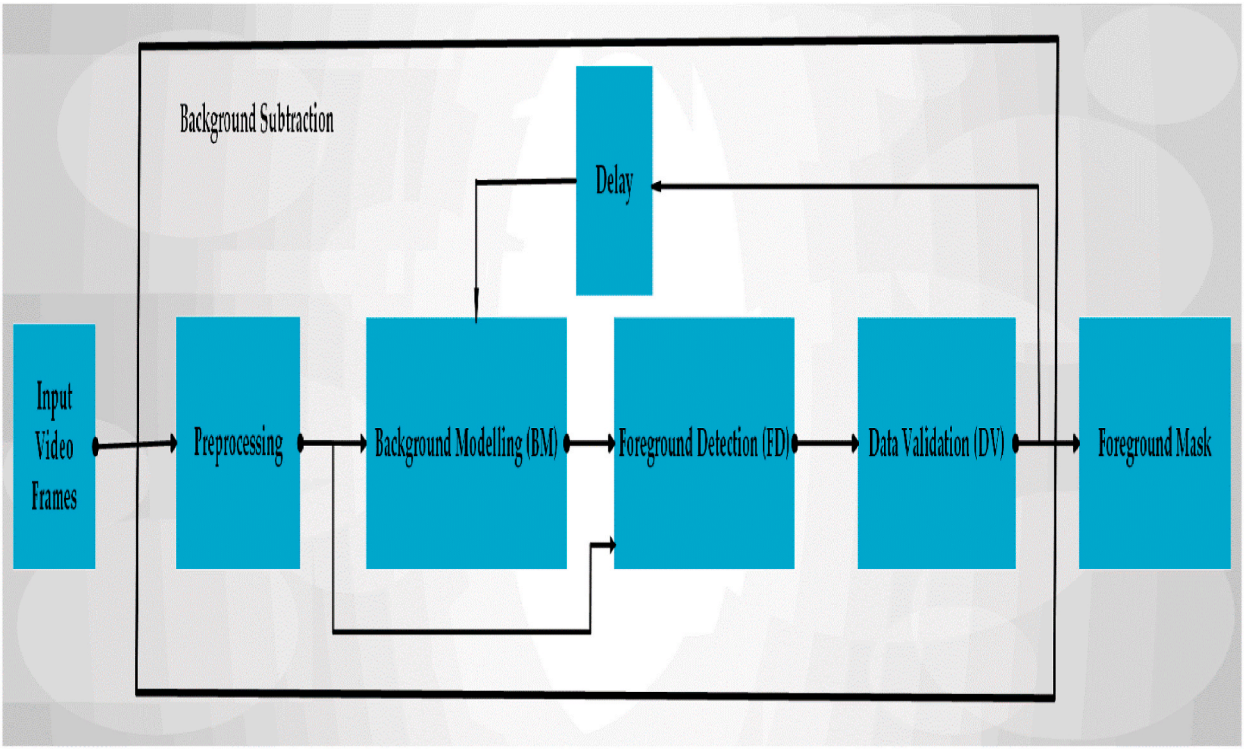


Fig. 2. Background subtraction algorithm (BSA).

proposed approach.

Despite the contributions of the above-mentioned respected authors, this study aims to provide a framework for the detection, monitoring, and surveillance of objects for security monitoring in a surveillance system. It illustrates how machine learning techniques can be used to overcome security challenges through video surveillance. These research results will be useful for (military) security personnel in detecting, tracking, and monitoring dynamic objects. It can also be used for security surveillance in public places by most organizations. In addition, research results may assist in monitoring the movement of objects in a security surveillance system.

3. Design of the proposed framework

The proposed framework uses background subtraction, component object labeling, approximate median filters, and, a Convolutional Neural Network (CNN), as shown in Fig. 1.

3.1. Background subtraction algorithm (BSA)

A critical task in video surveillance is the identification of moving objects from video sequences. These tasks include object identification in traffic monitoring and analysis, human detection and tracking, and gesture recognition at the human–machine interface. Background subtraction is a well-established method for identifying moving objects. This method compares each video frame with a reference background model. Any pixel in the current frame that deviates significantly from the background is regarded as a moving object. These pixels are known as foreground pixels and are further processed for object localization and tracking. Background subtraction is the most frequently performed first step in several computer-vision applications; therefore, the extracted foreground pixels must agree accurately with the moving objects of interest. Although several background-subtraction algorithms have been presented in the literature, the problem of identifying moving objects in a multifaceted environment remains challenging [19].

An effective *BSA* should be capable of correctly solving multiple problems. For instance, in an outdoor environment, considering a video sequence from a motionless camera overseeing a traffic intersection, a BSA should dynamically adapt to several illumination levels at different times of the day and be able to address harsh weather conditions, such as fog or snow, which change the background. To extract consistent features from objects during the subsequent processing, the shadows cast by moving objects should be changed. The complex traffic flow at intersections poses challenges to the BSA. In a road traffic-light scenario, vehicles travel at a normal speed when the light is green and then stop when it turns red. The vehicles remain stationary until the light turns green again. A good BSA must be able to address moving objects that first combine with the background and then become the foreground at a later time. Furthermore, to satisfy the real-time requirements of many applications, a BSA must be computationally inexpensive and demand low memory requirements while still being able to accurately identify moving objects in a video [19].

3.1.1. Basics of BSA

Different variations of the BSA exist; however, most adhere to a basic system flow, as shown in Fig. 2. As presented in the figure the four primary steps of the *BSA* are preprocessing, background modeling, foreground detection, and data validation. Each step is explained as follows [19].

i) The preprocessing step comprises different image-processing tasks that modify the raw input video into a form that can be acted upon in successive steps.
ii) The background-modeling step obtains a new video frame and acts upon it to compute and update the background model. Thus, this process provides a statistical description of the complete background scene.
iii) The foreground-detection step identifies each pixel in the video frame that is not clearly described by the background model. These identified pixels are output as binary entrant foreground masks.
iv) The data-validation step inspects the entrant mask, removes pixels that do not agree with actual moving objects, and yields the final mask. Domain knowledge and computationally intensive vision algorithms are used for validation. Additionally, real-time data processing is possible because these sophisticated algorithms are acted upon only on a few entrant foreground pixels.

3.2. Connected-component labeling (C-CL)

C-CL is a technique that includes associating a unique label with all pixels of individually connected components in a binary image (i.e., each object). It is crucial for characterizing different objects in a binary image and is a basic criterion for achieving better image analysis and object recognition in images. Hence, C-CL is among the most essential procedures for image analysis, image understanding, pattern recognition, and computer vision [36–38].

Considering an $N\,x\,N$ sized binary image, let the coordinate of the pixel at the location (x, y), where $0 \leq x \leq N-1$ and $0 \leq y \leq N-1$, be represented as $b\ (x, y)$. The same convention can be used to represent the value of a problem clearly. In other words, foreground pixels are similarly described as object pixels. In this study, we assume that the values of the object pixels and corresponding background pixels are set to 1 and 0, respectively. Furthermore, for clarity and ease, we assume that all pixels on the edge of an image are background pixels.

Therefore, for a pixel represented as $b\ (x, y)$, the four pixels $b\ (x-1, y)$, $b\ (x, y-1)$, $b\ (x+1, y)$, and $b\ (x, y+1)$ are named the *4-neighbors* of the pixel. Additionally, the four neighbors combined with the four pixels $b\ (x-1, y-1)$, $b\ (x+1, y-1)$, $b\ (x-1, y+1)$, and $b\ (x+1, y+1)$ are termed as the *8-neighbors* of the pixel. Assume that two object pixels p and q are termed as 8-connected (4-connected) if a path exists that contains object pixels $a1, a2, \ldots, an$ such that $a1 = p$ and $an = q$, and that for all $1 \leq i \leq n-1$, ai and $ai+1$ are 8-neighbor (4-neighbor) among themselves [36].

An image can be transformed into its corresponding binary image, where the pixels of the object to be recognized are transformed into pixels (object pixels), and the remaining pixels are transformed into background pixels. Therefore, to differentiate different objects in a binary image, C-CL is an essential procedure that involves assigning a unique label to all pixels of each object in the image for object detection, tracking, and recognition applications. Subsequently, the binary image is transformed into a labeled image. C-CL is time consuming because the connected components in an image may have complicated geometric shapes and complex connectivities. Most importantly, the labeling process cannot be completed via a simple parallel local operation; in fact, it requires sequential operations [39].

3.3. Approximate median filtering (AMF)

A grayscale image is an essential representation of image formation that combines black, white, and gray colors and can be used in object detection, tracking, and recognition. However, owing to the rapid and dynamic changes in environmental parameters and vision device accuracy, the actual image obtained continuously contains noise when processed from a grayscale image. These noises include Gaussian, salt-and-pepper, and other noises. The salt-and-pepper noise is a form of image impulse noise used to simulate imaging sensors or signal transmission errors that yield isolated bright or dark spots in images. Consequently, salt-and-pepper noise weakens the imaging-definition and visualization effects and reduces the accuracy of image segmentation, edge detection, and object identification. Hence, to ensure restoration of the original details of an image and eliminate discontinuities, the effects of noise associated with grayscale images must be reduced [40].

AMF is a nonlinear signal-processing algorithm that relies on statistics. In digital images, the noise value is changed by the median value of the neighborhood mask. This algorithm employs the correlation of an image to generate the features of the filtering mask on the image. This process adaptively scales a mask based on its noise levels. The pixels of the mask are ordered in the sequence of their gray-value levels, and the median value of the neighborhood is stored as a substitute for the noisy value [41].

Assume that the median-filtering output is expressed as $g\ (x, y) = \text{med}\ \{f\ (x - i, y - j), i, j \in W\}$, where f (x, y) and g (x, y) denote the original and output images, respectively, and the parameter W refers to the two-dimensional mask. The size of W is n x n, where n is typically expressed as an odd number. The structure of the mask can be linear, square, cross shaped, or circular.

The median-filtering algorithm offers an excellent noise-reducing effect; however, its time complexity is a significant challenge. Image noises may be presented during the capture and transmission processes, and they can be categorized as Gaussian, balanced, or impulse. Impulse noise in an image typically appears as light- and dark-noise pixels under a random distribution. This ultimately corrupts the actual image information and distorts the image's visual effects. Hence, removing impulse noise is crucial in object detection, tracking, and recognition. When an image is corrupted by noise, a linear or nonlinear filter method can be used to reduce

Algorithm 1: Multiple Object Detection Procedure

Input: Video frames $V_{Frames} = I_1, I_2, \ldots, I_{NoFrames}$
Output: Moving Object Detected

```
For I = 1 to NoFrames
Get the First Frame VFrames (I1) from BackgroundImage
Set ImagePixel = 1 to NoFrames
Set Difference = PixelValues - BackgroundImage
Perform Deep Learning Validation and Classifier
If      Difference > ThresholdValue Then
        {       Set Background = Foreground
                Set ImagePixel = 1 to Background
                }
Else:
        {       Set Foreground = 0
                }
If Foreground > Background Then
        {       Set Background = Background + 1
                Perform Deep Learning Validation and Classifier
                Perform Morphological Operation to Get TrueForeground
                }
Else
        {       Set Background = Background -1
                }
Increment I
```

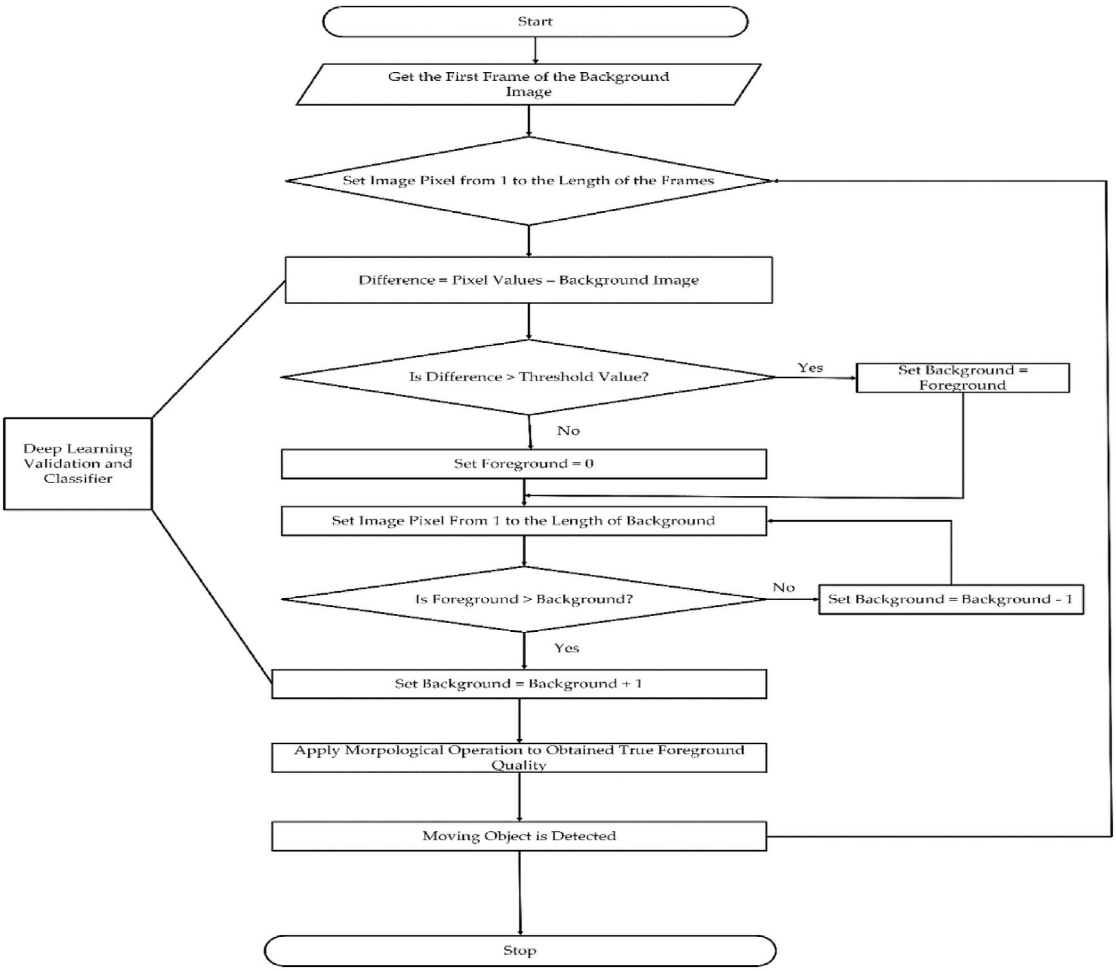


Fig. 3. Flowchart diagram of object detection procedure.

Algorithm 2: Multiple Object Tracking Procedure

Input: $Binary\ Images\ Binary_{Images} = I_1, I_2, \ldots, I_{LengthOfDetectedObjects}$

Output: *Tracked Objects*

```
For I = 1 to I_LengthOfDetectedObjects
Get Binary_Image
Scan Image = 1 to LengthOfDetected_Objects
Perform Deep Learning Validation and Classifier
If      Neighbor Labels = 0 Then
        {       Assign New Labels
                Find NumberOf_Objects
                Track Objects
                }
Else:
        {       // Labels are different
                Copy Highest_Value
                Find NumberOf_Objects
                Track Objects
                }
Increment I
```

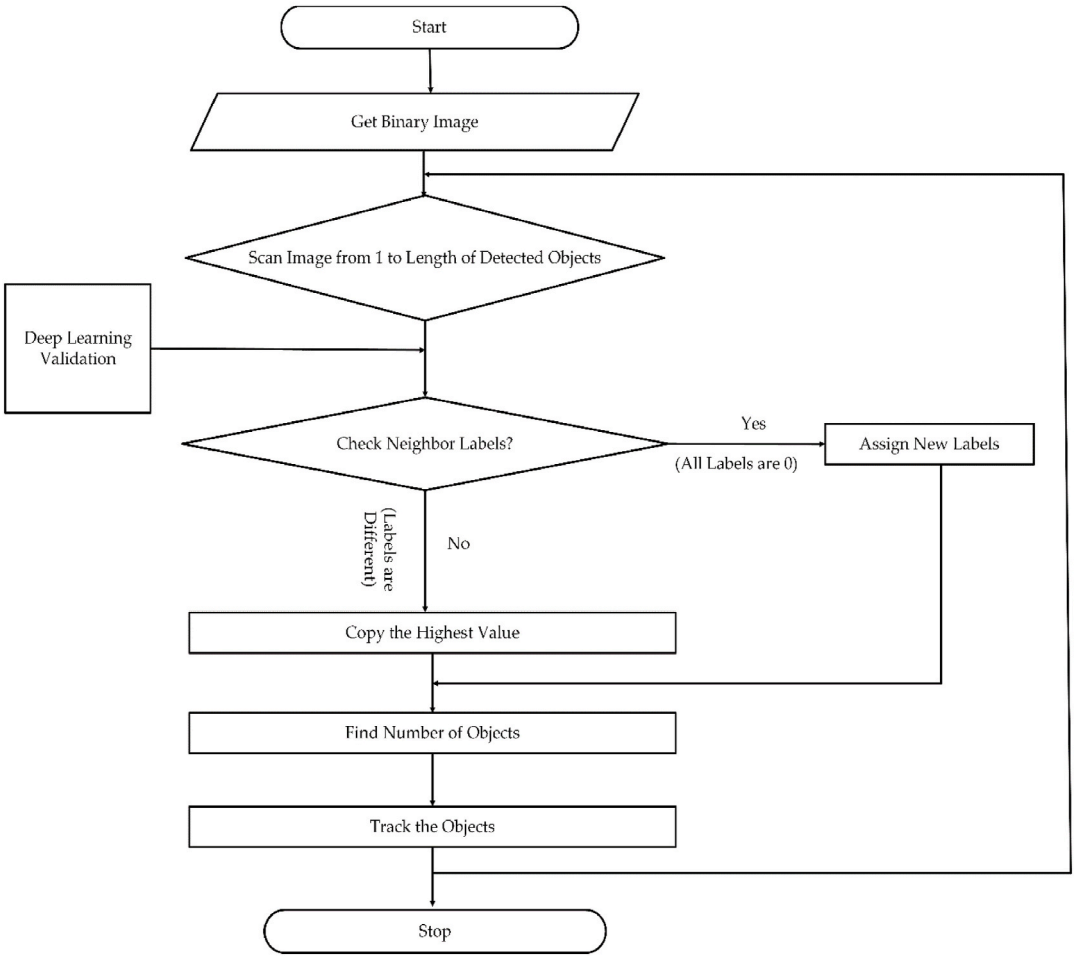


Fig. 4. Flowchart diagram of object tracking procedure.

Algorithm 3: Multiple Object Recognition Procedure

Input: Binary Images $Binary_{Images} = I_1, I_2, \ldots, I_{LengthOfDetectedObjects}$
Output: Face Recognition

```
For I = 1 to I_LengthOfDetectedObjects
Get Binary_Image
Detect Multiple_Faces
Scan Image = 1 to LengthOfDetected_Objects
If      Multiple_Faces is Detected Then
        {       Detect More Multiple_Faces
                Perform Deep Learning Validation and Classifier
                }
Else If NumberOf_Faces = Object Then
        {       Face Recognition
                Perform Network Training
                Compute Results
                }
Else:
        {       Detect More Multiple_Faces
                }
Increment I
```

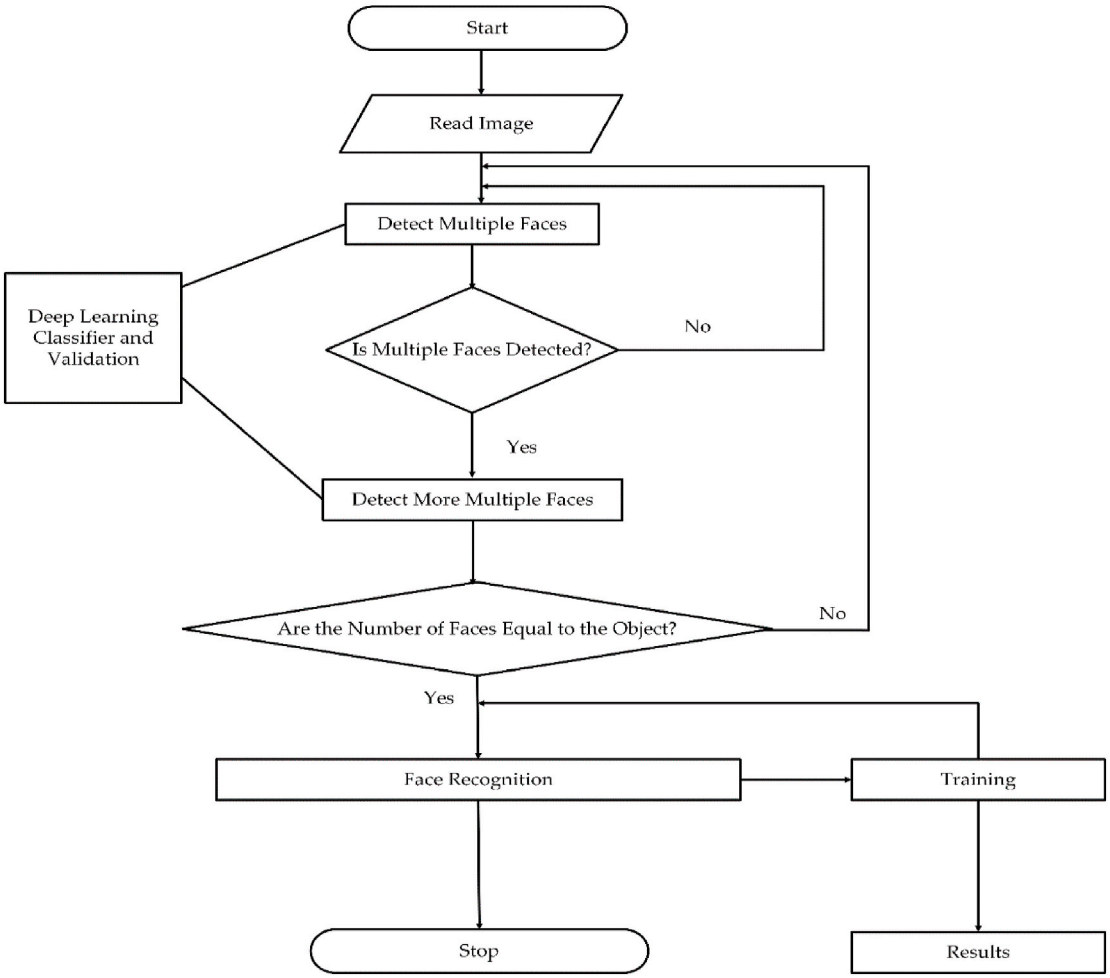


Fig. 5. Flowchart diagram of object recognition procedure.

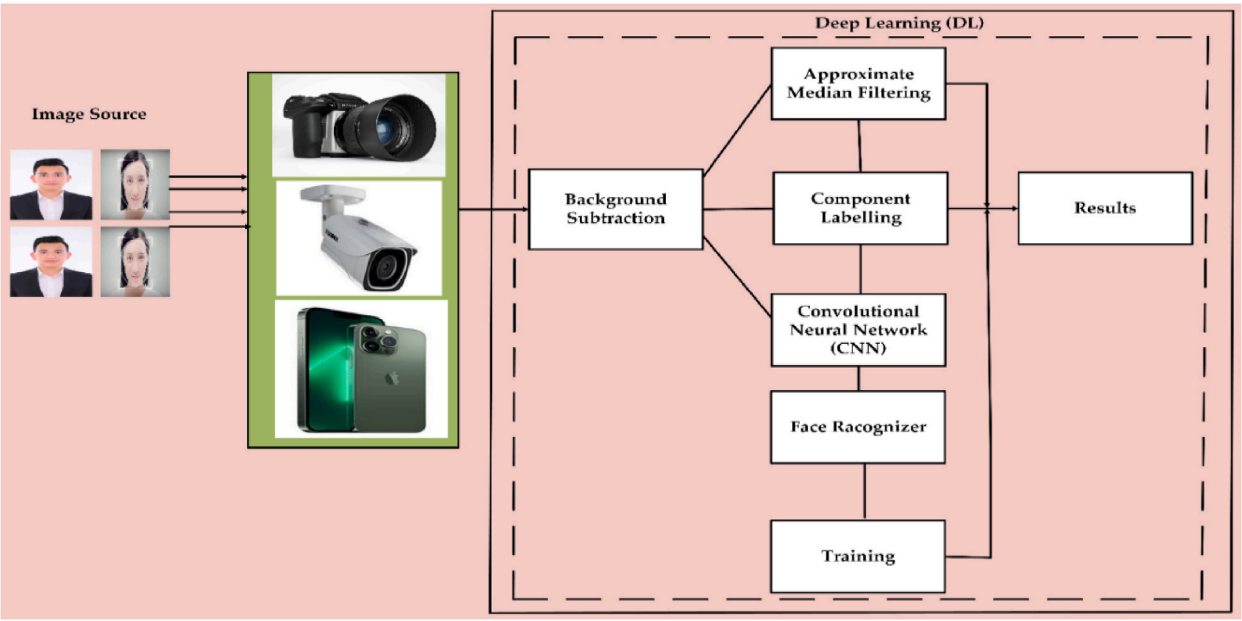


Fig. 6. Block diagram of a deep learning technique.

noise. The AMF is a nonlinear filter that is typically used in digital applications and is widely used in object detection, tracking, and recognition problems owing to its excellent edge-preserving features and ability to reduce impulse noise. The AMF is a nonlinear filter; therefore, its mathematical analysis is complex for images with random noise. To formulate an analysis for an image with zero mean noise under a normal distribution, the noise variance of the AMF is expressed as shown in Equation (1) [41].

$$\sigma_{med}^2 = \frac{1}{4n f^2(\overline{n})} \approx \frac{\sigma_i^2}{n + \frac{\pi}{2} - 1} \cdot \frac{\pi}{2}, \tag{1}$$

where.

σ_i^2 is the input noise power (the variance),

n is the size of the median-filtering mask, and

$f(n)$ is a function representing the noise density.

Therefore, the noise variance of the average filtering is expressed as shown in Equation (2) [41].

$$\sigma_o^2 = \frac{1}{n}\sigma_i^2 \tag{2}$$

Considering Equations (1) and (2), one can deduce that the median-filtering effects depend on two factors: 1) the mask size and 2) noise distribution. The median-filtering noise removal of random noise is significantly better than the average filtering performance. However, under impulse noise, the narrow pulses are farther apart and the pulse width is less than $\frac{n}{2}$; in this case, the median filter is highly effective. The effectiveness of AMF can be improved by combining both median and average filtering to adaptively resize the mask based on the noise density levels [40,41].

3.4. Video surveillance systems

Digital cameras, surveillance monitoring, and control software frameworks are promising solutions for automatic surveillance systems used to observe and monitor environments. The observed scenes and situations are analyzed using individual behavior, crowd behavior, interactions between individuals, motion detection, crowds, and their neighboring environments. The design and implementation of these automatic systems enables multiple tasks, including detection, interpretation, understanding, recording, and creating alarms as a result of the system analysis [42].

Over the past two decades, significant developments been achieved in different areas worldwide, owing to which life has become multifaceted in diverse aspects, including the safety and security of people. Therefore, monitoring and these aspects have become mandatory. In this regard, surveillance software frameworks and cameras installed in both public and private locations are appropriate solutions for ensuring comfort and safety. Humans are typically tasked to observe these cameras continuously 24 h a day, which is a tedious and expensive process. Therefore, an automated system that can monitor and control real-time events under different scenarios using software frameworks and surveillance cameras is highly desirable [42].

The design and implementation of software frameworks for video surveillance is a practical solution. The primary function of an

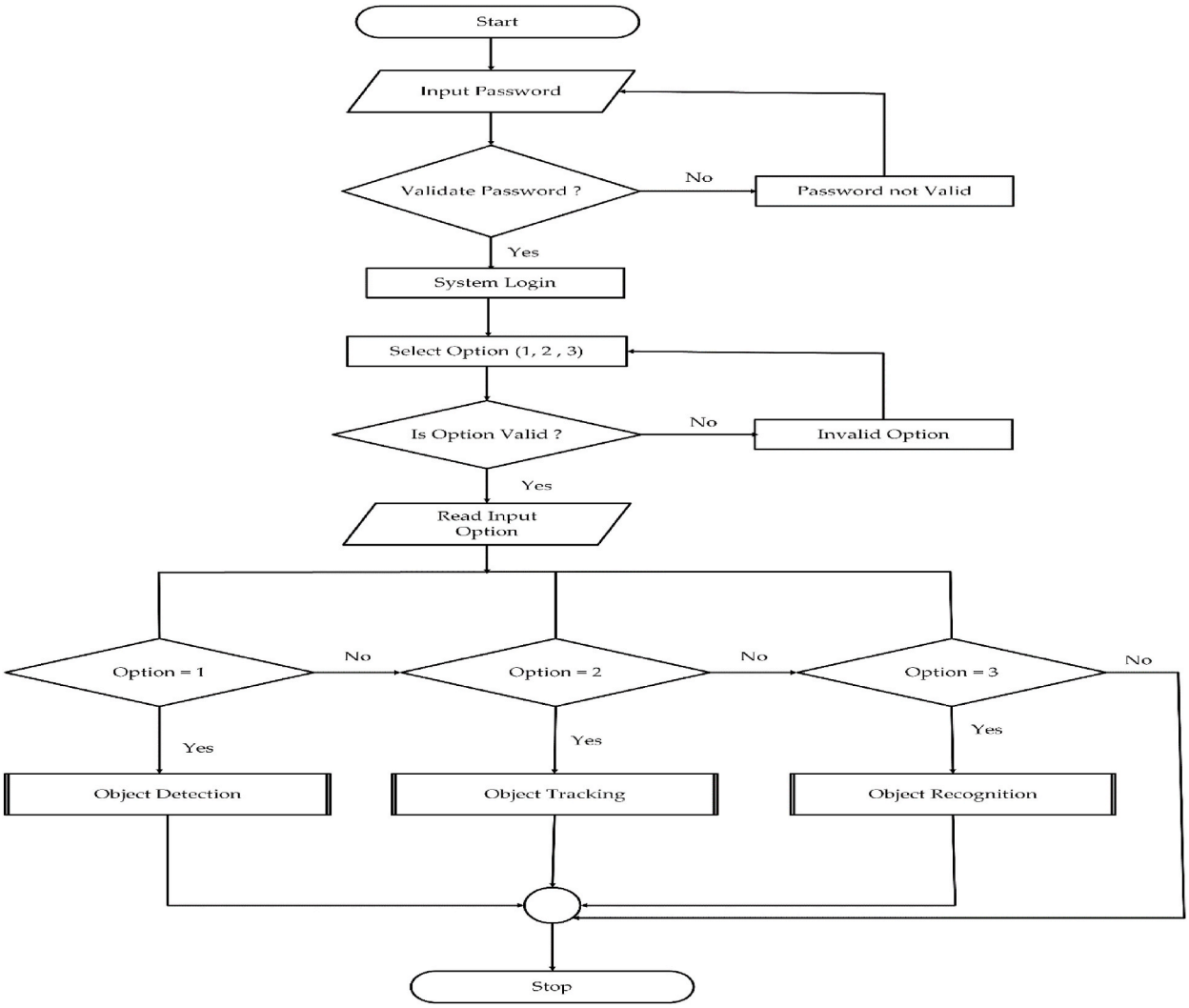


Fig. 7. The proposed framework system's logging flowchart.

automatic system is to assist security personnel in performing various tasks. The tasks to be performed can be related to different areas of application such as traffic control, accident prediction, crime prevention, motion detection, and homeland security. Additional application areas can be included to monitor indoor and outdoor scenes such as parking lots, highways, stores, shopping malls, airports, train stations, and offices. Software security and monitoring frameworks are being progressively deployed to control and prevent irregular events, particularly in situational-awareness applications, and to ensure public security. Therefore, a security-monitoring and control-software framework must be developed that can automatically monitor and control human lives [42,43].

3.5. The algorithm design

The proposed algorithms for object detection, tracking, and recognition were designed and implemented using Python and integrated with C# programming language for easy use, as they possess the programming capabilities and flexibility for handling image processing. In particular, C# contains tools for designing interactive user interfaces, whereas Python contains rich libraries and tools for efficient object detection and tracking. For example, ImageAI, YOLOV, Numpy, and OpenCV are the most commonly used Python libraries and tools used for object detection and tracking. Additionally, Python contains programming constructs for dynamically implementing machine-learning algorithms and techniques [44–47].

Algorithm 1 and Fig. 3 illustrate the algorithm and flowchart of object detection. In the first step of the flowchart, the procedure captures the first frame from the background image and repeatedly moves it to the next input frame. The pixel is set from one to the length of the frame. It then compares the image threshold values to determine the difference between the pixel intensities and the background. If the difference is greater than the threshold value, the foreground is set as the background; otherwise, it is set as zero (0). In the second step, the pixel is set from one to the length of the background. The foreground is then tested against the background. If the

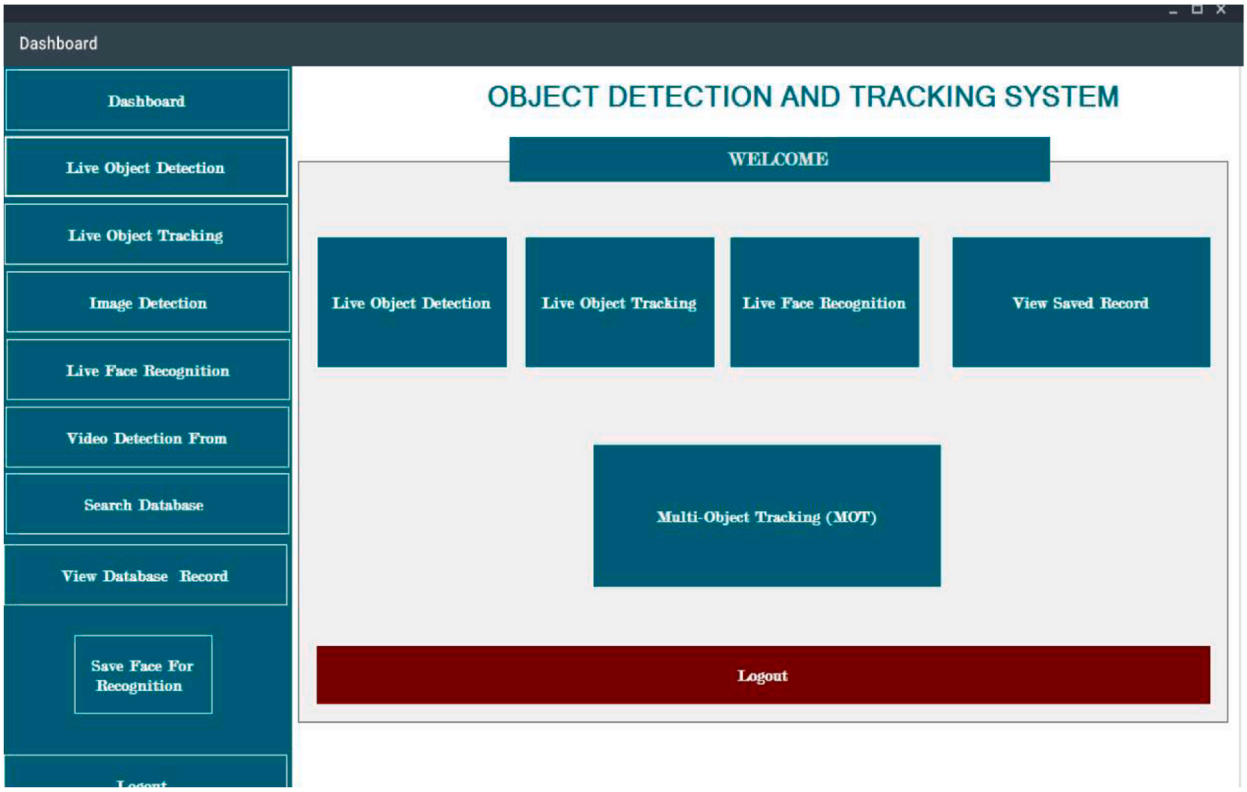


Fig. 8. The implemented software framework user interface.

foreground is greater than the background, then the background is incremented by 1, else the background is decremented by 1. The foreground quality is obtained by performing the morphological operation. Subsequently, a moving object is detected by determining the centroid of the foreground image [19–21]. The entire process is controlled and monitored using the deep learning technique namely, CNN via validation and classification, as shown in Algorithm 1 and Fig. 3.

The output of object detection from Fig. 3 and Algorithm 1 is fed as an input to Algorithm 2 and Fig. 4, and the image is read. The image is scanned from one to the length of the detected objects. The procedure assigns labels such that a new label is assigned if no neighbors have a label. If neighbors have different labels, it selects the highest value of the label and assigns it to the pixel via the CNN. When the scans are completed, the total number of objects is computed, and these objects are tracked.

Algorithm 3 and Fig. 5 present an object recognition procedure in which the algorithm captures the image of the object from the video and then uses a classifier to classify the objects based on the number of faces present via a CNN. The faces are then validated to ensure their reality. The resulting faces are recognized and then trained.

3.6. Deep learning (DL)

Currently, face-recognition techniques using deep learning or convolutional neural networks (CNNs) are highly effective in identifying people based on their facial features. The CNN model possesses kernels that detect a borderline function or the outline of an image and is characterized by weights organized in an array of values to obtain the desired characteristics. Therefore, each CNN model assigns a space to determine the control of the image to be recognized [24]. The use of datasets for general-purpose network applications has been proven challenging for deep and wide neural networks. However, CNNs have been demonstrated to be more effective for object detection and recognition. Furthermore, CNNs have transformed the fields of computer vision and audio processing [15]. For instance, smartphone devices are designed with AI-based object-recognition capabilities using the CNN architecture, thus enabling applications such as the recognition of objects in images, digital fingerprints, and voice commands to be used by end users [47]. CNNs offer excellent capabilities because they can solve some of the most complex computer-vision problems. Additionally, CNNs provide the unique ability to encode spatial relationships in datasets to extract and classify object features.

Additionally, CNN are employed in real-time video surveillance for automated weapon detection. In this approach, surveillance systems are used to monitor and classify weapons and track events in real time. In particular, three processing modules are used. First, a CNN is used for the object-detection module; second, a module is designated for weapon classification; and third, a module is designed for monitoring and alarm operations. The implemented surveillance system uses a closed-circuit television system to monitor a specified area of interest and perform basic monitoring and control functions. Two algorithms, namely shape and object-detection

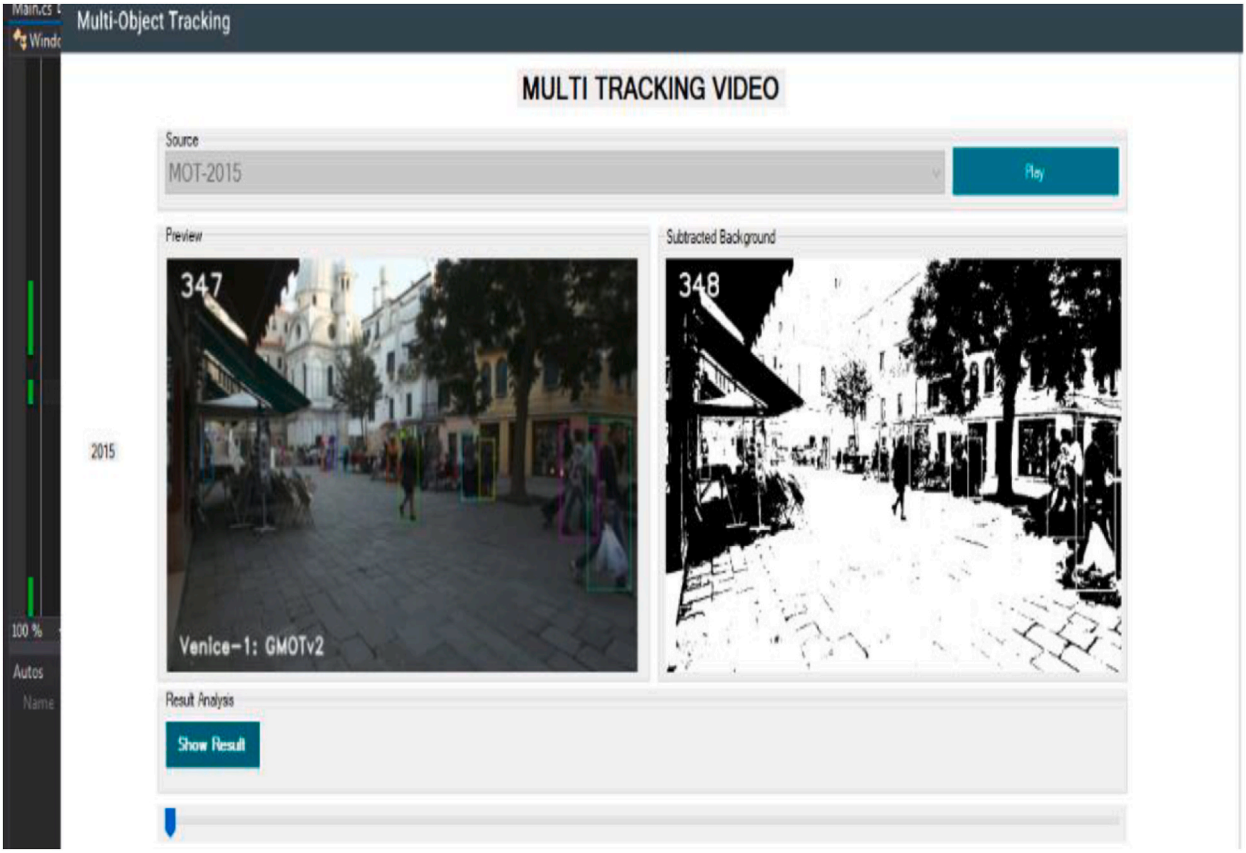


Fig. 9. Multiple-object tracking in original frame of video using MOT 15 dataset.

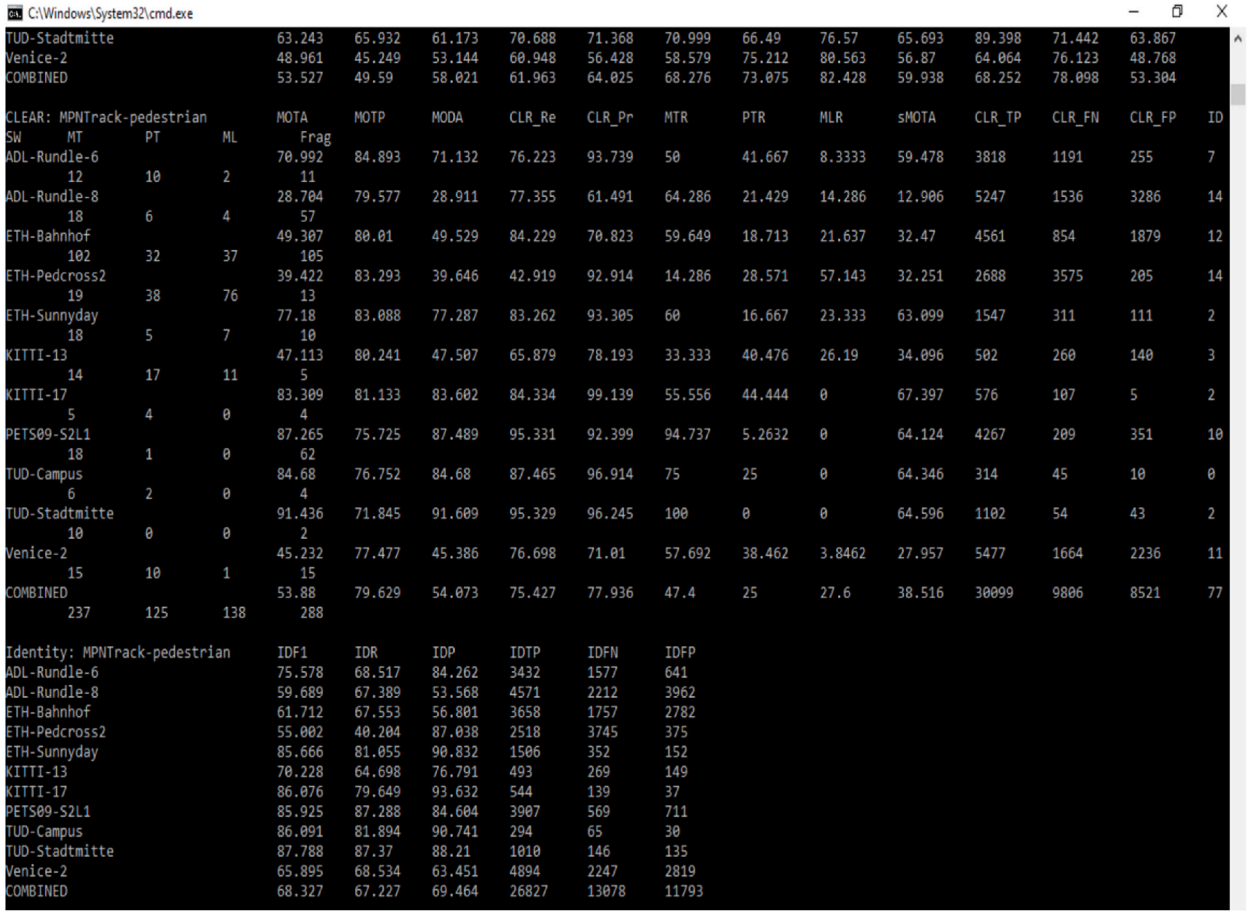
algorithms, were evaluated for their accuracy in terms of detection and processing time. The results show archived optimal accuracy in detecting weapons and object categories, names, and shapes in the ALEXNET dataset [48].

Deep residual learning for image recognition is a promising framework for simplifying the training of densely layered networks. This approach formulates the network layers as a learning residual function, where the input layers are regarded as reference points. This approach provides detailed proof that residual networks can achieve high accuracy based on a substantially increased network depth via simple optimization techniques. The ImageNet dataset was used to evaluate the residual networks with a depth of 158 layers, which is 8x deeper than that of VGG nets with reduced complexity. Experimental results showed that promising results were achieved based on a real-life application [49].

Another solution for real-time object detection and tracking is a CNN-based framework that employs real-time object detection and tracking using deep learning. This concept is based on a spatial–temporal mechanism. This approach addresses occlusion biases and target interactions. A software system that employs YOLO's technique and TensorFlow offers a better approach for real-time object detection, tracking, and counting in different datasets [50]. Therefore, a CNN was used in this study owing to the aforementioned advantages.

Fig. 6 presents a deep-learning technique for the proposed framework. As shown in Fig. 6, the algorithm captures an object using one of three devices: a digital camera, a surveillance video camera, or an iPhone. Background subtraction is performed to identify frames in the video sequence. The deepest learning algorithm layer classifies the approach to use as follows: the median filter for detection, component labeling for tracking, and the face recognizer for recognition. Furthermore, the algorithm monitors and controls the training process of the entire system. The target object is the outcome of the results.

Fig. 7 illustrates the system logging flowchart for the proposed framework. The algorithm begins by authenticating the user to secure the system against unwanted users. After a successful authentication process, a valid user can select from the following tasks: object detection, object monitroing, and recognition using the CNN. As shown in Fig. 6, a digital camera, a surveillance video camera, and an iPhone were the three main image sources of the system. The algorithm detects an object and subtracts the corresponding background for detection. Similarly, the same approach is used for tracking; however, a rectangular box is used for the tracked objects. For recognition, the faces of the objects are recognized based on training. An object is identified by its name and face using rectangular boxes, depending on the number of objects in the scene.



C:\Windows\System32\cmd.exe

TUD-Stadtmitte			63.243	65.932	61.173	70.688	71.368	70.999	66.49	76.57	65.693	89.398	71.442	63.867
Venice-2			48.961	45.249	53.144	60.948	56.428	58.579	75.212	80.563	56.87	64.064	76.123	48.768
COMBINED			53.527	49.59	58.021	61.963	64.025	68.276	73.075	82.428	59.938	68.252	78.098	53.304
CLEAR: MPNTrack-pedestrian			MOTA	MOTP	MODA	CLR_Re	CLR_Pr	MTR	PTR	MLR	sMOTA	CLR_TP	CLR_FN	CLR_FP	ID
SW	MT	PT	ML	Frag											
ADL-Rundle-6				70.992	84.893	71.132	76.223	93.739	50	41.667	8.3333	59.478	3818	1191	255	7
12	10	2		11												
ADL-Rundle-8				28.704	79.577	28.911	77.355	61.491	64.286	21.429	14.286	12.906	5247	1536	3286	14
18	6	4		57												
ETH-Bahnhof				49.307	80.01	49.529	84.229	70.823	59.649	18.713	21.637	32.47	4561	854	1879	12
102	32	37		105												
ETH-Pedcross2				39.422	83.293	39.646	42.919	92.914	14.286	28.571	57.143	32.251	2688	3575	205	14
19	38	76		13												
ETH-Sunnyday				77.18	83.088	77.287	83.262	93.305	60	16.667	23.333	63.099	1547	311	111	2
18	5	7		10												
KITTI-13				47.113	80.241	47.507	65.879	78.193	33.333	40.476	26.19	34.096	502	260	140	3
14	17	11		5												
KITTI-17				83.309	81.133	83.602	84.334	99.139	55.556	44.444	0	67.397	576	107	5	2
5	4	0		4												
PETS09-S2L1				87.265	75.725	87.489	95.331	92.399	94.737	5.2632	0	64.124	4267	209	351	10
18	1	0		62												
TUD-Campus				84.68	76.752	84.68	87.465	96.914	75	25	0	64.346	314	45	10	0
6	2	0		4												
TUD-Stadtmitte				91.436	71.845	91.609	95.329	96.245	100	0	0	64.596	1102	54	43	2
10	0	0		2												
Venice-2				45.232	77.477	45.386	76.698	71.01	57.692	38.462	3.8462	27.957	5477	1664	2236	11
15	10	1		15												
COMBINED				53.88	79.629	54.073	75.427	77.936	47.4	25	27.6	38.516	30099	9806	8521	77
237	125	138		288												

Identity: MPNTrack-pedestrian	IDF1	IDR	IDP	IDTP	IDFN	IDFP
ADL-Rundle-6	75.578	68.517	84.262	3432	1577	641
ADL-Rundle-8	59.689	67.389	53.568	4571	2212	3962
ETH-Bahnhof	61.712	67.553	56.801	3658	1757	2782
ETH-Pedcross2	55.002	40.204	87.038	2518	3745	375
ETH-Sunnyday	85.666	81.055	90.832	1506	352	152
KITTI-13	70.228	64.698	76.791	493	269	149
KITTI-17	86.076	79.649	93.632	544	139	37
PETS09-S2L1	85.925	87.288	84.604	3907	569	711
TUD-Campus	86.091	81.894	90.741	294	65	30
TUD-Stadtmitte	87.788	87.37	88.21	1010	146	135
Venice-2	65.895	68.534	63.451	4894	2247	2819
COMBINED	68.327	67.227	69.464	26827	13078	11793

Fig. 10. Runtime snapshot results of multiple-object tracking in original frame of video using MOT 15 dataset.

3.7. System implementation

Fig. 8 presents the implemented software framework for object detection, tracking, and recognition. After login, the user proceeds to the dashboard, where all functionalities of the software are defined for users' actions. On the dashboard, a user may wish to choose either to detect an object, track an object, recognize an object, or view saved records from the database, as shown in Fig. 8.

Fig. 9 illustrates the tracking of multiple objects within the original video frame using the MOT 15 dataset. Fig. 9 shows a live scene captured from the MOT 15 dataset and the subtracted background images of the scene. Multiple objects were tracked via a live video in real time. Fig. 10 shows the implemented framework at the execution time. The executed algorithms computed the MOT-Challenge metrics at runtime and yielded snapshot results of MOT in the original frame of the video using the MOT 15 dataset, as shown in Fig. 10.

Fig. 11 shows the MOT results in the original frame of the video using the MOT 16 dataset. Live scenes were captured, objects were montiored in real time, and background images were subtracted, as illustrated in Fig. 11. Fig. 12 shows the implemented framework at the execution time. The executed algorithms computed the MOT-Challenge metrics at the runtime and yielded snapshot results of MOT in the original video frame using the MOT 16 dataset, as shown in Fig. 12.

Fig. 13 illustrates MOT in the original frame of a video using the MOT 17 dataset. Fig. 14 shows the implemented framework at the execution time. The executed algorithms computed the MOT-Challenge metrics at runtime and yielded snapshot results of MOT in the original frame of the video using the MOT 16 dataset, as shown in Fig. 14.

As shown in Figs. 9–14, the performances of the proposed algorithms were computed using the MOT challenge metrics and the MOT 15, 16, and 17 datasets using live videos in real time. The performance-measurement results are presented in Figs. 10, 12 and 14 and in Tables 1–3, respectively.

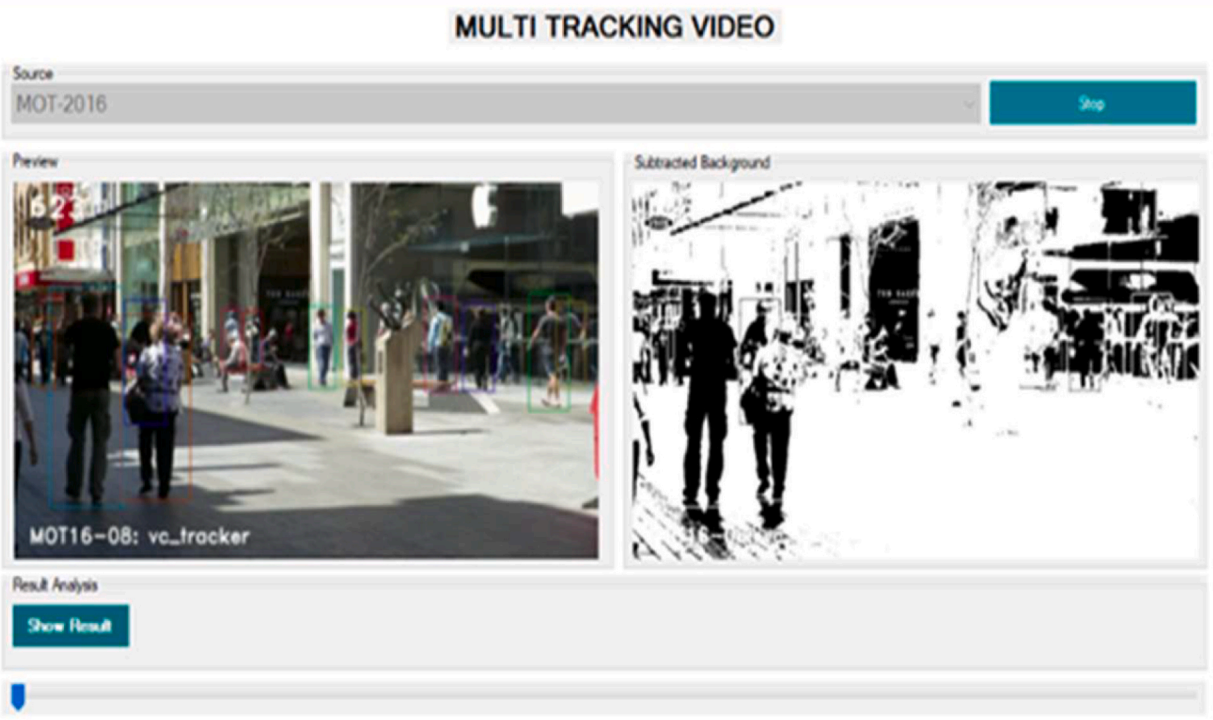


Fig. 11. Multiple-object tracking in original frame of video using MOT 16 dataset.

4. Experimental results and discussions

4.1. Experimental setup procedure

The experimentation, training, testing, and validation of the proposed framework were conducted using an Intel Core^{tm} i9 12,900 K CPU operating on a 3.2 GHz processor with 16 GB of RAM, 1 TB NVMe M.2, a solid-state drive, and an NVIDIA Quadro K4100 M GPU graphics card. Additionally, Microsoft Windows 10 and a 64-bit operating system were used. To train and test the proposed framework and network, we used the Python programming language (version 4.4.0.46) and the following libraries: Numpy version 1.18.1, SciPy version 1.4.1, Pycocotools version 2.0.2, SciKit version 0.16.2, PyTest version 6,0,1, Pillow version 8.1.2, and Matplotlib version 3.2.1.

4.2. Performance evaluation metrics

The following subsection presents the performance evaluation metrics for the proposed system as given in Ref. [14], and [16].

Mostly Tracked (MT) trajectories: These are defined as the number of ground-truth trajectories that are correctly tracked in at least 80 % of the frames.

Most Lost (ML) trajectories: This is defined as the number of ground-truth trajectories that are correctly tracked in less than 20 % of the frames.

ID switches (IDF1): This is defined as the number of times the object is correctly tracked, but the associated ID for the object is mistakenly changed.

False Positive (FP): This parameter is defined as the number of false positives in the entire video.

False Negative (FN): This parameter is defined as the number of false negatives in the entire video.

Multiple Object Tracking Accuracy (MOTA): This parameter records the number of all object configuration errors made by the tracker, false positives, misses, and mismatches over all frames.

Multiple Object Tracking Precision (MOTP): This parameter defines the total error in the estimated position for matched object-hypothesis pairs over all frames, averaged by the total number of matches made.

4.3. Training and optimization

The min batch gradient descent algorithm was used along with 3002 training sample images, a batch size of 32, and 500 epochs. To train the network, 32 samples were passed through all 3002 samples, and 94 iterations were required to obtain one epoch. This process was repeated 500 times (i.e., epochs). The learning rate was set to a default value of 0.001 based on the Adam class optimizer. The software development, including the design of the framework user interfaces, was coded in the C# programming language and

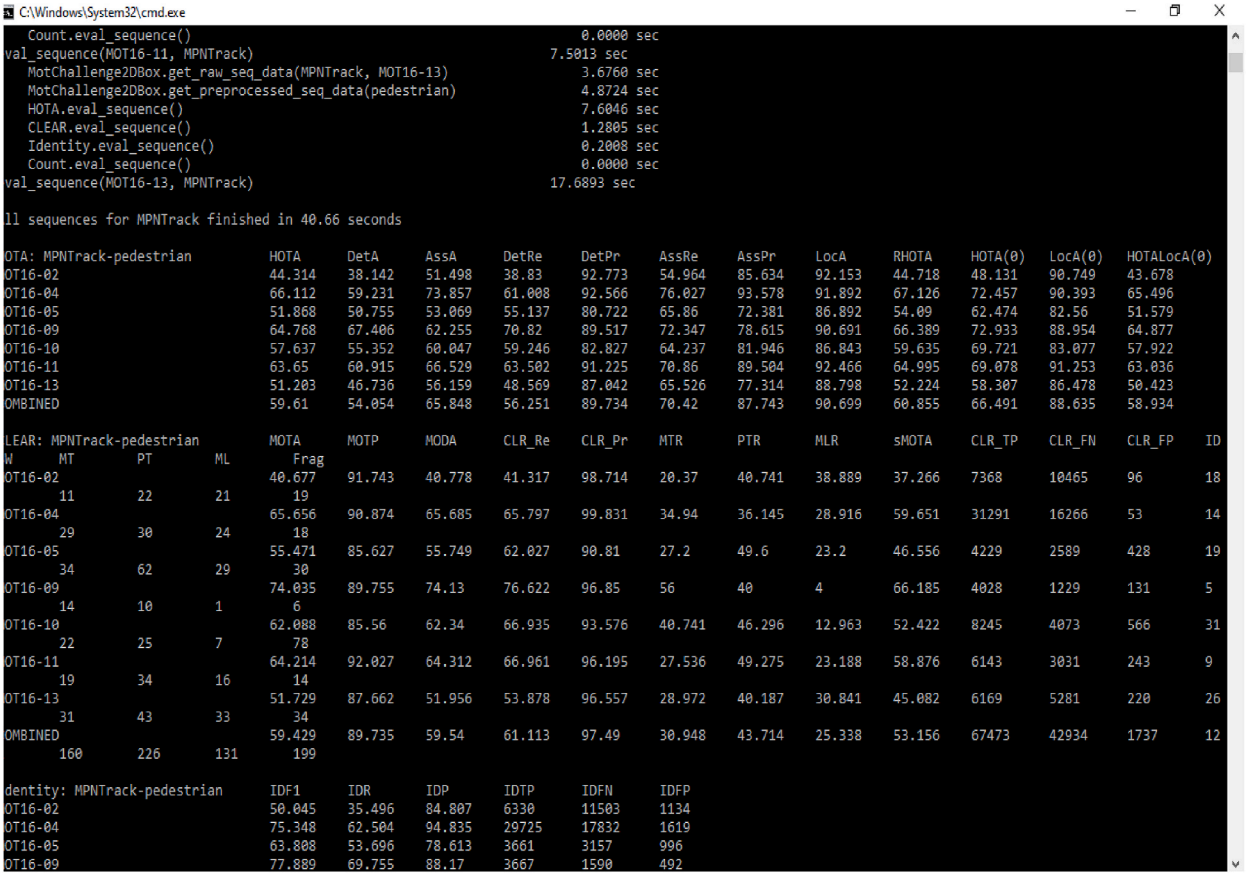


Fig. 12. Runtime snapshot results of multiple-object tracking in original frame of video using MOT 16 dataset.

integrated with Python using Microsoft Visual Studio (2019 edition). The integration was performed for ease of use, and the framework was executed as a standard software application, as shown in Fig. 7.

4.4. Results and discussion of MOT performance based on compared with state-of-the-art approaches

The proposed algorithms were tested on different Multiple Object Tracking (MOT) videos using the MOT15, MOT16, and MOT17 datasets [51]. The MOT metrics were computed by the algorithms, and the results obtained were evaluated against the works of different authors, as provided in Tables 1–3. The tables provide information regarding the authors' papers and the computed performance metrics. The following performance metrics based on multi-object tracking were used: (1) Mostly Tracked (MT), (2) Mostly Lost (ML), (3) ID Switches (IDF1), (4) False Positives (FP), (5) False Negatives (FN), (6) Multiple Object Tracking Accuracy (MOTA), (7) Multiple Object Tracking Precision (MOTP) as defined in Section 4.1. To understand the information contained in the tables, an upward arrow (↑) indicates that a higher score is better; in contrast, a downward arrow (↓) indicates that a lower score is better.

Table 1 provides the results of the comparison using the multiple object tracking (MOT15) datasets. As shown in Table 1, the proposed algorithms can track multiple objects from streams of video in a public detection scenario using the MOT15 dataset with the multiple objects tracking accuracy (MOTA) metric value of 53.9 % as against the works of Kim et al., Bewley et al., Bae and Yoon, and Chu et al. with MOTA values of 32.4 %, 33.4 %, 51.3 %, and 38.9 %, respectively. It can also be observed that using the multiple object precision metric (MOTP) the proposed algorithms provide a 75.8 % MOTP precision as against the works of Kim et al., Bewley et al., Bae and Yoon, Fang et al., and Chu et al. with 71.8 %, 72.1 %,74.2 %, 73.0 %, and 70.6 % MOTP precision values, respectively. It can also be observed that under the mostly tracked (MT) metric, the proposed algorithm provides an 18.0 % MT value compared to Kim et al., Bewley et al., Bae and Yoon, Fang et al., and Chu et al. with MT values of 16.0 %, 11.7 %, 36.3 %, 45.1 %, and 16.6 %, respectively. Under the IDF1 switch metric evaluation, the proposed algorithm provides a 55.0 % IDF1 value as against the works of Kim et al., Bewley et al., Bae and Yoon, Fang et al., and Chu et al. with IDF1 values of 45.3 %, 40.4 %, 54.1 %, 61.3 %, and a 44.5 %, respectively. It can also be observed that under the mostly lost (ML) metric, the proposed algorithms provided an ML value of 37.0 % as against the works of Kim et al., Bewley et al., Bae and Yoon, Fang et al., and Chu et al. which had ML values of 43.8 %, 30.9 %, 22.2 %, 14.6 %, and 31.5 %, respectively. In addition, it is interesting to observe that under the false positive (FP) metric, the proposed algorithm demonstrated a value of 6336 compared with the works of Kim et al., Bewley et al., Bae and Yoon, Fang et al., and Chu et al.,

Fig. 13. Multiple-object tracking in original frame of video using MOT 17 dataset.

having 9064, 32,615, 7110, 9386, and 7321 FP values, respectively. Finally, using the false negative (FN) metric, the proposed algorithm produced 15,990 false negatives compared to the works of Kim et., Bewley et al., Bae and Yoon, Fang et al., and Chu et al. with 32,060, 1001, 22,271, 16,921, and 29,501 FNs, respectively. Based on the analysis of the results, it is observed that the proposed algorithms have much better performance in terms of MOTA and MOTP metrics compared to the state-of-the-art approaches.

Table 2 presents the results of the comparison of multiple Object Tracking using the MOT16 dataset. As shown in Table 2, the proposed algorithms were able to track multiple objects from streams of video in a public detection scenario using the MOT16 dataset with a 51.2 % MOTA accuracy as against the works of Kim et al., Amir et al., Chen et al**.,** and Chu et al. with 35.3 %, 47.2 %, 47.6.3 %, and 48.8 % MOTA values, respectively. It can also be observed that under the MOTP metric, the proposed algorithms produced 85.6 % precision compared to the works of Kim et al., Amir et al., Chen et al**.,** and Chu et al. which showcased 75.2 %, 75.8 %, 74.8 %, and 75.7 % MOTP values, respectively. An interesting observation is under the MT metric, the proposed algorithms produced a 19.0 % MT value as against the works of Kim et al., Amir et al., Chen et al., and Chu et al. with MT values of 7.4 %, 14.0 %, 15.2 %, and 15.8 %, respectively. Similarly, considering the IDF1 metric, the proposed algorithms produced a value of 50.4 % as against the works of Kim et al., Amir et al., Chen et al**.,** and Chu et al. with IDF1 values of 46.3 %, 50.9 %, and 47.2 %, respectively. Another interesting observation is under the metric ML, the proposed algorithms produced a 33.0 ML value compared to the work of Kim et al., Amir et al., Chen et al., and Chu et al. with 51.1 %, 41.6 %, 38.3 %, and a 38.1 % ML values, respectively. In addition, the proposed algorithms produced 5433 values under the FP metric compared to the works of (Kim et al., Amir et al., Chen et al**.,** and Chu et al.) with 5592, 2681, 9253, and 5875 FP values respectively. In conclusion, the proposed algorithms produced 87,586 values under the FN metric compared to the works of (Kim et al., Amir et al., Chen et al**.,** and Chu et al.) with 110,778, 92,856, 85,431, and 86,567 FN values, respectively. Based on the analysis of the results, it is observed that the proposed algorithms perform much better in MOTA, MOTP, ML, MT, and IDF1 metrics than the state-of-the-art approaches.

Table 3 compares the results of the multiple object Tracking (MOT17) datasets. As shown in Table 3, the proposed algorithms were able to track multiple objects from streams of video in a public detection scenario using the MOT17 dataset with a 53.7 % MOTA accuracy compared to the works of (Fu et al., Chen et al., Sheng et al., and Lee and Kim) with a 46.5 %, 50.9 %, 51.8 %, and a 44, 9 % of MOTA values, respectively. It can also be observed that the proposed algorithms produced the highest percentage of precision of 84.8 % of MOTP metric compared to the works of (Fu et al., Chen et al., Sheng et al., and Sangyun et al.) with 77.2 %, 76.6 %, 77.0 %, and 76.6 % of MOTP value respectively. Similarly, under the MT metric, the proposed algorithms produced a 24.0 % MT value compared to the works of (Fu et al., Chen et al., Sheng et al., and Lee and Kim) with a 16.9 %, 17.5 %, 23.4 %, and a 16.5 % of MT values, respectively. An interesting observation is under the metric IDF1, it can be observed that the proposed algorithms produced a 55.8 % IDF1 value compared to the works of (Chen et al., Sheng et al., Lee and Kim) with a 52.7 %, 54.7 %, and 48.4 % of IDF1 values respectively. It can also be observed that the ML metric produced a 36.0 ML value compared to the work of (Fu et al., Chen et al., Sheng et al., and Lee and Kim) with 37.2 %, 35.7 %, 37.9 %, and 35.8 % of ML values, respectively. In addition, the FP metric produced

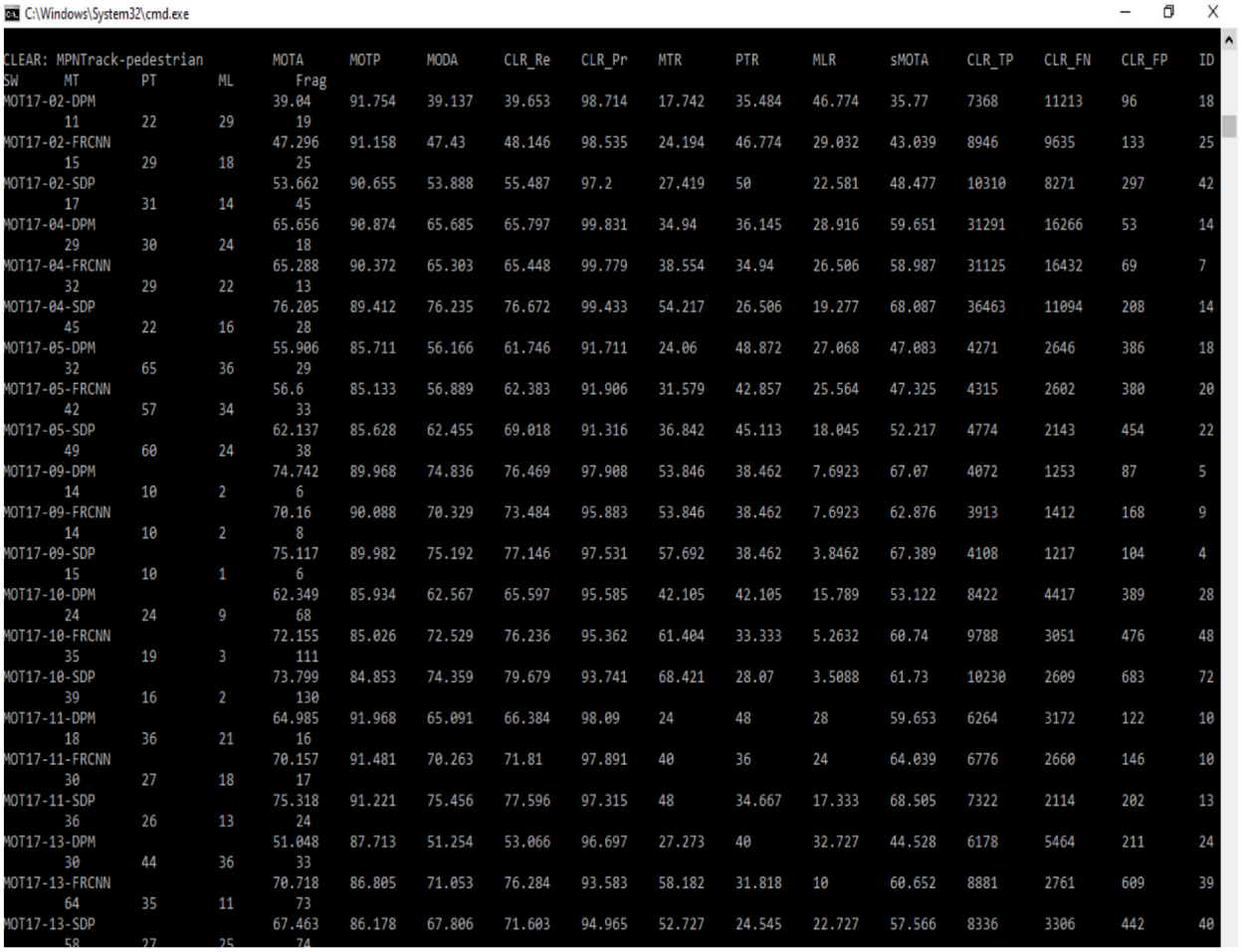


Fig. 14. Runtime snapshot results of multiple-object tracking in original frame of video using MOT 17 dataset.

Table 1
Results comparison of the proposed system on MOT15 dataset.

Author(s)	MOTA↑	MOTP↑	ML↓	MT↑	FP↓	FN↓	IDF1↑
Bewley et al. [25]	33.4	72.1	30.9	11.7	32,615	1001	40.4
Kim et al. [26]	32.4	71.8	43.8	16.0	9064	32,060	45.3
Fang et al. [27]	56.5	73.0	14.6	45.1	9386	16,921	61.3
Chu et al. [28]	38.9	70.6	31.5	16.6	7321	29,501	44.5
Bae and Yoon [29]	51.3	74.2	22.2	36.3	7110	22,271	54.1
Proposed work	53.9	75.8	37.0	18.0	6336	15,990	55.0

Note: N/A indicates result was not provided by the respective authors.

Table 2
Results comparison of the proposed system on MOT16 dataset.

Author(s)	MOTA↑	MOTP↑	ML↓	MT↑	FP↓	FN↓	IDF1↑
Chu et al. [28]	48.8	75.7	38.1	15.8	5875	86,567	47.2
Amir et al. [30]	47.2	75.8	41.6	14.0	2681	92,856	46.3
Chen et al. [31]	47.6	74.8	38.3	15.2	9253	85,431	50.9
Kim et al. [33]	35.3	75.2	51.1	7.4	5592	110,778	N/A
Proposed work	51.2	85.6	33.0	19.0	5433	87,586	50.4

Note: N/A indicates result was not provided by the respective authors.

Table 3
Results comparison of the proposed system on MOT17 dataset.

Author(s)	MOTA↑	MOTP↑	ML↓	MT↑	FP↓	FN↓	IDF1↑
Chen et al. [31]	50.9	76.6	35.7	17.5	24,069	250,768	52.7
Sheng et al. [32]	51.8	77.0	37.9	23.4	33,212	236,772	54.7
Lee and Kim [34]	44.9	76.6	35.8	16.5	33,757	269,952	48.4
Fu et al. [35]	46.5	77.2	37.2	16.9	23,859	274,430	N/A
Proposed work	53.7	84.8	36.0	24.0	31,220	263,821	55.8

Note: N/A indicates result was not provided by the respective authors.

31,220 FP values compared to the works of (Fu et al., Chen et al., Sheng et al., and Lee and Kim) with 23,859, 24,069, 33,212, and 33,757 FP values respectively. Finally, under the FN metric, the proposed algorithms produced FN values of 263,821 compared to the works of (Fu et al., Chen et al., Sheng et al., and Lee and Kim) with a 274,430, 250,768, 236,772, and 269,952 of FN values respectively. Based on the analysis of the results, it is observed that the proposed algorithms have much better performance in terms of the seven MOT challenge metrics compared to the state-of-the-art approaches. The high performance demonstrated by the proposed framework under the MOT challenge metrics and datasets indicates that the framework can be deployed in real-life situations to aid security and surveillance systems.

5. Conclusion and future directions

This study presents a real-time framework for object detection, tracking, and recognition for security surveillance in surveillance systems. The system has been implemented using an algorithm based on an approximate median filter, component labeling, background subtraction, and a Convolutional Neural Network (CNN). A software application framework was designed using Python and integrated with C# programming language for ease of use. Experimental results based on the MOT challenge benchmark and the MOT15, MOT16, and MOT17 datasets indicated that the proposed framework provided higher accuracy and precision performance than the state-of-the-art approaches. The high performance demonstrated by the proposed framework under the MOT challenge metrics and datasets indicates that the framework can be deployed in real-life situations to aid security and surveillance systems. The framework also provides an accurate and effective means of monitoring and recognizing moving objects.

The software development, including the design of the framework user interfaces, was coded in the C# programming language and integrated with Python using Microsoft Visual Studio (2019 edition). The integration was performed for ease of use and to execute the framework as a standard software application. Such an automated system can be used in applications where security is challenging.

Future studies will consider the dynamic scalability of the framework to accommodate different surveillance application areas in overcrowded scenarios. Multiple data sources shall be integrated to enhance the performance for different scene times, locations, and weather conditions. Furthermore, future studies will consider other object-detection techniques such as You Only Look Once and its variants. This allows the framework to adapt to complex situations in which security surveillance is challenging.

Data availability

Data included in article/supp. Material/referenced in the article.

Ethics statement

The authors declare the research did not involve human or animal experiments.

Funding

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2019R1I1A3A01058887) and in part by the Korea Institute of Energy Technology Evaluation and Planning and Ministry of Trade, Industry, and Energy of the Republic of Korea under Grant 20184010201650.

CRediT authorship contribution statement

Sani Abba: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ali Mohammed Bizi:** Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jeong-A Lee:** Writing – review & editing, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Souley Bakouri:** Writing – review & editing, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Maria Liz Crespo:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the National Research Foundation of Korea through the Ministry of ICT, the Korea Institute of Energy Evaluation and Planning and the Ministry of Trade, Industry, and Energy of the Republic of Korea for providing financial support for this research work. Special thanks are owed to the Abdussalam International Centre for Theoretical Physics (ICTP), Trieste, Italy and Dr. Iain Darby of the International Atomic Energy Agency (IAEA), Vienna, Austria, for the ICTP workshop/Seminar SMR 3143, "Joint ICTP-IAEA School on Zynq-7000 SoC and its Applications for Nuclear and Related Instrumentation", Aug–Sep 2017, Trieste, Italy.

References

[1] H. Ulusoy, Revisiting security communities after the cold war: the constructivist perspective, perceptions, J. Int. Aff. 8 (3) (2003) 1–22.
[2] D. Lohani, C. Crispim-Junior, Q. Barthélemy, S. Bertrand, L. Robinault, L. Tougne Rodet, Perimeter intrusion detection by video surveillance: a survey, Sensors 22 (3601) (2022), https://doi.org/10.3390/s2209360.
[3] F. Dumitrescu, C.-A. Boiangiu, M.-L. Voncilă, Fast and robust people detection in RGB images, Appl. Sci. 12 (1225) (2022), https://doi.org/10.3390/app12031225.
[4] H. Masood, A. Zafar, M.U. Ali, T. Hussain, M.A. Khan, U. Tariq, R. Damaševičius, Tracking of a fixed-shape moving object based on the gradient descent method, Sensors 22 (1098) (2022), https://doi.org/10.3390/s22031098.
[5] H. Qiao, X. Wan, Y. Wan, S. Li, W. Zhang, A novel change detection method for natural disaster detection and segmentation from video sequence, Sensors 20 (5076) (2020), https://doi.org/10.3390/s20185076.
[6] J. Wang, S. Simeonova, M. Shahbazi, Orientation- and scale-invariant multi-vehicle detection and tracking from unmanned aerial videos, Rem. Sens. 11 (2155) (2019), https://doi.org/10.3390/rs11182155.
[7] J.-H. Park, K. Farkhodov, S.-H. Lee, K.-R. Kwon, Deep reinforcement learning-based DQN agent algorithm for visual object tracking in a virtual environmental simulation, Appl. Sci. 12 (3220) (2022), https://doi.org/10.3390/app12073220.
[8] R. Opromolla, G. Inchingolo, G. Fasano, Airborne visual detection and tracking of cooperative UAVs exploiting deep learning, Sensors 19 (4332) (2019), https://doi.org/10.3390/s19194332.
[9] S. Zhang, L. Zhuo, H. Zhang, J. Li, Object tracking in unmanned aerial vehicle videos via multi-feature discrimination and instance-aware attention network, Rem. Sens. 12 (2646) (2020), https://doi.org/10.3390/rs12162646.
[10] D. Rodriguez, C. Aceros, J. Valera, E. Anaya, A framework for multiple object tracking in underwater acoustic MIMO communication channels, J. Sens. Actuator Netw. 6 (2) (2017), https://doi.org/10.3390/jsan6010002.
[11] F.S. Alsubaei, F.N. Al-Wesabi, A.M. Hilal, Deep learning-based small object detection and classification model for garbage waste management in smart cities and IoT environment, Appl. Sci. 12 (2281) (2022), https://doi.org/10.3390/app12052281.
[12] P. Kowalczyk, J. Izydorczyk, M. Szelest, Evaluation methodology for object detection and tracking in bounding box based perception modules, Electronics 11 (1182) (2022), https://doi.org/10.3390/electronics11081182.
[13] L. Wenhan, X. Junliang, M. Anton, Z. Xiaoqin, Wei Liu, k. T.-Tae, Multiple object tracking: a literature review, Artif. Intell. 293 (2021) 103448, https://doi.org/10.1016/j.artint.2020.103448.
[14] G. Ciparrone, F.L. Sanchez, L. Tabik, L. Troiano, R. Tagliaferri, F. Herrera, Deep learning in video multi-object tracking: a survey, (381). https://doi.org/10.1016/j.neucom.2019.11.023.
[15] Y.D. Li, Z.B. Hao, H. Lei, Survey of convolutional neural networks, J. Comput. Appl. 36 (9) (2016) 2508–2515.
[16] U. Chandrasekhar, T. Das, A survey of techniques for background subtraction and traffic analysis on surveillance video, 1(3), 107–113. Retrieved from: http://uniascit.in/files/documents/2011_18.pdf, 2011.
[17] Y. Alper, J. Omar, S. Mubarak, Object tracking: a survey, ACM Comput. Surv. 38 (4) (2013) 13.
[18] B. Drayer, T. Brox, Object detection, tracking, and motion segmentation for object-level video segmentation, Retrieved from, http://arxiv.org/abs/1608.03066, 2016.
[19] S. Sen Cheung, C. Kamath, Robust techniques for background subtraction in urban traffic, video, www.vis.uky.edu/~cheung/doc/UCRL-CONF-200706.pdf, 2004.
[20] G.M. Rao, Object tracking system using approximate median filter, kalman filter, and dynamic template matching, 83–89, https://doi.org/10.5815/ijisa.2014.05.09, 2014.
[21] A. Abdulmalik, A. Khalil, H. Ullah Khan, Object detection and tracking using background subtraction and connected component labeling, Int. J. Comput. Appl. 75 (13) (2013) 1–5.
[22] M. Alawi, O. Khalifa, R. Islam, Performance comparison of background estimation algorithms for detecting moving vehicle, 21 109–114, https://doi.org/10.5829/idosi.wasj.2013.21.mae.99934, 2013.
[23] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, A. Prati, The Sakbot system for moving object detection and tracking, in: Proceedings of 2nd European Workshop on Advanced Video-Based Surveillance Systems, 2001, pp. 159–162.
[24] S.A. Hussain, A.A.A. Salim, A real-time face emotion classification and recognition using a deep learning model, J. Phys.: Conferences series 1432 (2020) (2019) 012087, https://doi.org/10.1088/1742-6596/1432/1/012087.
[25] B. Alex, G. Zongyuan, O. Lionel, R. Fabio, U. Ben, Simple online and real-time tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), IEEE, 2016, pp. 3464–3468.
[26] K. Chanho, L. Fuxin, C. Arridhana, M.R. James, Multiple hypothesis tracking revisited, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4696–4704.
[27] F. Kuan, X. Yu, L. Xiaocheng, S. Silvio, Recurrent autoregressive networks for online multiobject tracking, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 466–475.
[28] C. Peng, F. Heng, C.T. Chiu, L. Haibin, Online multi-object tracking with an instance-aware tracker and dynamic model refreshment, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 161–170.
[29] H.B. Seung, Y. Kuk-J, Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 40 (3) (2017) 595–610.
[30] S. Amir, A. Alexandre, S. Silvio, Tracking the untrackable: learning to track multiple cues with long-term dependencies, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 300–311.
[31] L. Chen, H. Ai, Z. Zhuang, C. Shang, Real-time multiple people tracking with deeply learned candidate selection and person re-identification, in: ICME, 2018.

[32] Hao S., Yang Z., Jiahui C., Zhang X., Jun Z., Heterogeneous association graph fusion for target association in multiple object tracking, IEEE Trans. Circ. Syst. Video Technol, (29) (11) (2019) 3269 - 3280. DOI: 10.1109/TCSVT.2018.2882192.
[33] K. Minyoung, A. Stefano, R. Luca, Similarity mapping with enhanced siamese network for multiobject tracking, in: Machine Learning for Intelligent Transportation Systems (MLITS), NIPS Workshop, 2016.
[34] L. Sangyun, K. Euntai, Multiple object tracking via feature Pyramid siamese networks, IEEE Access 7 (2019) 8181–8194.
[35] F. Zeyu, A. Federico, M.N. Syed, A.C. Jonathon, Gm-PhD filter based online multiple human tracking using deep discriminative correlation matching, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2018, pp. 4299–4303.
[36] Lifeng He, Ren Xiwei, Qihang Gao, Zhao Xiao, Yao Bin, Chao Yuyan, The connected-component labeling problem: a review of state-of-the-art algorithms, Pattern Recogn. 70 (2017) 25–43, https://doi.org/10.1016/j.patcog.2017.04.018.
[37] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, Massachusetts, 1993.
[38] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.
[39] A. Rosenfeld, J.L. Pfalts, Sequential operations in digital picture processing, J. ACM 13 (4) (1966) 471–494.
[40] N. Cao, Y. Liu, High-noise grayscale image denoising using an improved median filter for the adaptive selection of a threshold, Appl. Sci. 14 (635) (2024), https://doi.org/10.3390/app14020635.
[41] Youlian Zhu, Huang Cheng, An improved median filtering algorithm for image noise reduction, 2012 international conference on solid state devices and materials science, Phys. Procedia 25 (2012) 609–616.
[42] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, A review of video surveillance systems, J. Vis. Commun. Image R. 77 (2021) 103116, https://doi.org/10.1016/j.jvcir.2021.103116.
[43] F. Porikli, F. Br′emond, S.L. Dockstader, J. Ferryman, A. Hoogs, B.C. Lovell, S. Pankanti, B. Rinner, P. Tu, P.L. Venetianer, Video surveillance: past, present, and now the future DSP forum, IEEE Signal Process. Mag. 30 (3) (2013) 190–198.
[44] B.K. Muhammad, E.M. Bashier, M. Mohssen, Machine Learning: Algorithm & Applications, International Standard Books, 2017, 13:978-1-4987 by Taylor &Francis Group, LLC.
[45] N. Svetlin, FUNDAMENTALS of COMPUTER PROGRAMMING with C# (The Bulgarian C# Programming Book), 2013, pp. 805–852. Sofia, ISBN 978-954-400-773-7.
[46] M. Mayo, P.S. Gregory, Data Science, Machine Learning Algorithm for Real-World Application, 2019.
[47] R. Pablo, Livery Place 35 Livery Street Birmingham B3 2PB, Deep Learning for Beginners: A Beginner's Guide to Getting up and Running with Deep Learning from Scratch Using Python, Packt Publishing Ltd, UK, 2020. ISBN 978-1-83864-085-9.
[48] P. Bhagyalakshmi, P. Indhumathi, R. Lakshmi, Dr Bhavadharini, Real-time video surveillance for automated weapon detection, International Journal of Trend in Scientific Research and Development (ijtsrd) (2019), 465-470.
[49] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[50] T. Kusuma, K. Ashwini, Real-time object detection and tracking design using deep learning with spatial-temporal mechanisms for video surveillance applications, in: H.S. Saini, R. Sayal, A. Govardhan, R. Buyya (Eds.), Innovations in Computer Science and Engineering. ICICSE 2022. Lecture Notes in Networks and Systems, Springer, Singapore, 2023, p. 565, https://doi.org/10.1007/978-981-19-7455-7_56.
[51] MOTChallenge: The Multiple Object Tracking Benchmark, Available at: https://motchallenge.net/.