

# Advanced Object Tracking in Video Surveillance Systems with Adaptive Deep SORT Enhancement

**M. Koteswara Rao**

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, India | Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India  
mailto:mkrao@gmail.com (corresponding author)

**P. M. Ashok Kumar**

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, India  
pmashokk@gmail.com

Received: 7 November 2024 | Revised: 2 December 2024 and 9 December 2024 | Accepted: 14 December 2024

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.9529>

## ABSTRACT

Object tracking is a crucial feature of video surveillance systems that are essential for maintaining awareness and detecting potential threats. Advanced solutions are needed to overcome the obstacles associated with video object tracking, including the complexity of everyday environments and the massive amount of data. Traditional tracking algorithms often struggle with the complexity of dynamic situations, necessitating the use of deep learning methods. This paper presents an innovative deep learning-based object tracking system that uses Multi-Level Glow-Worm Swarm Convolution Neural Networks (MLGS-CNNs) to detect objects in video frames. Subsequent object tracking is facilitated by the adaptive Deep Simple Online Real-time Tracking (DeepSORT) algorithm by incorporating an optimized Kalman filter instead of a conventional Kalman filter. The Waterwheel Plant Optimization (WPO) method is used to tune the noise covariances of the Kalman filter to further improve the tracking accuracy. Comprehensive performance criteria, including metrics such as Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Integrated Detection and False-alarm Rate (IDF1), Mostly Tracked (MT), and Mostly Lost (ML), are used to evaluate the effectiveness of our method.

*Keywords-Waterwheel Plant Optimization (WPO), Kalman filter, adaptive DeepSORT*

## I. INTRODUCTION

Traditional video surveillance systems rely on fiber and cable connections that are costly to deploy and maintain on a large scale, but today's video transmissions mostly use wireless connectivity. In addition, object motion tracking is an important aspect in video surveillance applications because it provides temporal information about moving objects [1], and it is beneficial for a variety of reasons, including security through the use of video feeds [2]. Some of the most challenging aspects of discriminating between moving objects include varying lighting conditions, extended occlusions between moving objects, shadows, and non-stationary background objects. At a second level, object motion tracking approaches operate through appearance-based object segmentation and motion feature clustering [3]. To accomplish tasks such as background removal, each frame is examined in conjunction with a reference or background model. The detection of accurate object features (spatial accuracy) and the temporal

stability of the detection (temporal coherency) are general requirements for a background removal algorithm. The minimum and maximum intensity values of each video clip are used to simulate the background scene during video tracking. Moreover, each pixel's greatest temporal component is immediately updated and recorded [4]. Shape analysis and tracking are combined to match each frontal object area to the current object collection [5]. The process of "object discovery" allows the identification of objects that are moving within an area, which is a prerequisite for a tracking method to work. Using existing object detection algorithms [6-11], tracking objects in images can be challenging due to a variety of factors, such as sudden object movements, scene or object changes, object shape changes, occlusion from the surrounding background, and illumination changes. Our work aims to proactively address these challenges by relying on deep learning to tackle the complexities of video object tracking. Deep learning helps our system detect intricate patterns, adapt

to dynamic conditions, and track objects across frames with unparalleled accuracy.

The growing importance of video surveillance in numerous real-time applications was discussed in a study by the authors in [12]. The study examined developments in machine learning, specifically in the field of Multi-Object Detection and Tracking (MODT). Using an ideal Kalman filtering technique, the presented methodology offered a novel approach to MODT for tracking moving objects within video frames. Using the region growth model, video clips were transformed into morphological operations. Kalman filtering was then used to optimize the parameters using the probability-based grasshopper algorithm. A major limitation of the presented methodology is the lack of motion estimation methods in the MODT analysis. To address the growing concerns about security and surveillance, the authors in [13] proposed a video surveillance system that uses knowledge-based deep learning for enhanced multi-object tracking and recognition. To maintain recognition accuracy while increasing efficiency, they developed a method that combines optical flow while maintaining the recognition performance through a knowledge-based Convolution Neural Network (CNN). The system's optical flow-based tracker can predict the position of objects in the next frame by combining a CNN-based detector with knowledge-based mining approaches for reliable object detection. However, further research is needed to create a surveillance system that can quickly identify numerous objects and detect their movements. Authors in [14] addressed people tracking in video surveillance by introducing a top-view-based approach. The technique uses a top-view camera and consists of four main modules: size estimation, tracking, standardization, and Binary Large Object (BLOB) detection. Through segmentation, statistical operations, connected component labeling, and morphological operations, the technique retrieves the foreground. To ensure rotation invariance, the retrieved BLOB was moved to an upright position using the radial symmetry of the top view. A drawback of this study is that it only considers scenarios with one person; therefore, further research is needed to cover multiple videos with different people. Authors in [15] presented a new metric for evaluating Multi-Object Tracking (MOT), Higher Order Tracking Accuracy (HOTA), which balances the importance of

accurate detection, association, and localization. To provide a thorough examination of tracking performance, HOTA was broken down into sub-metrics that independently evaluated each of the five fundamental error categories. The study evaluated HOTA's performance against the MOTChallenge benchmark, demonstrating how well it can capture important MOT performance factors that other metrics cannot. The MOTChallenge is the only benchmark dataset focused on in this study, which is a limitation. Authors in [16] presented FairMOT as a solution to the problem of competing item detection and re-identification tasks in MOT. Although there are computational benefits to expressing MOT as a multi-task learning problem, the fundamental conflict between the detection and re-identification tasks can lead to biased results. Based on CenterNet's anchor-free object detection architecture, FairMOT provided a thorough method for balancing the importance of each task. Authors in [17] suggested a long-term tracking strategy with deep tracklet association. The study focused on employing tracklets to generate more comprehensive trajectories, while acknowledging the influence of detector performance on tracking. A high-confidence tracklet generation technique was presented by means of an iterative clustering process, which addresses problems such as fragmentation in crowded environments. In addition, a deep association technique was presented that learns long-term features for tracklet association using a Motion Evaluation Network (MEN) and an Appearance Evaluation Network (AEN).

## II. METHODOLOGY

The present study proposes a new object tracking system using for video tracking in surveillance systems using deep learning. The adaptive Deep Simple Online Real-time Tracking (DeepSORT) method is introduced to improve tracking after object detection with the use of advanced neural networks. The performance of DeepSORT is greatly improved by an optimized Kalman filter. In addition, Waterwheel Plant Optimization (WPO), an approach that helps fine-tune noise covariances, is used to achieve further improvement. The effectiveness of the proposed approach is evaluated in depth utilizing important metrics. Figure 1 shows the structure of the proposed method.

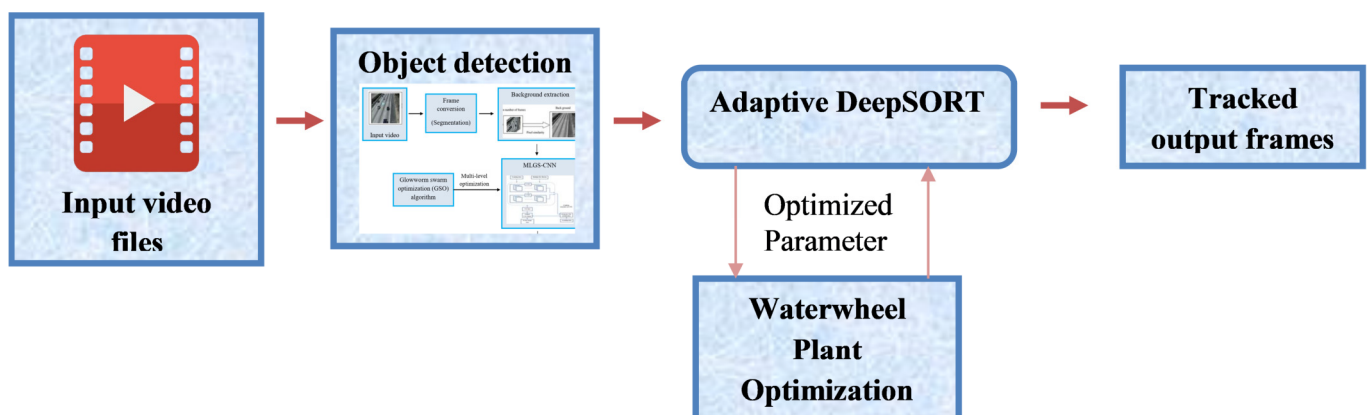


Fig. 1. Proposed methodology of object tracking in video surveillance systems.

### A. Detecting Anomalous Events in Video Surveillance Systems

The Multi-Level Glow-Worm Swarm Convolution Neural Network (MLGS-CNN) is used in the proposed approach for abnormal event detection in video surveillance systems. It enables automatic feature extraction and feature-based event class prediction. In order to identify dynamic foreground objects, the framework performs background extraction and frame conversion, which divides the input videos into segments. Then, MLGS-CNN is employed for prediction. Equations (1) and (2) define the video segments and the collection of all video events, respectively, and reflect the segmentation process in the computation.

$$S = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1K} \\ S_{21} & S_{22} & \dots & S_{2K} \\ & & \dots & \\ S_{H1} & S_{H2} & \dots & S_{HK} \end{bmatrix} \quad (1)$$

where each segment  $S_{ij}$  consists of a specific  $N$  number of frames. Due to the reduction in computational and comparison complexity, it is expected that each segment will have less than 200 frames in total. The entire set of the video events  $V_i$  is represented as:

$$V_i = \{e_i\}_{i=1}^N \quad (2)$$

The definition of anomalous events and the number of events in a one-minute video clip are given by (3) and (4), respectively. The anomalous event  $e_a$  is stated as follows:

$$e_a \in V_i \quad (3)$$

and the number of anomalous events is stated as follows:

$$M_{a,i} = M(e_a, e_i), \quad e_i \in V - \{e_a\} \quad (4)$$

The redundant information in these video frames is removed by assessing the similarity of the spatiotemporal variables across frames. The number of events in a one-minute video is estimated to be between 5 and 10, and the video volumes are approximations of long videos recorded over a long period of time. Static and dynamic pixel separation across frames is necessary for background extraction, which is essential for detecting anomalous events. To improve adaptation capacity and avoid overfitting, the prediction step uses the MLGS-CNN to iteratively optimize the CNN architecture and the hyperparameters using the Glow-Worm Swarm Optimization (GSO) algorithm. Unknown data are classified using the optimized CNN after the softmax classification layer of the CNN has determined the probability of events in each class. The suggested technique increases the accuracy of anomalous event detection in video surveillance systems by overcoming the shortcomings of traditional methods.

### B. Object Tracking in Video Surveillance Systems with Adaptive DeepSORT Algorithm

In the next stage of object tracking, the output of the object detection model which typically consists of bounding boxes indicating where objects have been detected in an image or video frame, becomes a crucial input to the adaptive DeepSORT algorithm. Once the object detection model has identified objects and provided their spatial coordinates, adaptive DeepSORT steps in to ensure smooth and accurate tracking. Figure 2 illustrates the structure of the adaptive DeepSORT algorithm. Employing a variety of novel techniques, adaptive DeepSORT incorporates a Kalman filter for state prediction and the Hungarian algorithm for effective data association between predicted and detected objects. The Kalman filter algorithm is expressed as:

$$P_k = A \cdot P \cdot A^{-1} + \rho \quad (5)$$

and it operates on the basis of the Kalman update given by:

$$K_u = P_k^{-1} \cdot H^T (H \cdot P_k^{-1} \cdot H^T + Q_c)^{-1} \quad (6)$$

$$z = z_0 + K_u \cdot \left( [z_c, y_c]^T - H \cdot z_0 \right) \quad (7)$$

where  $z_c$ ,  $y_c$  are the center coordinates of the object,  $Q_c$  is the measurement noise covariance matrix, and  $K_u$  is the Kalman update. The equation described above indicates that the measured values obtained by the Kalman filter are more significant and closer to the estimated or real values.

$$P_{k+1} = 1 - K_u \cdot H \cdot P_k \quad (8)$$

The measurement covariance matrix is expressed as:

$$Q_c = E(v_k v_k^T) \quad (9)$$

where  $v_k$  is the measurement noise standard deviation. These formulas directly show that the measurement noise covariance matrix  $Q_c$  determines the Kalman gain.

In the adaptive DeepSORT algorithm, the typical Kalman filter has been replaced by an optimized Kalman filter. This modification aims to improve the overall efficiency of the tracking algorithm, suggesting that the optimized version may provide more accurate and effective state predictions. The DeepSORT adaptive algorithm uses the optimized Kalman filter to estimate the track in each frame of a video sequence, where  $[u, v, \kappa, h, \hat{x}, \hat{y}, \hat{\kappa}, \hat{h}]$  represents the velocity along each coordinate and expresses in image coordinates the height, position, aspect ratio, and associated velocity information of the bounding box center. Specifically,  $(\hat{x}, \hat{y}, \hat{\kappa}, \hat{h})$  indicates the velocity in the relevant dimensions, whereas  $(u, v, \kappa, h)$  indicates the spatial location of the bounding box.

$$d = [u, v, \kappa, h, \hat{x}, \hat{y}, \hat{\kappa}, \hat{h}]^T \quad (10)$$

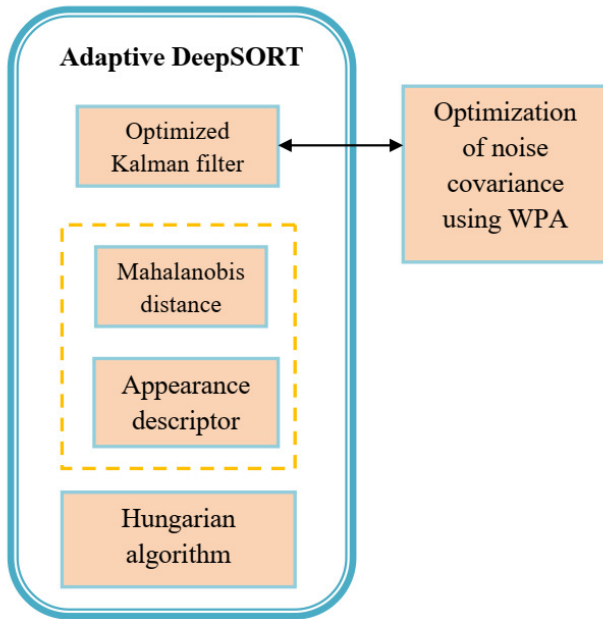


Fig. 2. The structure of adaptive DeepSORT algorithm.

An optimized Kalman filter with linear observation and constant velocity is used. Whenever a new frame arrives, the position of each track is calculated based on its past positions. Only spatial data are used for track estimation. The approach computes the total number of frames from the last successful measurement association ( $\alpha_k$ ) for each tracked object, denoted by track  $K$ . When a favorable prediction is obtained via the Kalman filter, a counter is increased. This number is reset to 0 upon a successful association with a measurement. The method considers an object to have left the scene and removes its associated track when the counter for that track reaches a certain maximum maturity, indicating that the object has been tracked for too long without a recent association. To obtain appearance information from both detections and tracks, an appearance descriptor is used to extract features from detection images and track images in previous frames. To improve the discriminative strength of the appearance representation, this descriptor can extract features so that features of the same identity are closest to each other in the feature space, while features of other identities are clearly separated. WPO method was used in the adaptive DeepSORT algorithm to fine-tune the noise covariances of the Kalman filter. WPO is primarily used to optimize the parameters of the Kalman filter, particularly the noise covariances that are important for accurate modeling and tracking of object states. Section II.B.1 provides a detailed explanation of how the WPO approach was developed to enhance the noise covariances of the Kalman filter in the adaptive DeepSORT algorithm.

The Hungarian algorithm is used to address the mapping problem between recently received measurements and predicted Kalman states. The Hungarian algorithm is useful in determining the best assignment of measurements to predicted states, taking into account the corresponding Mahalanobis distances. This mapping process considers both motion and

appearance information, and calculates the Mahalanobis distance between them using (11):

$$m_d^{(1)}(i, j) = (m_d^j - y_i)^T S_i^{-1} (m_d^j - y_i) \tag{11}$$

where  $(y_i, S_i)$  are the projection of the  $j$ -th track in measurement space, and  $m_d^j$  is the  $j$ -th new detection.

In order to account for uncertainty, the Mahalanobis distance calculates the number of standard deviations the detection point is far away from the average track location. Using this metric, improbable correlations can be removed by thresholding the Mahalanobis distance. The decision is expressed by an indicator, denoted in (12), which calculates to 1 if the association between the  $i$ -th track and the  $j$ -th detection is considered acceptable. In other words, if the Mahalanobis distance is less than a certain threshold, it means that the relationship between the track and the detection is considered acceptable, contributing to a more reliable data association phase in the tracking process.

$$g_{i,j}^{(1)} = 1 [m_d^{(1)}(i, j) < t^{(1)}] \tag{12}$$

Although the Mahalanobis distance works well in some cases, it has limitations when there is camera motion. To solve this problem, a new metric for the assignment problem is presented. This second measure computes the smallest cosine distance in appearance space between the  $i$ -th track and the  $j$ -th detection. This alternative measure considers the appearance features as well as their orientation in the feature space, providing a different perspective than the Mahalanobis distance.

$$m_d^{(2)}(i, j) = \min \{ 1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in \mathfrak{R}^2 \} \tag{13}$$

Again, a binary variable is introduced to indicate the acceptability of a relationship based on the provided metric. This binary variable acts as an indicator that evaluates to 1 or 0 depending on whether the association between items meets the criteria of the metric. This method promotes a clear and binary decision-making process, which improves the efficiency and effectiveness of the data association stage in the overall tracking algorithm.

$$g_{i,j}^{(2)} = 1 [m_d^{(2)}(i, j) < t^{(2)}] \tag{14}$$

The appropriate threshold for this binary indicator is obtained by measuring it on a separate training dataset. Both measurements are merged using a weighted total to create the association problem. This combination allows for a thorough examination by considering both the distance measured by Mahalanobis and the cosine distance in appearance space. The weighted sum provides a flexible framework for balancing the contributions of each indicator in the tracking algorithm, providing a resilient and adaptive approach to data association.

$$h_{i,j} = \gamma m_d^{(1)}(i, j) + (1 - \gamma) m_d^{(2)}(i, j) \tag{15}$$

where an association is acceptable if it falls between the boundary ranges of both metrics, such as:

$$g_{i,j} = \prod_{m=1}^n 2g_{i,j}^{(m)} \tag{16}$$

The influence of each metric on the total association cost can be controlled by hyperparameter  $\gamma$ . The Hungarian algorithm and an optimized Kalman filter are used by the adaptive DeepSORT method to provide effective data association. The mapping problem is handled by the Hungarian algorithm, which considers cosine and Mahalanobis distances. Data association is supported by a binary indicator whose thresholds are determined using a training dataset. The utilization of a weighted sum to combine measurements ensures a stable and adaptable method.

1) *Optimizing the Noise Covariance of the Kalman Filter using the Waterwheel Plant Optimization Algorithm*

This section examines how the WPO algorithm can be used to improve the performance of the Kalman filter by optimizing the noise covariance, thereby improving the accuracy and efficiency of the tracking and state estimation processes:

a) *Initialization*

Using a population-based optimization technique, the WPO uses a number of individuals to search through a given search space and identify the best possible solutions. Each waterwheel in the WPO population represents a distinct individual with a different value for each of the problem variables. Within the search space, specific locations of the waterwheels define these values. In order to get better results after several attempts, the algorithm iteratively searches this space by moving the waterwheels around. Initially, the solutions are populated as follows:

$$Z_M = \{z_1, z_2, \dots, z_p\}_M \tag{17}$$

where  $z_p$  depicts the  $M$ -th waterwheel (a solution of the candidate) and it can be expressed as:

$$z_p = [Q_c] \tag{18}$$

where  $Q_c$  represents the noise covariances of the Kalman filter.

b) *Fitness Estimation*

The fitness of each solution is determined after solution initialization. The fitness function is considered to have the highest accuracy and is expressed as follows:

$$Fit_n = Max (Accuracy) \tag{19}$$

$$Accuracy = \frac{TN + TP}{TP + FP + TN + FN} \tag{20}$$

c) *Solution Update*

The WPO algorithm combines exploration and exploitation phases to iteratively update the solution. To improve the overall solution, the waterwheels, which represent potential solutions, move throughout the search field. The algorithm can converge towards better solutions by simulating the hunting behavior of the waterwheel in the exploration phase and the process of capturing and transferring insects in the exploitation phase. The

process of iterative updating continues until the algorithm reaches its final iteration. Phase 1 consists of exploring the positions and hunting the insects. Waterwheels have an excellent sense of smell, which they use to locate insects with remarkable predatory ability. The waterwheel detects the exact location of the insect and launches an attack as soon as it comes in contact with it. By mimicking this hunting behavior, the WPO enhances its ability to discover the optimal region and avoid local optima. Significant location changes within the search space are the outcome of this modeling. The new location of the waterwheel is found using an equation and a simulation of the waterwheel's approach to the problem. If moving to this new location increases the objective function's value, the previous location disappears in favor of the new one.

$$\vec{W} = r_1 (\vec{p}(t) + 2K) \tag{21}$$

$$\vec{p}_{t+1} = \vec{p}(t) + \vec{W}(2K + r_2) \tag{22}$$

The following equation can be used to adjust the location of the waterwheel if the solution remains constant after three iterations:

$$\vec{p}_{t+1} = Gaussian(\mu_p, \sigma) + r_1 \frac{\vec{p}(t) + 2K}{\vec{W}} \tag{23}$$

The random variables in this situation are denoted by the variables  $r_1$  and  $r_2$ , which have values between 0 and 2.  $K$  is also an exponential variable, with values between 0 and 1. The diameter of the circle that the waterwheel plant will explore

and investigate is denoted by the vector  $\vec{W}$ . Phase 2 is the exploitation phase and it simulates how the waterwheels collect insects and moves them to a feeding tube. By promoting convergence towards solutions that are close to those already found, this simulated behavior improves the WPO's exploitation capabilities during local search. Throughout the process, the waterwheel's placement within the search space is slightly altered. To imitate the natural behavior of waterwheels, WPO selects a new random position for each waterwheel in the population. This is called a "good position for consuming insects." If the objective function value is higher at this new location, the waterwheel is moved there and substitutes the old one, based on the following calculations:

$$\vec{W} = r_3 * (K p_{best}(t) + r_3 \vec{p}(t)) \tag{24}$$

$$\vec{p}(t+1) = \vec{p}(t) + K \vec{W} \tag{25}$$

In this case, the values of the random variable denoted by  $r_3$  range from 0 to 2. The solution at iteration  $t$  is denoted by  $\vec{p}(t)$  and the optimal solution found up to that time is denoted by  $p_{best}(t)$ . As in the previous exploration phase, the next mutation is employed to guarantee the avoidance of local minima if the solution does not improve for three consecutive iterations.

$$\bar{p}(t+1) = (\bar{r}_1 + K) \sin\left(\frac{FC}{\theta}\right) \quad (26)$$

where the independent random variables  $F$  and  $C$  have values in the range  $(-5, 5)$ . In addition, the exponential decay of  $K$  can be shown using the following equation:

$$K = \left(1 + \frac{2t^2}{T_{\max}^3} + F\right) \quad (27)$$

The WPO is a repeatable process. After the first two phases of WPO are completed, all waterwheel positions must be adjusted.

#### d) Termination

The best candidate is improved after comparing the objective function values. Until the algorithm reaches its last iteration, the waterwheel positions are modified in each iteration. The best solution found and maintained by WPO is provided to us after enough iterations.

### III. RESULTS AND DISCUSSION

The proposed robust object tracking system was implemented and tested on the UCSD dataset [18], which contains video of pedestrians on UCSD walkways captured by a stationary camera, and it performed better than expected in overcoming the difficulties associated with video surveillance. The system's potential to set a new benchmark in object tracking for video surveillance systems was demonstrated by its thorough evaluation, which included accuracy, specificity, precision, recall, and F1-score.

#### A. Experimental Results

We have achieved promising results in the experimental evaluation of our robust object tracking system with the UCSD dataset. The suggested approach demonstrated significant gains in object tracking accuracy over conventional techniques by utilizing deep learning and the adaptive DeepSORT algorithm. Table I presents a detailed visual representation of the tracking results of the proposed object tracking system. This table is useful for evaluating the system's performance over various frames of the input video sequence. A systematic and easy to follow comparison is provided by sequentially grouping the input image frames with their corresponding tracked output images. Table I provides a detailed assessment of how well the tracking system maintains accuracy and consistency throughout the video sequence by showing the frames in sequence. Observers can easily follow the path of objects, identify successful tracking occurrences, and notice any differences or improvements in the tracked output compared to the original input frames.

#### B. Comparative Analysis

Simple Online Real-time Tracking (SORT), Global Optimization on Graph with Early Oscillation Check (GOG\_EOC), Siamese Convolutional Trackers (SCTrack), and the proposed DeepSORT are among the trackers evaluated in this section. The evaluation is based on various performance metrics such as Multiple Object Tracking Accuracy (MOTA),

Multiple Object Tracking Precision (MOTP), Integrated Detection and False-alarm Rate (IDF1), Mostly Tracked (MT), Mostly Lost (ML), False Positives (FP), False Negatives (FN), ID Switches (IDs), and Fragmentations (FM).

TABLE I. TRACKED OUTPUT FRAMES FOR CORRESPONDING INPUT FRAMES









| Frame number | Input image  | Tracked output image  |
|--------------|--|---|
| 1            |    |    |
| 2            |    |    |
| 3            |   |   |
| 4            |  |  |

Table II provides a thorough review of various object tracking techniques in the context of multiple object tracking. The proposed DeepSORT algorithm outperforms others in terms of MOTA, with the best score of 66.5% demonstrating higher accuracy in tracking multiple objects. The MOTP for the proposed DeepSORT is also noteworthy, with a MOTP score of 0.259, demonstrating precision in object tracking. The IDF1 score is 69, highlighting its efficiency. The metrics MT, ML, FP, FN, IDs, and FM demonstrate the robustness of the proposed DeepSORT. It excels in minimizing FP and FN with 4722 and 68060, respectively, and outperforms in the metrics MT, ML, IDs, and FM, demonstrating its ability to maintain accurate object trajectories. The comparison demonstrates the effectiveness of the proposed DeepSORT algorithm in overcoming problems associated with accurate and robust multiple object tracking, making it a promising solution for a wide range of tracking conditions. It highlights the proposed DeepSORT as a leading algorithm for advanced object tracking in video surveillance, combining high accuracy with precision and robustness across a wide range of tracking conditions.

TABLE II. EVALUATING THE PERFORMANCE OF VARIOUS TRACKERS USING DIFFERENT PERFORMANCE METRICS

| Tracker           | MOTA | MOTP  | IDF1 | MT   | ML   | FP    | FN    | IDs | FM   |
|-------------------|------|-------|------|------|------|-------|-------|-----|------|
| SORT              | 40.2 | 0.251 | 56.1 | 29.7 | 51.4 | 11838 | 74027 | 799 | 1380 |
| GOG_EOC           | 36.9 | 0.242 | 46.5 | 20.5 | 58.9 | 5445  | 86399 | 877 | 1090 |
| SCTrack           | 35.8 | 0.244 | 45.1 | 21.1 | 55.0 | 7298  | 85623 | 798 | 2042 |
| Proposed DeepSORT | 66.5 | 0.259 | 69   | 32.3 | 39.5 | 4722  | 68060 | 779 | 3717 |

#### IV. CONCLUSION

To address the difficulties posed by complex circumstances and enormous volumes of data, this work presents an advanced object tracking system for video surveillance systems. Our method demonstrates improved tracking capabilities by combining the adaptive Deep Simple Online Real-time Tracking (DeepSORT) algorithm with an optimized Kalman filter. The tracking accuracy is further improved by incorporating the Waterwheel Plant Optimization (WPO) approach. The effectiveness of the proposed deep learning-based method in improving object tracking in dynamic scenarios and overcoming the limitations of conventional tracking methods is demonstrated through evaluation using a wide range of performance metrics, such as Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Integrated Detection and False-alarm Rate (IDF1), Mostly Tracked (MT), and Mostly Lost (ML). Future research should focus on developing methods to improve the robustness of the system in dynamically changing situations, such as handling unexpected changes in illumination or adapting to unpredictable object movements.

#### REFERENCES

- [1] J. Luo, H. Chen, Q. Zhang, Y. Xu, H. Huang, and X. Zhao, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Applied Mathematical Modelling*, vol. 64, pp. 654–668, Dec. 2018, <https://doi.org/10.1016/j.apm.2018.07.044>.
- [2] S. M. A. Hasan and K. Ko, "Depth edge detection by image-based smoothing and morphological operations," *Journal of Computational Design and Engineering*, vol. 3, no. 3, pp. 191–197, Jul. 2016, <https://doi.org/10.1016/j.jcde.2016.02.002>.
- [3] Y. Tan, L. Liu, Q. Liu, J. Wang, X. Ma, and H. Ni, "Automatic breast DCE-MRI segmentation using compound morphological operations," in *2011 4th International Conference on Biomedical Engineering and Informatics*, Shanghai, China, 2011, pp. 147–150, <https://doi.org/10.1109/BMEI.2011.6098307>.
- [4] K. K. Verma, P. Kumar, and A. Tomar, "Analysis of moving object detection and tracking in video surveillance system," in *2015 2nd International Conference on Computing for Sustainable Global Development*, New Delhi, India, 2015, pp. 1758–1762.
- [5] R. Assaf, A. Goupil, V. Vrabie, T. Boudier, and M. Kacim, "Persistent homology for object segmentation in multidimensional grayscale images," *Pattern Recognition Letters*, vol. 112, pp. 277–284, Sep. 2018, <https://doi.org/10.1016/j.patrec.2018.08.007>.
- [6] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," *Machine Vision and Applications*, vol. 19, no. 5–6, pp. 345–357, Oct. 2008, <https://doi.org/10.1007/s00138-008-0132-4>.
- [7] Xiru W. U., Guoming H., and Lining S. U. N., "Fast Visual Identification and Location Algorithm for Industrial Sorting Robots Based on Deep Learning," *Robot*, vol. 38, no. 6, pp. 711–719, Nov. 2016, <https://doi.org/10.13973/j.cnki.robot.2016.0711>.
- [8] H. M. Hodgetts, F. Vachon, C. Chamberland, and S. Tremblay, "See no evil: Cognitive challenges of security surveillance and monitoring," *Journal of Applied Research in Memory and Cognition*, vol. 6, no. 3, pp. 230–243, Sep. 2017, <https://doi.org/10.1016/j.jarmac.2017.05.001>.
- [9] R. Verschae and J. Ruiz-del-Solar, "Object Detection: Current and Future Directions," *Frontiers in Robotics and AI*, vol. 2, Nov. 2015, Art. no. 29, <https://doi.org/10.3389/frobt.2015.00029>.
- [10] M. Haghghi and M. Abdel-Mottaleb, "Low Resolution Face Recognition in Surveillance Systems Using Discriminant Correlation Analysis," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition*, Washington, DC, USA, 2017, pp. 912–917, <https://doi.org/10.1109/FG.2017.130>.
- [11] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020, <https://doi.org/10.1016/j.neucom.2019.11.023>.
- [12] M. Elhoseny, "Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 611–630, Feb. 2020, <https://doi.org/10.1007/s00034-019-01234-7>.
- [13] H. Ahn and H.-J. Cho, "Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system," *Personal and Ubiquitous Computing*, vol. 26, no. 2, pp. 385–394, Apr. 2022, <https://doi.org/10.1007/s00779-019-01296-z>.
- [14] K. Ullah, I. Ahmed, M. Ahmad, A. U. Rahman, M. Nawaz, and A. Adnan, "Rotation invariant person tracker using top view," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 11, pp. 15343–15359, Nov. 2023, <https://doi.org/10.1007/s12652-019-01526-5>.
- [15] J. Luiten *et al.*, "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 548–578, Feb. 2021, <https://doi.org/10.1007/s11263-020-01375-2>.
- [16] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, Nov. 2021, <https://doi.org/10.1007/s11263-021-01513-4>.
- [17] Y. Zhang *et al.*, "Long-Term Tracking With Deep Tracklet Association," *IEEE Transactions on Image Processing*, vol. 29, pp. 6694–6706, 2020, <https://doi.org/10.1109/TIP.2020.2993073>.
- [18] A. Shah, "UCSD Pedestrian Database." Kaggle, [Online]. Available: <https://www.kaggle.com/datasets/aryashah2k/ucsd-pedestrian-database>.