International Journal of Advanced Technology and Engineering Exploration, Vol 11(111)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10102017

Research Article

ResNet50-deep affinity network for object detection and tracking in videos

Nandeeswar Sampigehalli Basavaraju[1*] and Pallavi Hallappanavar Basavaraja[2]
Professor, Department of Computer Science and Engineering, AI&ML, AMC Engineering College, Bangalore - 560083, India[1]
Associate Professor, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore-560082, India[2]

Received: 24-July-2023; Revised: 10-February-2024; Accepted: 11-February-2024

Abstract
Multiple-object tracking (MOT) plays a crucial role in addressing many fundamental challenges within the fields of computer vision and video analysis. The majority of MOT methods rely on two primary processes: object detection and data association. Initially, each video frame is analyzed to detect objects, followed by a subsequent step that establishes correlations among the detected objects across multiple frames to generate their tracks. However, the data association for tracking often relies on manually defined criteria such as motion, appearance, grouping, and spatial proximity, among others. In the study, the ResNet50-deep affinity network (DAN) was introduced, which had been designed for the detection and tracking of objects in videos, including those that appear and disappear between frames. The proposed method was evaluated using the widely recognized MOT17 dataset to address MOT challenges. During the preprocessing phase, photometric distortion correction, frame expansion, and cropping were performed. The ResNet50 model was utilized to extract features. The DAN was employed to identify object appearances in video frames and to calculate their cross-frame affinities (CFA). The approach was compared with existing research, including DAN, ByteTrack, the graph neural network for simultaneous detection and tracking (GSDT), the reptile search optimization algorithm with deep learning-based multiple object detection and tracking (RSOADL-MODT), the center graph network (CGTracker), the hybrid motion model, FlowNet2-deep learning, and the super chained tracker (SCT), to validate the efficiency of the ResNet50-DAN method. The ResNet50-DAN method achieved superior results, with a multiple-object tracking accuracy (MOTA) of 84.2%, an F1 score for identification metrics (IDF1) of 80.3%, 10,352 false positives (FP), and 1,284 identity switches (ID-Sw). The ResNet50-DAN method demonstrated higher MOTA compared to the existing approaches, including DAN, ByteTrack, GSDT, RSOADL-MODT, CGTracker, the hybrid motion model, FlowNet2-DL, and SCT.

Keywords
Deep tracking, Multiple object tracking, Object detection, Online tracking, Tracking challenge, Video surveillance.

1.Introduction
Multi-object detection and tracking (MODT) in videos is a significant task that is utilized in a wide range of application fields such as robot navigation, automated driving, and event detection. Machine vision systems (MVS) as well as image processing, are both used for object detection and tracking in multi-object real-time videos [1]. The detection of objects is one of the most basic tasks and current advances in deep convolutional neural networks (CNN) have achieved significant importance in object detection domains. Yet, directly using these object detectors in videos results in difficulties because of data-intensive media with huge complexities and variations [2].

Traffic flow data contains the total number of vehicles, speed, and vehicle entry and exit points to the area and period among entry and exit points. All these data are extracted from the trajectories of vehicles through vehicle detection and tracking approaches [3]. Approaches of computer vision like object detection and tracking, serve as a major tool in analyzing unmanned aerial vehicles (UAV)-related datasets [4]. In the conventional multiple-object tracking (MOT) approach, the data association issues are converted into different challenges [5]. The MOT is categorized based on the usage of video sequences: offline and online tracking [6]. Online video processing systems incorporate deep and machine learning-based image processing models, which are among recent data processing techniques [7].

*Author for correspondence

Many techniques including minimum output sum of squared error (MOSSE), intensities histogram, color names, histogram of gradients (HOG), and features of deep convolutions have been developed to improve the prototype tracker's performance by employing the filters of correlation. Incorporating these advanced features greatly enhances tracking accuracy [8]. Modern computer vision methodologies primarily depend on deep learning (DL), where deeper means greater accuracy and increased computer capacity for supporting the requirements of large memory, however, it also leads to longer processing time [9]. MOT mostly includes two sub-models, a detection technique to locate the targets and a re-identification (Re-ID) for associating them to the trajectory [10, 11].

For detecting objects, different kinds of sensors like light and camera detection and ranging are particularly employed in autonomous vehicles. Among these kinds, the camera image quality is affected by adverse weather conditions like sleeting rain, dusty blasts, heavy fog, and low light conditions. Due to this, visibility becomes very poor, that leads to ineffective vehicle detection as well as traffic accidents on the roads [12]. Multi-target tracking goals include calculating the object locations amidst noise, determining their identities in every frame video, and generating their trajectories [13]. Unsupervised video object segmentation (UVOS) has potential benefits for several applications such as object tracking and action recognition [14]. Enhancing the capability to monitor and track objects, is a primary essential topic in today's society [15]. MOT techniques are utilized in different applications like object collision avoidance and autonomous driving [16]. The MOT approach emerges from motion-color-texture tracking to motion-appearance-mixed tracking which is trained jointly with detection to simplify the process and minimize computational costs [17]. Specifically, MOT utilizes the Kalman filter to predict every object's location in the following frame [18] Basically, the tracking error generates from two parts: temporal association and object detection. [19]. Thus, MOT is known to have rapidly evolved and demonstrated great progress [20]. The MOT's application is very important in crowded places to evaluate people's movement in a video surveillance system [21]. The challenge in MODT is the association of data for tracking which still relies on handcrafted constraints like spatial proximity, motion, appearance, grouping, and so on, for determining affinities among the objects in various frames, as the objects emerge and disappear between the video frames. To overcome this challenge, the ResNet50-deep affinity network (DAN) is proposed for the detection and tracking of several objects in videos which is the main objective of this work. In the following, the primary contributions of this paper are summarized below:

- The proposed ResNet50-DAN carries out several object detections and trackings in videos wherein objects emerge and disappear between the frames of video.
- To extract the features, pairs of video frames and the locations of objects are passed across two streams of convolutional layers. These streams measure the parameters of identical models using the ResNet50 approach.
- A DAN is used to represent where objects appear in video frames and to calculate their cross-frame affinities (CFA).

The remainder of this research is structured as follows: Section 2 presents a review of the literature. Section 3 outlines the proposed methodology. Section 4 reports the results. Section 5 provides a discussion of the findings. The paper concludes with Section 6.

2.Literature survey

Elhoseny [22] implemented a MODT method using a technique of optimal Kalman filtering, to follow moving targets in the video frames and the model of region-expanding was employed to convert the clips of video into morphological action, evolved from the number of frames. Following object evolution, the probability-based grasshopper algorithm was employed to handle the filtering of Kalman for the optimization of parameters, and the chosen items were tracked by a similarity measure in all frames that employed the optimal parameters. The MODT method attained maximum tracking accuracy and detection with Kalman filtering. However, the implemented method had a fairly low detection rate.

Jha et al. [23] presented n-you only look once (N-YOLO)-based tracker with YOLO V3 and a correlation-based tracker. A tracking algorithm technique was combined with picture segmentation and merging of images for real-time detection and tracking of objects. Two potential boundary boxes were extracted from each grid of the image using YOLO, which were divided into equal-sized squares. The object was identified in the extracted candidate boundary box using the boundary box's class identifier, after which, the boundary box was merged

using non-maximum suppression (NMS). Scalability was achieved for real-time video monitoring in edge computing situations having limited processing capacity. However, the method did not manage to re-enter frames as the YOLO lacked re-entered frame detection feature.

Liu et al. [24] implemented a motion-aided feature calibration network (MFCN), an end-to-end learning method for the detection of video objects, and the essential idea was to use motion estimation to examine features' temporal coherence at both pixel levels and instance levels, across frames, to increase detection accuracy and provide greater robustness even in case of variations in appearance. Calibration of instance features, pixel-feature calibration, and adaptive weight computation were the three components that generated the MFCN. Utilization of more effective motion estimation and the feature extraction networks enabled the method to achieve desirable accuracy and run time speed. However, inadequate calibration ranges and lack of enough temporal information did not allow for enhancement in detection performance.

Yu et al. [25] introduced an end-to-end method for the detection of video text with online tracking that correlated with the textual characteristics. It contained multiple explainable features, including pyramidal histogram of characters (PHOC), geometry, and appearance, which helped the model to learn better representations. To create a bridge between the detection and tracking modules, an explainable descriptor and PHOC features were introduced. On the experimented Minetto dataset, the presented method raised the F-score and accurately detected the object. However, the recognition of video text was inappropriately performed by the presented method.

Sun et al. [26] introduced a DAN for MOT that utilized pre-detected objects and generated exhaustive pairing permutation to infer object affinities. It learnt concise yet comprehensive features of previously identified items at numerous abstraction levels. Using twelve distinct evaluation measures, the method was tested on three online multiple challenges of object tracking: MOT15, university at albany detection and tracking (UA DETRAC), and MOT17. In every challenge, the presented tracker performed efficiently and achieved the highest multiple objects tracking accuracy. However, the method was computationally expensive, requiring a significant amount of processing power and memory.

Ji et al. [27] implemented a novel cross-attention encoder-decoder model under the scheme of Siamese (CASNet) for the video salient object detection. To ensure an accurate intraframe object salient recognition, a baseline encoder-decoder design trained with the loss function of Lovasz SoftMax, was employed as a backbone network. The modules of cross-attention and self-attention were included in the model to protect the correlation of saliency and increase the consistency of intraframe salient detection. The image-based design achieved better execution in terms of mean absolute error (MAE) and maximum f-measure (MaxF). However, without the use of low level features, it was difficult to correctly forecast the edges and salient bound objects.

Zhang et al. [28] implemented ByteTrack for multi-object tracking by connecting each detection box. Initially, the detection box of high-score was associated with the tracklets and then, was related with the detection box of low-score. The unpaired tracklets were utilized to retrieve the detection box of low-score objects and the background was filtered at the same time. The Byte was applied easily with previous trackers which aided in consistent enhancement. The ByteTrack was robust enough to deter occlusion in its performance for accurate detection and assisted the detection box of low-score. However, the performance of ByteTrack was affected while objects had significant modifications in orientation and scale. Wang et al. [29] presented a graph neural network (GNN) for graph neural network for simultaneous detection and tracking (GSDT). The primary goal of the GNN was to model the relations among variable object sizes in both temporal and spatial domains, which was significant for learning the discriminative features for data association and detection. This approach had greater average precision that made it easier to determine existing objects. However, with this GNN approach it was harder to decrease false positive (FP) by using the relation data of an object.

Alagarsamy and Muneeswaran [30] developed reptile search optimization algorithm with deep learning-based multiple-object detection and tracking (RSOADL-MODT). Initially, this approach employed a path-augmented retinanet (PA-RetinaNet) which enhanced the process of feature extraction. Then, the reptile search optimization algorithm (RSOA) was employed as a

hyperparameter optimizer to enhance the PA-RetinaNet's potential. Finally, the quasi-recurrent neural network (QRNN) classifier was utilized for classification. The developed approach improved the efficiency significantly but the RSOA DL was computationally expensive.

Feng et al. [31] introduced a center graph network (CGTracker) for one-stage multi-pedestrian detection and tracking of objects. This approach predicted the target of the pedestrian as the object center point and extracted directly the object features from the representation of object center point features which was employed to detect the axis-aligned bounding box. The CGTracker achieved the optimal precision for the prediction of object location in tracking and this approach was employed in the application of real-time MOT. However, the CGTracker struggled with situations where pedestrians were greatly occluded by other individuals or objects.

Qureshi et al. [32] implemented the approach of super chained tracker (SCT) that depended on Kalman filtering and bipartite matching for online MOT, to improve feature extraction, data association, and the detection of objects. The informative regions were increased by employing SCT's regression of box pair with assistance of a joint attention unit to increase the performance. The implemented SCT approach was effective and generated relatively better results than the other approaches. However, the anchors of tracking were highly intricate and needed a lot of calculations, that lowered the system's efficacy.

Wu et al. [33] presented a hybrid motion model to enhance the accuracy of tracking in mobile devices. Initially, the approach determined the hypotheses of camera motion by computing transition smoothness and optimal flow similarity, to generate the estimation of camera trajectory. Then, the projection of smooth dynamic was employed with the camera trajectory to determine objects from an image. The spatiotemporal approach was established in tracklets association which achieved greater discriminability in the measurement of motion. However, this approach suffered from greater space costs and time for long video-dense objects.

Singh and Srivastava [34] developed a hybrid approach that employed flowNet-2 DL that computed target-wise motions for an unknown number of objects from the optical flows of pixel level. The matching approach directly employed the two associative frame objects for tracking and detection. The distance between the associate position of one boundary to the other was utilized for small switching of identities. This approach achieved greater accuracy and efficiency by a large margin on different datasets. However, it incurred high computational costs while scaling up for a large number of objects in the settings of MOT.

Xuan et al. [35] introduced a rotation-adaptive correlation filter (RACF) tracking approach to solve the issue caused by object rotation. This tracker was used to calculate the angle of object rotation. The tracker rotated the feature map of each frame to an identical angle according to the object rotation angle which maintained a stable feature map even if the object had a huge rotation range. Thus, by rotating the extracted image with calculated angles, the RACF approach solved the issue found in HOG-based trackers which did not properly track the rotation of the object. However, the object's initial angle was not calculated accurately when the background jitters occurred, which ultimately impacted the RACF's accuracy.

Suljagic et al. [36] implemented a similarity-based person re-id framework (SAT) utilizing a siamese neural network (SNN) for MOT by sharing weights. The SAT employed a Siamese feature extraction method once the detection was acquired from the backbone. Then, the similarity array was applied for evaluating the detection and tracklets. This method was enhanced by a precise and reliable object detector which avoided potential detection misses. However, sharing weights in SAT led to overfitting in particular object scenes which minimized generalizability.

Ma et al. [37] presented an associative affinity network (AAN) for MOT in videos. The AAN determined the associative affinity among detection and tracks through frames. The binary classifier was employed directly to examine the associative affinity of every pair of track-detection and then used a matching cardinality loss to obtain data between candidate pairs. The AAN handled the consistency of trajectory in the presence of missed detection effectively. However, the Siamese network-based single-object tracking (SOT) was time-consuming under dense trajectory conditions.

Xu et al. [38] developed TransCenter with a dense representation for accurately tracking every object. The usage of associated image-dense detection

queries and effective sparse tracking queries was established by employing query learning networks (QLN). The dense image enabled derivation of the target's location robustly and globally via the outcome of dense heatmap. The tracking queries efficiently communicated with feature images in the developed approach to relate the position of the object by time. The TransCenter generated a remarkably enhanced performance by a huge margin and had proven to be accurate and efficient. However, this approach struggled in scenarios with different-density objects, in which it was not able to adapt as per various levels of object congestion.

Wu et al. [39] introduced a dual-path transformation network (DTN) for MOT which minimized the contradictions of optimization with the network through decoupling sharing features into the representation of specific tasks. The pyramid non-local network (PNN) was employed to improve the representation of specific Re-ID scaling data. This approach enhanced the MOT efficiency and was appropriate for real-time applications. However, the interaction of frequent inter-objects and occlusions gave rise to significant constraints against robust tracking.

Hu and Jeon [40] implemented a feature-fused transformer for improved multiple-object tracking (FFTransMoT). The decoder incorporated enhanced features for accurately matching objects through frames which improved significantly the capabilities of the model's tracking. This decoder examined the association of matching data among frames and fused features. To obtain dependencies among input features, a self-attention approach was utilized which increased stability and accuracy of object detection. This feature procedure enhanced the feature set and reliability for the subsequent decoding of data association. However, while integrating different feature types, inconsistencies arose which impacted the fusion outcomes.

Chen et al. [41] presented a trajectory extraction and optimization approach that depended on multi-target tracking for multi-object video tracking. To pre-process and extract the crowd trajectory video data, the multi-target tracking approach was utilized. The trajectory was optimized by integrating the trajectory point extraction and Savitzky-Golay smoothing filtering approach. The presented approach obtained the real features of trajectories that increased the smoothing index of the trajectory. However, upstream trajectory data was greatly based on multi-tracking performance. The speed and accuracy also had essential room for enhancement, which was clearly not used by this presented model.

Xiang et al. [42] developed an efficient channel attention and switchable atrous convolution for MOT. The channel attention approach was employed to extract the data in images. Then, switchable atrous convolution was utilized in the network to adjust the associated field dynamically due to the object changes. The association effect was improved in the developed approach by saving the trajectory features of minimum occlusion. However, combining both approaches increased computational demands which affected the performance of real-time tracking.

Lee et al. [43] introduced a graph convolutional network (GCN) for MOT to extract the initial features and modify the features that compute the affinity among nodes. The object pose features were employed from the object joint to extract the features. Then, the node features were updated by measuring the updated edge features and the related strength between the detection and tracker. The introduced GCN approach was lighter and achieved maximum tracking significantly. However, due to the graph complexity, as the number of trackers and detection became more, the speed of tracking minimized.

Li et al. [44] presented motion estimation and multi-stage association (MEMA) for multi-target tracking. Initially, the bounding box of the ground-true aspect ratio was established to maximize the detection fit. The elliptical gaussian kernel was employed to increase the object center point accuracy. Then, the Kalman filter was utilized to detect the velocity and width directly, and finally, to combine various confidence boxes, the multi-stage association (MA) was created. This approach decreased the occlusion impact and maximized the performance of tracking without employing appearance features. However, the MEMA struggled to manage intricate scenes with occlusion which led to insignificant performance in such situations.

Chen et al. [45] implemented a Zone-based clustering for improving pedestrian group tracking and detection. Initially, the approach of object detection was to extract pedestrians and their box of bounds from an image. Then, to divide the image into various zones, zone detection was employed. To detect the group of pedestrians, the clustering approach was established with each zone and finally, an object tracking approach was utilized to track pedestrians.

The tracking of various pedestrian groups by this group detection approach, was greatly more time-efficient than individual tracking. However, this approach faced challenges in accurately defining zones in crowded scenes which resulted in missed detection.

Liang et al. [46] developed a NMS for MOT. Due to the intersection over union (IOU) among detected and predicted boxes, the irrelevant data was removed. To increase the adaptability of the domain, an unsupervised pre-trained person ReID was utilized. Furthermore, bicubic interpolation was employed to enhance the low-scoring resolution boxes. By establishing pre-training on the person ReID, the domain gap was increased in the ability of the ReID network simultaneously. However, the NMS approach potentially removed the overlapping object detection. Hence, some objects were missed from being detected.

Li et al. [47] introduced a lightweight Re-ID network which was termed as fast omni-scale network (Fast-OSNet) for MOT. The hierarchical adaptive exponential moving average (HAEMA) was employed to decrease the occlusion noise effect on the trajectory's appearance, by use of adaptive modified weights with a two-stage linear transformation. The introduced Fast-OSNet approach generated a robust tracking performance in intricate scenarios. However, the fast OSNet had constraints in adapting to different object variations and appearance, which affected the robustness.

2.1 Review

There are some limitations of object detection and tracking in the videos, as mentioned above, such as computational complexity, objects that have significant modifications in orientation and scale, and the difficulty in decreasing FP by using the relation data of an object. To overcome these issues, the ResNet50-DAN is proposed for object detection and tracking in videos.

3.Methods

The online MOT was executed by utilizing the representational strength of DL schemes. The overview of object detection and tracking in video is shown as a block diagram in *Figure 1*.

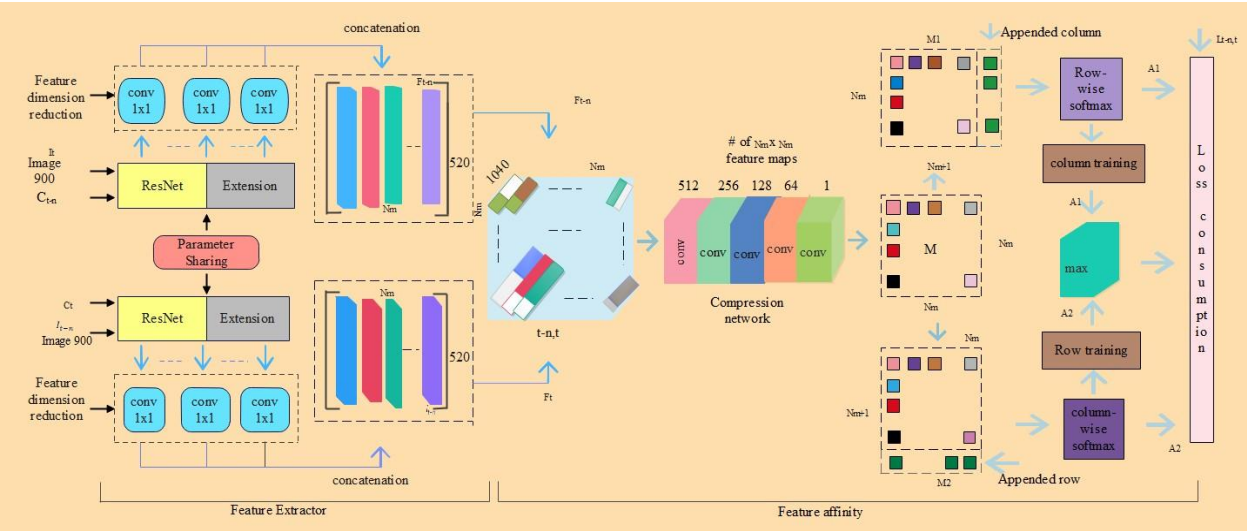

Figure 1 Block diagram of the proposed method

3.1Dataset

MOT 17 [48] is one of the most recent tracking challenges to be offered online. Similar to MOT 16, this challenge featured seven various indoor and outdoor scenarios of public spaces with pedestrians. Each scene had two segments, one for training and the other for testing, in a separate video. By using three detectors: deformable part models (DPM), faster-region-based convolutional neural network (Faster-RCNN), and scale-dependent pooling (SDP), the dataset offered object detection in video frames. The challenge accepted both online as well as offline tracking systems, with the latter permitted to use predicted tracks from upcoming video frames. The dataset was divided into training, testing, and validation sets in the ratio of 70:20:10. The training video clips were not included in the testing data, which only included clips from the sequences. There

were 17757 testing data frames and 2355 tracks with 564228 boxes in those clips.

3.2Pre-processing

In photometric distortions, all pixels of a video frame were scaled by incidental values between (0.7, 0.15). The image was translated to the scheme of hue saturation value (HSV), and the channel of saturation was scaled by an incidental value in the range (0.7, 0.15). The frame was then rescaled by a randomly picked value in the equal range and translated back to the format of red green blue (RGB). By applying a random ratio selected from the interval (1, 1.2), the frames were enlarged. As a result of this growth in frame size, the original frame was extended with additional pixels to get a new size. These additional pixels had a value equal to the value of the mean pixel of the training data. Utilizing cropping ratios that were incidentally picked in the (0.8, 1) range, the frames were cropped and the crops that had the centers of every box detected in the original frames, were maintained. With a probability of 0.3, each step was applied to the frame pairings in sequences and the frames were then horizontally reversed with a frequency of 0.5 and enlarged to a fixed dimension H×W×3. The final processed frames and the associated object centers were calculated by the detector and were fed into the DAN. Object occlusions were considered at this step as well. At this stage, the training technique ignored fully occluded objects in the training data. The visibility threshold of 0.3 was chosen for determining an object as completely occluded and positive samples of occluded items were the remaining partially occluded objects. By placing a bound of upper N_m on the maximum object number established in a specified frame, the data association matrices were built for the frames. $N_m = 80$ is known to be a very proven and suitable bound for experiments concerning benchmark problems. For consistency, extra columns and rows were added to the matrix of data association that correlates to the dummy bounding boxes in every frame of the video, that resulted in a matrix of $N_m \times N_m$ in size and eventually included N_m objects in every frame. Here N_m refers to maximum number of objects present in a frame.

3.3Deep affinity network (DAN)

DAN is used to represent where objects appear in video frames and to calculate their CFA. The proposed method included two parts, an extractor of features, and an affinity estimator. Moreover, the entire network was trained end to end, the frame of video I_t and its centers of the object C_t were considered for the DAN training, and the two frames were not required to play back one after the other in a video. Instead, these were enabled to be separated by n times such that $n \in N \overset{rand}{\sim} [1, N_v]$. Despite that the network was mainly used to track items in the successive video frames, training it using a non-consecutive frame improved the total strategy by enabling reliable association between items in a current frame and those in several prior frames. To calculate the cost of the network during training, DAN needs the input frame pair's matrix of ground truth association of binary data $L_{t-n,t}$. In contrast to DAN, where the extractor of the feature element is trained as a network of two-stream, the method employed it as a one-stream design, and the parameters were shared by the two streams. The deployment of DAN by separating its two key parts was demonstrated. The network predicted one frame and used the location of the object center C_t as its input and video frame I_{t-n} with its $C_t - n$ object centers. The latter calculated the permutation tensor $\varphi_{t-n,t}$ for the frame pair using the prior frame matrix of the feature I_{t-n}. Then, using the network's basic forward pass and concatenation process as previously mentioned, the tensor was translated to an affinity matrix. As a result, all frames were processed over the detector of the object and feature extractor. However, the features were employed more than once to compute affinities with numerous additional frames in pairs.

$L_{t-n,t}$ is a matrix of binary data association used to represent the relationship between the objects found in frames I_{t-n} and I_t, I_{t-n} is video frame, f_t is feature matrix linked with the t^{th} frame, $\varphi_{t-n,t}$ is the permutation tensor, and $t-n, t$- entity is computed for the pair of frames I_{t-n} and I_t. The currently displayed picture frame I_t and the preceding frame picture I_{t-n} were fed into the front-end convolution layer (FE-Conv) for the feature extraction, together with the center points of the previously recognized objects. In DAN, the FE-Conv layer is a shortened name for the network extension of ResNet50. Then, from nine locations in the layer of FE-Conv, a certain number of feature maps were dimensionally minimized to 520. Following that, the features in each frame that correspond to the centre points of each detected item were removed to generate features maps f_t and f_{t-n}. These were combined in the FF module, whose output served as the input for the back-end convolution layer (BE-Conv), which produced an affinity matrix $M_{t,t-n}$ as the tracking outcome. It should be noted that the BE-Conv, which

is the central component that handles DAN functions, is a shortened name of the compression network. In *Figure 2*, the feature maps are extracted using four dilated kernels of various scales, following which the dilated convolution outputs are connected, and lastly, the feature maps are combined from 320×4 to 256 using a squashing layer. The feature map size decreases as the number of layers rises. Small feature maps have the drawback of losing data regarding limited objects, and huge ones lack sophisticated semantic data. In the FE-Conv, the atrous spatial pyramid pooling (ASPP) modules are required to be arranged in various locations for extracting the holistic features. First, a dilated convolution is produced using four various kernels that are scaled according to the dilated rate λ. Then, various scale feature maps are created, and these feature maps are displayed in various shades of violet, red, blue, and black, after which all of these feature maps are concatenated together. To connect the feature maps and lighten the computational load, a squashing layer is added. This layer is made up of batch normalization, rectified linear units (ReLU) and BE-Conv. The ASPP modules are arranged in the FE-Conv in as many different locations as feasible apart from one another to extract the holistic properties.

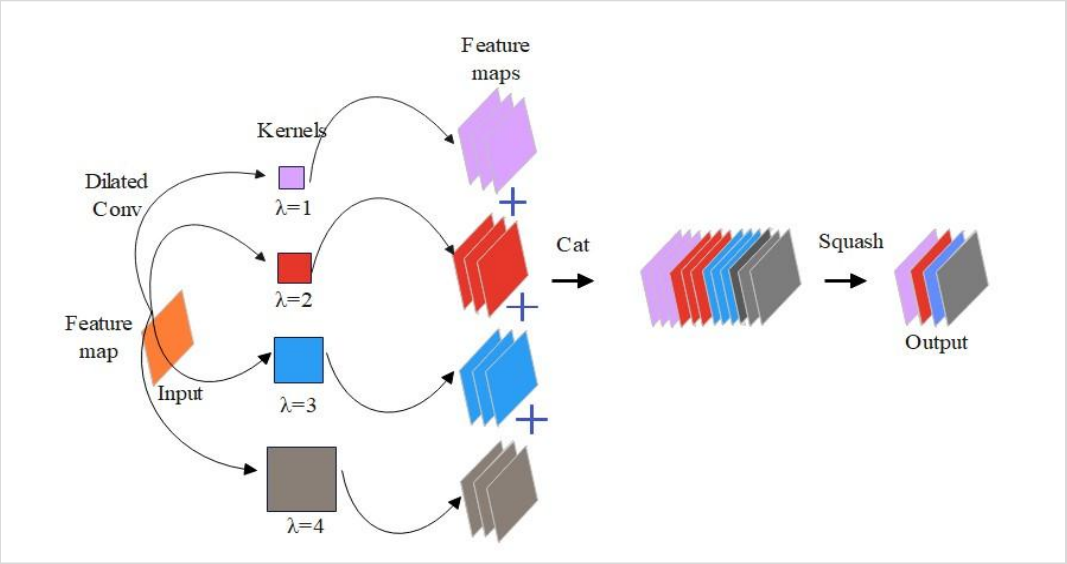

Figure 2 Module of ASPP

3.4Feature extractor

To extract the features, pairs of video frames and the locations of objects were passed across two streams of convolutional layers and these streams measure the parameters of identical models, but their architecture is modeled by the ResNet-50 network. After transforming the fully connected and SoftMax layers to convolutional layers, ResNet-50 architecture was used. This change was done because convolution layers are better at encoding spatial properties of objects, which are more important, and the network size of the input frame is significantly greater which is 3×900×900 as opposed to the original ResNet. ResNet-50 [49, 50] is a 50-layer CNN out of which 48 are convolutional layers, one is average pool layer, and the other remaining one is max pool layer. Identity and convolutional blocks are used in the remaining states. A convolutional layer with activation functions and further batch normalization, was included in these two blocks. The input layer had an additional bridge to the output layer to enhance the residuals of the convolutional blocks. The residual block on ResNet 50 is expressed in Equation 1.

$$y = F(x, W + x) \quad (1)$$

Where x and y denote the input and output layers respectively, function F refers to the residual map and W represents weight of the corresponding layer. When the input and output values on the ResNet 50 are identical, the residual block is processed. According to the convention, the activation of ReLU and Batch normalization are counted as distinct layers and indexed in the last layer of the ResNet50.

3.5Affinity estimator

The purpose of the ResNet50-DAN model is to encode affinities between objects by using the retrieved features. To achieve this, the network set the columns of f_t and f_{t-n} in a tensor $\varphi \in R^{N_m \times N_m \times (520 \times 2)}$ in such a way that the columns of the two feature matrices were connected across the

dimension of tensor depth in $N_m \times N_m$ permutations. A compression network was utilized with 5 convolution layers and 11 kernels, to transfer this tensor on the matrix $M \epsilon R^{N_m \times N_m}$. The physical relevance of the output and input signals inspired the compression network's architecture. A tensor that stored object features mixture, was converted by the network into a matrix, which coded similarity among features. A forward pass by DAN was assumed, until the matrix $M \epsilon R^{N_m \times N_m}$ was calculated. To calculate losses, the rest of the network correlated this ground truth association matrix of data $L_{t-n,t} \epsilon R^{(N_m+1)\times(N_m+1)}$. The items that enter or exit the video between the two input frames were not taken into consideration unlike $L_{t-n,t,} M$. An additional column and row was added to M to create the matrices $M_1 \epsilon R^{N_m \times (N_m+1)}$ and $M_2 \epsilon R^{(N_m+1)\times N_m}$ to take care of those items. To maintain a precise and physically understandable loss computation, the column and row vectors were distinctly added to M and these vectors were of the form $V \in R^{N_m} = \gamma 1$, where 1 is the vector of ones, and γ is the DAN's hyperparameter.

3.6Network loss

The m^{th} row of M_1 in the formulation, linked the m^{th} frame identity I_{t-n} to $N_m + 1$ recognized in frame I_t, where +1 came from the unknown objects in I_t. By performing a row-wise SoftMax operation over the resulting matrix, a different probability distribution was fitted to each row of M_1. As a result, each resulting matrix row $A_1 \in R^{N_m\, x\, (N_m+1)}$ represented probabilistic correlations between an object in frame I_{t-n} and all identities of frame I_t. To compute $A_2 \in R^{(N_m+1)x\, N_m}$, whose columns represent identical associations backward from the frame I_t to I_{t-n}, a column-wise SoftMax operation was appropriately executed over M_2. It implied that several objects arrived or exitted the video between the two frames due to the probabilistic object association. The highest number of permitted items in a frame N_m served as the upper boundary for the number of objects that entered or exitted the video. To evaluate loss function for DAN, four sub-losses were used, and these were, (i) forward-direction loss L_f that promotes accurate identification from I_{t-n} to I_t, (ii) backward-direction loss L_b, which assures that the connections between I_t and I_{t-n} are accurate, (iii) loss of consistency L_c to rebuff any disparity between L_f and L_b and, (iv) total loss L_a which suppressed non-maximum backward and forward associations for affinity forecasts. The above definitions of the losses are provided in Equations 2 to 6.

$$L_f(L_1, A_1) = \frac{\sum_{coeff}(L_1 \odot (-\log A_1))}{\sum_{coeff}(L_1)} \quad (2)$$

$$L_b(L_2, A_2) = \frac{\sum_{coeff}(L_2 \odot (-\log A_2))}{\sum_{coeff}(L_2)} \quad (3)$$

$$L_c(\widehat{A_1}, \widehat{A_2}) = \left\|\widehat{A_1} - \widehat{A_2}\right\| 1 \quad (4)$$

$$L_a(L_3, \widehat{A_1}, \widehat{A_2}) = \frac{\sum_{coeff}(L_3 \odot (-\log(max(\widehat{A_1}, \widehat{A_2}))))}{\sum_{coeff}(L_3)} \quad (5)$$

$$L = \frac{L_f + L_b + L_a + L_c}{4} \quad (6)$$

Where L_1, L_2 are $L_{t-n,t}$ trimmed versions that were created by removing the last row and column, A_1, A_2 represent the matrix A_1, A_2 reduced to size $N_m\, x\, N_m$, $\odot$ is Hadamard product, $\sum_{coeff}(.)$ adds up all the matrix's coefficients to produce a scalar value, and Max, log is used to perform element-wise operations.

The four sub-losses's mean value was used to calculate the overall loss L and the expected value of the training loss data was used to define the overall network cost function. The backward and forward loss direction, increased the probabilities represented by the A_q related coefficients where $q \in \{1,2\}$, to approach equivalent L_q by applying metrics of distance. This approach makes more sense than minimizing the gap between a probability matrix A_q and a binary matrix L_q. Similar to this, l_1 distance was used rather than the usual l_2 distance for the loss consistency, since the difference between $\widehat{A_1}$ and $\widehat{A_2}$ was predicted to be small. The matrix of affinity for an input pair frame was computed by the DAN using the formula $A \in R^{N_m x\, N_m+1} = A(max(\widehat{A_1}, \widehat{A_2}))$ where $A(.)$ added the $(N_m + 1)$ column of A_1 to the matrix in its argument. The performance of max was carried out to compute the assemble loss which was also justified by the maximization employed by the affinity matrix. As a result, the four sub-losses were correlative and produced an accurate estimate of the ground truth association data.

4.Results

In this paper, the ResNet50-DAN was simulated using MATLAB 2018 environment with the system requirements: Intel Core i7, 16 GB of RAM, Windows 10 (64-bit) operating system, graphics processing unit (GPU): 22gb, framework: YOLOv4, and Libraries: OpenCV, TensorFlow. The performance evaluation of the suggested method was considered by means of multiple-object tracking accuracy (MOTA), multiple -object tracking precision (MOTP), f1 score of correctly identified detection (IDF1), identity detection prediction (IDP), and identity detection recall (IDR) which assessed the

suggested technique on the popular MOT17 task. After the novel technique was submitted for examination, tracking results were computed by a hosting server. For generating the models, annotated training data was provided however, the test data labels were not made accessible. The servers used a number of common metrics to thoroughly evaluate the submitted techniques.

4.1Evaluation metrics

The evaluation of the approach was thoroughly conducted utilizing twelve common evaluation criteria, including the MOTA, MOTP, IDF1 metrics.

- **MOTA** – It indicates the total number of misses, mismatch errors, FP, etc. that the tracker system has produced which is expressed in Equation 7.

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + ID_{Sw})}{\sum_t GT_t} \quad (7)$$

Where FP represents false positive, FN denotes false negative (FN), ID_{Sw} represents identity switches, GT denotes stands for ground truth, and t represents computed value at the t^{th} value.

- **MOTP**-By averaging the overlap among all correctly matched forecasts and their ground truth and calculates the localization accuracy. It is expressed in Equation 8.

$$MOTP = \frac{1}{|TP|} \sum_{TP} S \quad (8)$$

Where TP represents true positive and S denotes sum of all values.

- **IDF1**-It determines whether trajectories are present by computing a bijective mapping between the gtTrajs and prTrajs sets and combines IDP and IDR. It is expressed in Equation 9 to 11.

$$ID - Recall = \frac{|IDTP|}{|IDTP| + |IDFN|} \quad (9)$$

$$ID - Precision = \frac{|IDTP|}{|IDTP| + |IDFP|} \quad (10)$$

$$IDF1 = \frac{|IDTP|}{|IDTP| + 0.5\,|IDFN| + 0.5\,|IDFP|} \quad (11)$$

Where $IDTP$ represents ID true positive, $IDFN$ denotes ID false positive, $IDFN$ represents ID false negative, and $IDFP$ denotes ID false positive.

4.2Performance analysis

Pytorch framework was used to create the ResNet50-DAN and the NVIDIA GeForce GTX Titan GPU was used for training. Hyperparameter of DAN was optimized with assistance of MOT-17 dataset for specifying the set of validation to train the model. For parameter optimization, MOT17 was used because of its manageable size and the following were the values of hyperparameters that were eventually used in the implementation. The frame's input size for the network was 900×900 and before being transmitted through the network, all training and testing data were reduced to these dimensions. DAN was trained using the stochastic gradient descent (SGD) optimizer that updates weights depending on the gradient of the whole training dataset. The momentum 0.9 and parameter of weight decay 5e-4 were used subsequently. The rate of learning started at 0.01 and decreased to $1/10^{th}$ of the prior value at epochs 50, 80, and 100. The size of batch refers the number of trainings employed in iterations. A higher batch size speeds up training but needs more memory. It selects based on dataset size and available memory. The common choices were 32, 64, 128, etc. The parameter values like learning rate $\delta_w$and bias term $\delta_b$were examined once the network was trained. The grid is created using multiples of three in the range [3, 30] and the final implementation used $\delta_w = 12$ and $\delta_b = 15$. Optimizing hyperparameters like δ_w, δ_b significantly influenced model convergence and accuracy.

Table 1 represents MOT17's training data attributes in which $move$ denotes a yes (Y) or no (N) to indicate whether or not a moving camera was used to record the video. The hosting server was established by to train the ResNet50-DAN and the method was evaluated by the server after submission. The most recent results of the approach were taken from the challenge server's leaderboard and made available. Although it was an online challenge task, it also included the greatest results of offline techniques, which were taken for comparing with the proposed model to justify its relatively highly effective strategy.

The proposed method ResNet50-DAN exceeded the online and offline approaches on five criteria while generally maintaining competitive performance on the remaining metrics. In particular, it outperformed the currently available online techniques by means of MOTA and MOTL, which are commonly recognized as multiple objects tracking measures. The computational complexity of feeding a complex large video through a pre-trained ResNet model is primarily determined by the forward pass through the neural network. For a single image, the complexity is roughly proportional to the number of operations in the forward pass, which is typically measured in terms of floating-point operation (FLOP). The region proposal network (RPN) introduces additional complexity, as it involves running a set of

convolutions and generating region proposals. The computational complexity depends on the number of proposals generated and the spatial dimensions of the feature maps. The DAN introduces its own set of convolutions and pairwise affinity computations. The complexity is influenced by the architecture of the network and the size of the affinity predictions. Training involves multiple iterations (epochs) over the large dataset. The computational complexity during training includes the forward and backward passes for each batch, parameter updates, and possibly regularization steps. The complexity is influenced by the size of the dataset, the number of training epochs, and the batch size. NMS is a relatively lightweight operation, and its computational complexity is proportional to the number of bounding boxes generated by the model.

Table 1 MOT17's attributes training data

Video index	Resolutions	FPS	Length (frames)	Boxes	Tracks	Density	Move	MODT	MFCN	DAN	CASNet	VGG-DAN	Proposed method ResNet50-DAN
02	1920 × 1080	30	600	18581	62	31.0	N	4162	4214	3520	4958	2014	5347
04	1920 × 1080	30	1050	47557	83	45.3	N	34587	34617	24712	14887	5217	45498
05	640 × 480	14	837	6917	133	8.3	Y	5887	5992	3147	4285	6375	6525
09	1920 × 1080	30	525	5325	26	10.1	N	5992	5994	6102	6204	6347	6487
10	1920 × 1080	30	654	12839	57	19.6	Y	11475	6518	6634	5756	7867	16812
11	1920 × 1080	30	900	9436	75	15.5	Y	6485	5574	4778	6843	6920	6935
13	1920 × 1080	25	750	11642	110	8.3	Y	7712	6865	6974	7987	9991	10994

4.3Comparative analysis

This section provides the comparative analysis of proposed ResNet50-DAN as per the metrics of MOTA, IDF1, FP, and ID-Sw, to evaluate the proposed method. *Table 2* shows the comparative analysis with the existing methods DAN [26], ByteTrack [28], GSDT [29], RSOADL-MODT [30], CGTracker [31], Hybrid motion model [33], FlowNet2-DL [34], and SAT [36]. When compared with the existing methods, the proposed ResNet50-DAN achieves better MOTA of 84.2%, IDF1 of 80.3%, FP of 10352, and ID-Sw of 1284 respectively.

Table 2 Comparative analysis with existing methods

Author	Method	MOTA (%)	IDF1 (%)	FP	ID-S_w
Sun et al. [26]	DAN	52.4224	49.4934	25423	8431
Zhang et al. [28]	ByteTrack	80.3	77.3	25491	N/A
Wang et al. [29]	GSDT	73.2	66.5	N/A	N/A
Alargarsamy and Muneeswaran [30]	RSOADL-MODT	74.67	72.31	N/A	4331
Feng et al. [31]	CGTracker	69.3	62.8	22434	5682
Wu et al. [33]	Hybrid motion model	62.0	65.7	25628	1457
Singh and Srivastava [34]	FlowNet2-DL	74.2	73.1	N/A	N/A
Suljagic et al. [36]	SAT	58.9	58.1	11125	N/A
Proposed method	ResNet50-DAN	84.2	80.3	10352	1284

5.Discussion

The computational efficiency of the method namely, ResNet-50 feature extraction combined with DAN for object detection, varies based on several factors, including model architecture, hardware, and implementation details. The choice of ResNet architecture (e.g., ResNet-50, ResNet-101) and the design of the DAN significantly impact computational efficiency. Smaller models may be faster but might sacrifice some accuracy. Utilizing GPU acceleration significantly speeds up the computations involved in DL tasks. GPUs are well-suited for the parallelized operations common in CNN. Processing multiple frames in a batch leads to better GPU utilization. The time to process a single frame might be higher than processing multiple

frames simultaneously due to parallelization benefits. Efficient implementation practices, such as using optimized DL frameworks (e.g., TensorFlow, PyTorch), model quantization, and low-level optimizations, contribute to computational efficiency. Techniques like quantization (reducing the precision of weights and activations) and pruning (removing certain connections or neurons) are applied to decrease model size and speed up inference at the cost of minimal loss in accuracy. Comparing the computational efficiency of this method with other object detection methods such as YOLO, single shot detector (SSD), faster R-CNN, would require benchmarking on a specific dataset and hardware configuration. Some methods may be faster but sacrifice accuracy, while others prioritize accuracy at the expense of speed. For real-time processing, the frame processing time should be within the desired frame rate. Achieving real-time performance of 30 frames per second necessitates a balance between model complexity and hardware capabilities.

The main objective of this work was to carry out MODT in videos that emerge and disappear between the video frames, by using ResNet50-DAN approach. The advantages of the proposed method and the limitations of existing methods are discussed here. The existing approaches such as DAN [26] was computationally expensive, requiring a significant amount of processing power and memory. ByteTrack [28] was impacted with objects having significant modifications in orientation and scale. GSDT [29] was harder to decrease FP by using the relation data of object. RSOADL-MODT [30] was computationally expensive. CGTracker [31] struggled with situations where pedestrians were greatly occluded by other individuals or objects. Hybrid motion model [33] suffered from space costs and large time for long video dense objects. The proposed ResNet50-DAN approach overcame the existing models' limitations. The ResNet50 approach enabled significant feature extraction in videos which improved the accuracy for object detection. Its skip connections, reduced vanishing gradient issues, and generated robust and stable tracking over the video frames. DAN effectively captured the relationship of spatial-temporal and increased MODT's efficacy in videos. By combining both ResNet50 and DAN, better results were provided. The results were evaluated for ResNet50-DAN to compute the MODT in videos. The obtained results represent that ResNet50-DAN achieved better MOTA of 84.2% respectively. Moreover, the proposed approach was evaluated in MOT17 dataset. The comparative results represent that ResNet50-DAN achieved better outcomes in terms of MOTA measured at 84.2% which is comparatively higher than the existing approaches DAN (52.4224%), ByteTrack (80.3%), GSDT (73.2%), RSOADL-MODT (74.67%), CGTracker (69.3%), hybrid motion model (62.0%), FlowNet-DL (74.2%), and SAT (58.9%) respectively. Hence, the overall results show the efficiency of proposed approach that helps in better MODT in videos, based on overall metrics.

5.1Limitation

The proposed ResNet50-DAN model exhibits a limitation in accurately detecting overlapping objects. This issue stems from the feature extraction process in deep networks, wherein the features of overlapping objects tend to merge. In scenarios featuring objects that are closely or densely positioned, the ResNet50-DAN model may misclassify or fail to identify individual objects.

A complete list of abbreviations and symbols is summarized in *Appendices I and II*, respectively.

6.Conclusion

In this study, the ResNet50-DAN model was introduced for object detection and tracking in videos. The evaluation of the model's performance utilized images from the MOT17 dataset. During the preprocessing stage, operations such as photometric distortion correction, frame expansion, and cropping were carried out. The ResNet50 model was used for feature extraction, while the DAN was employed to determine the locations of objects in video frames and to compute their Cross-Frame Affinities (CFA). The proposed ResNet50-DAN model achieved superior results in Multiple Object Tracking Accuracy (MOTA), reaching 84.2%, compared to existing methods. Future work will focus on addressing occlusion challenges during detection.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The MOT17 dataset utilized in this study is publicly accessible and can be found at https://motchallenge.net/data/MOT17/.

Author's contribution statement

The background work, conceptualization, methodology,

dataset collection, implementation, result analysis and comparison, draft preparation and editing, and visualization for the paper were conducted by Nandeeswar Sampigehalli Basavaraju. Supervision, work review, and project administration were provided by Pallavi Hallappanavar Basavaraja.

References

[1] Pramanik A, Pal SK, Maiti J, Mitra P. Granulated RCNN and multi-class deep sort for multi-object detection and tracking. IEEE Transactions on Emerging Topics in Computational Intelligence. 2021; 6(1):171-81.

[2] Deng J, Pan Y, Yao T, Zhou W, Li H, Mei T. Single shot video object detector. IEEE Transactions on Multimedia. 2020; 23:846-58.

[3] Mao H, Chen Y, Li Z, Chen P, Chen F. SCTracker: multi-object tracking with shape and confidence constraints. IEEE Sensors Journal. 2023; 24(3):3123-30.

[4] Ariza-sentís M, Baja H, Vélez S, Valente J. Object detection and tracking on UAV RGB videos for early extraction of grape phenotypic traits. Computers and Electronics in Agriculture. 2023; 211:108051.

[5] Gao X, Wang Z, Wang X, Zhang S, Zhuang S, Wang H. DetTrack: an algorithm for multiple object tracking by improving occlusion object detection. Electronics. 2023; 13(1):1-16.

[6] Li J, Piao Y. Multi-object tracking based on re-identification enhancement and associated correction. Applied Sciences. 2023; 13(17):1-16.

[7] Azimjonov J, Özmen A. A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. Advanced Engineering Informatics. 2021; 50:101393.

[8] Lu X, Ma C, Ni B, Yang X. Adaptive region proposal with channel regularization for robust object tracking. IEEE Transactions on Circuits and Systems for Video Technology. 2019; 31(4):1268-82.

[9] Fernández-sanjurjo M, Mucientes M, Brea VM. Real-time multiple object visual tracking for embedded GPU systems. IEEE Internet of Things Journal. 2021; 8(11):9177-88.

[10] Yu E, Li Z, Han S, Wang H. Relationtrack: relation-aware multiple object tracking with decoupled representation. IEEE Transactions on Multimedia. 2022; 25: 2686-97.

[11] Gu S, Ma J, Hui G, Xiao Q, Shi W. STMT: spatio-temporal memory transformer for multi-object tracking. Applied Intelligence. 2023; 53(20):23426-41.

[12] Hassaballah M, Kenk MA, Muhammad K, Minaee S. Vehicle detection and tracking in adverse weather using a deep learning framework. IEEE Transactions on Intelligent Transportation Systems. 2020; 22(7):4230-42.

[13] Baisa NL. Occlusion-robust online multi-object visual tracking using a GM-PHD filter with CNN-based re-identification. Journal of Visual Communication and Image Representation. 2021; 80:103279.

[14] Wang W, Shen J, Lu X, Hoi SC, Ling H. Paying attention to video object pattern understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020; 43(7):2413-28.

[15] Hou J, Li B. Swimming target detection and tracking technology in video image processing. Microprocessors and Microsystems. 2021; 80:103535.

[16] Razzok M, Badri A, El MI, Ruichek Y, Sahel A. Pedestrian detection and tracking system based on deep-SORT, YOLOv5, and new data association metrics. Information. 2023; 14(4):1-16.

[17] Feng W, Bai L, Yao Y, Yu F, Ouyang W. Towards frame rate agnostic multi-object tracking. International Journal of Computer Vision. 2023:1-20.

[18] Wang S, Li WX, Wang L, Xu LS, Deng QX. VGT-MOT: visibility-guided tracking for online multiple-object tracking. Machine Vision and Applications. 2023; 34(4):1-6.

[19] Wang G, Wang Y, Gu R, Hu W, Hwang JN. Split and connect: a universal tracklet booster for multi-object tracking. IEEE Transactions on Multimedia. 2022; 25:1256-68.

[20] Zhou Y, Chen J, Wang D, Zhu X. Multi-object tracking using context-sensitive enhancement via feature fusion. Multimedia Tools and Applications. 2023:1-20.

[21] Boragule A, Jang H, Ha N, Jeon M. Pixel-guided association for multi-object tracking. Sensors. 2022; 22(22):1-14.

[22] Elhoseny M. Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. Circuits, Systems, and Signal Processing. 2020; 39:611-30.

[23] Jha S, Seo C, Yang E, Joshi GP. Real time object detection and trackingsystem for video surveillance system. Multimedia Tools and Applications. 2021; 80:3981-96.

[24] Liu D, Cui Y, Chen Y, Zhang J, Fan B. Video object detection for autonomous driving: motion-aid feature calibration. Neurocomputing. 2020; 409:1-11.

[25] Yu H, Huang Y, Pi L, Zhang C, Li X, Wang L. End-to-end video text detection with online tracking. Pattern Recognition. 2021; 113:107791.

[26] Sun S, Akhtar N, Song H, Mian A, Shah M. Deep affinity network for multiple object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019; 43(1):104-19.

[27] Ji Y, Zhang H, Jie Z, Ma L, Wu QJ. CASNet: a cross-attention siamese network for video salient object detection. IEEE Transactions on Neural Networks and Learning Systems. 2020; 32(6):2676-90.

[28] Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, et al. Bytetrack: multi-object tracking by associating every detection box. In European conference on computer vision 2022 (pp. 1-21). Cham: Springer Nature Switzerland.

[29] Wang Y, Kitani K, Weng X. Joint object detection and multi-object tracking with graph neural networks. In international conference on robotics and automation 2021 (pp. 13708-15). IEEE.

[30] Alagarsamy R, Muneeswaran D. Multi-object detection and tracking using reptile search optimization algorithm with deep learning. Symmetry. 2023; 15(6):1-17.
[31] Feng X, Wu HM, Yin YH, Lan LB. CGTracker: center graph network for one-stage multi-pedestrian-object detection and tracking. Journal of Computer Science and Technology. 2022; 37(3):626-40.
[32] Qureshi SA, Hussain L, Chaudhary QU, Abbas SR, Khan RJ, Ali A, et al. Kalman filtering and bipartite matching based super-chained tracker model for online multi object tracking in video sequences. Applied Sciences. 2022; 12(19):1-19.
[33] Wu Y, Sheng H, Zhang Y, Wang S, Xiong Z, Ke W. Hybrid motion model for multiple object tracking in mobile devices. IEEE Internet of Things Journal. 2022; 10(6):4735-48.
[34] Singh D, Srivastava R. An end to end trained hybrid CNN model for multi-object tracking. Multimedia Tools and Applications. 2022; 81(29):42209-21.
[35] Xuan S, Li S, Zhao Z, Zhou Z, Zhang W, Tan H, et al. Rotation adaptive correlation filter for moving object tracking in satellite videos. Neurocomputing. 2021; 438:94-106.
[36] Suljagic H, Bayraktar E, Celebi N. Similarity based person re-identification for multi-object tracking using deep Siamese network. Neural Computing and Applications. 2022; 34(20):18171-82.
[37] Ma L, Zhong Q, Zhang Y, Xie D, Pu S. Associative affinity network learning for multi-object tracking. Frontiers of Information Technology & Electronic Engineering. 2021; 22(9):1194-206.
[38] Xu Y, Ban Y, Delorme G, Gan C, Rus D, Alameda-pineda X. TransCenter: transformers with dense representations for multiple-object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022; 45(6):7820-35.
[39] Wu H, Nie J, Zhu Z, He Z, Gao M. Learning task-specific discriminative representations for multiple object tracking. Neural Computing and Applications. 2023; 35(10):7761-77.
[40] Hu X, Jeon Y. FFTransMOT: feature-fused transformer for enhanced multi-object tracking. IEEE Access. 2023; 11:130060-71.
[41] Chen S, Hu X, Jiang W, Zhou W, Ding X. Novel learning framework for optimal multi-object video trajectory tracking. Virtual Reality & Intelligent Hardware. 2023; 5(5):422-38.
[42] Xiang X, Ren W, Qiu Y, Zhang K, Lv N. Multi-object tracking method based on efficient channel attention and switchable atrous convolution. Neural Processing Letters. 2021; 53(4):2747-63.
[43] Lee J, Jeong M, Ko BC. Graph convolution neural network-based data association for online multi-object tracking. IEEE Access. 2021; 9:114535-46.
[44] Li Y, Wu L, Chen Y, Wang X, Yin G, Wang Z. Motion estimation and multi-stage association for tracking-by-detection. Complex & Intelligent Systems. 2023:1-4.
[45] Chen M, Banitaan S, Maleki M. Enhancing pedestrian group detection and tracking through zone-based clustering. IEEE Access. 2023; 11:132162-79.
[46] Liang H, Wu T, Zhang Q, Zhou H. Non-maximum suppression performs later in multi-object tracking. Applied Sciences. 2022; 12(7):1-11.
[47] Li Y, Liu Y, Zhou C, Xu D, Tao W. A lightweight scheme of deep appearance extraction for robust online multi-object tracking. The Visual Computer. 2023:1-7.
[48] https://motchallenge.net/data/MOT17/. Accessed 15 December 2023.
[49] Walia IS, Kumar D, Sharma K, Hemanth JD, Popescu DE. An integrated approach for monitoring social distancing and face mask detection using stacked Resnet-50 and YOLOv5. Electronics. 2021; 10(23):1-15.
[50] Li B, Lima D. Facial expression recognition via ResNet-50. International Journal of Cognitive Computing in Engineering. 2021; 2:57-64.

Nandeeswar Sampigehalli Basavaraju completed his Bachelor of Engineering in Computer Science and Engineering (CSE) in 2002 from Bangalore University, his Master of Technology in CSE in 2007 from Visvesvaraya Technological University, and his PhD in CSE in 2022 from Presidency University, Bengaluru, India. With over 22 years of rich experience in both industry and academia, he is continuously engaged in research within the areas of Artificial Intelligence & Machine Learning, Data Science & Analytics, and Computational Theory.
Email: nandeeswarsb@gmail.com

Pallavi Hallappanavar Basavaraja completed her Bachelor of Engineering in Computer Science and Engineering (CSE) in 2012 from Visvesvaraya Technological University, her Master of Technology in CSE in 2014 from the same university, and her PhD in CSE in 2022 from Presidency University, Bengaluru, India. With over 10 years of experience in teaching, she is continuously engaged in research in the areas of Artificial Intelligence & Machine Learning and Computer Networks.
Email: pallavihb.7@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	AAN	Associative Affinity Network
2	ASPP	Atrous Spatial Pyramid Pooling
3	BE-Conv	Back-End Convolution Layer
4	CASNet	Cross-Attention Encoder-Decoder Model Under The Scheme Of Siamese
5	CFA	Cross-Frame Affinities
6	CGTracker	Center Graph Network
7	CNN	Convolutional Neural Network
8	DAN	Deep Affinity Network

9	DL	Deep Learning
10	DPM	Deformable Part Models
11	DTN	Dual-Path Transformation Network
12	Faster-RCNN	Faster-Region-Based Convolutional Neural Network
13	Fast-OSNet	Fast Omni-Scale Network
14	FE-Conv	Front-End Convolution Layer
15	FFTransMoT	Feature-Fused Transformer for Improved Multiple-Object Tracking
16	FLOP	Floating-Point Operation
17	FN	False Negative
18	FP	False Positive
19	GCN	Graph Convolutional Network
20	GNN	Graph Neural Network
21	GPU	Graphics Processing Unit
22	GSDT	Graph Neural Network For Simultaneous Detection And Tracking
23	HAEMA	Hierarchical Adaptive Exponential Moving Average
24	HOG	Histogram of Gradients
25	HSV	Hue Saturation Value
26	IDF1	F1 Score Of Correctly Identified Detection
27	IDP	Identity Detection Prediction
28	IDR	Identity Detection Recall
29	ID-Sw	Identity Switches
30	IOU	Intersection Over Union
31	MA	Multi-Stage Association
32	MAE	Mean Absolute Error
33	MaxF	Maximum F-Measure
34	MEMA	Motion Estimation And Multi-Stage Association
35	MFCN	Motion-Aided Feature Calibration Network
36	MOT	Multiple-Object Tracking
37	MOTA	Multiple-Object Tracking Accuracy
38	MOTAL	Multiple-Object Tracking Accuracy With Logarithmic Decay
39	MODT	Multiple-Object Detection and Tracking
40	MOTP	Multiple -Object Tracking Precision
41	MOSSE	Minimum Output Sum Of Squared Error
42	MVS	Machine Vision Systems
43	NMS	Non-Maximum Suppression
44	N-YOLO	n-You Only Look Once
45	PA-RetinaNet	Path-Augmented Retinanet
46	PHOC	Pyramidal Histogram of Characters
47	PNN	Pyramid Non-Local Network
48	QLN	Query Learning Networks
49	QRNN	Quasi-Recurrent Neural Network
50	RACF	Rotation-Adaptive Correlation Filter
51	RGB	Red Green Blue
52	DAN	Deep Affinity Network
53	Re-ID	Re-Identification
54	ReLU	Rectified Linear Units
55	RPN	Region Proposal Network
56	RSOA	Reptile Search Optimization Algorithm
57	RSOADL-MODT	Reptile Search Optimization Algorithm With Deep Learning-Based Multiple Object Detection and Tracking
58	SAT	Similarity-Based Person Re-ID Framework
59	SCT	Super Chained Tracker
60	SDP	Scale-Dependent Pooling
61	SGD	Stochastic Gradient Descent
62	SNN	Siamese Neural Network
63	SOT	Single-Object Tracking
64	SSD	Single Shot Detector
65	TP	True Positive
66	TN	True Negative
67	UAV	Unmanned Aerial Vehicles
68	UA DETRAC	University At Albany Detection And Tracking
69	UVOS	Unsupervised Video Object Segmentation

Appendix II

Symbol	Description
N_m	Maximum objects number present in a frame
C_t	Object center
$L_{t-n,t}$	A matrix of binary data association is used to represent the relationship between the objects found in frames I_{t-n} and I_t
I_{t-n}	Video frame
f_t	Feature matrix linked with the t^{th} frame
$\varphi_{t-n,t}$	Permutation tensor
$t-n,t$	The entity is computed for the pair of frames I_{t-n} and I_t
1	Vector of ones
γ	DAN's hyperparameter
L_1, L_2	$L_{t-n,t}$ trimmed versions that were created by removing the last row and column
A_1, A_2	Represents the matrix A_1, A_2 reduced to size N_m x N_m
$\odot$	Hadamard product
$\sum_{coeff}(.)$	adds up all the matrix's coefficients to produce a value of the scalar
$Operations\ of\ Max\ and\ log$	perform element-wise
I_t	t^{th} video frame under the index of 0-based
$t-n{:}\,t$	Time period between $t-n$ to t
$A_{t-n,t}$	Matrix of Affinity which represents resemblance among the bounding boxes in frames $(t-n)^{th}$ frame and t^{th} frame
B	Size of batch during training