Multi-Object Detection and Tracking Using Reptile Search Optimization Algorithm with Deep Learning

Ramachandran Alagarsamy 1,* and Dhamodaran Muneeswaran 2

1 Department of Electronics and Communication Engineering, SSM Institute of Engineering and Technology, Dindigul 624002, India
2 Department of Electronics and Communication Engineering, M.Kumarasamy College of Engineering, Karur 639113, India
* Correspondence: ramachandran061185@gmail.com

Abstract: Multiple-Object Tracking (MOT) has become more popular because of its commercial and academic potential. Though various techniques were devised for managing this issue, it becomes a challenge because of factors such as severe object occlusions and abrupt appearance changes. Tracking presents the optimal outcomes whenever the object moves uniformly without occlusion and in the same direction. However, this is generally not a real scenario, particularly in complicated scenes such as dance events or sporting where a greater number of players are tracked, moving quickly, varying their speed and direction, along with distance and position from the camera and activity they are executing. In dynamic scenes, MOT remains the main difficulty due to the symmetrical shape, structure, and size of the objects. Therefore, this study develops a new reptile search optimization algorithm with deep learning-based multiple object detection and tracking (RSOADL–MODT) techniques. The presented RSOADL–MODT model intends to recognize and track the objects that exist with position estimation, tracking, and action recognition. It follows a series of processes, namely object detection, object classification, and object tracking. At the initial stage, the presented RSOADL–MODT technique applies a path-augmented RetinaNet-based (PA–RetinaNet) object detection module, which improves the feature extraction process. To improvise the network potentiality of the PA–RetinaNet method, the RSOA is utilized as a hyperparameter optimizer. Finally, the quasi-recurrent neural network (QRNN) classifier is exploited for classification procedures. A wide-ranging experimental validation process takes place on DanceTrack and MOT17 datasets for examining the effectual object detection outcomes of the RSOADL–MODT algorithm. The simulation values confirmed the enhancements of the RSOADL–MODT method over other DL approaches.

Keywords: multi-object tracking; deep learning; reptile search optimization algorithm

1. Introduction

Nowadays, object detection and tracking has not only grabbed the interest of many because of its recent breakthrough research and wide range of applications but also its equal significance in academia and real-time applications [1], including monitoring security, transportation surveillance, autonomous driving, and robotic vision. Various sensing modalities such as computer vision (CV) radar and Light Detection and Ranging (LIDAR) are available for object tracking and detection. Unlike tracking a single specific object, multiple-object tracking (MOT) can be very complex [2]. In addition to the problems in tracking single objects, MOT in a single category should make new tracked objects utilizing identification outcomes, terminate objects whenever they go out of the field of camera view, or recognize lost objects when they appear again [3]. On top of that, pose changes; the problems of occlusion and background clutter are more complicated than those in tracking single objects. To manage such difficulties, certain DL-related techniques were devised. For instance, it is possible to replace traditional features with features derived from deep
neural networks for associating detection outcomes, though the features were learned from tasks of recognition or classification [4]. In addition, it has been proven that the performance becomes enhanced if the attributes of MOT, such as temporal order or spatial attention maps, are explored. As well, certain end-to-end DL structures were devised for feature extraction not merely for appearance descriptors but for motion data. Although DL techniques are potentially applicable to MOT issues [5], there is much room to enhance the tracking efficiency using the power of DL because of its great achievements in the domain of image recognition and classification [6].

In DL, the problem of object detection refers to the task of labeling various objects in an image frame with their accurate classes and estimating the bounding box with a higher degree of probability. The learning accuracy in DL rests on previous experiences or the number of samples [7]. Higher accuracy in the performance can be achieved with a greater number of samples. These days’ data are abundantly available, which makes DL the right choice. Unlike conventional (shallow) learning, thousands of images are needed by DL to gain optimal outcomes [8]. The term “shallow” is contrary to “deep”. Hence, DL can be computationally intensive and problematic to engineers. It requires high-performing GPUs to offer fast motion and object detection [9]. DL methods are availed in domain-specific and generic object tracking and detection. Deep CNN was employed as a backbone in the detection network for the extraction of crucial features from input images or video frames. Such attributes were utilized for classifying and localizing objects in similar frames [10].

Wang et al. [11] recommend a fast and robust camera–LiDAR fusion-based MOT technique that accomplishes better tradeoffs between speed and accuracy. Based on the features of LiDAR sensors and cameras, a deep association mechanism was devised and embedded in the presented MOT model. The proposed method realized MOT in a 2D domain if the object is distant and detected only by the camera, and the updating of 2D trajectory includes 3D data that can be attained if the object appeared in the LiDAR field of view to accomplish a smooth fusion of 2D and 3D trajectories. Wang et al. [12] introduced an MOT technique based on GNN. The major concept is that GNN relationships between different size objects in spatial and temporal domains are crucial for learning discriminatory features for data association and detection.

The author in [13] developed an object detection technique by revamping YOLOv3, and real-time MOT can be performed by using Deep SORT for tracking the target through these data association and movement representation models. It is a tracking-by-detection technique. Guo and Zhao [14] proposed a new architecture for online 3D MOT to remove the impact of unknown biases and the inherent uncertainty in point clouds. A constant turn rate and velocity (CTRV) motion were used for estimating future motion states that can be smoothed using the cubature Kalman filter (CKF) model. An adaptive cubature Kalman filter (ACKF) is presented to update the tracked state powerfully and to remove the effect of unknown bias.

Rafique et al. [15] developed a Maximum Entropy Scaled Super-Pixels (MEsSP) Segmentation technique that encapsulates super-pixel segmentation depending on Entropy Model and exploits local energy terms for labeling the pixel. After pre-processing and acquisition, the image is initially segmented using two various approaches: MEsSP and Fuzzy C-Means (FCM). Lastly, based on the categorized objects, a DBN allocates the related label to the scene, intersection over dice similarity coefficient, and union scores. Lusardi et al. [16] developed a GNN-based architecture for MOT that integrates association and detection with the usage of novel re-detection features. The combination of multiple appearance features within the architecture is explored to improve tracking accuracy and obtain the best representation. Data augmentation with random noise and random erase are used for improving the robustness during tracking. Jiang et al. [17] proposed the Residual Neural Network (RNN) for target tracking and achieved high accuracy when compared with Multi-Domain (MDnet) in three complex problems such as deforms or rotates, similar target interference, and complex scenes. Wang et al. [18] proposed a solution for motion blur using the Motion Enhance Attention (MEA) module to detect far-away objects from
the camera for target tracking using a Dual Correlation Attention (DCA) module, but the tracking performance is not better due to the above-combined module. CenterTrack [19] calculates the object displacements with inputs to associate the objects. FairMOT [20] finds each object in a particular detection dataset and find out to discriminate between those to learn tracking from stationary images. TraDes [21] and GTR [22] is a new method proposed to attempt multiple-object tracking. MOTR [23] and QDTrack [24] carried out the multi-object tracking with transformer and Quasi-Tense similarity learning, but both fall in classification errors such as large-scale datasets. Multi-object detection and tracking are categorized into neural networks, fuzzy logic, and meta-heuristic algorithms based on the approach in soft computing studies [25]. To solve the optimization problems, meta-heuristic algorithms are proposed that seek to exit local optimal points. Recently, the exploit of meta-heuristic optimization algorithms has occupied many research studies to detect and track objects because of their facility to estimate the location of objects accurately [25]. In contrast to the traditional optimization method [26–28] that performs an iterative search using a set of elements in video frames, this approach carries the efficient evolutionary computing method into object tracking. The elements are updated using a fitness function and make some iterations in the object location.

In addition, an object tracker can be divided into three core works, namely feature representation, observation, and motion features [29]. It generates the feature representation of the object, which is then used for comparison under the observation. Based on the association results, the motion feature makes a prediction of the object’s location. This, without uncertainty, can bring better tracking performance due to more effective searches in the search space.

Nonetheless, the need for improvements in other works of object tracking is mainly unseen in the literature. A possible solution to the weak feature problem is the adoption of deep learning-based feature representation, which could potentially improve the accuracy and robustness of the tracking.

This study develops a new reptile search optimization algorithm with deep learning-based multi-object detection and tracking (RSOADL–MODT) technique. The presented RSOADL–MODT model applies path augmented RetinaNet (PA–RetinaNet) based object detection module, which improves the feature extraction process. To improvise the network potentiality of the PA–RetinaNet method, the RSOA is utilized as a hyperparameter optimizer. Finally, the quasi-recurrent neural network (QRNN) classifier is exploited for classification procedures. A wide-ranging experimental validation process takes place on DanceTrack and MOT datasets for examining the effectual object detection outcomes of the RSOADL–MODT approach.

2. Materials and Methods

In this study, an automated MOT using the RSOADL–MODT technique has been developed for the recognition and tracking of the objects that exist with position estimation, tracking, and action recognition. It follows a series of processes, namely PA–RetinaNet-based object detection, RSOA-based hyperparameter tuning, QRNN object classification, and object tracking. Figure 1 represents the workflow of the RSOADL–MODT approach.

2.1. Object Detection Using PA–RetinaNet

In this work, the presented RSOADL–MODT technique exploited the PA–RetinaNet-based object detection module. The PA–RetinaNet incorporates two parallel sub-networks, a backbone network, and a path augmentation model with certain tasks [30]. The two sub-networks are used for bounding box regression and object classification. The backbone network adopts the Feature Pyramid Network (FPN), a new convolution network, to evaluate the feature maps of input images. A path augmentation model can be improved to make lower layer data easier for transmission correspondingly. The FPN can be adopted as a backbone network of PA–RetinaNet. FPN increases a typical convolution network with lateral and top-down pathway connections, which allows the effective creation of a
multiscale, rich FPN from the single-resolution input images. Every level of the pyramid detects an object at various scales.

**Figure 1.** Workflow of RSOADL–MODT approach—Detection, Tracking and Classification.

The proposal of the PA–RetinaNet detector improves and inherits the architecture of RetinaNet. With the addition of the bottom-up path makes the lowest layer data easier to transmit, reduces the data transmission path, and produces a good result. The lower-level feature map mostly perceives the local information, edge, and corner of an image and additional information, whereas the higher-level feature map reflects semantical data of the entire objects. A bottom-up path can be improved to make lower-layer data easier to
transmit. Thus, the addition of a bottom-up path is essential to increase the reasonable classification ability of each feature and semantically propagate strong features in FPN.

Based on the description in FPN, layers of a similar network phase produce a feature map of similar spatial size, and every feature level corresponds to a single phase. ResNet as a fundamental framework and characterize feature level of the FPN as \( \{P_2; P_3; P_4; P_5\} \). The space size can be downsampled gradually to 1/4 of the prior layer size. \( \{N_2; N_3; N_4; N_5\} \) is recently produced feature maps respective to \( \{P_2; P_3; P_4; P_5\} \). It should be noted that \( N_2 \) is \( P_2 \). Now, this signifies the lateral connections of the high-layer fine feature map \( N_i \) and low-layer coarse feature map \( P_i + 1 \), which produce a novel feature map \( N_i + 1 \). First, every feature map \( N_i \) can be down-sampled using 3 \( \times \) 3 convolution layers with the stride of 2, providing a similar resolution as \( P_i + 1 \) and added to it through a lateral connection. Next, the combined feature map produces \( N_i + 1 \) via another 3 \( \times \) 3 convolution layers as input to the following two subnetworks. This is an iteration procedure until \( N_5 \) is produced. In these modules, 256 channel feature map is used, and a ReLu operation can be implemented afterward in every convolution layer. The newest feature map, \( \{N_2; N_3; N_4; N_5\} \), is later pooled to attain the feature grid. The two subnets are the box regression subnet and the classification subnet.

The proposal of the box regression subnet can be the same as the classification subnet, but it dismisses in 4A linear output for every spatial position. The input feature mapping of 256 channels can be attained from the presented layer in the classification subnet. First, the subnet uses four 3 \( \times \) 3 convolution layers. Every convolution layer contains 256 filters, the ReLu activation, 3 \( \times \) 3 convolution layer with KA filters, and sigmoid activation output KA binary prediction for every spatial position. The classification subnet forecasts the probability that every K object class and A anchor will exist at every spatial location, and if one exists, then the box regression subnet regresses the offset from every anchor box to the nearer ground-truth object.

2.2. RSOA-Based Hyperparameter Tuning

For enhancing the network potentiality of the PA–RetinaNet method, the RSOA is utilized as a hyperparameter optimizer. RSOA is a metaheuristic algorithm based on the natural hunting strategies of crocodiles [31]. The functioning of RSOA is based on two stages: the encircling and the hunting phases. The RSOA switch between the hunting search and the encircling phases and the shift between dissimilar stages are implemented by splitting the number of iterations into four different parts. RSOA begins by stochastically producing a candidate solution based on the subsequent formula:

\[
X_{jk} = \text{rand} \times (U_b - L_b) + L_bk = 1, 2, \ldots, n
\]

In Equation (1), \( X_{jk} \) = initialization matrix, \( j = 1, 2, \ldots, P \). \( P \) signifies population size (rows of initialization matrix), \( \text{rand} \) randomly generated values, \( n \) signifies dimensions (columns of initialization matrix) of the optimization problems, and \( L_b \) and \( U_b \), represents the lower and upper bounds.

The encircling stage is an exploration of a higher-density area. During the encircling stage, belly and high walking are stimulated by the crocodile movement, which plays a crucial role. This movement does not assist in catching the prey however assists in discovering a search space.

\[
X_{jk}(t+1) = \text{Best}_k(t) \times \left( -\mu_{(jk)}(\tau) \right) \times \beta - \left( R_{(jk)}(\tau) \times \text{rand} \right), \tau \leq \frac{T}{4}
\]

\[
x_{jk}(\tau + 1) = \text{Best}_k(\tau) \times X_{(r1,k)} \times ES(\tau) \times \text{rand}, \tau \leq \frac{2T}{4} \text{ and } \tau > \frac{T}{4}
\]

From the expression, \( \text{Best}_k(\tau) \) denotes the optimum solution attained at \( k \)th location, \( \text{rand} \) characterizes a randomly generated value, \( \tau \) displays the existing iteration count, and the maximal amount of iterations is characterized by \( T \). \( \mu_{(jk)} \) indicates the value of
the hunting operator of \( j \)th solution at \( k \)th location. The value of \( \mu(j,k) \) can be defined by Equation (4):

\[
\mu(j,k) = \text{Best}_k(\tau) \times P(j,k)
\]

where \( \beta \) denotes the sensitivity parameter and describes the exploration results. Additional function termed \( R(j,k) \), whose objective is to decrease the search region, is evaluated by Equation (5):

\[
R(j,k) = \frac{\text{Best}_k(\tau) - P(r_2,k)}{\text{Best}_k(\tau) + e}
\]

where \( r_1 \) denotes the randomly generated integer within \([1, N]\). Now, \( N \) signifies the overall quantity of solution candidates. \( z_{(r_1,l)} \) characterizes a random location for the \( k \)th solution. \( r_2 \) denotes a random (arbitrary) integer ranges amongst \([1, N]\), \( e \) signifies a value of smaller magnitude. Modification is made to incorporate RSOA into the framework of object tracking. Prior to the initiation of tracking, a representation of the target object is extracted using the feature representation from the first frame and is stored as a reference object tracking. Prior to the initiation of tracking, a representation of the target object is extracted and compared to the object using an exact observation model called a fitness function in RSOA Equation (11).

In Equation (6), \( r_3 \) denotes a randomly generated value. \( P(j,k) \) is evaluated by Equation (7):

\[
P(j,k) = \alpha + \frac{\chi(j,k) - M(\chi_j)}{\text{Best}_k(\tau) x (U_{b(k)} - (L_{b(k)}) + e}
\]

where \( \alpha \) denotes the sensitivity boundary that controls the exploration accuracy, \( M(\chi_j) \) indicates the average location of the \( j \)th solution and is evaluated by Equation (8):

\[
M(\chi_j) = \frac{1}{n} \sum_{k=1}^{n} \chi(j,k)
\]

In Equation (6), \( r_3 \) denotes a randomly generated value. \( P(j,k) \) is evaluated by Equation (7):

The hunting stage, similar to the encircling stage, has two approaches, such as hunting cooperation and coordination. These two approaches are used for locally traversing the searching space and assisting in targeting the prey (search for an optimal solution). In addition, based on the iteration, the hunting stage can be divided into two parts. The hunting coordination approach can be performed for iteration ranges from \( \tau \leq 3T^4 \) to \( > 2T^4 \), whereas the hunting cooperation can be performed from \( \tau \leq T \) to \( > 3T^4 \). A stochastic coefficient is used for traversing the local search space to produce optimum solutions. Equations (9) and (10) are utilized for the exploitation stage:

\[
x(j,k)(\tau + 1) = \text{Best}_k(\tau) \times \left( P(j,k)(\tau) \right) \times \text{rand}, \tau \leq 3T^4 \text{ and } \tau > 2T^4
\]

\[
x(j,k)(\tau + 1) = \text{Best}_k(\tau) - \mu(j,k)(\tau) \times e - R(j,k)(\tau) \times \text{rand}, \tau \leq T \text{ and } \tau > 3T^4
\]

where \( \text{Best}_k(\tau) \) denotes the \( k \) location in the optimum solution obtained in the existing iteration. Similarly, \( \mu(j,k) \) signifies the hunting operator that is evaluated using Equation (4). Figure 2 depicts the flowchart of RSA. During the update process, a feature representation is extracted and compared to the object using an exact observation model called a fitness function in RSOA Equation (11).
2.3. Classification Using QRNN Model

For object classification purposes, the QRNN model is used in this work (Algorithm 1). The classification problem is hard to extend to large-scale datasets. The QRNN is proposed to decrease the computation efforts of the recurrent step in LSTM [32]. QRNN is a hybrid neural network of CNNs and LSTMs, which combines the benefits of both. In contrast with

![Flowchart of RSOA](image)

**Figure 2.** Flowchart of RSOA.

The RSOA method has derived a fitness function from having enhanced outcomes. It has determined a positive value for signifying the better outcome of the candidate solutions. Here, the reduced classifier error rate is treated as the fitness function, as presented in Equation (11).

$$fitness(x_i) = \frac{\text{number of misclassified samples}}{\text{total number of samples}} \times 100$$  \hspace{1cm} (11)

2.3. Classification Using QRNN Model

For object classification purposes, the QRNN model is used in this work (Algorithm 1). The classification problem is hard to extend to large-scale datasets. The QRNN is proposed to decrease the computation efforts of the recurrent step in LSTM [32]. QRNN is a hybrid neural network of CNNs and LSTMs, which combines the benefits of both. In contrast with
LSTM, QRNN is extremely parallelizable to CNN. Blue represents a parameterless function running parallel alongside the channel or feature dimension. Brown represents convolution or matrix multiplication. Contiguous blocks imply that computation is implemented simultaneously. The LSTM is decomposed into a blue element and brown linear blocks, whereas computation at every time step depends on the prior outcomes.

Every layer of QRNN integrates two seed components, such as pooling and convolution layers in CNN. Both layers permit fully parallel computation: the pooling layer supports parallelization across feature dimensions and mini-batch; the convolutional layer supports spatial dimensions (viz., sequence dimension) and parallelization across minibatches.

The equation of the QRNN unit has been demonstrated below:

$$\begin{align*}
\hat{x}_t &= \tanh(W_s X_t) \\
f_t &= \sigma(W_f X_t) \\
0_t &= \sigma(W_o X_t) \\
c_t &= f_t c_{t-1} + (1 - f_t) \odot \hat{x}_t \\
h_t &= 0_t \odot \tanh(c_t)
\end{align*}
$$

From the expression, $$x_{t-K+1} \ldots x_t$$ where $$X_t \in \mathbb{R}^{k \times n}$$ denotes the input series of $$kn$$-dimensional vectors, “$$*$$” signifies the mask convolution and the timestep dimension, $$W, W_f, W_o$$ indicates the convolutional filter bank in $$\mathbb{R}^{d \times n \times k}$$, and $$k$$ represents the width of the filter. The first three expressions are the convolutional part of QRNN and generate m-dimensional sequence $$\hat{x}, f, c$$, and $$0_t$$. The symbol signifies component-wise multiplication. Especially, QRNN exploits a forget gate that is “dynamic average pooling”.

A single QRNN implements three multiply-vector operations, as indicated in Equation (2) that depend on the input series $$X$$ without dependence on preceding outputs, such as $$h_{t-1}$$. With known input, this multiply-vector operation $$W_s X_t$$ is predefined in various timesteps. Consequently, weight matrices with an enormous quantity of memory no longer have to be loaded at every timestep. In the presented method, the cost of DRAM is decreased once the timestep needed to conduct grows as follows:

$$u^T = \begin{pmatrix} W & W_f \\ W_o & \end{pmatrix} [X_{t-k+1} \ldots X_{t+k-1}]$$

where $$U \in \mathbb{R}^{L \times 3d}$$ denotes the combined result matrix; $$d$$ indicates the number of hidden layer neurons; $$L = T - k + 1$$ signifies the input sequence length. $$U \in \mathbb{R}^{L \times B \times 3d}$$ becomes a tensor while considering minibatch $$B$$.

**Algorithm 1: QRNN Model**

1. Input: merged matrix $$U[k, j, t]$$ initial hidden state $$c_0[i, j]$$, bias vector $$b_f[i]$$, $$b_r[i]$$
2. Output: hidden state tensor $$c[t', j]$$ and output tensor $$h[t', j]$$
3. Parameter settings: number of hidden layer neurons, $$B$$ mini batch, L output sequence length
4. for = 1, B, i = 1, d do
5. for $$k = 1 \ldots L$$ do
6. $$\hat{c} = \tanh(U[k, j, i])$$
7. $$f = \sigma(U[k, j, i + d] + b_f[i])$$
8. $$r = \sigma(U[k, j, i + d \times 2] + b_r[i])$$
9. $$c = f \times c + (1 - f) \times \hat{c}$$
10. $$h = r \times \tanh(c)$$
11. $$c[k, j, i] = c$$
12. $$h[k, j, i] = h$$
13. return $$h[t', j]$$ and $$c[t', j]$$
3. Performance Validation

This paper presents the details of an experimental dataset and evaluation metrics. It demonstrates the effectiveness of the proposed method by comparing its performance to state-of-the-art methods on benchmark datasets. Next, an ablation study was conducted to investigate how our PA–Retinanet, reptile search optimization, and classification method were used to measure the performance of the approach.

DanceTrack [33] and MOT17 [34] datasets are used to evaluate the proposed method. DanceTrack [33] is a large-scale dataset for multi-object tracking in complex scenes such as uniform appearance and diverse motion, and it has 40 videos for training, 25 videos for validation, and 35 videos for testing. MOT17 [34] is a widely used dataset containing seven sequences for training and seven for testing, and it consists of crowded street scenes with linear object motion.

In order to evaluate the proposed method, the following tracking metrics are utilized. MOTA—The Multiple Object Tracking Accuracy (MOTA) [35] metric computes tracking accuracy with detection accuracy. MOTA weighs detection performance more greatly than association performance. MOTA is computed using the FP (False Positive), FN (False Negative), ID switches (IDs), and Mostly Tracked (MT). IDF1—The Identification F1 Score (IDF1) [36] matches ground truth and calculations on the trajectory level and computes a corresponding F1 score. IDF1 focuses on measuring association performance. HOTA—Higher Order Tracking Accuracy (HOTA) [37] aims to combine the evaluation of detection and association relatively. Figure 3 shows the sample images.

![Sample Images](image_url)
Experiments are conducted on PyTorch with the runtime machine in eight NVIDIA 2080 Ti GPUs. It utilizes the variant of ResNet as the backbone network for fast convergence and object detection. The training model is trained with RSOA for the tuning of hyperparameters, and it boosts the accuracy. The initial learning rate is 0.0001 and produces the length of eight video clips and then trains with a batch size of eight videos on eight GPUs, ensuing in an effective batch size of 64.

In DanceTrack, 70 epochs on the training sets and the learning rate decreases by a factor of 10 at the 10th epoch. From the initial length of the video clip, it gradually increases the clips to 3, 4, and 5 at the 30th, 40th, and 50th epochs, respectively. The gradual increment of video clip length improves training efficiency and constancy. The training step takes about 8 h on two 2080 Ti GPUs.

In Table 1 and Figure 4, the overall tracking performance of the RSOADL–MODT method is examined in terms of evaluation metrics on the DanceTrack dataset. The results indicate that the RSOADL–MODT technique accomplishes improved performance under each iteration. It is noticed that the RSOADL–MODT technique gains an average MOTA of 87.67, MOTP of 0.325, IDF1 of 51.21%, Idsw of 4272, recall of 98.52, precision of 97.73, and MT of 537.

Table 1. Overall tracking result of RSOADL–MODT approach on the DanceTrack dataset.

<table>
<thead>
<tr>
<th>DanceTrack Dataset</th>
<th>No. of Iterations</th>
<th>MOTA</th>
<th>MOTP</th>
<th>IDF1</th>
<th>Idsw</th>
<th>Recall</th>
<th>Precision</th>
<th>MT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration-1</td>
<td>87.32</td>
<td>0.340</td>
<td>51.48</td>
<td>3953</td>
<td>98.70</td>
<td>97.45</td>
<td>537</td>
<td></td>
</tr>
<tr>
<td>Iteration-2</td>
<td>88.27</td>
<td>0.341</td>
<td>50.27</td>
<td>4461</td>
<td>98.86</td>
<td>97.62</td>
<td>539</td>
<td></td>
</tr>
<tr>
<td>Iteration-3</td>
<td>87.12</td>
<td>0.317</td>
<td>51.04</td>
<td>4590</td>
<td>98.70</td>
<td>97.87</td>
<td>537</td>
<td></td>
</tr>
<tr>
<td>Iteration-4</td>
<td>87.16</td>
<td>0.301</td>
<td>50.85</td>
<td>4050</td>
<td>98.20</td>
<td>97.73</td>
<td>538</td>
<td></td>
</tr>
<tr>
<td>Iteration-5</td>
<td>88.52</td>
<td>0.324</td>
<td>52.42</td>
<td>4304</td>
<td>98.14</td>
<td>97.96</td>
<td>535</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>87.67</td>
<td>0.325</td>
<td>51.21</td>
<td>4272</td>
<td>98.52</td>
<td>97.73</td>
<td>537</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Average result of RSOADL–MODT approach on the DanceTrack dataset.

The TACY and VACY of the RSOADL–MODT approach on the DanceTrack dataset is defined in Figure 5. The Figure implied that the RSOADL–MODT technique had improved performance with increased values of TACY. Visibly, the RSOADL–MODT method has reached maximal TACY outcomes.
Figure 4. Average result of RSOADL–MODT approach on the DanceTrack dataset.

The TACY and VACY of the RSOADL–MODT approach on the DanceTrack dataset is defined in Figure 5. The Figure implied that the RSOADL–MODT technique had improved performance with increased values of TACY. Visibly, the RSOADL–MODT method has reached maximal TACY outcomes.

Figure 5. TACY and VACY outcome of RSOADL–MODT approach on the DanceTrack dataset.

The TLOS and VLOS of the RSOADL–MODT method on the DanceTrack dataset is defined in Figure 6. The Figure inferred that the RSOADL–MODT method has better performance with minimal values of TLOS and VLOS. Particularly, the RSOADL–MODT approach has minimal VLOS outcomes.

Figure 6. TLOS and VLOS outcome of RSOADL–MODT approach on the DanceTrack dataset.

A comparison study is made on the DanceTrack dataset in Table 2 to highlight the enhanced performance of the RSOADL–MODT technique [31].

Figure 7 shows the comparative assessment of the RSOADL–MODT technique in terms of MOTA, IDF1, recall, and precision. The results indicate that the RSOADL–MODT technique reaches improvised results. Based on the MOTA value RSOADL–MODT technique results in a higher MOTA of 87.67 while the CenterTrack [19], FairMOT [20], TraDes [21], GTR [22], and MOTR [23] models attain decreased MOTA values.

Table 2. Comparative outcome of RSOADL–MODT approach with other systems on the DanceTrack dataset.

<table>
<thead>
<tr>
<th>Models</th>
<th>MOTA</th>
<th>HOTA</th>
<th>IDF1</th>
<th>DetA</th>
<th>AssA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSOADL–MODT</td>
<td>87.6</td>
<td>54.3</td>
<td>51.2</td>
<td>80.6</td>
<td>37.1</td>
</tr>
<tr>
<td>CenterTrack</td>
<td>86.8</td>
<td>41.8</td>
<td>35.7</td>
<td>78.1</td>
<td>22.6</td>
</tr>
<tr>
<td>FairMOT</td>
<td>82.2</td>
<td>39.9</td>
<td>40.8</td>
<td>66.7</td>
<td>23.8</td>
</tr>
<tr>
<td>TraDes</td>
<td>86.2</td>
<td>43.3</td>
<td>41.2</td>
<td>74.5</td>
<td>25.4</td>
</tr>
<tr>
<td>GTR</td>
<td>84.7</td>
<td>48.0</td>
<td>50.3</td>
<td>72.5</td>
<td>31.9</td>
</tr>
<tr>
<td>MOTR</td>
<td>79.7</td>
<td>47.7</td>
<td>51.5</td>
<td>73.5</td>
<td>40.2</td>
</tr>
<tr>
<td>QDTrack</td>
<td>87.7</td>
<td>54.2</td>
<td>50.4</td>
<td>80.1</td>
<td>36.8</td>
</tr>
</tbody>
</table>
Table 2. Comparative outcome of RSOADL–MODT approach with other systems on the DanceTrack dataset.

<table>
<thead>
<tr>
<th>Models</th>
<th>MOTA</th>
<th>HOTA</th>
<th>IDF1</th>
<th>DetA</th>
<th>AssA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSOADL–MODT</td>
<td>87.6</td>
<td>54.3</td>
<td>51.21</td>
<td>80.6</td>
<td>37.1</td>
</tr>
<tr>
<td>CenterTrack [19]</td>
<td>86.8</td>
<td>41.8</td>
<td>35.7</td>
<td>78.1</td>
<td>22.6</td>
</tr>
<tr>
<td>FairMOT [20]</td>
<td>82.2</td>
<td>39.9</td>
<td>40.8</td>
<td>66.7</td>
<td>23.8</td>
</tr>
<tr>
<td>TraDes [21]</td>
<td>86.2</td>
<td>43.3</td>
<td>41.2</td>
<td>74.5</td>
<td>25.4</td>
</tr>
<tr>
<td>GTR [22]</td>
<td>84.7</td>
<td>48.0</td>
<td>50.3</td>
<td>72.5</td>
<td>31.9</td>
</tr>
<tr>
<td>MOTR [23]</td>
<td>79.7</td>
<td>47.7</td>
<td>51.5</td>
<td>73.5</td>
<td>40.2</td>
</tr>
<tr>
<td>QDTrack [24]</td>
<td>87.7</td>
<td>54.2</td>
<td>50.4</td>
<td>80.1</td>
<td>36.8</td>
</tr>
</tbody>
</table>

Figure 7. Comparative outcome of RSOADL–MODT approach on the DanceTrack dataset.

In Table 3 and Figure 8, the overall tracking performance of the RSOADL–MODT method technique is examined in terms of MOTA, MOTP, IDF1, IDsw, and MT on the MOT17 dataset. The results indicate that the RSOADL–MODT technique accomplishes improved performance under each iteration. It is noticed that the RSOADL–MODT technique gains an average MOTA of 74.67, MOTP of 0.321, IDF1 of 72.31, Idsw of 4331, recall of 98.40, a precision of 97.63, and MT of 623.

Figure 8. Average result of RSOADL–MODT approach on the MOT17 dataset.
Table 3. Overall tracking result of RSOADL–MODT approach on the MOT17 dataset.

<table>
<thead>
<tr>
<th>MOT17 Dataset</th>
<th>No. of Iterations</th>
<th>MOTA</th>
<th>MOTP</th>
<th>IDF1</th>
<th>Idsw</th>
<th>Recall</th>
<th>Precision</th>
<th>MT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iteration-1</td>
<td>73.22</td>
<td>0.328</td>
<td>73.22</td>
<td>4488</td>
<td>98.40</td>
<td>97.70</td>
<td>617</td>
</tr>
<tr>
<td></td>
<td>Iteration-2</td>
<td>74.11</td>
<td>0.304</td>
<td>71.56</td>
<td>4559</td>
<td>98.24</td>
<td>97.64</td>
<td>623</td>
</tr>
<tr>
<td></td>
<td>Iteration-3</td>
<td>78.23</td>
<td>0.343</td>
<td>74.26</td>
<td>3991</td>
<td>98.73</td>
<td>97.57</td>
<td>623</td>
</tr>
<tr>
<td></td>
<td>Iteration-4</td>
<td>73.46</td>
<td>0.307</td>
<td>72.11</td>
<td>4039</td>
<td>98.14</td>
<td>97.91</td>
<td>629</td>
</tr>
<tr>
<td></td>
<td>Iteration-5</td>
<td>74.36</td>
<td>0.324</td>
<td>70.44</td>
<td>4580</td>
<td>98.48</td>
<td>97.32</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>74.67</td>
<td>0.321</td>
<td>72.31</td>
<td>4331</td>
<td>98.40</td>
<td>97.63</td>
<td>623</td>
</tr>
</tbody>
</table>

The TACY and VACY of the RSOADL–MODT model on the MOT17 dataset can be defined in Figure 9. The Figure implied that the RSOADL–MODT enrich methodology has displayed enhanced performance with maximal increased values of TACY and VACY, notably that the RSOADL–MODT technique has reached maximal TACY outcomes.

Figure 9. TACY and VACY outcome of RSOADL–MODT approach on the MOT17 dataset.

The TLOS and VLOS of the RSOADL–MODT algorithm on the MOT17 dataset can be defined in Figure 10. The Figure inferred that the RSOADL–MODT technique had revealed better performance with the smallest values of TLOS and VLOS. Visibly, the RSOADL–MODT technique has reduced VLOS outcomes.

Figure 10. TLOS and VLOS outcome of RSOADL–MODT approach on the MOT17 dataset.
To point out the enhanced performance of the RSOADL–MODT method, a comparative analysis is made on the MOT17 dataset in Table 4.

Table 4. Comparative outcome of RSOADL–MODT approach with other systems on the MOT17 dataset.

<table>
<thead>
<tr>
<th>Models</th>
<th>MOTA</th>
<th>HOTA</th>
<th>IDF1</th>
<th>DetA</th>
<th>AssA</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSOADL–MODT</td>
<td>74.6</td>
<td>55.6</td>
<td>72.3</td>
<td>60.7</td>
<td>56.3</td>
</tr>
<tr>
<td>CenterTrack [19]</td>
<td>67.8</td>
<td>52.2</td>
<td>64.7</td>
<td>53.8</td>
<td>51</td>
</tr>
<tr>
<td>FairMOT [20]</td>
<td>73.7</td>
<td>59.3</td>
<td>72.3</td>
<td>60.9</td>
<td>58</td>
</tr>
<tr>
<td>TraDes [21]</td>
<td>69.1</td>
<td>52.7</td>
<td>63.9</td>
<td>55.2</td>
<td>50.8</td>
</tr>
<tr>
<td>MOTR [23]</td>
<td>73.4</td>
<td>57.8</td>
<td>68.6</td>
<td>60.3</td>
<td>55.7</td>
</tr>
<tr>
<td>QDTrack [24]</td>
<td>68.7</td>
<td>53.9</td>
<td>66.3</td>
<td>57.2</td>
<td>52.7</td>
</tr>
</tbody>
</table>

Figure 11 exhibits the comparative assessment of the RSOADL–MODT approach in terms of MOTA, IDF1, recall, and precision. The results indicate that the RSOADL–MODT technique reaches improvised results. Based on MOTA, the RSOADL–MODT algorithm result in a higher MOTA of 74.67, while the CenterTrack [19], FairMOT [20], TraDes [21], GTR [22], MOTR [23] and QDTrack [24] methods attain decreased MOTA values. In the meantime, based on IDF1, the RSOADL–MODT technique results in a higher IDF1 of 72.31 while the CenterTrack [19], FairMOT [20], TraDes [21], GTR [22], MOTR [23] and QDTrack [24] models attain decreased IDF1 values.

![Figure 11. Comparative outcome of RSOADL–MODT approach on the MOT17 dataset.](image)

**Table 4.** Comparative outcome of RSOADL–MODT approach with other systems on the MOT17 dataset.

**Ablation Study**

To demonstrate the effectiveness of the proposed method, an ablation study on the validation of DanceTrack and MOT17 train sets by performing 3-fold cross-validation following [38]. This study analyzes the main aspects of our method: (i) the advantages of using a PA–RetinaNet object detector and (ii) the hyper-parameter optimization method across different levels.

At the initial stage, the presented RSOADL–MODT technique applies path augmented RetinaNet (PA–RetinaNet) based object detection module, which improves the feature extraction process. For optimization and to improvise the network potentiality of the PA–RetinaNet method, the Reptile Search Optimization Algorithm (RSOA) is utilized as a hyperparameter optimizer. The optimization difficulty for object detection is greatly relieved. It combines the feature extractor with a PA–RetinaNet detector and reptile search optimization and observes that it achieves the best performance with 87.6 MOTA and 51.2 IDF1. MOTA and IDF1 were improved, and the IDs were significantly reduced. MOTA
and IDF1 are calculated for overall evaluations. Nevertheless, this initiative approach can be used to progress MOTA and IDF1 further. YOLOx proposal does not perform well enough in small datasets such as MOT17 due to its data-hungry nature, and it works well in large datasets such as DanceTrack [24]. When compared to the QDTrack, the proposed method gives a sufficient improvement in MOT17.

The results indicate that base object detectors with classification feature embeddings are adequate for multiple-object tracking with effective optimization. This method produces reasonable results and achieves the best overall scores with MOTA and IDF1. Additionally, an optimization algorithm with deep learning boosts the performance on all metrics while tuning hyperparameters and, finally, slight computation in the method. In addition, better computational outcomes have been obtained by metaheuristics with the precise optimization algorithm. When compared to the state-of-the-art methods given in Table 4, the proposed method is slightly less computationally expensive.

The proposed method finds that if it concerns the above features with hyper-parameter tuning, the result on the DanceTrack dataset shows slight improvement, and long-time occlusion between the objects should be improved using meta-heuristic optimization algorithms in the future.

Table 4 reports a significant improvement of MOTA, IDF1, and HOTA over state-of-the-art techniques, and it shows that similar tracking performance can be achieved by using optimization when performing the tracking. Given the sole features of the datasets, these results highlight the versatility of the approach to utilize the right methods for different scenarios.

4. Conclusions

In this study, an automated MOT using the RSOADL-MODT technique has been developed for the recognition and tracking of the objects that exist with position estimation, tracking, and action recognition. It follows a series of processes, namely object detection, hyperparameter tuning, object classification, and object tracking. Primarily, the presented RSOADL-MODT technique exploited the PA–RetinaNet-based object detection module, which improves the feature extraction process. For enhancing the network potentiality of the PA–RetinaNet method, the RSOA is utilized as a hyperparameter optimizer. At last, the QRNN classifier is exploited for classification procedures. A wide-ranging experimental validation process takes place on DanceTrack and MOT datasets for examining the effectual object detection outcomes of the RSOADL-MODT algorithm. The simulation values confirmed the enhancements of the RSOADL-MODT method over other DL models. In the future, feature fusion-based DL techniques can be employed to improve the object detection results of the proposed model.

Author Contributions: Conceptualization, R.A. and D.M.; Formal Analysis, R.A. and D.M.; Investigation, R.A.; Writing—original draft, R.A.; Writing—Review and Editing, D.M.; Supervision, D.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References


17. Jiang, T.; Zhang, Q.; Yuan, J.; Wang, C.; Li, C. Multi-Type Object Tracking Based on Residual Neural Network Model. Symmetry 2022, 14, 1689. [CrossRef]

18. Wang, Y.; Zhang, Z.; Zhang, N.; Zeng, D. Attention Modulated Multiple Object Tracking with Motion Enhancement and Dual Correlation. Symmetry 2021, 13, 266. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.