IEICE TRANS. FUNDAMENTALS, VOL.E107–A, NO.9 SEPTEMBER 2024

PAPER

DETrack: Multi-Object Tracking Algorithm Based on Feature Decomposition and Feature Enhancement

Feng WEN[†], **Haixin HUANG**[††], **Xiangyang YIN**[†], **Junguang MA**[†], ***and*** **Xiaojie HU**[†a)], ***Nonmembers***

SUMMARY Multi-object tracking (MOT) algorithms are typically classified as one-shot or two-step algorithms. The one-shot MOT algorithm is widely studied and applied due to its fast inference speed. However, one-shot algorithms include two sub-tasks of detection and re-ID, which have conflicting directions for model optimization, thus limiting tracking performance. Additionally, MOT algorithms often suffer from serious ID switching issues, which can negatively affect the tracking effect. To address these challenges, this study proposes the DETrack algorithm, which consists of feature decomposition and feature enhancement modules. The feature decomposition module can effectively exploit the differences and correlations of different tasks to solve the conflict problem. Moreover, it can effectively mitigate the competition between the detection and re-ID tasks, while simultaneously enhancing their cooperation. The feature enhancement module can improve feature quality and alleviate the problem of target ID switching. Experimental results demonstrate that DETrack has achieved improvements in multi-object tracking performance, while reducing the number of ID switching. The designed method of feature decomposition and feature enhancement can significantly enhance target tracking effectiveness.

key words: *multi-object tracking, feature decomposition, feature enhancement*

1. Introduction

Multi-object tracking (MOT) technique combines image processing, computer vision, and machine learning to identify and track specific targets. This method first detects whether the image or video frame contains the desired object and then retrieves its location and ID. Multi-object tracking is a crucial tool for analyzing behavior and identifying abnormal postures in various fields such as intelligent transportation, crowd counting, public safety, intelligent surveillance, and autonomous driving. This technique has gained widespread usage due to its ability to improve situational awareness and enhance decision-making processes.

MOT algorithms can be categorized into two types: Separate Detector and Embedding model (SDE) and Jointly learns the Detector and Embedding model (JDE), also known as two-step and one-shot MOT algorithms respectively. In the two-step approach, detection and Re-IDentification (re-ID) tasks are performed sequentially. First, the object detection algorithm annotates various types of class objects in the image with bounding boxes. Then, the re-ID model extracts the appearance features of the object while matching the object identity by using a data association algorithm. However, this approach has poor real-time performance, as the two task models are trained and deployed separately. In contrast, the one-shot MOT algorithm uses multi-task learning to fuse the two tasks into a single network. This kind of algorithm can simultaneously obtain both the bounding boxes and appearance features of the target, resulting in high real-time performance. This approach has gained popularity due to its ability to handle complex tracking scenarios, such as occlusions and pose changes, with high accuracy. Overall, the one-shot MOT algorithm represents a promising approach to tracking multiple objects in real-time, and it is likely to find widespread use in various applications, such as autonomous driving and video surveillance.

The One-shot MOT algorithm employs a multi-task learning approach that integrates object detection and re-ID into a single model by sharing feature maps. This approach reduces the number of network parameters, thus increasing the model's speed. By combining the two tasks, this method strikes a balance between detection speed and accuracy. FairMOT pointed out the existence of conflicts for feature maps between the object detection task and the Re-ID task [1]. We experimentally verified these conflicts between the two tasks, and the results are shown in Sect. 4. During object detection, the network's goal is to enhance the similarity of objects in the same category, while the re-ID task aims to maximize the differences between objects of the same category. This conflicting optimization purpose can reduce the effectiveness of multi-object tracking.

In MOT the data association method typically employs both motion and appearance models to match targets. However, the re-ID task, which extracts the appearance features, often lacks representativeness, exacerbating the ID switching problem. The key to alleviating the ID switching problem lies in improving the feature characterization capability of the re-ID task. To address these issues, this paper proposes an efficient multi-object tracking algorithm based on feature decomposition and feature enhancement, namely DETrack.

In this study, we propose DETrack to address the optimization conflict and ID switching problems of the one-shot MOT algorithm. Our approach introduces two novel modules: the feature decomposition module and the feature enhancement module. The feature decomposition module is designed to alleviate the conflict by identifying the correlation and difference between feature maps of each task.

Manuscript received December 20, 2023.
Manuscript revised March 18, 2024.
Manuscript publicized April 22, 2024.

[†]School of Information Science and Engineering, Shenyang Ligong University, Liaoning, China.
[††]School of Automation and Electrical Engineering, Shenyang Ligong University, Liaoning, China.
a) E-mail: xiaojie.hu.wmu@gmail.com

DOI: 10.1587/transfun.2023EAP1162

It uses parallel convolution and feature fusion to decompose the feature maps into detection feature maps and re-ID feature maps. By doing so, we separate the two tasks and reduce the conflicts that arise from combining them into a single model. The second module, the feature enhancement module, is proposed to ease the ID switching problem. It enhances the feature representation of the input feature map by using a combination of the large kernel attention, channel attention, and spatial attention mechanisms. By employing these attention mechanisms, we can extract more informative feature, which improves the re-ID task's feature characterization capability.

In summary, the proposed DETrack algorithm provides an effective solution to the one-shot MOT conflict and ID switching problems. By decomposing the feature maps and enhancing the feature representation, we improve the performance of the one-shot MOT algorithm in multi-object tracking scenarios. Our proposed modules can be easily integrated into existing one-shot MOT algorithms, making them more practical and applicable for real-world applications.

2. Related Work

One-shot multi-object tracking algorithms have been shown to improve tracking efficiency by fusing detection and re-ID models into the same backbone network. For instance, Retinatrack [2] and JDE [3] incorporated a re-ID branch into their one-stage detection methods to achieve a balance of tracking performance and speed. These studies revealed that the anchor-based detection method was unsuitable for multi-object tracking tasks and that the high-dimensional re-ID features were too redundant for multi-object tracking. To address these issues, FairMot was proposed, and CenterNet was utilized in the object detection branch to produce lower-dimensional re-ID appearance features [1]. In addressing these challenges, CenterTrack [4] replaced the anchor-based Faster R-CNN [5] with the anchor-free CenterNet [6]. This adaptation signifies the instantiation of a multi-object tracking algorithm grounded in anchor-free object detection principles. FairMOT is also among the multi-object tracking algorithms built upon CenterNet. However, the one-shot multi-object tracking network that simultaneously performs the tasks of object detection and re-ID can result in conflict. To mitigate this issue, RelationTrack [7] was proposed with a feature decoupling module and a feature learning module with global information, which improved the Identity F1 score (IDF1) [8] by 3% on MOT16 dataset [9].

The SORT [10] algorithm employs the Kalman filter and Hungarian matching algorithm to achieve a MOTA [11] of 33.4 on the MOTChallenge dataset [12] and a tracking speed of 260 FPS. However, SORT only utilizes the Kalman filter to estimate the target's motion state, disregarding the target's inherent similarity and resulting in a significant number of ID switches. The DeepSort [13] algorithm improves upon SORT by incorporating a feature extraction module. By combining geometric and feature similarity, DeepSort yields superior results to SORT. The ByteTrack [14] algorithm recognizes that simultaneous data association of detection frames of varying quality can adversely impact tracking results. Accordingly, detection frames are categorized into two groups, with priority given to high-quality frames followed by low-score frames for matching, resulting in a new SOTA algorithm.

Attention mechanisms have been utilized to enhance the feature representation extracted by convolutional neural networks. SE-Net [15] predicts a weight for each output channel and applies weights to each channel to improve detection performance. For instance, CBAM [16] incorporates both spatial and channel attention mechanisms to achieve superior results. The ECA-Net model presents advancements over the SE-Net by introducing a local cross-channel interaction strategy and an adaptive selection of the one-dimensional convolutional kernel size, which leads to an overall improvement in performance [17]. Furthermore, the Large Kernel Attention (LKA) mechanism enables adaptability in both spatial and channel dimensions, enhancing the perceptual field capture and remote dependence [18]. Studies also shown that fusing YOLOv4 with CBAM [19], [20], as well as integrating SE-Net into YOLOv5 [21], significantly improved detection accuracy and recall. These studies demonstrate that feature enhancement based on attention mechanisms can effectively improve feature map representation and detection performance. In the context of multi-object tracking scenarios, attention mechanisms can further enhance object tracking performance by improving the detection performance.

3. Method

The DETrack algorithm is comprised of four main components: the backbone network, feature decomposition module, detection network, and re-ID network, as illustrated in Fig. 1. This algorithm builds on the anchor-free object detection approach of FairMot, which is particularly suited for multi-object tracking. DETrack employs DLASeg [1] as the backbone network, takes three channels RGB images as input, and generates output that includes the bounding box and embedding of the target. The feature decomposition module is responsible for extracting task-relevant and distinctive features from the upper layer features. The detection network is used to identify the category and location of the target, while the re-ID network is responsible for computing the target's embedding.

3.1 Feature Decomposition Module

The one-shot MOT algorithm employs a multi-task learning approach, which can create conflicts between different tasks and negatively impact tracking performance. To address this issue, the DETrack algorithm decomposes the output feature map of the backbone network into a detection feature map and a re-ID feature map, as illustrated in Fig. 2. The detection task emphasizes commonalities among targets, whereas the re-ID task focuses on their differences. In multi-object

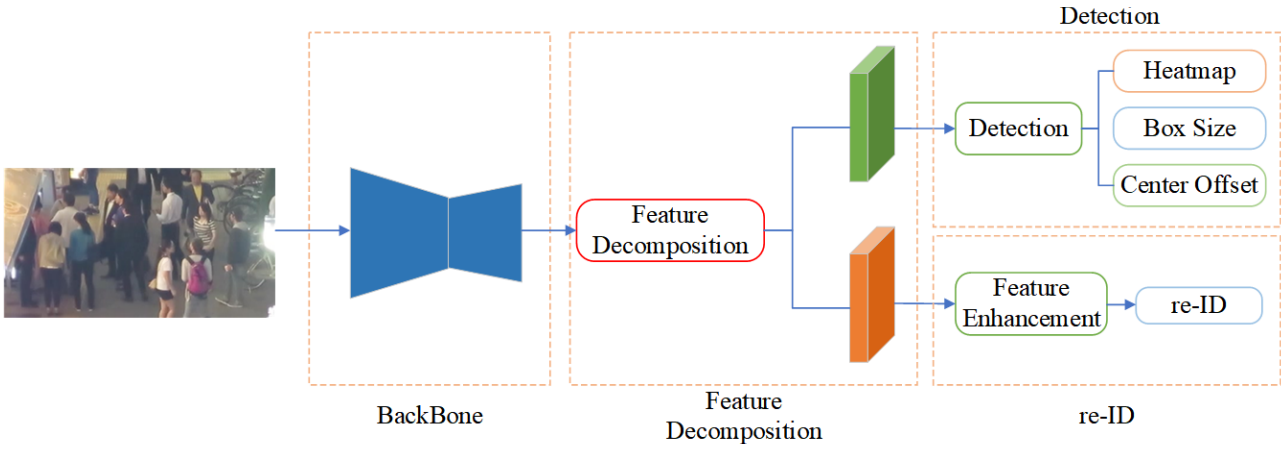


Fig. 1 Overall structure of DETrack.

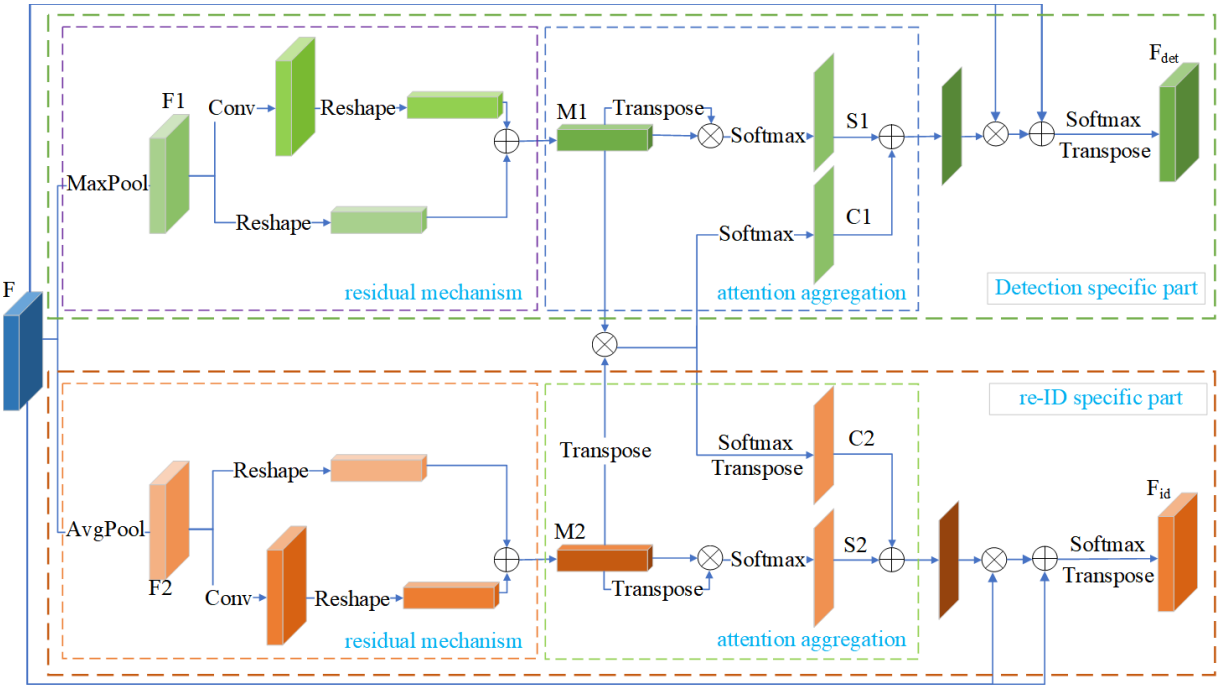


Fig. 2 Feature decomposition module framework.

tracking, the detection tasks and re-ID tasks are correlated, with information such as target texture and color being separated into foreground and background in the detection task, while distinctive appearance features are used in the re-ID task to differentiate between targets. Because these two tasks are interrelated, the feature decomposition module needs to fuse the difference and correlation features across tasks to optimize the synergies between them.

The feature decomposition module is composed of detection-specific and re-ID-specific parts, see Fig. 2. Both of the parts share the same structural design and utilize the residual mechanism and attention fusion to learn feature representations that possess relevant task characteristics. The backbone network generates feature maps represented by $F \in R^{C\times H\times W}$. Initially, these two steps conduct maximum pooling and average pooling to downsample F, leading to the formation of $F1$ and $F2$. Subsequently, $F1$ and $F2$ generate tensors $M1$ and $M2$, respectively, via different residual modules. Thereafter, $M1$ and $M2$ are subjected to multiplication with their corresponding transpose matrices, followed by multiplication with each other. The resulting products are subjected to Softmax and summation to obtain a tensor that encapsulates both task discrepancy and task relevance. Finally, the tensors are multiplied with the input F, and the resulting product is subjected to Softmax, Transpose, and summation to obtain the detection feature map and the re-ID feature map.

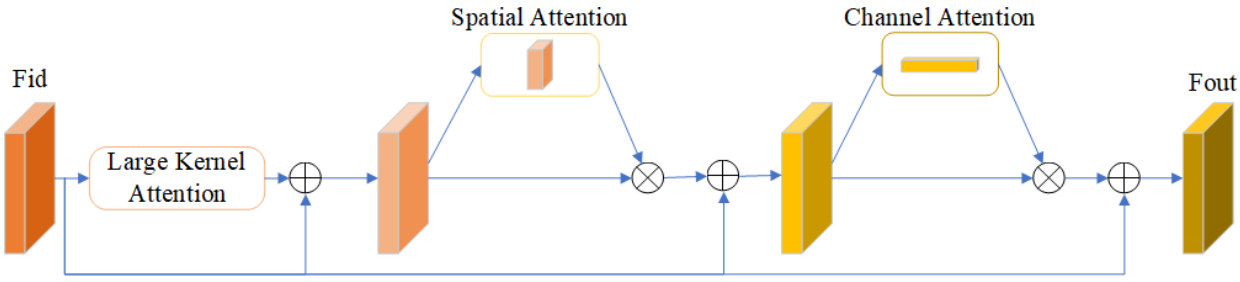


Fig. 3 Feature enhancement module framework.

3.2 Feature Enhancement Module

In multi-object tracking scenarios, ID switching often occurs on neighboring targets due to inadequate appearance feature extraction, despite most objects being detectable in position. In a typical one-step MOT tracking scenario, the detection network and re-ID network employ the same network inference structure. However, simple convolutional network for appearance feature extraction has limited ability to accurately characterize features, necessitating the use of deeper networks. While, employing complex networks may result in less accurate feature learning and exponentially increased inference time consumption. Hence, it is necessary to design an effective module to extract appearance feature.

The feature enhancement module proposed in this study is composed of three sub-modules: large kernel attention, spatial attention, and channel attention, see Fig. 3. These sub-modules estimate the attention weights on the spatial and channel dimensions, thereby enhancing the characterization of appearance features. This process reduces the number of ID switches and mitigates the risk of false matches. Moreover, this feature enhancement module is able to maintain the inference time while effectively enhancing the semantics of the features.

The feature enhancement module is as follows.

$$\begin{cases} \mathbf{F}_{\text{lka}} = \mathbf{F}_{\text{id}} + \text{LKA}\left(\mathbf{F}_{\text{id}}\right) \\ \mathbf{F}_{\text{spatial}} = \mathbf{F}_{\text{id}} + \mathbf{F}_{\text{lka}} \times \text{SA}\left(\mathbf{F}_{\text{lka}}\right) \\ \mathbf{F}_{\text{channel}} = \mathbf{F}_{\text{id}} + \mathbf{F}_{\text{spatial}} \times \text{CA}\left(\mathbf{F}_{\text{spatial}}\right) \\ \mathbf{F}_{\text{out}} = \mathbf{F}_{\text{id}} + \mathbf{F}_{\text{channel}} \end{cases} \tag{1}$$

The large kernel attention module proposed in this study includes depth-wise convolution and point-wise convolution. This module can efficiently capture long-range relationships using a relatively small number of computational efforts and parameters. By calculating the importance of each pixel, an attention map can be generated, allowing for the identification of salient features. The detailed structure of the large kernel attention module is depicted in Fig. 4.

Specifically, assuming a dilation interval of d, a $K \times K$ standard convolution is disassembled into a $\frac{K}{d} \times \frac{K}{d}$ deep dilation convolution, a $(2d-1)\times(2d-1)$ depth-wise convolution, and a 1×1 point-wise convolution. This disassembly technique not only saves computational overhead but also yields an attention map that can effectively capture long-range dependencies. After undergoing large kernel attention

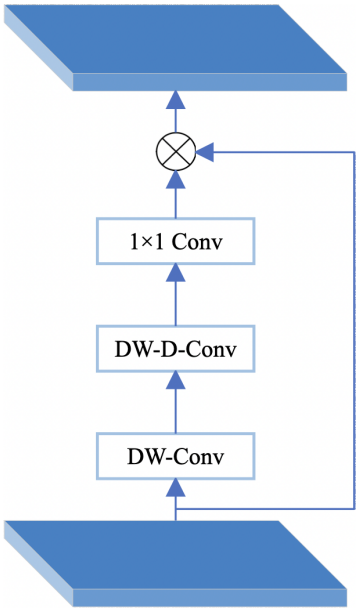


Fig. 4 Constructing large kernal attention.

enhancement, the feature map is then fed into the spatial attention enhancement module. In this module, the input feature map is subjected to max-pooling and average-pooling in the channel dimension, resulting in two single-channel feature maps, each with dimensions of $1 \times H \times W$. These output feature maps are then combined to form a feature map with two channels, which is subsequently transformed into a single-channel feature map using convolution calculation. Finally, the spatial attention weights are obtained by applying the ReLU activation function, as illustrated in Fig. 5. Our proposed approach offers a computationally efficient solution that is capable of capturing long-range dependencies, and therefore has the potential to enhance the performance of a wide range of applications.

The channel attention mechanism can obtain channel attention weights by enhancing the category sensitivity of feature maps, see Fig. 6. In this approach, spatial features are subjected to average and maximum pooling to compute channel feature information, resulting in output dimensions of $1 \times C$. These output features are then fully connected and convolved, and the resulting values are summed. ReLU is used to obtain the channel attention weights. These weights are multiplied by the input feature maps at corresponding

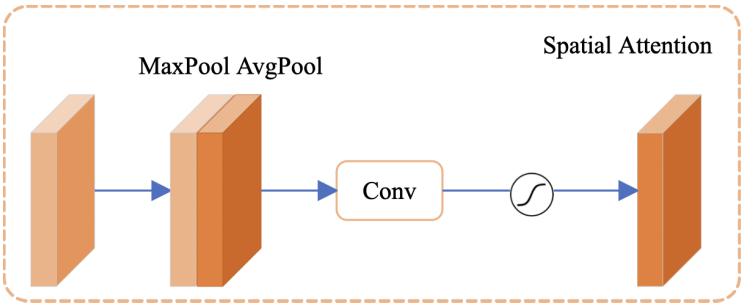


Fig. 5 Spatial attention framework.

$$L_{\text{heatmap}} = -\frac{1}{N}\sum_{xyz}\begin{cases}\left(1-\hat{Y}_{xyc}\right)^{\alpha}\log\left(\hat{Y}_{xyc}\right), & Y_{xyc}=1\\ \left(1-Y_{xyc}\right)^{\beta}\left(\hat{Y}_{xyc}\right)^{\alpha}\log\left(1-\hat{Y}_{xyc}\right), & \text{otherwise}\end{cases} \tag{2}$$

positions and are then fused with the input feature maps of the feature enhancement module.

3.3 Loss Function

The loss function utilized in the DETrack algorithm is composed of both detection and re-ID branch loss functions. The detection branch outputs Heatmap, Box Size, and Center Offset. Heatmap represents the probability of a pixel point being the target centroid, with values ranging from 0 to 1. Box Size represents the size of the bounding box with the current pixel point as the centroid. However, since the backbone of the algorithm utilizes downsampling to compute features, there may be deviations in mapping points in the feature map to the original input size. Therefore, Center Offset is designed to represent the centroid offset. The focal loss function [22] of Heatmap is shown in Eq. (2).
where $\hat{Y}$ is the estimated heatmap. With the (x,y) pixel as the centroid, $\hat{Y}_{xyc}$ is the predicted probability of the target category C and Y_{xyc} is the true probability of the target category being C. The focal loss function was employed with predetermined parameters α and β, where α=2 and β=4.

For the Center Offset, we use the Mean Absolute Error (MAE), also known as the L1, for the loss function.

$$L_{\text{offset}} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\widetilde{p}} - \left(\frac{p}{\mathrm{R}} - \widetilde{p}\right)\right| \tag{3}$$

where N is where is the training parameter batch size, p is the centroid coordinate of the labeled box, $\widetilde{p}$ is the result of downward rounding of the centroid coordinate after downsampling, R is the downsampling factor, $\hat{O}_{\widetilde{p}}$ is the prediction error value of the network output, and $\frac{p}{\mathrm{R}} - \widetilde{p}$ is the position error of the real centroid and the predicted centroid.

The size prediction branch utilizes the L1 function to predict the width and height of the detection frame:

$$L_{\text{size}} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{p_k} - S_k\right| \tag{4}$$

where $\hat{S}_{p_k}$ is the width and height of the predicted detection box. The coordinates of the upper left and lower right corners of the real box are $\left(x_1^k, y_1^k, x_2^k, y_2^k\right)$, and the bounding box size is

$$S_k = \left(x_2^k - x_1^k, y_2^k - y_1^k\right) \tag{5}$$

The loss function for the target detection subtask is calculated as follows:

$$L_{\text{detection}} = L_{\text{heatmap}} + L_{\text{offset}} + 0.1 \times L_{\text{size}} \tag{6}$$

The re-ID features are learned through a classification task. In this study, object instances with the same identity in the training set were regarded as same class. For each detection box in the image, we obtain the object center on the heatmap, then extract the re-ID embedding and map it to a class distribution matrix $\mathbf{P} = \{\mathbf{p}(k), k \in [1, K]\}$, where K represents the number of classes, and classes labeled as $L^i(k)$. The re-ID loss is computed as:

$$L_{\text{identity}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} L^i(k)\log(p(k)) \tag{7}$$

Uncertainty loss [23] is used to automatically balance the detection and re-ID tasks:

$$L_{total} = \frac{1}{2}\left(\frac{1}{e^{\omega_1}}L_{detection} + \frac{1}{e^{\omega_2}}L_{identify} + \omega_1 + \omega_2\right) \tag{8}$$

where ω_1 and ω_2 are learnable parameters.

4. Experiments and Results

4.1 Data and Metrics

This research experiment was conducted using an Ubuntu 18.04 LTS operating system, two Tesla V100 graphics cards, and the Python programming language with the PyTorch deep learning framework for model training and testing.

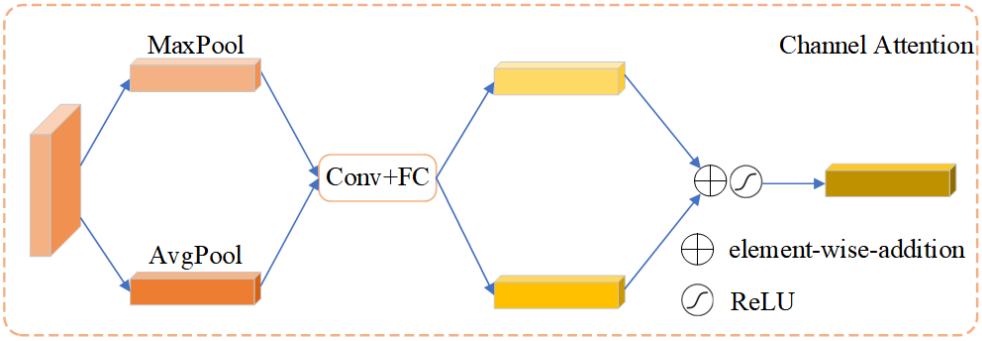


Fig. 6 Channel attention framework.

In this study, we used a variety of datasets for the experiments, collectively referred to as the MIX dataset. The datasets used include ETH [24], CityPersons [25], Caltech Pedestrian [26], MOT17 [9], CUHK-SYSU [27], PRW [28], and CrowdHuman [29]. The ETH and CityPersons datasets were used for training the detection model, while the Caltech Pedestrian, MOT17, CUHK-SYSU, and PRW datasets were used to train both the detection and re-ID subtasks. The CrowdHuman dataset provided bounding box information and was used for model pre-training. Finally, the model was evaluated using the MOT16 and MOT17 test datasets to determine its effectiveness.

The pre-trained model was trained for 60 epochs using the CrowdHuman dataset, with a batch size of 8. Then, the model was fine-tuned for 45 epochs on the MIX dataset using the Adam optimizer with a learning rate of 1e-4 and a batch size of 12. The learning rate was decayed to 1e-5 at round 20. Data enhancements, including image rotation, translation, scaling, and color dithering, were used during training. The input image size was converted to 1088*608 and processed by the backbone network. The output feature map size was reduced to 1/4 size. The total training process took 72 hours.

4.2 Experiment of Conflict between the Object Detection Task and the Re-ID Task

We experimentally verified the conflicts for feature maps between the object detection task and the Re-ID task, as shown in Table 1. FairMOT was trained for 60 epochs on the Crowdhuman dataset [2]. Thirty epochs were dedicated to training using half of the MOT17 dataset [3], with the other half reserved for validation. FairMOT's default configuration included detection (Det) and Re-ID loss ratios set at 1 and 0.1, respectively. Additionally, we introduced two contrasting sets of configurations with Det and Re-ID loss ratios set at 0.5 and 0.1, and 2 and 0.1, respectively. Experimental results indicate that as the Re-ID to Det loss ratio increases, the MOTA values tend to decrease, while the IDF1 values tend to increase. The experiments indicate that this conflict will reduce the effectiveness of multi-object tracking.

Table 1 A comparison of FairMOT's Detection/Re-ID loss ratios using the MOT17 dataset.

FairMOT Configuration	MOTA↑	IDF1↑
Det=0.5, Re-ID=0.1	71.39	74.64
Det=1, Re-ID=0.1 (default)	71.54	74.55
Det=2, Re-ID=0.1	71.80	74.03

Table 2 Ablation analysis of feature decomposition and feature enhancement modules.

Method	IDF1↑	IDs↑	MOTA↑	FPS↑
Baseline	80.3	559	82.3	28.6
Baseline+FD	82.1	551	83.5	26.5
Baseline+FE	82.9	527	82.8	26.8
Baseline+FD+FE	83.7	512	83.9	24.0

4.3 Ablative Studies

In this study, we propose feature decomposition and feature enhancement modules and use ablation experiments to validate their effectiveness in improving the detection performance of the DETrack algorithm. We use the MOT17 and the MIX dataset for training and validation. The MOT17 training dataset is divided into two equal parts, with one part used for training with MIX dataset, and the other part for validation. The One-shot MOT algorithm is used as the baseline, and we test the effectiveness of the feature decomposition (FD) and feature enhancement modules (FE) in improving detection performance.

First, experimental results show that the introduction of the feature decomposition module increases the Multiple Object Tracking Accuracy (MOTA) from 82.3% to 83.5%, see Table 2. This increase indicates that the feature decomposition module plays an important role in the target detection subtask. Additionally, the feature decomposition module optimizes the appearance extraction subtask, as evidenced by the decrease in the number of Identity Switches (IDs) from 559 to 551. Moreover, the feature decomposition module increases the IDF1 score from 80.3% to 82.1%, indicating that it improves the accuracy of identity classification. However, the feature decomposition module also increases the computational time consumption to some extent, as evidenced by the decrease in Frames Per Second (FPS) from 28.6 to 26.5. Despite this, this module effectively allevi-

Fig. 7 Tracking performance of DETrack on the MOT17 test dataset.

ates the conflict problem of the One-shot MOT algorithm, making it a valuable addition to the algorithm.

Second, results revealed that introducing the feature enhancement module on the baseline method resulted in a decrease in the number of IDs from 559 to 527, indicating that the module improves the expression of appearance features and reduces the number of false matches in the data association phase. Furthermore, the MOTA improved from 82.3% to 82.8%, and the IDF1 score improved from 80.3% to 82.9%, demonstrating that the feature enhancement module effectively alleviates the problem of ID switching, which is commonly faced by MOT algorithms.

Third, after introducing both feature decomposition and feature enhancement modules to the baseline method, the MOTA improved from 82.3% to 83.9%, the IDs decreased from 559 to 512, and the IDF1 score improved from 80.3% to 83.7%. These experimental results indicate that combining both modules improves the tracking performance much more than using either module alone. The DETrack algorithm effectively alleviates the conflict problem in the One-shot MOT algorithm and improves the quality of extracted appearance features. Although the FPS decreased from 28.6 to 24.0, the DETrack algorithm still meets the requirements of real-time multi-object tracking.

4.4 Qualitative Results

This section presents the qualitative results of DETrack on the MOT17 benchmark dataset, along with an analysis of the tracking outcomes, as illustrated in Fig. 7. The results demonstrate that DETrack performs well in capturing objects even when they undergo long-distance movements within the camera's field of view, see Fig. 8. In crowded scenarios, DETrack exhibits proficient prediction of detection boxes and trajectories, see Fig. 9. Moreover, DETrack demonstrates capability in tracking small distant objects, thereby extending its applicability across diverse domains, see Fig. 10.

However, DETrack exhibits limitations in scenarios where targets are excessively occluded or move out of the camera's field of view, rendering the detector ineffective in detecting objects, as shown in Fig. 11. Subsequently, when the objects reappear within the camera's field of view, DETrack erroneously assigns a new ID to the tracked objects. Such occurrences pose challenges to conventional MOT methods as prolonged disappearance may incorrectly infer the object is out of range, and its reappearance after some time may disrupt trajectory consistency, leading to the assignment of new trajectory IDs. In future research, we aim to further enhance the performance of the tracker and focus on addressing such challenges.

4.5 Comparison with Other SOTA Algorithms

In this study, we also compared the performance of the DETrack algorithm with state-of-the-art One-shot, Two-step, and other types of MOT algorithms. For the comparison, we used the MOT16 and MOT17 datasets and a private detector for detection. The Two-step algorithms included in the comparison were DeepSORT [7], RAR16wVGG [30], TAP [31], CNNMTT [32], and POI [33]. The One-shot algorithms compared were JDE, FairMot, CSTrack [34], and CSTrack++ [35]. We also included other none re-ID algorithms for joint detection and tracking, such as CTrackerV1 [36], TubeTK [37], CenterTrack [38], and PermaTrack [39]. By comparing the DETrack algorithm with these state-of-the-art MOT algorithms, we aimed to demonstrate its su-

Fig. 8 Tracking performance of DETrack on object long-distance movements trajectories.

Fig. 9 Tracking performance of DETrack in densely crowded areas.

periority in terms of tracking accuracy and computational efficiency.

DETrack stands out from traditional Two-step MOT algorithms due to its advanced re-ID appearance feature extraction and data association method, see Table 3. In comparison to other state-of-the-art Two-step MOT algorithms, DETrack achieved superior performance in terms of MOTA and IDF1 metrics, with a significant 9.2% and 10.2% improvement respectively when compared to POI. Additionally, DETrack also demonstrated faster inference speed, outperforming POI by 18.8 FPS.

In the realm of One-shot MOT algorithms, DETrack has demonstrated superior performance when compared to other state-of-the-art approaches such as FairMot and CSTrack++.

Fig. 10 Tracking performance of DETrack on small objects.

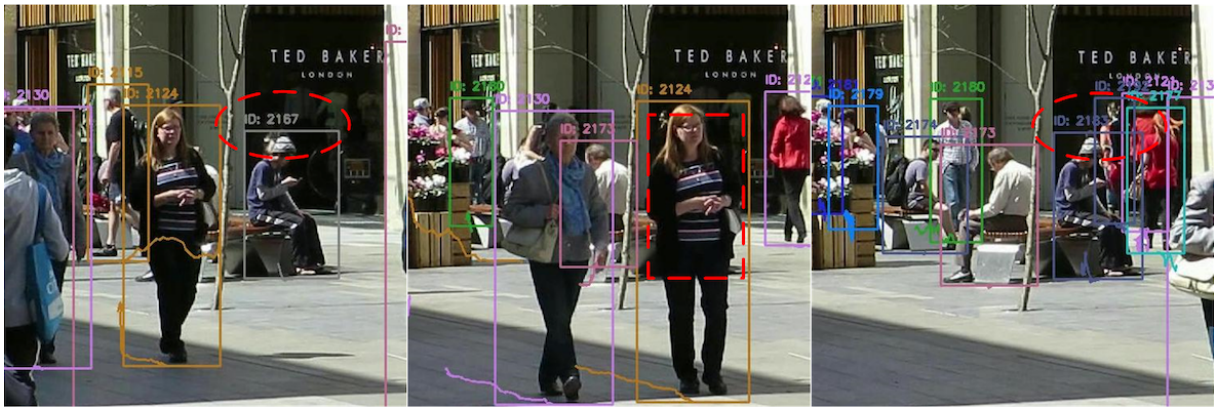


Fig. 11 DETrack failure cases.

On the MOT17 test dataset, DETrack outperformed FairMot in terms of MOTA, IDF1, MT, and ML metrics by 1.1%, 2.3%, 1%, and 1.5% respectively. Additionally, DETrack achieved a reduction of 858 ID switches and a decrease in inference speed of 1.9 FPS in comparison to FairMot. The improvements in tracking accuracy can be attributed to the cascade matching of different quality detection results on the DeepSORT data association, which enhances detection and data association metrics in the tracking results. When compared to CSTrack++, the results on the MOT17 test dataset showed that DETrack was 1.5% lower in MOTA, but 0.8% higher in IDF1, and demonstrated an 11.2 FPS improvement in inference speed. These findings suggest that DETrack may be more effective in handling multi-task learning conflicts, balancing subtasks, and data associations.

In comparison to other none re-ID algorithms, DETrack outperformed in terms of MOTA and IDF1 metrics on both test datasets. On the MOT16 test dataset, DETrack showed MOTA leads of 8.3%–11.9% and IDF1 leads of 15.9%–18.1% in comparison to other none re-ID algorithms. Similarly, on the MOT17 test dataset, DETrack demonstrated MOTA leads of 0.7%–11.8% and IDF1 leads of 5.7%–16.0% in comparison to none re-ID algorithms. These results suggest that incorporating re-ID appearance feature information in object tracking algorithms can significantly improve tracking accuracy, particularly in scenarios where high-quality appearance features are crucial for matching detection results and trajectories.

5. Conclusion

This study presents the DETrack algorithm, a novel ap-

Table 3 Comparison of DETrack with state-of-art methods.

Data	Methods	MOTA↑	IDF1↑	MT↑	ML↓	IDs↓	FPS↑
MOT16	EAMTT[40]	52.5	53.3	19.9%	34.9%	910	12.2
	SORT	59.8	53.8	25.4%	22.7%	1423	59.5
	DeepSORT	61.4	62.2	32.8%	18.2%	781	<6.7
	RAR16wVGG	63	63.8	39.9%	22.1%	482	<1.5
	VMaxx[41]	62.6	49.2	32.7%	21.1%	1389	6.5
	TubeTK	64	59.4	33.5%	19.4%	1117	1
	JDE*	64.4	55.8	35.4%	20%	1544	18.8
	TAP	64.8	73.5	38.5%	21.6%	571	39.4
	HOGM[31]	64.8	73.5	40.6%	22%	794	<8.0
	CNNMTT	65.2	62.2	32.4%	21.3%	946	<5.2
	POI	66.1	65.1	34%	20.8%	805	<5.2
	CTracker v1	67.6	57.2	32.9%	23.1%	1897	6.8
	CSTrack*	69.4	69.3	35%	22.3%	958	16.9
	CSTrack++*	76.4	74.1	46.1%	13.3%	-	12.8
	FairMOT*	74.9	72.8	44.7%	15.9%	1074	25.9
	DETrack*	75.9	75.3	45.8%	14.2%	836	24
MOT17	STT	52.4	49.5	21.4%	30.7%	8431	6.3
	TubeTK	63	58.6	31.2%	19.9%	4137	3
	CenterTrack	67.8	64.7	34.6%	24.6%	3039	22
	CTracker v1	66.6	57.4	32.2%	24.2%	5529	6.8
	CSTrack*	67.3	67.9	34.2%	24.1%	2994	16.9
	CSTrack++*	76.3	73.8	44.7%	13.6%	-	12.8
	FairMOT*	73.7	72.3	43.2%	17.3%	3303	25.9
	LMOT[42]	72	70.3	45.4%	17.3%	3071	28.6
	PermaTrack	73.8	68.9	43.8%	17.2%	3699	11.9
	SGT[43]	76.3	72.4	47.9%	11.7%	4578	62.5
	TrackFormer[44]	74.1	68	47.3%	10.4%	2829	5.7
	OUTrack[45]	73.5	70.2	43.3%	15%	4122	25.9
	DETrack*	74.8	74.6	44.2%	15.8%	2445	24

proach for addressing the feature map optimization conflict issue in One-shot MOT algorithms. The algorithm incorporates feature decomposition and enhancement modules, which effectively mitigate the problem of frequent identity switching in complex multi-object tracking scenarios, leading to improved MOT performance. Experimental results demonstrate that the DETrack algorithm outperforms state-of-the-art approaches on the challenging MOT16 and MOT17 datasets, showcasing its superior tracking performance. Additionally, compared to other One-shot MOT algorithms, the proposed DETrack algorithm achieves superior detection and tracking results.

Acknowledgments

This work was supported by the 2020 Program for Liaoning Excellent Talents (LNET) in University, the 2021 Shenyang Ligong University Research Team Innovation Project, SYLUTD202105, the Research Support Program for Inviting High-Level Talents grant of Shenyang Ligong University (1010147000825), and National Key Research and Development Program of China (2022YFC3302500).

References

[1] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," Int. J. Comput. Vis., vol.129, pp.3069–3087, 2021.

[2] Z. Lu, V. Rathod, R. Votel, and J. Huang, "RetinaTrack: Online single stage joint detection and tracking," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.14668–14678, 2020.

[3] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, Aug. 2020, Proceedings, Part XI 16, pp.107–122, Springer, 2020.

[4] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," Proc. European Conference on Computer Vision (ECCV), Glasgow, UK, pp.474–490, 2020.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, vol.28, 2015.

[6] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," arXiv preprint arXiv:1904.07850, 2019.

[7] E. Yu, Z. Li, S. Han, and H. Wang, "RelationTrack: Relation-aware multiple object tracking with decoupled representation," IEEE Trans. Multimedia, vol.25, pp.2686–2697, 2023.

[8] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," European Conference on Computer Vision, pp.17–35, Springer, 2016.

[9] A. Milan, L. Leal-Taix'e, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," arXiv preprint arXiv:1603.00831, 2016.

[10] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), pp.3464–3468, IEEE, 2016.

[11] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," EURASIP Journal on Image and Video Processing, vol.2008, pp.1–10, 2008.

[12] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," arXiv preprint arXiv:1504.01942, 2015.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), pp.3645–3649, IEEE, 2017.

[14] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, Oct. 2022, Proceedings, Part XXII, pp.1–21, Springer, 2022.

[15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.7132–7141, 2018.

[16] S. Woo, J. Park, J.Y. Lee, and I.S. Kweon, "CBAM: Convolutional block attention module," Proc. European Conference on Computer Vision (ECCV), pp.3–19, 2018.

[17] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11534–11542, 2020.

[18] M.H. Guo, C.Z. Lu, Z.N. Liu, M.M. Cheng, and S.M. Hu, "Visual attention network," arXiv preprint arXiv:2202.09741, 2022.

[19] H. Fu, G. Song, and Y. Wang, "Improved YOLOv4 marine target detection combined with CBAM," Symmetry, vol.13, no.4, p.623, 2021.

[20] A. Bochkovskiy, C.Y. Wang, and H.Y.M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[21] X. Jiang, H. Hu, X. Liu, R. Ding, Y. Xu, J. Shi, Y. Du, and C. Da, "A smoking behavior detection method based on the YOLOv5 network," J. Phys.: Conf. Ser., vol.2232, p.012001, IOP Publishing, 2022.

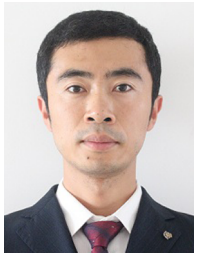
[22] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," Proc. IEEE International Conference on Computer Vision, pp.2980–2988, 2017.

[23] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.7482–7491, 2018.

[24] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, Oct. 2012, Proceedings, Part IV 12, pp.215–230, Springer, 2012.

[25] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse

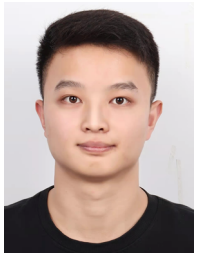
dataset for pedestrian detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3213–3221, 2017.
[26] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp.304–311, IEEE, 2009.
[27] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3415–3424, 2017.
[28] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1367–1376, 2017.
[29] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "CrowdHuman: A benchmark for detecting human in a crowd," arXiv preprint arXiv:1805.00123, 2018.
[30] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.466–475, IEEE, 2018.
[31] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," 2018 24th International Conference on Pattern Recognition (ICPR), pp.1809–1814, IEEE, 2018.
[32] N. Mahmoudi, S.M. Ahadi, and M. Rahmati, "Multi-target tracking using CNN-based features: CNNMTT," Multimed. Tools Appl., vol.78, pp.7077–7096, 2019.
[33] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple object tracking with high performance detection and appearance feature," Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, Oct. 2016, Proceedings, Part II 14, pp.36–42, Springer, 2016.
[34] C. Liang, Z. Zhang, X. Zhou, B. Li, S. Zhu, and W. Hu, "Rethinking the competition between detection and reid in multiobject tracking," IEEE Trans. Image Process., vol.31, pp.3182–3196, 2022.
[35] C. Liang, Z. Zhang, X. Zhou, B. Li, and W. Hu, "One more check: making "fake background" be tracked again," Proc. AAAI Conference on Artificial Intelligence, vol.36, no.2, pp.1546–1554, 2022.
[36] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, Aug. 2020, Proceedings, Part IV 16, pp.145–161, Springer, 2020.
[37] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.6308–6318, 2020.
[38] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, Aug. 2020, Proceedings, Part IV, pp.474–490, Springer, 2020.
[39] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon, "Learning to track with object permanence," Proc. IEEE/CVF International Conference on Computer Vision, pp.10860–10869, 2021.
[40] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, Oct. 2016, Proceedings, Part II 14, pp.84–99, Springer, 2016.
[41] X. Wan, J. Wang, Z. Kong, Q. Zhao, and S. Deng, "Multi-object tracking using online metric learning with long short-term memory," 2018 25th IEEE International Conference on Image Processing (ICIP), pp.788–792, IEEE, 2018.
[42] R. Mostafa, H. Baraka, and A. Bayoumi, "LMOT: Efficient lightweight detection and tracking in crowds," IEEE Access, vol.10, pp.83085–83095, 2022.
[43] J. Hyun, M. Kang, D. Wee, and D.Y. Yeung, "Detection recovery in online multi-object tracking with sparse graph tracker," Proc. IEEE/CVF Winter Conference on Applications of Computer Vision, pp.4850–4859, 2023.
[44] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.8844–8854, 2022.
[45] Q. Liu, D. Chen, Q. Chu, L. Yuan, B. Liu, L. Zhang, and N. Yu, "Online multi-object tracking with unsupervised re-identification learning and occlusion estimation," Neurocomputing, vol.483, pp.333–347, 2022.



Feng Wen received his B.S. degree from the Jilin University of Technology, China, in 1999, and his M.S. degree from Northeastern University, China, in 2005. He received his Ph.D. degree from Waseda University, Japan, in 2010. He is currently an professor at the Graudate College, Shenyang Ligong University, China.

Haixin Huang received her B.S. degree from the Northeastern University, China, in 1995, and her M.S. degree from Northeastern University, China, in 2001. She received her Ph.D. degree from Northeastern University, China, in 2012. She is currently an professor at the School of Automation and Electrical Engineering, Shenyang Ligong University, China.

Xiangyang Yin received his B.S. degree from the Heilongjiang University, China, in 2017, and his M.S. degree from Shenyang Ligong University, China, in 2023. He is currently an algorithm engineer in Lalamove company.



Junguang Ma received his B.S. degree from the Shandong Agricultural University, China, in 2022, he is currently a postgraduate student at Shenyang Ligong University. His research interests include object detection, semantic segmentation and few shot learning.

Xiaojie Hu received his B.A. degree from the Portsmouth University, UK, and his M.A. degree from Western Michigan University, USA, in 2011. He received his Ph.D. degree from Western Michigan University, USA, in 2018. He is currently a lecturer at School of Information Science and Engineering, Shenyang Ligong University, China.