IAMPDNet: Instance-aware and Multi-part Decoupled Network for Joint Detection and Embedding

Pan Yang[1], Xiong Luo[1,2,3*], Jiankun Sun[1]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, China
[2] Shunde Innovation School, University of Science and Technology Beijing, China
[3] Beijing Key Laboratory of Knowledge Engineering for Materials Science, China
G20208845@xs.ustb.edu.cn, xluo@ustb.edu.cn, B20170330@xs.ustb.edu.cn

Abstract

Video surveillance applications play an important role in smart cities. Recently, intelligent video surveillance methods have been widely investigated to address large-scale video data, among which multi-object tracking (MOT) is the most popular method, which aims to track every object appearing in the video for monitoring. MOT is essential for deep intelligent perception. Considering inference speed, joint detection and embedding (JDE) has become a new paradigm for MOT. JDE is to obtain detection results and object features through one forward propagation. However, most JDE models lack instance awareness ability and multi-part feature extraction ability, which may lead to the lack of discrimination of extracted instance features. To address these problems, in this paper we propose an instance-aware and multi-part decoupled network (IAMPDNet), which can perceive all instances in the environment and extract multi-part features from the instances. Specifically, our IAMPDNet consists of three key modules: a complementary attention module used to perceive all instances in the environment, a feature extraction module used to decouple multi-part features from the instances, and an adaptive aggregation module used to fuse multi-level features of instances. Extensive experiments on multiple MOT benchmarks demonstrate that our IAMPDNet achieves higher tracking accuracy and lower identity switches against recent MOT methods.

Keywords: Video surveillance applications, Joint Detection and Embedding (JDE), Instance awareness, Multi-part feature decouple

1 Introduction

Multi-object tracking (MOT) aims to detect all objects and track them automatically in a period of continuous video frames, which has great application significance in intelligent video surveillance. In the past, with the rapid development of object detection models, MOT usually followed the paradigm of tracking-by-detection [1-3]. This approach is in high accuracy but low inference speed. Considering the demand for real-time computation, joint detection and embedding (JDE) has become a new paradigm [4-7].

Generally speaking, JDE aims to integrate object detection and feature extraction into one model, and then the detected boxes and object features can be obtained after one forward propagation. Typical JDE models include FairMOT [4], CenterTrack [5], and TRMOT [6], all of which follow a similar framework, as shown in Figure 1. Given an image frame, generic visual features are extracted by a strong backbone, and multiple heads perform object box detection and feature extraction on generic visual features separately. Although this framework has achieved a good trade-off between speed and accuracy, it still has two limitations. 1) There is a lack of instance awareness ability. Each extracted instance feature only depends on the corresponding instance but does not interact with other instance features. Then, the extracted instance features are usually not optimal. 2) There is a lack of multi-part feature extraction ability. Each instance feature is extracted according to the overall appearance of the instance, and therefore the extracted instance features are not fine-grained enough.

To extract more discriminative instance features, the above two limitations of JDE are expected to be addressed. Therefore, three key modules are proposed in this paper to remedy them. 1) A complementary attention module is proposed for modeling the interaction between instance features. This is beneficial to perceive other instances in the environment for each instance. 2) A module is presented for extracting multi-part features of an instance. This module can offer more fine-grained instance features. 3) An adaptive aggregation module is developed for fusing multi-part features of an instance. This module can make better use of both the overall features and multi-part features of the instances.

Then, our contributions can be summarized as follows.

1) We propose an instance-aware and multi-part decoupling network (IAMPDNet), which can make up for the lack of instance feature interaction and the lack of fine-grained features in the joint detection and embedding models.

2) We propose a complementary attention module for perceiving all instances in the environment.

3) Our proposed multi-part feature extraction module can offer more fine-grained instance features.

4) We conduct extensive experiments on multiple MOT challenges [8], verifying the effectiveness of our proposed IAMPDNet for multi-object tracking.

The remainder of this paper is organized as follows. Section 2 overviews some recent works on JDE. The proposed method is described in Section 3. Our experimental results are presented in Section 4. The conclusion of this paper is provided in Section 5.

*Corresponding Author: Xiong Luo; E-mail: xluo@ustb.edu.cn
DOI: 10.53106/160792642022112306024

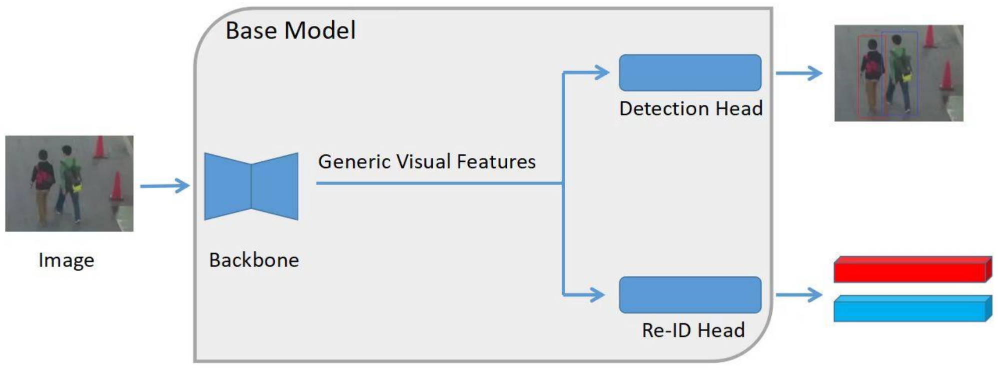

Figure 1. The general framework for joint detection and embedding

2 Related Work

2.1 Tracking-by-detection

Tracking-by-detection [9-11] models regard object detection and instance feature extraction as two independent tasks. Firstly, an object detection model is used to detect all objects, then the detected object areas are cropped and inputted into the instance features extraction networks. Deepsort [3] is a classic tracking-by-detection model, which detects the objects in the image by an object detector firstly and then adopts a deep network to extract instance features. MAT [11] is an enhanced MOT framework, which introduces a motion localization module, a dynamic reconnection context module, and a 3D integral image module to perform multi-object tracking. MOTR [12] introduces a tracking query into DETR [13] to better model the tracked objects in the input video. TrackFormer [14] treats MOT tasks as a frame-to-frame set prediction problem, and an end-to-end MOT framework was proposed using an encoder-decoder network. However, the tracking-by-detection models cannot meet the real-time requirement generally in some cases.

2.2 Joint Detection and Embedding

JDE models integrate object detection and feature extraction into a single model, which greatly improves the inference speed. While some advancements have been made in the applications of various deep learning models [15-17], then the recently developed JDE models also achieve some good performance. Track-RCNN [7] adds a feature extraction head on top of the backbone network and regresses a bounding box and an instance feature for each proposal. Similarly, TRMOT [6] is built on top of YOLOv3 [18] and greatly improves the inference speed. FairMOT [4] is a classic JDE model based on the CenterNet [5]. FairMOT is trained on six public datasets and has achieved state-of-the-art performance on MOT17 [8]. GTREID [19] builds a graph neural network and combines detection and association along with a re-identification feature for multi-object tracking. TGraM [20] rethinks MOS tasks from the multi-task learning perspective and models MOT as a graph information reasoning. However, most of the above JDE models cannot perceive other instance features for an instance feature and cannot offer more fine-grained features.

3 Methodology

3.1 Overview

For MOT, most JDE models cannot perceive other instance features for an instance feature and cannot extract fine-grained instance features. Our method tackles the above limitations by integrating three key modules, including a complementary attention module, a multi-part feature extraction module, and an adaptive aggregation module. The whole pipeline of our method is shown in Figure 2.

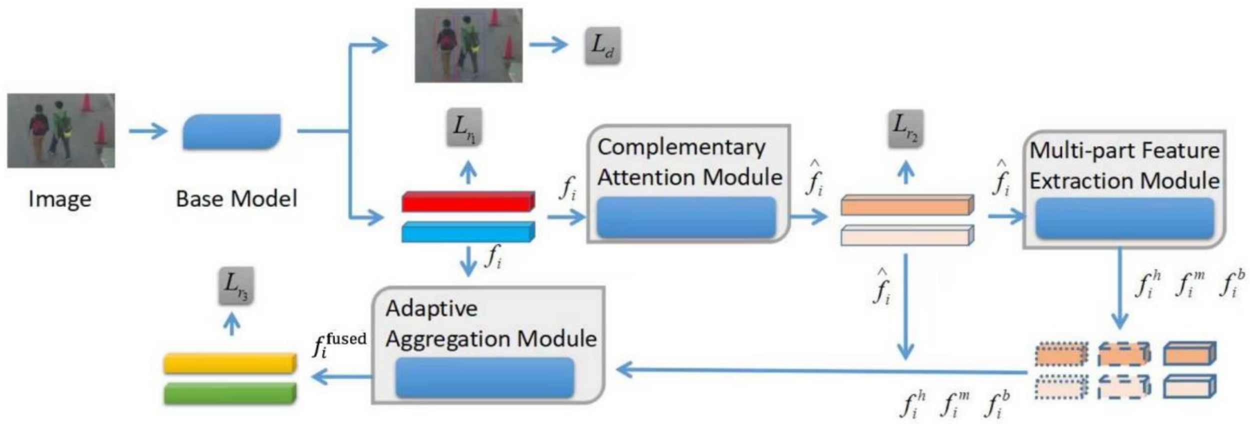

Figure 2. The architecture of our proposed IAMPDNet, which consists of a base model and three key additive modules: 1) complementary attention module, 2) multi-part feature extraction module, and 3) adaptive aggregation module

3.2 Base Model

Considering feature misalignment introduced by the anchor-based detector and the superior performance of FairMOT [4], we follow FairMOT to build our base model. As shown in Figure 1, our base model consists of a backbone, a detection head, and a re-identification (ReID) head. We adopt DLASeg [5] as our backbone for extracting generic visual features. DLASeg adopts ResNet-34 and an enhanced deep layer aggregation (DLA) to extract and fuse multi-scale features, and it is beneficial to extract more semantic features and achieve a good balance between accuracy and speed. In DLASeg, deformable convolution is used to replace all up-sampling layers, adjusting the receptive field dynamically according to the size of objects. The detection head consists of a heatmap head, a box size head and a center offset head. All heads are implemented via several convolution layers. In addition, the loss function L_d and L_{r_1} are utilized to jointly supervise the training process of the base model, as shown in Figure 2.

3.3 The Complementary Attention Module

The attention mechanism can effectively aggregate similar features and is not limited by the receptive field, thus it is widely used in the field of computer vision. However, the attention mechanism directly discards dissimilar features, which are valuable in some scenarios. For example, in MOT, dissimilar features also contribute to the extraction of instance features. Therefore, we propose the complementary attention module to aggregate similar and dissimilar instance features respectively. Using aggregated similar and dissimilar features, we can further refine the instance features extracted by the base model.

To model the interaction between instance features, we propose a complementary attention module on top of the base model, as shown in Figure 3. Given normalized instance features $\{f_i\}_{i=1}^{N}$ where N denotes the total number of instances, we firstly calculate cosine similarity $\mathrm{Sim} \in \mathbb{R}^{N\times N}$ between instance features as the formula below:

$$\mathrm{Sim}_{ij} = f_i^{\mathrm{T}} f_j, \tag{1}$$

where Sim_{ij} represents the cosine similarity between i-th and j-th instance features.

Then, an aggregation operator is performed for perceiving all instance features in the environment and the i-th aggregated instance feature f_i^1 can be calculated by:

$$f_i^1 = \sum_{j=1}^{N} \mathrm{Sim}_{ij} f_j. \tag{2}$$

It is noted that when Sim_{ij} is close to 0, the aggregated instance features f_i^1 is hardly affected by the j-th normalized instance features. However, all instance features should be perceived and thus an improved version is proposed here. Specifically, we calculate the complementary aggregated instance features f_i^2 by:

$$f_i^2 = \sum_{j=1}^{N} \phi\left(\mathrm{Sim}_{ij}\right) f_j, \tag{3}$$

where $\phi(*)$ is a transform function defined as:

$$\phi(x) = \begin{cases} 1-x, & x>0, \\ x-1, & \text{otherwise.} \end{cases} \tag{4}$$

Due to the complementary aggregated instance features, all instance features can be perceived. Then the final instance feature $\hat{f}_i$ can be calculated via the concatenation of the original instance feature f_i, the aggregated instance feature f_i^1 and the complementary aggregated instance feature f_i^2. Specifically, the following equation is given for the final instance feature:

$$\hat{f}_i = \Phi(f_i \oplus f_i^1 \oplus f_i^2), \tag{5}$$

where $\Phi(*)$ is implemented by two fully connected layers, and $\oplus$ denotes the concatenation operation. Here, the instance features $\hat{f}_i$ is supervised by the cross entropy loss L_{r_2}.

3.4 The Multi-part Feature Extraction Module

The fine-grained features are beneficial to improving the accuracy of feature matching, which is critical for MOT. To extract features from different parts, spatial attention mechanisms are usually employed. The spatial attention mechanism can effectively make the neural network pay attention to different parts of an instance. However, the spatial attention mechanism has a large amount of calculation and is difficult to meet the real-time requirements. Thus, we propose the multi-part feature extraction module to extract features from different parts. This module is lightweight and utilizes an extra person re-identification model for supervision, which makes our models focus on different parts of each instance. Furthermore, in the inference stage, the person re-identification model can be discarded without incurring additional inference time.

To offer more fine-grained instance features, we propose a multi-part feature extraction module, which can extract the head part, middle part, and bottom part of an instance separately. As shown in Figure 2, the multi-part feature extraction module is built on top of the complementary attention module. Given aggregated instance features $\{\hat{f}_i\}_{i=1}^{N}$, three decoupled heads are used to extract the head part f_i^h, middle part f_i^m and bottom part f_i^b of an instance respectively by:

$$f_i^h = \Phi_h(\hat{f}_i), \tag{6}$$

$$f_i^m = \Phi_m(\hat{f}_i), \tag{7}$$

$$f_i^b = \Phi_b(\hat{f}_i), \tag{8}$$

where $\Phi_h(*)$, $\Phi_m(*)$ and $\Phi_b(*)$ are implemented by fully connected layers.

In order to focus our model's attention on different parts of an instance, we use a person re-identification model [21] for deep supervision, as shown in Figure 4.

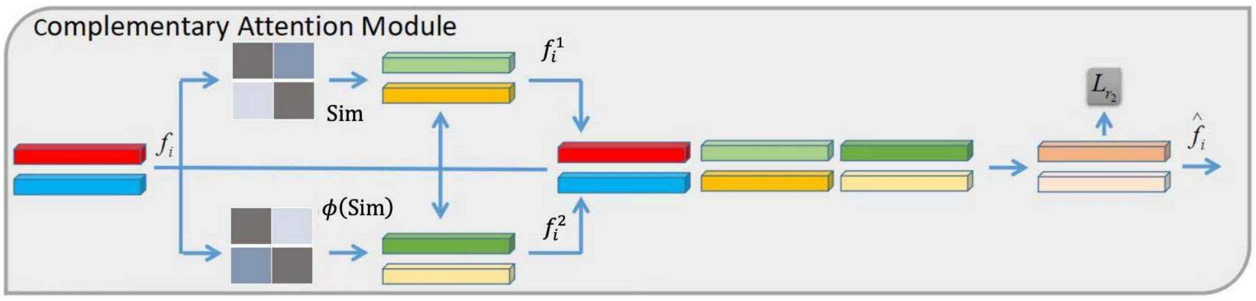

Figure 3. Details of the complementary attention module

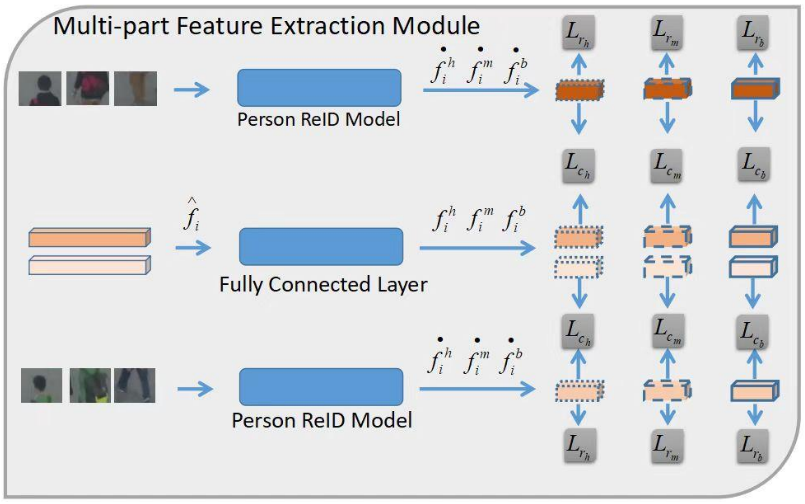
Multi-part Feature Extraction Module
Person ReID Model
Fully Connected Layer
Person ReID Model
$\dot{f}_i^h$ $\dot{f}_i^m$ $\dot{f}_i^b$
$\hat{f}_i$
f_i^h f_i^m f_i^b
L_{r_h} L_{r_m} L_{r_b}
L_{c_h} L_{c_m} L_{c_b}

Figure 4. Details of the multi-part feature extraction module

Note. We use a person ReiD model to auxiliary supervise the training. Specifically, each instance region is cropped into three parts, which are passed through a person ReID model to generate fine-grained features of three parts for auxiliary supervision.

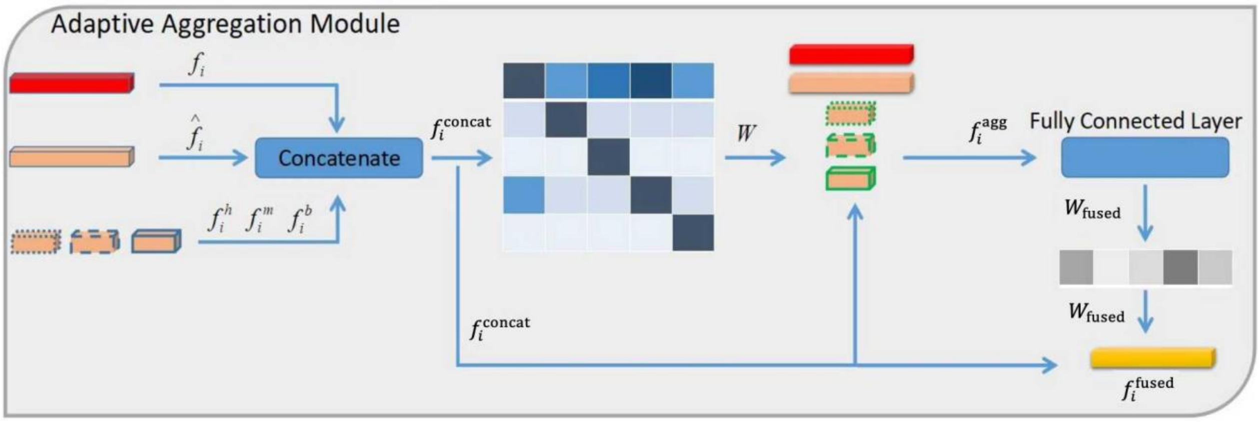

Figure 5. Details of the adaptive aggregation module

We feed different parts of an instance into the person re-identification model, and then use the cosine loss function L_{c_h}, L_{c_m} and L_{c_b} to supervise the features $\dot{f}_i^h$, $\dot{f}_i^m$ and $\dot{f}_i^b$ extracted by the person re-identification model and the features f_i^h, f_i^m and f_i^b extracted by our multi-part feature extraction module:

$$L_{c_h} = -\frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Norm}\left(f_i^h\right)^{\mathrm{T}}\mathrm{Norm}\left(\dot{f}_i^h\right)\right), \quad \textbf{(9)}$$

$$L_{c_m} = -\frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Norm}(f_i^m)^{\mathrm{T}}\mathrm{Norm}(\dot{f}_i^m)\right), \quad \textbf{(10)}$$

$$L_{c_b} = -\frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Norm}\left(f_i^b\right)^{\mathrm{T}}\mathrm{Norm}\left(\dot{f}_i^b\right)\right), \quad \textbf{(11)}$$

where Norm$(*)$ denotes the normalization operation. Furthermore, we use L_{r_h}, L_{r_m} and L_{r_b} to supervise the personal re-identification model.

3.5 The Adaptive Aggregation Module

To fuse various features, the adaptive aggregation methods can adaptively generate different weights according to the importance of various features, which is more efficient than manually searching weights. To this end, we propose the adaptive aggregation module. It is worth noting that our method performs a self-attention operation before outputting the weights, which strengthens the interaction between the features from different levels and aggregates rich information for generating the weights.

Through the above complementary attention module and multi-part feature extraction module, we can obtain multi-level features f_i, $\hat{f}_i$, f_i^h, f_i^m and f_i^b. How to utilize these features for better tracking performance? An intuitive approach is to directly perform a weighted average, which lacks the interaction between multi-level features and requires manual assignment for weights. Therefore, we propose an adaptive aggregation module shown in Figure 5.

Specifically, given normalized multi-level features f_i, $\hat{f}_i$, f_i^h, f_i^m and f_i^b, we concatenate them as $f_i^{\text{concat}} \in \mathbb{R}^{5\times C}$, where C denotes the feature dimension:

$$f_i^{\text{concat}} = f_i \oplus \hat{f}_i \oplus f_i^h \oplus f_i^m \oplus f_i^b. \quad \textbf{(12)}$$

Then, a cosine similarity $W \in \mathbb{R}^{5\times 5}$ between multi-level features can be calculated by:

$$W = f_i^{\text{concat}} f_i^{\text{concat}^{\mathrm{T}}}. \quad \textbf{(13)}$$

In addition, an aggregated feature can be calculated through the aggregation operation:

$$f_i^{\text{agg}} = f_i^{\text{concat}} + W f_i^{\text{concat}}. \quad \textbf{(14)}$$

Finally, the multi-level features are adaptively fused by:

$$W_{\text{fused}} = \text{Softmax}\left(\Phi_w\left(f_i^{\text{agg}}\right)\right), \quad \textbf{(15)}$$

$$f_i^{\text{fused}} = {W_{\text{fused}}}^{\mathrm{T}} f_i^{\text{concat}}, \quad \textbf{(16)}$$

where $\Phi_w(*)$ is implemented by three fully connected layers. It is noted that the calculation of f_i^{fused} is supervised by the cross entropy loss L_{r_3} here. Moreover, f_i^{concat} contains multi-level instance features and f_i^{agg} aggregates different levels of features for each level of features, which provides rich information to generate the importance weights of each level of features. In (15), three full connection layers and a Softmax operation are implemented on f_i^{agg} to generate importance weights of each level of features. Then, f_i^{concat} containing multi-level features can be efficiently aggregated by using generated importance weights in (16).

3.6 Training Loss

The training loss L_{t_1} and L_{t_2} for supervising our IAMPDNet are defined as:

$$L_{t_1} = \frac{1}{2}\left(\frac{1}{e^{w_1}} L_d + \frac{1}{e^{w_2}}\left(L_{r_1} + L_{r_2}\right) + w_1 + w_2\right), \quad \textbf{(17)}$$

$$\begin{aligned} L_{t_2} = {} & L_{r_1} + \left(L_{r_2} + L_{r_h} + L_{r_m} + L_{r_b}\right) w_r \\ & + L_{r_3} w_f + \left(L_{c_h} + L_{c_m} + L_{c_b}\right) w_c, \end{aligned} \quad \textbf{(18)}$$

where w_1 and w_2 are the trainable parameters to balance the detection task and the re-identification task. Moreover, w_r, w_f and w_c are hyper-parameters to weight different loss. All $L_{r\_}$ are the cross entropy loss. The definition of L_d is the same as FairMOT [4].

In multi-task learning, deep neural networks usually learn different tasks simultaneously. The weights of different tasks have a very large impact on the results. Thus, how to assign the weights is an important issue. Then, an uncertain method was proposed to learn the weights [22], and it is widely used in MOT. In (17), we also adopt the uncertain method to balance the object detection task and the re-identification task by two trainable parameters w_1 and w_2.

The network weight update process of our proposed method is described in Algorithm 1. In Line 1, the base model detects all objects in the input image and extracts corresponding features. To generate more discriminative features, the complementary attention module and the multi-part feature extraction module are utilized to aggregate features and refine features in Lines 2-3. Then, multi-level features should be fused for subsequent feature matching. In Line 4, the adaptive aggregation module can adaptively search weights of different features and fuse them. To update the weights of the network, the training losses are calculated in Lines 5-9. Finally, we can calculate gradients and update parameters by back-propagation algorithm in Line 10.

Algorithm 1. Network weights update process

```
Input: image $x$.
Output: updated network weights $w_{\text{network}}$.
1. Generate boxes $\{\text{box}_i\}_{i=1}^N$ and instance features $\{f_i\}_{i=1}^N$ of detected objects in image $x$ by the base model;
2. Generate aggregated instance features $\{\hat{f}_i\}_{i=1}^N$ by the complementary attention module;
3. Generate fine-grained instance features $\{f_i^h\}_{i=1}^N$, $\{f_i^m\}_{i=1}^N$ and $\{f_i^b\}_{i=1}^N$ by the multi-part feature extraction module;
4. Generate fused instance features $\{f_i^{\text{fused}}\}_{i=1}^N$ by the adaptive aggregation module;
5. Calculate loss $L_d$ and $L_{r_1}$ using $\{\text{box}_i\}_{i=1}^N$ and $\{f_i\}_{i=1}^N$;
6. Calculate loss $L_{r_2}$ using $\{\hat{f}_i\}_{i=1}^N$;
7. Calculate loss $L_{r_h}$, $L_{r_m}$ and $L_{r_b}$ using $\{f_i^h\}_{i=1}^N$, $\{f_i^m\}_{i=1}^N$ and $\{f_i^b\}_{i=1}^N$;
8. Calculate loss $L_{c_h}$, $L_{c_m}$ and $L_{c_b}$ using (9)-(11);
9. Calculate total loss $L_{t_1}$ and $L_{t_2}$ using (17) and (18);
10. Update network weights $w_{\text{network}}$ using $L_{t_1}$ and $L_{t_2}$;
11. Return $w_{\text{network}}$.
```

4 Experiments

4.1 Experimental Settings and Metrics

All experiments are implemented with the Pytorch framework and Python 3.7.3 on GeForce GTX 1080 Ti GPUs. Specifically, the Adam optimizer is utilized to train our model with 50 epochs. The initial learning rate is set as 0.0001, and it reduces in steps to 0.00001 at the 20th epoch. The batch size is 4. In addition, w_1 and w_2 are initialized to -1.85 and -1.05, respectively. Moreover, w_r, w_f, and w_c are set as 0.1, 0.01, and 0.01.

Due to the authority and comprehensiveness of the CLEAR metrics [23], they are often used to evaluate MOT algorithms. Following those works in relation to MOT [4-6], we also adopt some common CLEAR metrics to evaluate our proposed method.

Several common CLEAR metrics [23], including multi-object tracking accuracy (MOTA), identity F1 score (IDF1), mostly tracked (MT), mostly lost (ML), identity switch (IDSW) and frame per second (FPS), are computed to evaluate the overall tracking performance.

In Table 1 to Table 9, the arrows appearing on the first row are defined as follows. The upward arrow indicates that the higher the value, the better the performance. The downward arrow indicates that the smaller the value, the better the performance.

4.2 Datasets

In this paper, we conduct experiments on two MOT challenges, i.e., MOT16 and MOT17 [8], to verify the effectiveness of our proposed IMAPDNet.

Like most methods, we firstly use half of the train splits in MOT17 for training. Then, the performance evaluation is conducted on the remaining train splits. Finally, extra supporting experiments for performance comparison are conducted on the test splits of MOT16 and MOT17.

4.3 Quantitative Results

In this section, we train our IMAPDNet only using train splits of MOT17 and compare the performance with the SOTA methods on the test splits of MOT16 and MOT17 [8]. As depicted in Table 1 and Table 2, our IAMPDNet achieves better performance in MOT16 and MOT17 compared with other methods.

4.3.1 Discriminative Instance Features

First, in our IAMPDNet, the complementary attention module can perceive all instances in the environment, which is beneficial to extracting more discriminative instance features due to the interaction between all instances.

Second, the multi-part feature extraction module can decouple instance features into three different parts, which provides more fine-grained features.

Third, the adaptive aggregation module can adaptively fuse multi-level features. Therefore, on test splits of MOT16 and MOT17, our IAMPDNet ranks 1 in IDF1 and ranks 2 in IDSW, which demonstrates that the instance features extracted by our IAMPDNet are more discriminative.

4.3.2 Tracking Accuracy

Due to discriminative instance features extracted by our IAMPDNet, better tracking accuracy has been achieved by our IMAPDNet. As shown in Table 1 and Table 2, our IAMPDNet ranks 1 in MOTA, IDF1, MT and ML and ranks 2 in IDSW.

Table 1. Performance comparison on test splits of MOT16 [8]

Method	Publication	Year	JDE	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
SORT [14]	ICIP	2016		0.598	0.538	0.254	0.227	1423
MCMOT-HDM [15]	ECCV	2016		0.624	0.516	0.315	0.242	1394
Vmaxx [24]	ICIP	2018		0.626	0.492	0.327	0.211	1389
TRMOT [6]	ECCV	2020	✓	0.644	0.558	0.354	0.200	1544
CtrackerV1 [25]	ECCV	2020	✓	0.676	0.572	0.329	0.231	1897
TraDeS [26]	CVPR	2021	✓	0.701	0.647	0.373	0.200	**1144**
IAMPDNet (Ours)			✓	**0.711**	**0.711**	**0.381**	**0.197**	1233

Table 2. Performance comparison on test splits of MOT17 [8]

Method	Publication	Year	JDE	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
SST [27]	PAMI	2019	✓	0.524	0.495	0.214	0.307	8431
CTrackerV1 [25]	ECCV	2020	✓	0.666	0.574	0.322	0.242	5529
CenterTrack [5]	ECCV	2020	✓	0.673	0.599	0.349	0.248	2898
FairMOT [4]	IJCV	2021	✓	0.698	0.699	-	-	3996
IAMPDNet (Ours)			✓	**0.701**	**0.703**	**0.372**	**0.215**	3803

Note. Here, '-' denotes that the result is not available from the corresponding paper.

4.3.3 Inference Speed

Table 3 lists the inference speed of our IAMPDNet and other methods. As shown in Table 3, our IAMPDNet can achieve 24.8 frames per second (FPS) while the fastest method FairMOT only achieves 25.9 FPS, which illustrates that our IAMPDNet infers faster than most of the methods and can meet real-time requirements.

4.4 Ablation Analysis

We conduct extensive comparative experiments on the validation split of MOT17 [8] to verify the effectiveness of our IMAPDNet from three aspects: the complementary attention module (CAM), the multi-part feature extraction module (MFEM) and the adaptive aggregation module (AAM). As depicted in Table 4, we design 4 different experiment settings: 1) original FairMOT [4] without CAM, MFEM and AAM; 2) original FairMOT with CAM; 3) original FairMOT with CAM and MFEM; and 4) original FairMOT with CAM, MFEM and AAM.

4.4.1 The Complementary Attention Module

As shown in Table 4, adding CAM into original FairMOT [4] improves MOTA (0.685 vs 0.675), IDF1 (0.717 vs 0.699), and IDSW (364 vs 408). This is because the complementary attention module is beneficial to perceive other instances for each instance. As shown in Figure 6, the complementary attention module can aggregate similar instances and dissimilar instances. Accepting similar instance features and dissimilar instance features as input, our model can generate more discriminative instance features, and it is key to identity association and thus improve the performance of multi-object tracking.

4.4.2 The Multi-part Feature Extraction Module

Comparing experiment settings 2) and 3), it is obvious that adding MFEM into FairMOT with CAM furthermore decreases IDSW (358 vs 364), since the MFEM generates more fine-grained features for different parts of an instance. However, adding MFEM hinders the overall performance of tracking. Specifically, adding MFEM decreases MOTA (0.684 vs 0.685), IDF1 (0.708 vs 0.717), MT (144 vs 145) and increase ML (61 vs 60). This may be due to the lack of the adaptive aggregation module. Instead of the adaptive aggregation module, we simply average all features, which may lead to poor performance. To evaluate the effect of feature weights, we compare various feature weights combination w_{f_i}, $w_{\hat{f}_i}$, $w_{f_i^h}$, $w_{f_i^m}$ and $w_{f_i^b}$ in Table 9. The results show that different weights combination can lead to different performances and the average strategy indeed hinders the performance.

4.4.3 The Adaptive Aggregation Module

Comparing experiments setting 3) and 4), we can find that original FairMOT with CAM, MFEM and AAM achieves the better performance on MOTA (0.689 vs 0.684), IDF1 (0.722 vs 0.708), MT (150 vs 144), ML (60 vs 61) and IDSW (353 vs 358). Especially, IDSW is decreased from 408 (original FariMOT) to 353, which verifies the effectiveness of our AAM. To furthermore illustrate the role of AAM, we compare different weights combinations w_{f_i}, $w_{\hat{f}_i}$, $w_{f_i^h}$, $w_{f_i^m}$ and $w_{f_i^b}$. As shown in Table 9, the first row represents the weights combinations generated by our AAM, which achieves superior performance. The results show that the weights combinations are key to improving performance and searching weights manually is time-consuming and difficult to find suitable values.

4.4.4 Ablation Study on Loss Weight

To explore more loss weight combinations, we search different w_r, w_f, and w_c. The results are listed in Table 5, Table 6, and Table 7. The results show that when setting w_r, w_f and w_c as 0.1, 0.01 and 0.01, the best performance can be achieved.

4.4.5 Ablation Study on Backbones

To illustrate that our proposed modules can be applied in different backbones, we conducted experiments on extra two backbones like HRNet-W18 and ResNet-34-FPN. As shown in Table 8, using our proposed modules on HRNet-W18 and ResNet-34-FPN both improve the performance, which demonstrates that our proposed modules can be utilized in various backbones with better performance.

4.5 Qualitative Results

In this section, we use figures to more intuitively illustrate the role of the complementary attention module and the adaptive aggregation module.

4.5.1 Visualization of Complementary Attention Module

In order to illustrate the effect of the complementary attention module more directly, we visualize the cosine similarity Sim_i and $\phi(\mathrm{Sim}_i)$ of the i-th object in various scenes. In Figure 6(a), the i-th objects in three scenes are marked in red dots. In Figure 6(b), the cosine similarity Sim_i of the i-th object in three scenes are marked in blue. The complementary attention module can assign high weights to similar objects. In Figure 6(c), the cosine similarity $\phi(\mathrm{Sim}_i)$ of i-th object in three scenes are marked in yellow. The complementary attention module can assign high weights to dissimilar objects. Due to the complementary attention module, each object can perceive similar objects and dissimilar objects simultaneously, which is beneficial to generating more discriminative features.

4.5.2 Fusion Weights Visualization

To illustrate the role of the adaptive aggregation module, we analyze each image in validation sets. Specifically, for i-th instance in each image, we generate f_i, $\hat{f}_i$, f_i^h, f_i^m and f_i^b sequentially through the base model, the complementary attention module and the multi-part feature extraction module. Then, the importance weights W_{fused} of each level of features can be calculated according to (12)-(15). After generating all importance weights W_{fused} of images in validation sets, we average the weights W_{fused} and visualize it as shown in Figure 7. The visualization shows that the weights of f_i and $\hat{f}_i$ are almost three times the weights of f_i^h, f_i^m and f_i^b, which is reasonable since f_i and $\hat{f}_i$ are global features while f_i^h, f_i^m and f_i^b are local features used to assist the global features. The f_i and $\hat{f}_i$ have similar weights since they are both global features and assigning them similar weights can act as an ensemble model, achieving better results than using extreme weights as shown in Table 9.

As shown in Figure 7, the weights assigned to f_i, $\hat{f}_i$, f_i^h, f_i^m and f_i^b are 0.360, 0.330, 0.098, 0.110 and 0.098, and it is different from the simple average. We also compare different weight combinations w_{f_i}, $w_{\hat{f}_i}$, $w_{f_i^h}$, $w_{f_i^m}$ and $w_{f_i^b}$ to evaluate the effectiveness of our AAM. As shown in Table 9, f_i and $\hat{f}_i$ have dominant contributions for multi-object tracking. Increasing weights of f_i^h, f_i^m and f_i^b slightly can improve the performance, which verifies that the fine-grained features are beneficial to multi-object tracking.

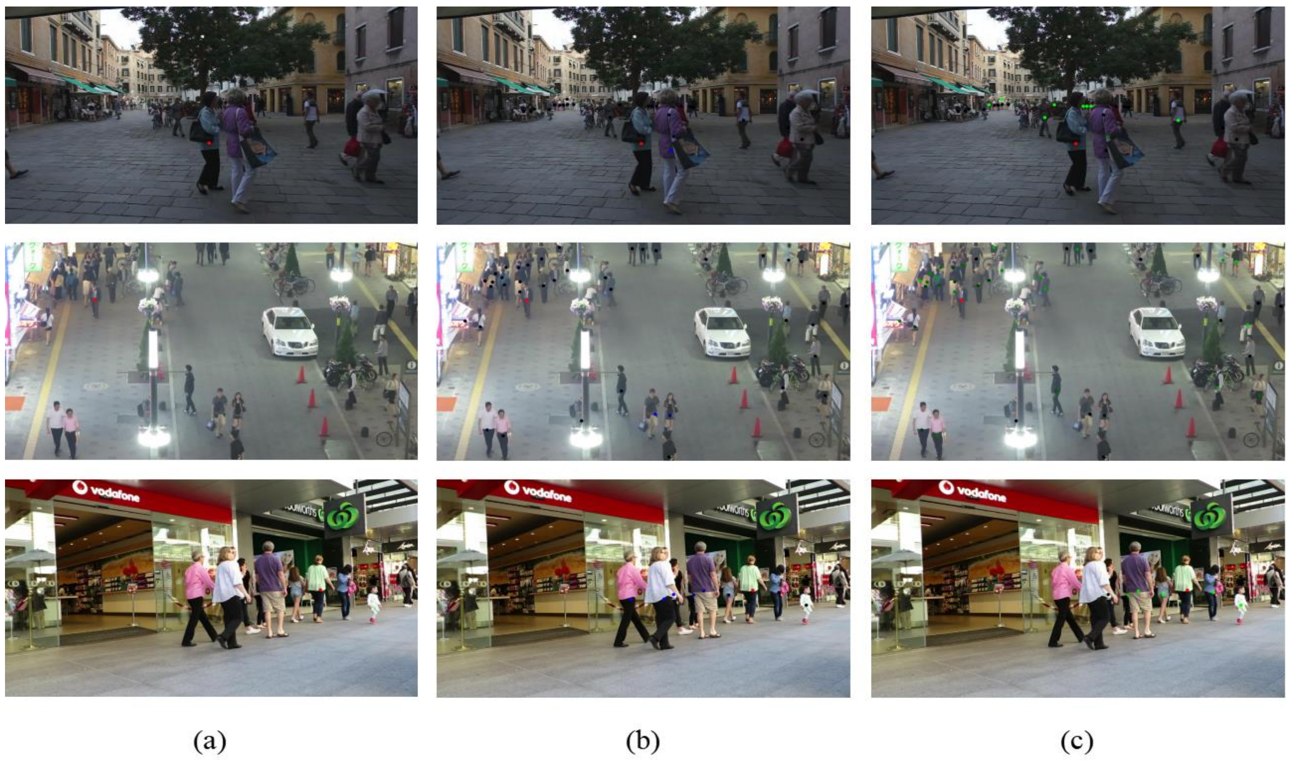

Figure 6. The visualization of cosine similarity Sim_i and $\phi(\text{Sim}_i)$ of i-th object in various scenes

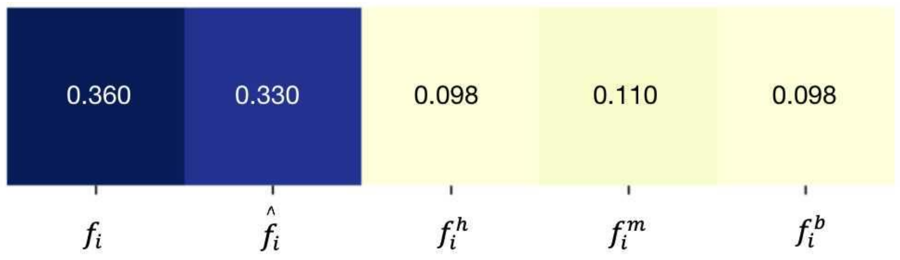

Figure 7. Fusion weights visualization

Table 3. Inference speed comparison

Method	Publication	Year	JDE	FPS↑
SST [27]	PAMI	2019	✓	3.9
CTrackerV1 [25]	ECCV	2020	✓	6.8
CenterTrack [5]	ECCV	2020	✓	22.0
TRMOT [6]	ECCV	2020	✓	22.2
FairMOT [4]	IJCV	2021	✓	**25.9**
TraDeS [18]	CVPR	2021	✓	17.5
MAT [21]	Neurocomputing	2022		9.0
IAMPDNet (Ours)			✓	24.8

Table 4. Ablation study on the validation splits of MOT17 [8]

CAM	MFEM	AAM	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
			0.675	0.699	-	-	408
✓			0.685	0.717	145	60	364
✓	✓		0.684	0.708	144	61	358
✓	✓	✓	**0.689**	**0.722**	**150**	**60**	**353**

Table 5. Ablation study on loss weight w_r

w_r	w_f	w_c	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
0.100	0.010	0.010	**0.689**	**0.722**	**150**	60	**353**
0.010	0.010	0.010	0.687	0.716	144	**59**	355
1.000	0.010	0.010	0.686	0.715	150	59	364

Table 6. Ablation study on loss weight w_f

w_r	w_f	w_c	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
0.100	0.010	0.010	**0.689**	**0.722**	**150**	**60**	**353**
0.100	0.001	0.010	0.688	0.717	145	60	368
0.100	0.100	0.010	0.685	0.711	147	**59**	383

Table 7. Ablation study on loss weight w_c

w_r	w_f	w_c	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
0.100	0.010	0.010	**0.689**	**0.722**	**150**	60	**353**
0.100	0.010	0.001	0.685	0.708	147	**58**	389
0.100	0.010	0.100	0.687	0.715	143	59	386

Table 8. Ablation study on backbones

Backbone	Using proposed module	MOTA↑	IDF1↑	MT↑	ML↓	IDSW ↓
HRNet-W18		0.595	0.634	102	77	533
HRNet-W18	✓	**0.611**	**0.661**	**124**	**66**	**519**
ResNet-34-FPN		0.582	0.636	118	69	570
ResNet-34-FPN	✓	**0.597**	**0.649**	**124**	**69**	**566**

Table 9. Ablation study on different weight combinations

w_{f_i}	$w_{\hat{f}_i}$	$w_{f_i^h}$	$w_{f_i^m}$	$w_{f_i^b}$	MOTA↑	IDF1↑	MT↑	ML↓	IDSW↓
0.360	0.330	0.098	0.110	0.098	**0.689**	0.722	**150**	60	**353**
1.000	0.000	0.000	0.000	0.000	0.687	**0.726**	143	60	414
0.000	1.000	0.000	0.000	0.000	0.687	0.720	142	59	394
0.000	0.000	0.330	0.330	0.330	0.688	0.695	146	58	434
0.500	0.500	0.000	0.000	0.000	0.688	0.724	143	60	396
0.450	0.450	0.030	0.030	0.030	0.688	0.723	143	60	390
0.200	0.200	0.200	0.200	0.200	0.688	0.710	145	**57**	410
0.100	0.100	0.260	0.260	0.260	0.689	0.703	145	58	422
0.050	0.050	0.300	0.300	0.300	0.689	0.701	146	58	416

5 Conclusion

In this paper, we propose a novel model IAMPDNet with three key modules for MOT. Firstly, a complementary attention module is designed to model the interaction between instance features, which remedies the lack of feature interaction ability in JDE models. Secondly, a multi-part feature extraction module is proposed to extract more fine-grained instance features. Finally, an adaptive aggregation module is designed to interact between multi-level features and adaptively fuse them. Compared with state-of-the-art methods in MOT16 and MOT17 [8], our IAMPDNet ranks 1 on MOTA, IDF1, MT, ML and ranks 2 on IDSW. The ablation study and qualitative results demonstrate that: 1) the complementary attention module is beneficial to perceive other instance features for an instance feature; 2) the multi-part feature extraction module can extract more fine-grained instance features and the adaptive aggregation module can use multi-level instance features better than the weighted average.

However, our proposed method in this paper is only trained on a single image and does not utilize temporal information. In the future, a transformer model can be used to extract temporal features, and then we extend 2D MOT to 3D MOT method.

Acknowledgments

This work was supported in part by the Beijing Natural Science Foundation under Grants 19L2029 and L211020, in part by the National Natural Science Foundation of China under Grant U1836106, and in part by the Scientific and Technological Innovation Foundation of Foshan under Grants BK21BF001 and BK20BF010.

References

[1] J. Shi, C. Tomasi, Good Features to Track, *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 1994, pp. 593-600.

[2] L. Ren, J. Lu, Z. Wang, Q. Tian, J. Zhou, Collaborative Deep Reinforcement Learning for Multi-object Tracking, *European Conference on Computer Vision*, Munich, Germany, 2018, pp. 605-621.

[3] H. Gao, K. Xu, M. Cao, J. Xiao, Q. Xu, Y. Yin, The Deep Features and Attention Mechanism-Based Method to Dish Healthcare Under Social IoT Systems: An Empirical Study With a Hand-Deep Local-Global Net, *IEEE Transactions on Computational Social Systems*, Vol. 9, No. 1, pp. 336-347, February, 2022.

[4] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking, *International Journal of Computer Vision*, Vol. 129, No. 11, pp. 3069-3087, November, 2021.

[5] X. Zhou, V. Koltun, P. Krähenbühl, Tracking Objects as Points, *European Conference on Computer Vision*, Glasgow, United Kingdom, 2020, pp. 474-490.

[6] Z. Wang, L. Zheng, Y. Liu, Y. Li, S. Wang, Towards Real-time Multi-object Tracking, *European Conference on Computer Vision*, Glasgow, United Kingdom, 2020, pp. 107-122.

[7] J. Xiao, H. Xu, H. Gao, M. Bian, Y. Li, A Weakly Supervised Semantic Segmentation Network by Aggregating Seed Cues: The Multi-Object Proposal Generation Perspective, *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 17, No. 1s, pp. 1-19, January, 2021.

[8] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler, MOT16: A Benchmark for Multi-object Tracking, May, 2016. https://arxiv.org/abs/1603.00831

[9] N. Wojke, A. Bewley, D. Paulus, Simple Online and Realtime Tracking With a Deep Association Metric, *IEEE International Conference on Image Processing*, Beijing, China, 2017, pp. 3645-3649.

[10] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, P. K. Rhee, Multi-class Multi-object Tracking Using Changing Point Detection, *European Conference on Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 68-83.

[11] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, X. Pan, MAT: Motion-aware Multi-object Tracking, *Neurocomputing*, Vol. 476, pp. 75-86, March, 2022.

[12] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, Y. Wei, MOTR: End-to-End Multiple-object Tracking With Transformer, *European Conference on Computer Vision*, Tel Aviv, Israel, 2022.

[13] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, L. Zhang, Dynamic DETR: End-to-End Object Detection With Dynamic Attention, *IEEE/CVF International Conference on Computer Vision*, Virtual, Online, Canada, 2021, pp. 2988-2997.

[14] T. Meinhardt, A. Kirillov, L. Leal-Taixe, C. Feichtenhofer, TrackFormer: Multi-object Tracking With Transformers, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 2022, pp. 8844-8854.

[15] X. Luo, J. Sun, L. Wang, W. Wang, W. Zhao, J. Wu, J. H. Wang, Z. Zhang, Short-term Wind Speed Forecasting via Stacked Extreme Learning Machine With Generalized Correntropy, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 11, pp. 4963-4971, November, 2018.

[16] X. Luo, J. Li, M. Chen, X. Yang, X. Li, Ophthalmic Diseases Detection via Deep Learning With a Novel Mixture Loss Function, *IEEE Journal of Biomedical and Health Informatics*, Vol. 25, No. 9, pp. 3332-3339, September 2021.

[17] H. Gao, J. S. Xiao, Y. Y. Yin, T. Liu, J. G. Shi, A Mutually Supervised Graph Attention Network for Few-shot Segmentation: The Perspective of Fully Utilizing Limited Samples, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-13, 2022, doi: 10.1109/TNNLS.2022.3155486.

[18] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, April, 2018. https://arxiv.org/abs/1804.02767

[19] C. Lusardi, A. M. N. Taufique, A. Savakis, Robust Multi-object Tracking Using Re-identification Features and Graph Convolutional Networks, *IEEE/CVF International Conference on Computer Vision*, Montreal, BC, Canada, 2021, pp. 3868-3877.

[20] Q. He, X. Sun, Z. Yan, B. Li, K. Fu, Multi-object Tracking in Satellite Videos With Graph-based Multitask Modeling, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 60, pp. 1-13, February, 2022, doi: 10.1109/TGRS.2022.3152250.

[21] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, J. Gu, A Strong Baseline and Batch Normalization Neck for Deep Person Re-identification, *IEEE Transactions on Multimedia*, Vol. 22, No. 10, pp. 2597-2609, October, 2020.

[22] A. Kendall, Y. Gal, R. Cipolla, Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7482-7491.

[23] K. Bernardin, R. Stiefelhagen, Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics, *EURASIP Journal on Image and Video Processing*, Vol. 2008, Article No. 246309, 2008.

[24] X. Wan, J. Wang, Z. Kong, Q. Zhao, S. Deng, Multi-object Tracking Using Online Metric Learning With

Long Short-term Memory, *IEEE International Conference on Image Processing*, Athens, Greece, 2018, pp. 788-792.

[25] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Fu, Chained-tracker: Chaining Paired Attentive Regression Results for End-to-End Joint Multiple-object Detection and Tracking, *European Conference on Computer Vision*, Glasgow, United Kingdom, 2020, pp. 145-161.

[26] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, J. Yuan, Track to Detect and Segment: An Online Multi-object Tracker, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, Online, USA, 2021, pp. 12352-12361.

[27] S. Sun, N. Akhtar, H. Song, A. Mian, M. Shah, Deep Affinity Network for Multiple Object Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 1, pp. 104-119, January, 2021.

Biographies

Pan Yang is currently working toward the Master degree at the University of Science and Technology Beijing, Beijing, China. His research interests include machine learning and computational intelligence.

Xiong Luo received the Ph.D. degree in computer applied technology from the Central South University, Changsha, China, in 2004. He is currently a Professor at University of Science and Technology Beijing. His current research interests include machine learning and computational intelligence.

Jiankun Sun is currently working toward his Ph.D. degree at the University of Science and Technology Beijing, Beijing, China. His current research interests include computational intelligence and Internet of Things security.