Multi-object tracking based on deep associated features for UAV applications*

XIONG Lingyu and TANG Guijin**

School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

(Received 24 April 2022; Revised 17 October 2022)

Multi-object tracking (MOT) techniques have been increasingly applied in a diverse range of tasks. Unmanned aerial vehicle (UAV) is one of its typical application scenarios. Due to the scene complexity and the low resolution of moving targets in UAV applications, it is difficult to extract target features and identify them. In order to solve this problem, we propose a new re-identification (re-ID) network to extract association features for tracking in the association stage. Moreover, in order to reduce the complexity of detection model, we perform the lightweight optimization for it. Experimental results show that the proposed re-ID network can effectively reduce the number of identity switches, and surpass current state-of-the-art algorithms. In the meantime, the optimized detector can increase the speed by 27% owing to its lightweight design, which enables it to further meet the requirements of UAV tracking tasks.

Document code: A **Article ID:** 1673-1905(2023)02-0105-7

DOI https://doi.org/10.1007/s11801-023-2070-9

The techniques of vision-based multi-object tracking (MOT) exploit the particular similarity metrics to match different objects in video sequence[1,2]. They have been proposed over the years and have been widely used in many applications, such as surveillance[3], traffic monitoring[4], autonomous driving, and unmanned aerial vehicle (UAV) tracking.

Throughout the years, MOT tasks were mainly performed with tracking by detection paradigm, where objects ware detected by an object detector and fed to the object tracking method, which then dealt with the object association between previous frames and present one. With the emergence of deep learning-based neural networks (DNNs)[5], new state-of-the-art methods have been proposed in object vision-based tasks. Therefore, to improve the object association step of tracking algorithm, convolutional neural networks (CNNs) have been applied to extract object appearance features, which are used to compute similarity probability between two object's feature maps[6].

Currently, more works focus on pedestrian target tracking, and UAV tracking for small targets still needs further research. However, for different general tracking tasks, UAV vision tracking has some new challenges. When the UAV flies at a certain altitude, the resolution and clarity become low, the scale of the tracked target on the ground becomes very small, and the target features and textures become blurred, which makes it difficult to extract the target features, resulting in the difficulty of target detection and tracking. Due to structural characteristics of UAVs, most UAVs have limited computing resources. So how to develop a low complexity tracking algorithm with a high accuracy is a great challenge. To address these problems, we propose a new re-identification (re-ID) module learning mechanism. This is achieved by designing an inverted block composed of multiple convolutional streams and attention mechanism, each detecting features at a certain scale. Moreover, a novel detector with lightweight module is used to balance the model complexity and performance to reach real-time requirement.

We first provide a brief overview about the popular tracking by detection paradigm for MOT, and then introduce the re-ID for data association in MOT.

With the emergence of deep learning-based object detectors, tracking by detection has become the most popular approach community[2,7]. Specifically, an object detector is used to detect objects in each frame, then a subsequent tracker is utilized to associate the objects across different frames. In terms of temporal information usage, existing MOT methods can be categorized into online[8-10] and offline methods[11,12]. Online methods process video sequences frame-by-frame and track objects by only using information up to the current frame. By contrast, offline methods process video sequences in a batch and can even utilize the whole video information. In this work, the newly proposed tracking model can be naturally integrated into the online tracking by the detection

* This work has been supported by the Research Foundation of Nanjing University of Posts and Telecommunications (No.NY219076).
** E-mail: tanggj@njupt.edu.cn

MOT system.

Learning discriminative representations for objects is crucial to identity association in tracking[13]. The representation can be used to re-identity lost objects that reappear after disappearing for a long period of time. Appearance similarity can be measured by the cosine similarity of the re-ID features. Ref.[14] adopts a stand-alone re-ID model to extract appearance features from the detection boxes. Different from Ref.[14], our proposed re-ID model improves the representation power of features by adding multi-branch architecture and attention mechanism. Deployment of factorized convolution keeps the model lightweight with competitive performance.

The main contributions of this work can be summarized as follows. We propose a novel re-ID learning network, which adopts some effective strategies such as multi-scale learning, inverted block. We design a detector with lower computation. Compared to other detectors, the parameters and computations of the proposed detector are significantly reduced, while the performance keeps competitive. Both the re-ID learning and compressed detector can be applied to existing MOT methods in a natural way. Experimental results demonstrate the effectiveness of the proposed network.

In order to improve tracking ability for small targets, this paper designs a more powerful re-ID network termed aggregation network of inverted bottleneck (ANIB) to extract more robust features as appearance feature metric when employing association algorithm. The designed network follows the following principles.

1. Using multi-path convolution to obtain features of different receptive fields is helpful to make model recognize tasks from different scales.

2. Using depth separable convolution replaces traditional convolution operation. After that, parameters and computation are significantly reduced while obtaining same receptive field.

3. Using inverted blocks module replaces traditional bottleneck module. It is more difficult to lose information when extracting features by convolution operations.

4. A channel attention mechanism is introduced to make model more focused on useful channel features.

We use multi-path convolution to extract features of different scales to adapt the model to different sizes of objects. We use a different number of 3×3 convolutions in each branch to obtain different receptive fields, while ensuring that feature maps of each output have the same size, so that they can be aggregated directly. In the meanwhile, in order to reduce the model parameters, a cheap 3×3 convolution is employed. The computation and parameters have been greatly reduced. This module enables our designed network to learn features of different scales and enhance the robustness of model.

ANIB is based on depth-wise separable convolution which factorize a standard convolution into a depth-wise convolution and a 1×1 convolution named point-wise convolution. The depth-wise applies a single filter to each channel feature. The point-wise convolution then applies a 1×1 convolution to combine the output of depth-wise convolution. The depth-wise separable convolution splits this into two layers. This factorization has the effect of drastically reducing computation and model size. Fig.1 shows how a standard convolution (a) is factorized into a depth-wise convolution (b) and a 1×1 point-wise convolution (c).

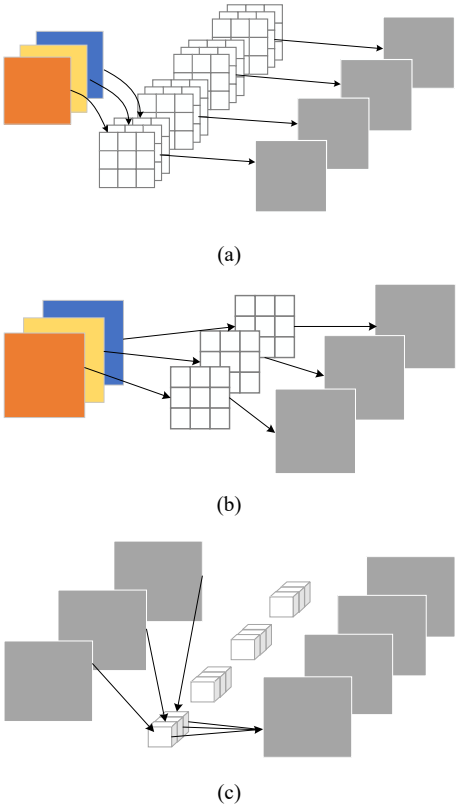

Fig.1 (a) Standard convolution filters; (b) Depth-wise convolution filters; (c) 1×1 convolution filters called point-wise convolution in the context of depth-wise separable convolution

Assuming that the input feature is $x \in R^{h\times w\times c}$, for traditional convolution $w \in R^{k\times k\times c\times c'}$, the computation is $h\cdot w\cdot k^2\cdot c\cdot c'$, where h and w are the height and width of input feature respectively, k is the size of the kernels, c is the number of input channels, and c' is the number of output channels. For depth-wise convolution $u \in R^{k\times k\times 1\times c'}$, the computation is $h\cdot w\cdot k^2\cdot c'$. For point-wise convolution $v \in R^{1\times 1\times c\times c'}$, the computation is $h\cdot w\cdot c\cdot c'$. Therefore, the total computation changes from $h\cdot w\cdot k^2\cdot c\cdot c'$ to $h\cdot w\cdot(k^2+c)\cdot c'$, and parameters change from $k^2\cdot c\cdot c'$ to $(k^2+c)\cdot c'$. As can be seen from Eq.(1) and Eq.(2), under the condition of obtaining same receptive field, the use of depth separable convolution reduces the number of parameters and computation to $1/k^2$ compared with traditional convolution. Generally, we take k as 3. We call this module light 3×3. Fig.2 shows the structure.

$$\frac{(k^2+c)\cdot c'}{k^2\cdot c\cdot c'}\approx\frac{1}{k^2}, \tag{1}$$

$$\frac{h\cdot w\cdot(k^2+c)\cdot c'}{h\cdot w\cdot k^2\cdot c\cdot c'}\approx\frac{1}{k^2}. \tag{2}$$

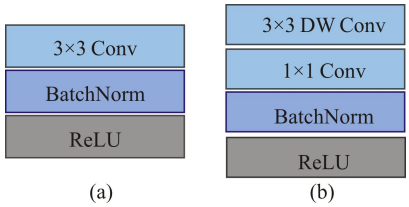

Fig.2 (a) Standard 3×3 convolution; (b) Light 3×3 convolution (DW: depth-wise)

The inverted bottleneck block is originally proposed in MobileNetv2[15]. Its core idea is that using bottleneck block will make model lose many information features by compressing and then amplifying, while information features will be retained as much as possible by amplifying and then compressing. With this idea, we introduced inverted bottleneck block. A basic version is shown in Fig.3(a). We first use 1×1 convolution for dimensions upgrading, and then use lightweight convolution light 3×3 extract features. Finally, we use 1×1 convolution to compress convolution dimensions for extracting effective features. Fig.3(b) shows the aggregation network combined baseline inverted block with multi-path convolution. ANIB is built by a simple stack of this module.

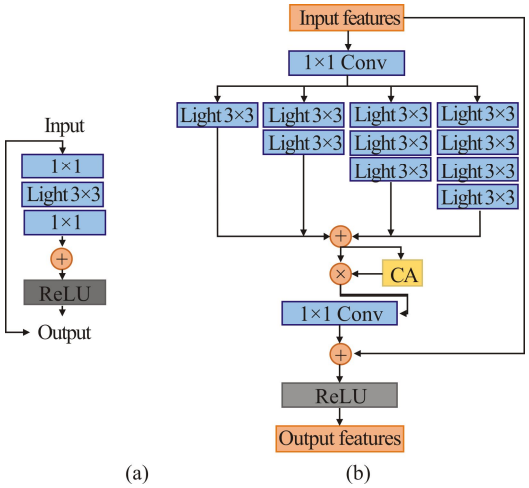

Fig.3 (a) Baseline inverted bottleneck blocks; (b) Inverted bottleneck blocks (CA: channel attention)

We added channel attention (CA) layer[16] after aggregating features of different paths in each inverted block, which is used to assign different weights to corresponding channels. By adding channel attention mechanism, the model parameter values are more focused on effective features. It is also conducive to pruning unimportant channels during model deployment, achieving the purpose of lightweight network. After CA module in Fig.3(b) obtains the input features of previous aggregation layer, global average pooling is conducted in channel dimension, then reduce dimension through the first full connection layer, and upgrade dimension through the second full connection layer. After the first full connection layer, rectified linear unit (ReLU) is used as activation function, and the second full connection layer uses Sigmoid as activation function, so that output values are between [0,1]. Finally, multiply each channel with channel weight and output them.

ANIB is built by a simple stack of inverted blocks, without carefully designing the width and depth for each layer. Details of the network structure are shown in Tab.1. ANIB can easily be deepened and enlarged to tradeoff between performance and speed.

Tab.1 ANIB architecture with input image size of 256×128 (*s*: stride)

Stage	Output	ANIB
conv1	256×128×32	3×3 inverted blocks, *s*=2
	128×64×32	Max pooling, *s*=2
conv2	128×64×48	3×3 inverted blocks, *s*=2
	64×32×48	Max pooling, *s*=2
conv3	64×32×48	3×3 inverted blocks, *s*=2
conv4	64×32×64	3×3 inverted blocks, *s*=2
	32×16×64	Max pooling, *s*=2
conv5	32×16×64	3×3 inverted blocks, *s*=2
conv6	32×16×96	3×3 inverted blocks, *s*=2
	16×8×96	Max pooling, *s*=2
conv7	16×8×96	3×3 inverted blocks, *s*=2
conv8	16×8×128	3×3 inverted blocks, *s*=2
	8×4×128	Max pooling, *s*=2
GAP	1×1×128	Global average pooling
FC	512	FC
# Params		1.94M

For multi-path convolution, ANIB is similar to ResNext[17] and Inceptionv1[18], but it is significantly different from them. For each convolution, they have different receptive fields, but they all use the same light 3×3 modules. We also follow the principle of inverted blocks to make the network more effective in extracting different scale features and lose less information in down sampling. In addition, ANIB uses CA module to extract different channel features, which makes the model pay more attention to valuable features. Finally, we also use the depth separable convolution to replace the standard convolution, so the whole model is lightweight.

The main lightweight ideas for parameter optimization in deep neural network include lightweight network, network pruning[19], knowledge distillation[20], quantization[21], etc. In this paper, we choose to compress the model with lighter modules.

Because YOLOv5 network structure contains a large number of residual modules, the feature maps generated by convolution contain many similar feature maps. These

feature maps can be obtained by a simpler method without expensive convolution operation, which can reduce model parameters and computation. In order to reduce convolution operations, this paper replaces residual block with Ghost-Bottleneck module in GhostNet[22].

Fig.4 shows the structure of Ghost bottleneck. Its main principle is to obtain similar feature maps by simple linear transformation and the idea of group convolution is adopted. Changing the number of groups into the number of channels becomes depth-wise convolution. Here, we learn from the idea of inverted residual module. The first ghost module is responsible for increasing the number of channels, and the second module is responsible for reducing the number of channels. Ghost module can use fewer parameters to obtain same number and size of feature maps as ordinary convolution, which reduces model parameters and computation to a certain extent.

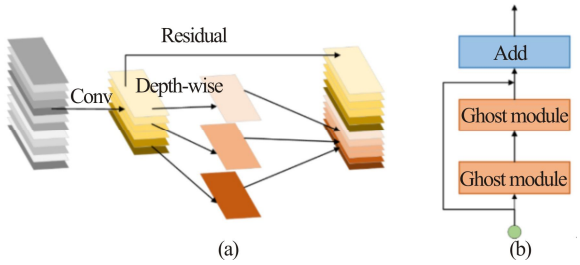

Fig.4 (a) Ghost module; (b) Ghost bottleneck module

We use the VeRi-776 vehicle re-ID dataset[23] as our performance evaluation benchmark. VeRi-776 contains more than 50 000 images of 776 vehicles. These images are taken by 20 cameras and cover an area of 1.0 km^2 in 24 h, which makes the dataset scalable enough for vehicle re-ID related research. VeRi-776 dataset is divided into 37 778 training images, 11 579 test images and 1 678 query images, respectively.

A classification layer is employed on the top of ANIB. Training follows the standard classification paradigm where each person identity is regarded as a unique class. Cross entropy loss with label smoothing is used for supervision. Training batch size and weight decay are set to 64 and 5×10^{-4} respectively, while test batch size is set to 128. Learning rate is set to 0.001 5. We train model from scratch with stochastic gradient descent (SGD) optimization algorithm for 100 epochs. Image is resized to 256×128. Data augmentation includes random flip and random erasing.

Tab.2 shows the evaluation results of mainstream algorithms and our proposed re-ID network on VeRi-776 dataset. Experiments show that the re-ID network proposed in this paper not only maintains lightweight but also has better performance than the network models with larger parameters and current mainstream lightweight network model. Fig.5 shows the exemplary test results of our proposed model. It can be seen that most of queries are accurate, and there are a small number of errors in vehicles that are difficult to distinguish.

Tab.2 Performance evaluation results

Method	VeRi		Market1501		Duke		Params
	R1	mAP	R1	mAP	R1	mAP	
Ref.[17]	69.9	41.1	63.9	43.5	46.7	28.4	22.03M
Ref.[24]	78.7	41.2	70.6	48.8	54.5	36.4	24.69M
Ref.[25]	80	45.4	**83.2**	55.6	65.2	45.2	2.97M
Ref.[26]	81.4	44.6	73.3	51.1	**73.6**	48.4	1.46M
ANIB (ours)	**87.2**	**49.6**	82.9	**62.7**	73.1	**51.5**	1.94M

Fig.5 Test results in VeRi-776 dataset

From Tab.2, it can be seen that ANIB achieves state-of the-art performance on most datasets, outperforming most published methods by a clear margin. Moreover, our method uses a small-scale model. Compared with Ref.[24] of backbone network with nearly 24.7M parameters, ANIB has only 1.9M parameters. This proves the effectiveness of multi-scale feature learning. ANIB outperforms mainstream lightweight networks in Ref.[15] and Ref.[26] by a large margin. It is noted that the three networks have considerable model parameters, which justify that the network we designed not only employs lightweight, but also achieves competitive performance. Compared with Ref.[17], which also uses multi-branch feature extraction structure, our method has better performance, mainly due to the use of CA layer, which makes model pay more attention to important features.

Tab.3 evaluates our architectural design choices where our primary model is model 1. Compared with standard convolution, factoring convolution reduces the R1 marginally by 0.3% (model 3 vs. model 2). This means our architecture design maintains the representation power even though the model size is reduced by more than 3 times. Compared with ResNeXt-like design, ANIB is transformed into a ResNeXt-like architecture by fusing different sizes of features, which refers to model 1. We observe the variant is clearly outperformed by the primary model, with 6.1%/2.7% difference in R1/mean average precision (mAP). This further validates the necessity of our multi-scale design. Employment of attention improves the representational power of a network by enabling it to perform dynamic channel-wise feature recalibration, with an increment of 2.3%/1.2% at R1/mAP. By turning the bottleneck block into inverted block, both the R1 and mAP increase by 1.5%/0.8%. As inverted block is enabled to retain more feature information following the upper module, it is advantageous to use inverted block.

Tab.3 Ablation experiment results (MS: multi-scale; DW: depth-wise separable convolution; IB: inverted block)

Model	MS	DW	Attention	IB	VeRi	
					R1	mAP
1					77.6	45.4
2	✓				83.7	48.1
3	✓	✓			83.4	47.6
4	✓	✓	✓		85.7	48.8
5	✓	✓	✓	✓	87.2	49.6

To reduce the parameters and computation in detection stage, the CSP bottleneck is replaced with Ghost bottleneck module. Comparison of model parameters before and after replacement is shown in Tab.4. The amount of model parameters is reduced from 7.28M to 4.13M, and computation is reduced from 17.1 giga floating-point operations per second (GFLOPS) to 9.9 GFLOPS, nearly half. At the same time, processing speed enhanced from 20.8 ms to 15.9 ms while performance metric mAP is slightly lower than that of the primary detection network.

Tab.4 Comparison of model parameters

Model	Module	Params	Computation	mAP	Speed
YOLOv5s	CSP	7.28M	17.1 GFLOPs	0.802	20.8 ms
	Ghost	4.13M	9.9 GFLOPs	0.792	15.9 ms

In general, compared with the original network, YOLOv5 model with Ghost bottleneck still achieves competitive performance, while reducing the parameters and computation to about half of the former, which proves the effectiveness and practicability of our improved model.

This paper uses the UAVDT dataset[27] as the benchmark dataset. The benchmark includes three tasks, target detection, single target tracking and multi-target tracking. The UAVDT benchmark consists of 10 h of original videos, from which 100 video sequences of about 80 000 representative frames are selected. Each sequence contains 83 to 2 970 frames and annotates about 840 000 bounding boxes for more than 2 700 vehicles totally. The selected video frames comprehensively consider the weather factors, flight altitude, camera angle, vehicle type and vehicle occlusion, which is more in line with the actual scene requirements and more challenging.

We use the improved object detection network to extract targets, use the DeepSORT algorithm as the basic tracking algorithm, and embed ANIB to extract the features as appearance metric for data association. During training, the initial learning rate is 0.01, SGD is used as the optimizer, and the momentum factor is 0.937. Set the weight decay to 5×10^{-4}. Set the training batch size to 16 and the input picture size to 640×640. Do not use pre-training weight. Start training from scratch, 100 epochs totally.

Evaluation is carried out according to the following metrics.

- Multi-object tracking accuracy (MOTA): Summary of overall tracking accuracy in terms of false positives, false negatives and identity switches.
- Multi-object tracking precision (MOTP): Summary of overall tracking precision in terms of bounding box overlap between ground-truth and reported location.
- Mostly tracked (MT): Percentage of ground-truth tracks that have the same label for at least 80% of their life span.
- Mostly lost (ML): Percentage of ground-truth tracks that are tracked for at most 20% of their life span.
- Identity switches (IDSW): Number of times the reported identity of a ground-truth track changes.
- Fragmentation (Frag): Number of times a track is interrupted by a missing detection.

Compared with baseline algorithm, ANIB improves detector and satisfies the requirement of real-time detection. Moreover, ANIB embeds an SOTA re-ID network to improve tracking performance. An overall comparison below evaluates the effectiveness of ANIB. Tab.5 shows the results of the different tracking algorithms on the UAVDT dataset. It can be seen that the proposed algorithm is superior to other algorithms in MOTA, IDSW, Frag, and Hz metrics. And it's competitive to the original algorithm in other metrics. On MOTA, our proposed method is improved by 2% and outperforms other methods. On IDSW, we reduce the number of identity switches by nearly 59% compared with other algorithms. IDSW is reduced by 43% with and without ANIB, which proves the validity of ANIB feature extraction. On the Frag metric, it is reduced by nearly 14% compared with other algorithms. In addition, the speed of the improved algorithm is increased by 27%, which can basically meet the real-time processing requirements. Fig.6 shows a tracking example of the proposed algorithm.

Tab.5 Test results of tracking algorithms

Method	MOTA ↑	MOTP ↑	IDSW ↓	MT ↑	ML ↓	Frag ↓	Hz ↑
R-FCN[28]	30.87	77.0	427	284	108	1 186	17.4
SSD[29]	32.69	76.7	1 149	216	105	2 162	23.4
Faster R-CNN[30]	**40.68**	75.2	567	324	94	**812**	21.4
YOLOv5 (resnet50)	39.88	**82.2**	**268**	**394**	**83**	818	**25.4**
YOLOv5 (ANIB)	40.21	82.1	**151**	392	84	772	26.5
YOLOv5-ghost (resnet50)	42.18	80.4	312	390	86	734	31.2
YOLOv5s-ghost (ANIB)	**42.41**	80.4	171	389	85	**698**	**32.3**

Fig.6 Tracking example on UAVDT dataset

Aiming at the challenges in UAV tracking tasks, this paper designs a lightweight re-ID network to extract target appearance features for trajectory association, which can improve tracking algorithm performance for small targets. To solve the problem of limited computing resources of UAV platforms, we greatly reduce the model parameters by using lighter modules. Experimental results show that our model greatly reduces the number of identity switches, and outperforms other methods on multiple metrics.

Statements and Declarations

The authors declare that there are no conflicts of interest related to this article.

References

[1] CIAPARRONE G, SÁNCHEZ F L, TABIK S, et al. Deep learning in video multi-object tracking：a survey[J]. Neurocomputing, 2020, 381：61-88.

[2] KAMAL R, CHEMMANAM A J, JOSE B A, et al. Construction safety surveillance using machine learning[C]//2020 International Symposium on Networks, Computers and Communications (ISNCC), October 20-22, 2020, Montreal, QC, Canada. New York：IEEE, 2020：1-6.

[3] XU Y, OSEP A, BAN Y, et al. How to train your deep multi-object tracker[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 13-19, 2020, Seattle, WA, USA. New York：IEEE, 2020：6787-6796.

[4] BEHRENDT K, NOVAK L, BOTROS R. A deep learning approach to traffic lights：detection, tracking, and classification[C]//2017 IEEE International Conference on Robotics and Automation (ICRA), May 29-June 3, 2017, Singapore, Singapore. New York：IEEE, 2017：1370-1377.

[5] PEREIRA R, GARROTE L, BARROS T, et al. A deep learning-based indoor scene classification approach enhanced with inter-object distance semantic features[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 27-October 1, 2021, Prague, Czech Republic. New York：IEEE, 2021：32-38.

[6] WU J, CAO J, SONG L, et al. Track to detect and segment：an online multi-object tracker[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 20-25, 2021, Nashville, TN, USA. New York：IEEE, 2021：12352-12361.

[7] LIU Y, LI X, BAI T, et al. Multi-object tracking with hard-soft attention network and group-based cost minimization[J]. Neurocomputing, 2021, 447：80-91.

[8] LIU Q, CHU Q, LIU B, et al. GSM：graph similarity model for multi-object tracking[C]//International Joint Conferences on Artificial Intelligence (IJCAI), January 7-15, 2021, Yokohama, Japan. California：IJCAI, 2020：530-536.

[9] ZHOU X, KOLTUN V, KRÄHENBÜHL P. Tracking objects as points[C]//European Conference on Computer Vision, August 23-28, 2020, Virtual. Cham：Springer, 2020：474-490.

[10] ZHANG Y, WANG C, WANG X, et al. Fairmot：on the fairness of detection and re-identification in multiple object tracking[J]. International journal of computer vision, 2021, 129(11)：3069-3087.

[11] BRASÓ G, LEAL-TAIXÉ L. Learning a neural solver for multiple object tracking[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 13-19, 2020, Seattle, WA, USA. New York：IEEE, 2020：6247-6257.

[12] HORNAKOVA A, HENSCHEL R, ROSENHAHN B, et al. Lifted disjoint paths with application in multiple object tracking[C]//International Conference on Machine Learning, July 12-18, 2020, Virtual. IMLS, 2020：4364-4375.

[13] KIM C, FUXIN L, ALOTAIBI M, et al. Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 20-25, 2021, Nashville, TN, USA. New York：IEEE, 2021：9553-9562.

[14] WOJKE N, BEWLEY A, PAULUS D. Simple online and realtime tracking with a deep association met ric[C]//2017 IEEE International Conference on Image Processing (ICIP), September 17-20, 2017, Beijing, China. New York：IEEE, 2017：3645-3649.

[15] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2 ： inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York：IEEE, 2018：4510-4520.

[16] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York：IEEE,

2018：7132-7141.

[17] XIE S, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York：IEEE, 2017：1492-1500.

[18] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015, Boston, MA, USA. New York：IEEE, 2015：1-9.

[19] FRANKLE J, CARBIN M. The lottery ticket hypothesis：finding sparse, trainable neural networks[EB/OL]. (2018-03-09) [2022-03-13]. https：//arxiv.org/abs/1803.03635v5.

[20] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[EB/OL]. (2015-03-09) [2022-03-13]. https：//arxiv.org/abs/1503.02531.

[21] JACOB B, KLIGYS S, CHEN B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York：IEEE, 2018：2704-2713.

[22] HAN K, WANG Y, TIAN Q, et al. Ghostnet：more features from cheap operations[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 13-19, 2020, Seattle, WA, USA. New York：IEEE, 2020：1580-1589.

[23] LIU X, LIU W, MEI T, et al. A deep learning-based approach to progressive vehicle re-identification for urban surveillance[C]//European Conference on Computer Vision, October 8-16, 2016, Amsterdam, The Netherlands. Cham：Springer, 2016：869-884.

[24] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016, Las Vegas, NV, USA. New York：IEEE, 2016：770-778.

[25] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets：efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17) [2022-03-13]. https：//arxiv.org/abs/1704.04861.

[26] ZHANG X, ZHOU X, LIN M, et al. Shufflenet：an extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York：IEEE, 2018：6848-6856.

[27] DU D, QI Y, YU H, et al. The unmanned aerial vehicle benchmark：object detection and tracking[C]//Proceedings of the European Conference on Computer Vision, September 8-14, 2018, Munich, Germany. Cham：Springer, 2018：370-386.

[28] DAI J, LI Y, HE K, et al. R-FCN：object detection via region-based fully convolutional networks[J]. Advances in neural information processing systems, 2016, 29.

[29] LIU W, ANGUELOV D, ERHAN D, et al. SSD：single shot multibox detector[C]//European Conference on Computer Vision, October 8-16, 2016, Amsterdam, The Netherlands. Cham：Springer, 2016：21-37.

[30] REN S, HE K, GIRSHICK R, et al. Faster R-CNN：towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28：91-99.