Dynamic Fusion Network for RGBT Tracking

Jingchao Peng, Haitao Zhao, and Zhengwei Hu

Abstract—Since both visible and infrared images have their own advantages and disadvantages, RGBT tracking plays an important role in intelligent transportation systems. The key points of RGBT tracking lie in feature extraction and fusion of visible and infrared images. Current RGBT tracking methods mostly pay attention to both individual features (features extracted from images of a single camera) and common features (features extracted and fused from an RGB camera and a thermal camera). Still, they pay less attention to different and dynamic contributions of the individual and common features for different sequences of registered image pairs. This paper proposes a novel RGBT tracking method, called Dynamic Fusion Network (DFNet), which adopts a two-stream structure, in which two non-shared convolution kernels are employed in each layer to extract individual features. Besides, DFNet has shared convolution kernels for each layer to extract common features. Since non-shared and shared convolution kernels are adaptively weighted and summed according to different image pairs, DFNet can deal with different contributions for different sequences. DFNet has a fast speed, which is 28.658 FPS. The experimental results show that when DFNet only increases the Multi-Adds by 0.02% compared with the non-shared-convolution-kernel-based fusion method, Precision Rate (PR) and Success Rate (SR) reach 88.1% and 71.9%, respectively. The model and dataset are available at https://github.com/PengJingchao/DFNet.

Index Terms—Object tracking, fusion tracking, dynamic convolution, deep learning, intelligent perception.

I. INTRODUCTION

O
bject tracking is a prevalent computer vision task, whose purpose is to continuously track the position of the object in the subsequent frames when given in the first frame [1], [2], [3]. Tracking under a complex visual scenery, including rain, smoke, or night, is one of the most challenging computer vision tasks [4], [5], especially for visible-image-based tracking [5], [6]. However, infrared sensors can work around the clock. For example, infrared has a strong ability to penetrate smoke, which can supplement the deficiencies of visible images in bad visual conditions [7], [8], [9], [10]. Therefore, RGBT tracking is widely used in comprehensive intelligent transportation systems (ITS), such as autonomous driving [11], [12], surveillance [13], [14], pedestrian detection [15], [16], and so on.

Since 2018, due to its powerful learning ability, Deep Learning (DL) models, especially Convolutional Neural Networks (CNN), have been widely used to address RGBT tracking [17], [18], [19], [20], [21], [22]. DL-based tracking methods have demonstrated their capabilities over traditional fusion tracking methods [23], [24], [25], [26] or other tracking methods (e.g., sparse-representation-based methods [27] and graph-based methods [28]). The advantage of DL-based tracking methods is their ability to learn more effective and robust features than hand-crafted features [6], [29], [30]. DL-based tracking methods can be divided into pixel-level [17], feature-level [18], [19], [20], [21], and decision-level [22] fusion tracking. Compared with pixel-level fusion, feature-level fusion has lower requirements for image registration and can tolerate a certain amount of noise [19], [20]. Compared with decision-level fusion, it has lower computational complexity and faster speed [6], [31]. Recently research works on DL-based RGBT tracking mainly focus on feature-level fusion [6].

Due to different imaging properties of visible light reflection and infrared radiation, visible and infrared images have different individual features [32], which can be used to track the object based on single-modal images. In visible-image-based tracking, the object can be distinguished through rich textures and colors. While in infrared-image-based tracking, the object can be distinguished by high-contrast light-dark changes that reflect the heat of the object. In order to fully utilize the individual features from the two different modalities, feature-level fusion methods are often adopted, in which two Convolutional Neural Networks (CNNs) were often employed to handle visible and infrared images, respectively. For example, Zhang et al. [18] utilized two different CNNs to extract individual features separately from visible and infrared images. In their work, the visible and infrared features were concatenated and sent to follow layers for tracking the object. ConvNet [20] and SiamFT [19] employ fusion sub-networks to select discriminative features after extracting the individual features. DSiamMFT [33] and FANet [34] focus on multi-layer feature fusion to achieve more effective hierarchical feature aggregation. For simplicity, this paper denotes the basic feature-level fusion method without any bells and whistles, which only uses two different CNNs to separately handle visible and infrared images, as the non-shared-convolution-kernel-based fusion method.

In addition to individual features, since visible and infrared images are shot in the same scene and used to track the same object, there are common features in the two modalities [35]. Common features reflect the size, location, contour, and so on, which are also important information in object tracking [36]. When individual features are not enough to achieve good tracking performances, it is necessary to use common features, such as the semantics of the object and other characteristics.
of the object at the corresponding position of the visible and infrared images, for object tracking. MANet [21], CAT [32], IVFuseNet [36], and SiamIVFN [31] use a shared convolution kernel to extract common features. Their experimental results show that the shared-convolution-kernel-based fusion methods can extract common features that are more informative than the non-shared-convolution-kernel-based fusion method. Nevertheless, in their networks, the contributions of individual and common features are prefixed and have no consideration of adaption to the registered image pairs captured in different scenes.

However, the contributions of individual features and common features are not permanently fixed. Visible images are greatly affected by illumination and prone to overexposure or underexposure. Infrared images are easily interfered with by the external scenery and internal systems and are prone to noise [37], [38]. In other words, the reliability of different modalities is not always the same. For different reliable modalities, individual and common features contribute to different degrees. When one modality is reliable, the individual features of the reliable modality contribute more, as shown in Figure 1(a). When tracking based on single-modal images is impossible, more attention needs to be paid to common features, as shown in Figure 1(b). Therefore, the tracker needs to adaptively calculate different contributions of individual and common features in different scenes.

Dynamic convolution has natural advantages to solving the network’s performance limitation in changing scenes. The concept of dynamic convolution (e.g., CondConv [39], Dynamic Convolution [40], and WeightNet [41]) usually adopts the method of attention over convolution kernels. Dynamic convolution has been applied in scene segmentation, scene synthesizing, image inpainting, biomedical imaging, and so on [42], [43], [44], [45]. Due to aggregating multiple convolution kernels adapted to each input, dynamic convolution has more representation power without increasing the width and depth of the network. The aggregation of multiple convolution kernels in convolution kernel space makes it possible to make full use of multiple convolution kernels only by one convolution operation. Therefore, dynamic convolution is computationally efficient. However, dynamic convolution is designed to integrate into existing CNN architectures and cannot aggregate individual and common features in fusion tasks.

Motivated by the above analysis, we propose a novel RGBT tracking method called dynamic fusion network (DFNet). DFNet adopts a two-stream structure with non-shared convolution kernels to extract individual features. One CNN is utilized to extract features from infrared images, and the other is for handling visible images. Besides, DFNet has shared convolution kernels to extract common features. DFNet adaptively merges the shared and non-shared convolution kernels in convolution kernel space through the dynamic convolution. To satisfy strict latency requirements for object tracking, DFNet only needs two convolution operations in each layer to extract the individual and common features of visible and infrared images. Since the weights of shared and non-shared convolution kernels are dynamically computed, it can adaptively calculate the contributions of individual and common features to different scenes.

Specifically, the proposed method has the following advantages:

1) DFNet has shared and non-shared kernels, which separately extract the common and individual features. DFNet has a strong representation power.
2) DFNet adaptively calculates the contributions of individual and common features according to different registered image pairs.
3) As a general fusion module, DFNet can be easily implemented on other tracking frameworks.

II. RELATED WORK

Section I overviews DL-based RGBT tracking. This section focuses on three of the most related works to ours: ConvNet [20], MANet [21], and IVFuseNet [36]. Their simplified feature extraction layer diagrams are shown in Figure 2.

A. ConvNet

ConvNet [20] uses different convolutional networks to extract the individual features of visible and infrared images and then fuses them. The expression of its feature extraction layer is shown in the first row of Table I, where $W_{RGB}$ and $W_T$ represent the convolution kernel for RGB and thermal features, respectively; $F_{RGB}$ and $F_T$ represent the input of visible and infrared branch, respectively; $*$ represents convolution operation; and $\sigma(\cdot)$ represents activation function, such as ReLU.

In ConvNet, different convolution kernels extract individual features from visible and infrared images. Then these two features are fused and sent to domain-specific layers for binary classification and target identification. Besides, ConvNet designs a fusion sub-network, which adaptively fuses two individual features to remove redundant noise. The feature extraction process performed two convolution operations in
Fig. 2. Simplified feature extraction layer diagrams of ConvNet, MANet, IVFuseNet, and DFNet (ours). The blue and gray branches separately handle visible and infrared images. The yellow block represents the shared convolution kernel used to extract common features. (a) Conv Layer. (b) MA Layer. (c) IVFuse Layer. (d) DF Layer (ours).

### TABLE I

| Fusion method | Expression |
|---------------|------------|
| ConvNet       | $\sigma\left(\begin{bmatrix} W_{RGB} & W_T \end{bmatrix} \ast \begin{bmatrix} F_{RGB} \\ F_T \end{bmatrix}\right)$ |
| MANet         | $\sigma\left(\begin{bmatrix} W_{RGB} & W_{share} & W_T \end{bmatrix} \ast \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \ast \begin{bmatrix} F_{RGB} \\ F_T \end{bmatrix}\right)$ |
| IVFuseNet     | $\sigma\left(\begin{bmatrix} W_{RGB} & W_{share} & W_T \end{bmatrix} \ast F_{RGB} \ast \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \ast \begin{bmatrix} F_{RGB} \\ F_T \end{bmatrix}\right)$ |
| DFNet         | $\sigma\left(\begin{bmatrix} aW_{RGB} + bW_{share} & cW_{share} + dW_T \end{bmatrix} \ast \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \ast \begin{bmatrix} F_{RGB} \\ F_T \end{bmatrix}\right)$ |

one layer; therefore, the speed of ConvNet is fast. ConvNet focuses on individual features but does not fully consider the common features.

### B. MANet

Li et al. [21] argue that common features of visible and infrared images are crucial to the effectiveness of the fusion. Therefore, MANet employs a shared convolution kernel to extract the common features of visible and infrared images. The expression of its feature extraction layer is shown in the second row of Table I, where $W_{share}$ represents the shared convolution kernel.

Before respective convolution operations, visible and infrared features must both undergo a shared convolution operation using the same convolution kernel. It is worth noting that MANet fuses shared and non-shared features in the feature space. Please note that the activation function $\sigma(\cdot)$ is not a linear operation, so we have:

$$\sigma (W_{RGB} \ast F_{RGB}) + \sigma (W_{share} \ast F_{RGB})$$

$$\sigma (W_{share} \ast F_T) + \sigma (W_T \ast F_T)$$

Without shared convolution kernels, one registered image pair only needs to be convolved with non-shared convolution kernels. While the registered image pair in MANet needs extra convolution operations with shared convolution kernels. MANet has double the number of convolution operations.
than methods without shared convolution kernels, which is not conducive to the real-time requirements of the tracking task. In addition, the weights of the shared and non-shared convolution kernels are equal, so they cannot be adjusted in real-time in the face of different contributions of individual and common features. CAT [32] is an RGBT tracking method inherited from MANet; both CAT and MANet fuse shared and non-shared features in the feature space by adding. The difference is that the shared convolution kernel of MANet has only one kernel, while the corresponding part of CAT has five kernels. The multi-branch design of CAT makes it difficult to implement and customize and increases the memory access cost [46].

C. IVFuseNet

Unlike the fusion of shared and non-shared features in feature space, IVFuseNet [36] merges the shared and non-shared convolution kernels in convolution kernel space. IVFuseNet concatenates two small-sized convolution kernels, one of which is a shared convolution kernel. The visible and infrared images are separately convolved with different spliced convolution kernels. The expression of its feature extraction layer is shown in the third row of Table I. Since the shared and non-shared convolution kernels are fused in convolution kernel space, IVFuseNet only needs two convolution operations. However, because the shared and non-shared convolution kernels are concatenated in advance, the size of the convolution kernel is smaller than that of MANet, which means IVFuseNet has weaker representation power than MANet. For example, in MANet, the shared and non-shared convolution kernel size in the first layer is $96 \times 3 \times 7 \times 7$ and $96 \times 3 \times 3 \times 3$, respectively; while in IVFuseNet, the size of them is $24 \times 3 \times 7 \times 7$ and $72 \times 3 \times 3 \times 3$, respectively. Besides, the channel size of the shared and non-shared convolution kernel needs to be prefixed, so the coupling rate cannot be adjusted in real-time in the face of different contributions of individual and common features.

We summarize the related works below:

1) The speed of ConvNet is fast, but ConvNet does not have shared convolution kernels to extract common features.
2) Although MANet has both shared and non-shared convolution kernels, the speed of MANet is much slower than that of ConvNet. Moreover, MANet has no design to deal with the different contributions of the individual and common features.
3) IVFuseNet has both shared and non-shared convolution kernels, and the speed of IVFuseNet is fast. However, compared with MANet, IVFuseNet has weaker representation power. Moreover, IVFuseNet also has no procedure to handle the different contributions of the individual and common features.

III. OUR METHOD

In this section, we will introduce a novel RGBT tracking method called dynamic fusion network (DFNet). We first introduce the dynamic fusion layer. Dynamic convolution is used in the convolution kernel space to fuse the shared and non-shared convolution kernels. Then we use the dynamic fusion layer as the basic module to construct DFNet for RGBT tracking. DFNet has the advantages of MANet and IVFuseNet, which have shared convolution kernels to extract common features. The differences in network structures and formula expressions between DFNet and the related models are highlighted in Table I and Figure 2.

A. Dynamic Fusion Layer

Dynamic fusion layer fuses the shared convolution kernel and non-shared convolution kernels in convolution kernel space. First, the convolution kernels are merged, then the convolution operation is performed. The expression of its feature extraction layer is shown in the last row of Table I. Its structure diagram is shown in Figure 2(d). In terms of convolution computation, DFNet reduces the computation by half compared with fusing shared and non-shared features in feature space.

The fusion of shared and non-shared convolution kernels is a weighted addition:

$$
\begin{align*}
\bar{W}_{\text{RGB}} &= aW_{\text{RGB}} + bW_{\text{share}} \\
\bar{W}_T &= cW_{\text{share}} + dW_T \\
\text{s.t. } 0 < &\{a, b, c, d\} < 1 \quad a + b = 1 \quad c + d = 1.
\end{align*}
$$

In this way, the size of the convolution kernels is not changed, and no additional artificially set parameters are introduced. The weights $a, b, c,$ and $d$ are adaptive, which can be obtained from the input through Global Average Pooling (GAP), two-layer full connection (FC), and softmax:

$$
\begin{align*}
[a, b] &= \mathcal{F}(F_{\text{RGB}}) \\
[c, d] &= \mathcal{F}(F_T) \\
\mathcal{F}(X) &= \text{Softmax} \circ \text{FC} \circ \text{ReLU} \circ \text{FC} \circ \text{GAP}(X).
\end{align*}
$$

The weights are input-dependent so that the dynamic fusion layer can use different convolution kernels for different image pairs, enhancing the expressive ability of the model. Compared with the convolution layer without weights, the dynamic fusion layer only increases the Mult-Adds of 0.02%, which can guarantee the real-time performance of the network. For details, please refer to Section V-E.

In the dynamic fusion layer, the convolution kernel is updated through the back-propagation algorithm. In each iteration, the convolution kernel of visible and infrared features iterates as follows:

$$
\begin{align*}
\bar{W}^{(i)}_{\text{RGB}} &= a\left(W^{(i-1)}_{\text{RGB}} + \text{lr} \frac{\partial L}{\partial W^{(i-1)}_{\text{RGB}}} \right) \\
&+ b\left(W^{(2i-1)}_{\text{share}} + \text{lr} \frac{\partial L}{\partial W^{(2i-1)}_{\text{share}}} \right),
\end{align*}
$$

$$
\bar{W}^{(i)}_T &= c\left(W^{(2i-2)}_{\text{share}} + \text{lr} \frac{\partial L}{\partial W^{(2i-2)}_{\text{share}}} \right) \\
&+ d\left(W^{(i-1)}_T + \text{lr} \frac{\partial L}{\partial W^{(i-1)}_T} \right),
$$

where $i$ is the iteration numbers, lr is the learning rate, and $L$ is the loss function. In each iteration, the non-shared convolution
kernels are updated once, and the shared convolution kernel are updated twice.

B. The Architecture

The overall architecture of DFNet is shown in Figure 3. The features of visible and infrared images are first extracted and fused through three dynamic fusion layers. After PrRoiPooling [47], the features of different sizes are unified into \(3 \times 3\). Then, features enter three fully connected networks to determine the object or background. At the end of the tracking process, DFNet takes the candidate with the highest network output score as the object:

\[
x^*_t = \arg \max_{x_t^i} F(x_t^i),
\]

(7)

where \(x_t^i\) represents the \(i\)-th candidate frame in the \(t\)-th frame, \(F()\) represents the score of network output, and \(x^*_t\) represents the final object result of the \(t\)-th frame.

Based on RT-MDNet [48], DFNet adopts a multi-domain learning framework, replacing RoIAlign with PrRoiPooling [47], which can precisely extract fixed-size feature maps of the region of interest from the whole feature maps. During training, all video sequences share three dynamic fusion layers, FC4 and FC5. Each video sequence uses a domain-specific FC6. During testing, the multiple domain-specific FC6s are replaced with a reinitialized FC6.

IV. IMPLEMENTATION DETAILS

We train and test DFNet on the PyTorch platform with an i7-10700K CPU and a TITAN RTX GPU. This section will introduce the training and online tracking process details.

A. Offline Training

The pre-trained network in VGG-M [49] is adopted to initialize the model and use ImageNet [50] and RGBT (GTOT [51] or RGBT234 [52]) mixed datasets to train DFNet. We train the network using stochastic gradient descent with momentum. The momentum is set to 0.9, the learning rate is set to 1e-4, and the weight decay is set to 5e-4. The number of epochs is set to 60. More details can be referred to [48].

B. Online Tracking

In the online tracking phase, we initialize the model with the trained three dynamic fusion layers, FC4, and FC5. We reinitialize a new FC6 and use the first frame to train FC6. Specifically, we collect 500 positive samples (IOU>0.7) and 5000 negative samples (IOU<0.3) from the first frame as training samples, and we use stochastic gradient descent with momentum for training. The momentum is set to 0.9, the learning rate is set to 1e-4, and the weight decay is set to 5e-4. More details can be referred to [48].

In the follow-up tracking phase, three dynamic fusion layers are fixed, but FC4, FC5, and FC6 are fine-tuned online. We collect 50 positive and 200 negative samples, perform long-term updates every 10 frames, and perform short-term updates when tracking fails. The learning rate of FC6 is set to 1e-3, and the learning rate of FC4 and FC5 is set to 5e-4. At the \(t\)-th frame, we use a Gaussian sampler to collect 256 candidates around the object position in the previous frame and calculate their classification scores through the network. Then, the Multi-layer Perceptron (MLP) is used to regress the average value of the top five bounding boxes of the classification score to obtain the final bounding box.

V. EXPERIMENT

A. Dataset and Evaluation Matrix

We use two RGBT datasets, GTOT [51] and RGBT234 [52], to compare DFNet with other tracking methods. We use ImageNet and GTOT mixed dataset as the training set when evaluating on RGBT234, and we use ImageNet and RGBT234 mixed dataset as the training set when evaluating on GTOT. The GTOT and RGBT234 datasets have 50 and 234 RGBT sequences of image pairs aligned in space and time, respectively. In one-pass evaluation (OPE), we use Precision Rate (PR) and Success Rate (SR) as evaluation indicators to evaluate the tracking results. PR refers to the proportion of frames
TABLE II
PR/SR SCORES OF RGBT234 BASED ON ATTRIBUTES. THE BEST, SECOND-BEST, AND THIRD-BEST PR/SR ARE SHOWN IN RED, YELLOW, AND BLUE

| Tracker | SiamFC | SiamRPN | MDNet | RTMDNet | RTMDNet -pixel | RTMDNet -feature | RTMDNet -IVFuseNet | RTMDNet -MANet | DFNet |
|---------|--------|---------|-------|---------|----------------|-----------------|------------------|----------------|-------|
| BC      | 0.496/0.333 | 0.187/0.116 | 0.683/0.462 | 0.630/0.402 | 0.582/0.375 | 0.705/0.439 | 0.659/0.397 | 0.697/0.437 | 0.714/0.452 |
| CM      | 0.564/0.407 | 0.321/0.226 | 0.689/0.497 | 0.637/0.438 | 0.626/0.429 | 0.699/0.467 | 0.663/0.438 | 0.676/0.448 | 0.692/0.471 |
| DEB     | 0.591/0.431 | 0.281/0.212 | 0.685/0.497 | 0.654/0.451 | 0.611/0.419 | 0.679/0.466 | 0.651/0.448 | 0.682/0.445 | 0.661/0.462 |
| FM      | 0.518/0.374 | 0.276/0.155 | 0.690/0.448 | 0.679/0.404 | 0.595/0.330 | 0.637/0.365 | 0.618/0.356 | 0.621/0.374 | 0.640/0.378 |
| HO      | 0.521/0.367 | 0.261/0.164 | 0.654/0.459 | 0.634/0.422 | 0.586/0.369 | 0.621/0.403 | 0.592/0.381 | 0.634/0.409 | 0.641/0.412 |
| LI      | 0.498/0.336 | 0.231/0.154 | 0.674/0.451 | 0.605/0.391 | 0.609/0.404 | 0.727/0.464 | 0.794/0.492 | 0.715/0.442 | 0.756/0.497 |
| LR      | 0.605/0.404 | 0.295/0.159 | 0.734/0.502 | 0.683/0.447 | 0.727/0.464 | 0.794/0.492 | 0.787/0.471 | 0.730/0.446 | 0.797/0.496 |
| MB      | 0.554/0.405 | 0.310/0.209 | 0.702/0.517 | 0.669/0.467 | 0.633/0.442 | 0.676/0.470 | 0.670/0.450 | 0.635/0.433 | 0.702/0.489 |
| NO      | 0.765/0.564 | 0.404/0.282 | 0.862/0.636 | 0.842/0.576 | 0.828/0.557 | 0.859/0.582 | 0.856/0.564 | 0.868/0.569 | 0.871/0.599 |
| PO      | 0.629/0.446 | 0.275/0.188 | 0.810/0.567 | 0.754/0.513 | 0.714/0.508 | 0.856/0.569 | 0.838/0.554 | 0.817/0.544 | 0.857/0.575 |
| SV      | 0.634/0.461 | 0.308/0.210 | 0.767/0.549 | 0.747/0.508 | 0.697/0.461 | 0.751/0.499 | 0.743/0.494 | 0.762/0.505 | 0.749/0.501 |
| TC      | 0.681/0.488 | 0.233/0.155 | 0.801/0.585 | 0.763/0.551 | 0.727/0.487 | 0.771/0.520 | 0.737/0.490 | 0.743/0.490 | 0.796/0.543 |
| ALL     | 0.610/0.435 | 0.295/0.197 | 0.756/0.536 | 0.718/0.485 | 0.691/0.438 | 0.761/0.504 | 0.741/0.483 | 0.756/0.493 | 0.772/0.513 |

whose difference between the output position and the ground truth bounding box is within the threshold. The threshold of GROT is set to 5, and the threshold of RGBT234 is set to 20. SR is the proportion of frames where the overlap ratio between the output position and the ground truth bounding box is greater than the threshold. The area under the curves (AUC) is employed to calculate the SR score.

B. Comparison With Other Methods

We compared DFNet with visible tracking methods (MDNet [53], RT-MDNet [48], SiamFC [54], and SiamRPN [55]) and fusion tracking methods (pixel-level fusion [17], feature-level fusion [18], [19], MANet [21], and IVFuseNet [36]). To be fair, all fusion methods have been implemented on the RT-MDNet tracking framework, represented below as RTMDNet-pixel, RTMDNet-feature, RTMDNet-MANet, and RTMDNet-IVFuseNet, respectively. The overall tracking performance is shown in Figure 4. For three indicators of these two benchmarks, our DFNet has better results than other tracking methods. Specifically, on the GROT benchmark, our DFNet reached 88.1%/71.9% on PR/SR. While on the RGBT234 benchmark, our DFNet reached PR/SR 77.2%/51.3%. To further show the effectiveness of DFNet, we list the performance of each attribute of the RGBT234 and GROT datasets. The specific tracking results are shown in Table II and Table III. It can be concluded from the table that our proposed DFNet is better than other trackers in 15 cases with higher PR or SR.

C. Ablation Study

In order to show the importance of fusion for tracking, we compared the tracking performance of DFNet+RGB, DFNet+T, and DFNet. DFNet+RGB and DFNet+T indicate that DFNet solely relies on visible or infrared images for tracking. The tracking performance of DFNet is shown in Figure 5. The PR/SR of RGBT tracking is 8.6%/8.0% higher than tracking using visible image alone, and 21.4%/16.8% higher than tracking using infrared image alone. Experimental results show that the performance of DFNet is significantly better than that of methods based on single-modal images.


### Table III

| Tracker | SiamFC | SiamRPN | MDNet | RTMDNet | RTMDNet | RTMDNet | RTMDNet |
|---------|--------|---------|--------|---------|---------|---------|---------|
|         |        |         |        |         | -pixel | -feature | -FuseNet | -MANet |
| OCC     | 0.665/0.539 | 0.337/0.282 | 0.684/0.669 | 0.829/0.636 | 0.737/0.592 | 0.850/0.666 | 0.864/0.674 | 0.859/0.668 |
| LSV     | 0.760/0.609 | 0.412/0.395 | 0.857/0.679 | 0.828/0.658 | 0.792/0.635 | 0.814/0.670 | 0.826/0.671 | 0.823/0.672 |
| FM      | 0.680/0.562 | 0.352/0.356 | 0.851/0.663 | 0.788/0.645 | 0.728/0.610 | 0.768/0.630 | 0.809/0.631 | 0.773/0.627 |
| LI      | 0.595/0.494 | 0.337/0.316 | 0.853/0.630 | 0.755/0.618 | 0.820/0.671 | 0.882/0.704 | 0.835/0.698 | 0.896/0.709 |
| TC      | 0.734/0.579 | 0.371/0.348 | 0.864/0.677 | 0.837/0.658 | 0.762/0.621 | 0.862/0.687 | 0.899/0.697 | 0.873/0.686 |
| SO      | 0.709/0.525 | 0.383/0.318 | 0.896/0.663 | 0.874/0.636 | 0.827/0.625 | 0.928/0.687 | 0.940/0.702 | 0.928/0.687 |
| DNF     | 0.539/0.462 | 0.303/0.283 | 0.847/0.687 | 0.738/0.614 | 0.860/0.698 | 0.917/0.726 | 0.929/0.735 | 0.924/0.726 |
| ALL     | 0.645/0.530 | 0.344/0.328 | 0.841/0.681 | 0.790/0.639 | 0.800/0.650 | 0.860/0.695 | 0.831/0.699 | 0.871/0.696 |

The best, second-best, and third-best PR/SR are shown in **red**, yellow, and blue, respectively.

---

**Fig. 6.** Weights for different video sequences across the GTOT dataset in DFNet. (a) Weights of visible images. (b) Weights of infrared images.

---

### Table IV

**Dynamic Calculating the Contributions and Fusion in the Kernel Space Implemented on RT-MDNNet**

| Network   | Dynamic calculating the contributions | Fusion in the kernel space | PR  | SR  | PPS |
|-----------|--------------------------------------|---------------------------|-----|-----|-----|
| RTMDNet-MANet | √                                      | √                          | 0.756 | 0.493 | 25.221 |
| DFNet     | √                                      | √                          | 0.753 | 0.485 | 28.871 |

| Network | C1  | C2  | C3  | PR  | SR  |
|---------|-----|-----|-----|-----|-----|
| Feature-level fusion | √   | √   | √   | 0.860 | 0.693 |
| Feature-level fusion | √   | √   | √   | 0.863 | 0.695 |
| Dynamic fusion layer | √   | √   | √   | 0.866 | 0.691 |
| Dynamic fusion layer | √   | √   | √   | 0.867 | 0.699 |
| DFNet   | √   | √   | √   | 0.881 | 0.709 |

Using dynamic fusion layers for all three layers produces the best results. And the later the dynamic fusion layer is used in the modal, the better the performance is.

---

**Fig. 7.** Visible and infrared images and dynamic fusion layer weights of different frames of (a) OccCar-2 and (b) FastMotorNig. (a) Visible images and their weight trend. (b) Infrared images and their weight trend.

We visualized the weights of the dynamic fusion layer in the order of the videos in GTOT, as shown in Figure 6.
In Figure 6(a), the solid blue line represents the average of the visible non-shared convolution kernel weights, and the blue shading represents the range of the visible non-shared convolution kernel weights. The solid red line represents the average of the visible shared convolution kernel weights, and the red shade represents the range of the visible shared convolution kernel weights. In Figure 6(b), the solid red line represents the average of the infrared non-shared convolution kernel weights, and the red shade represents the range of the infrared non-shared convolution kernel weights. The solid blue line represents the average of the infrared shared convolution kernel weights, and the blue shading represents the range of the infrared shared convolution kernel weights. It can be found that the dynamic fusion layer can calculate different weights according to videos. In this way, the dynamic fusion layer makes the fusion tracker adaptively calculates the contributions of individual and common features.

In addition, we visualized the weights of the dynamic fusion layer in two video sequences, as shown in Figure 7. Figure 7(a) is from OccCar-2. It can be found that in this video sequence, the contributions of individual and common features are also different. At the beginning of the video, the car is clear in the visible images, and the contributions of individual features of visible images are significant. When the car is blocked by leaves, visible images cannot clearly distinguish the car, so the contributions of individual features reduce. In contrast, the contributions of the common features increase. In the later stage of the video sequence, because the bicycle is close to the crowd, thermal crossover [51] occurs, so the contributions of the individual features of the infrared images are low. As the bicycle moves away from the crowd, the contributions of the individual features of the infrared images increase.

**D. Comparison Over Time**

We select three videos from GTOT, including car, motorcycle, and pedestrian, calculate the IOU of different methods and compare them over time. The IOU for cars (the first row), motorbikes (the second row), and pedestrians (the third row) over time are shown in Figure 8. The first row shows IOUs for cars (the first column), motorbikes (the second column), and pedestrians (the third column) over time. The second row shows the corresponding boxplots. It can be found from the figure that DFNet can maintain a high level of IOU compared with other methods, and DFNet also has less fluctuation. This is because DFNet can adaptively calculate the contributions of individual and common features.

**E. Efficiency Analysis**

The speed of DFNet is 28.658 FPS. We compared the speed and performance of DFNet to other fusion methods on RGBT234. The results are shown in Figure 9. The computational cost of DFNet is $O(X) = 2(HWC_{in} + C_{in}Ch_{idden} + 2C_{hidden} + HWC_{out}C_{out}k^2)$ Mult-Adds, where $H$, $W$ are the...
height and width of the input; \( C_{in} \), \( C_{out} \), and \( C_{hidden} \) are the channel numbers of the input, output, and hidden layer, respectively; \( k \) is the size of the convolution kernel. Correspondingly, the computational cost of baseline (RTMDNet-feature) is \( O(X) = 2HW C_{in} C_{out} k^2 \) Mult-Adds. Since the fusion of shared and non-shared convolution kernels is performed in convolution kernel space, compared with the non-shared-convolution-kernel-based fusion method, DFNet here has no additional calculations to increase.

The increase in computational cost is from the attention module, which calculates weights according to the input. However, the computational cost caused by the attention module is much smaller than convolution. In DFNet, it is less than 0.02%. Compared with the baseline, MANet fuses the shared and non-shared features in feature space, which increases 8.85% of calculations. The specific computational cost is shown in Table VI, where “C1”, “C2”, “C3”, and “total” indicate the computational cost of the first, second, third, and all convolutional layers, respectively; “percent” indicate computational cost expressed as a percentage of the non-shared-convolution-kernel-based fusion method.

### Table VI

| Model                       | C1 (MB) | C2 (MB) | C3 (MB) | total (MB) | percent |
|-----------------------------|---------|---------|---------|------------|---------|
| RTMDNet-Manet               | 323.14M | 768.00M | 265.47M | 1376.51M   | 100.00% |
| RTMDNet-IVFuseNet          | 323.14M | 768.00M | 265.47M | 1376.51M   | 100.00% |
| RTMDNet-MANet              | 323.14M | 768.00M | 265.47M | 1376.51M   | 100.00% |
| DFNet (ours)               | 323.14M | 768.00M | 265.47M | 1376.51M   | 100.00% |

Based on all the experiments performed in this section, we conclude that:

1) Compared with the visible tracking method (MDNet, RT-MDNet, SiamFC, and SiamRPN) and the fusion tracking method (pixel-level fusion, feature-level fusion, MANet, and IVFuseNet), DFNet achieves the best PR and SR.

2) The performance of the fusion method is better than that of methods based on single-modal images, which shows the advantage of fusion.

3) With consideration of the contributions of individual and common features, DFNet can adaptively calculate the weights of shared and non-shared convolution kernels to cope with changes in modality reliability.

4) Compared with the fusion of shared and non-shared features in the feature space, the fusion of shared and non-shared convolution kernels in the convolution kernel space can effectively reduce the computational complexity and improve the tracking speed.

### VI. Conclusion

In this paper, we propose a novel RGBT tracking method called dynamic fusion network (DFNet). DFNet is essentially a feature-level fusion method, which can use non-shared convolutions to respectively extract individual features according to the different characteristics of visible and infrared images. Furthermore, DFNet takes advantage of shared convolution kernels to extract common features. In addition, because attention is used to adaptively calculate different convolution kernel weights according to inputs, DFNet can dynamically calculate the contributions of individual and common features in the face of changes in modality reliability. The shared convolution kernels and non-shared convolution kernels are concatenated in convolution kernel space, so the computational cost is small. Extensive experiments on two RGBT datasets validate the effectiveness of DFNet. Future work will focus on adopting more advanced architectures, designing other adaptive weighting methods, and reducing the redundancy of features between different modalities.
We performed DFNet implemented on MDNet and compared it with other SOTA approaches. The overall tracking performance is shown in Figure 10. It can be seen from the figure that DFNet achieves the best results when implemented on MDNet, which is often used by other SOTA approaches. Our DFNet reached 82.5%/60.8% on RGBT234, and 90.2%/74.0% on GTOT.

Fig. 10. Overall performance compared with SOTA approaches on RGBT234 (a) and GTOT (b). (a) Comparison on RGBT234. (b) Comparison on GTOT.

REFERENCES

[1] Z. Li, K. Nai, G. Li, and S. Jiang, “Learning a dynamic feature fusion tracker for object tracking,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 2, pp. 1479–1491, Feb. 2022.

[2] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, “Deep learning for visual tracking: A comprehensive survey,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 5, pp. 3943–3968, May 2022.

[3] Y. Wang, X. Wei, X. Tang, H. Shen, and H. Zhang, “Adaptive fusion CNN features for RGBT object tracking,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 7, pp. 7831–7840, Jul. 2022.

[4] S. Zhai, P. Shao, X. Liang, and X. Wang, “Fast RGB-T tracking via cross-modal correlation filters,” Neurocomputing, vol. 334, pp. 172–181, Mar. 2019.

[5] C. Wang et al., “Cross-modal pattern-propagation for RGB-T tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 7671–7679.

[6] X. Zhang, P. Ye, H. Leung, K. Gong, and G. Xiao, “Object fusion tracking based on visible and infrared images: A comprehensive review,” Inf. Fusion, vol. 63, pp. 166–187, Nov. 2020.

[7] A. Berg, J. Ahlberg, and M. Felsberg, “A thermal object tracking benchmark,” in Proc. 12th IEEE Int. Conf. Adv. Video Signal Based Survell. (AVSS), Aug. 2015, pp. 1–6.

[8] X. Li, Q. Liu, N. Fan, Z. He, and H. Wang, “Hierarchical spatial-aware Siamese network for thermal infrared object tracking,” Knowl.-Based Syst., vol. 166, pp. 71–81, Feb. 2019.

[9] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen, “Deep convolutional neural networks for thermal infrared object tracking,” Knowl.-Based Syst., vol. 134, pp. 189–198, Oct. 2017.

[10] L. Zhang, A. Gonzalez-Garcia, J. Van De Weijer, M. Danelljan, and F. S. Khan, “Synthetic data generation for end-to-end thermal infrared tracking,” IEEE Trans. Image Process., vol. 28, no. 4, pp. 1837–1850, Apr. 2019.

[11] L. Peng, X. Hu, W. Tian, H. Wang, D. Cao, and F.-Y. Yang, “Parallel planning: A new motion planning framework for autonomous driving,” IEEE/CAA J. Autom. Sinica, vol. 6, no. 1, pp. 236–246, Jan. 2019.

[12] Z. Chen et al., “Vehicle detection in high-resolution aerial images based on fast sparse representation classification and multiorder feature,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 8, pp. 2296–2309, Aug. 2016.

[13] D. Shen, T. Spisak, Y. Chen, and R. Tian, “A practical view of the similarity and differences among the impaired driver states in legal driving,” in HCI International-Late Breaking Papers, Digital Human Modeling and Ergonomics, Mobility and Intelligent Environments (Lecture Notes in Computer Science), C. Stephanidis, V. G. Duffy, N. Streitz, S. Konomi, and H. Kromker, Eds. Cham, Switzerland: Springer, 2020, pp. 207–220.

[14] Y. Wang, M. Papageorgiou, and A. Messmer, “Renaissance: A real-time freeway network traffic surveillance tool,” in Proc. IEEE Intell. Transp. Syst. Conf., Sep. 2006, pp. 839–844.

[15] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, “Person re-identification by multi-channel parts-based CNN with improved triplet loss function,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1335–1344.

[16] Z. Zhou, W. Pan, Q. M. Jonathan Wu, C. Yang, and Z. Lv, “Geometric-recification-based neural network architecture for image manipulation detection,” Int. J. Intell. Syst., vol. 36, no. 12, pp. 6993–7016, Dec. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22577

[17] X. Zhang, P. Ye, D. Qiao, J. Zhao, S. Peng, and G. Xiao, “Object fusion tracking based on visible and infrared images using fully convolutional Siamese networks,” in Proc. 22nd Int. Conf. Inf. Fusion, Jul. 2019, pp. 1–8.

[18] X. Zhang, X. Zhang, X. Du, X. Zhou, and J. Yin, “Learning multi-domain convolutional network for RGB-T visual tracking,” in Proc. 11th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI), Oct. 2018, pp. 1–6.

[19] X. Zhang, P. Ye, S. Peng, J. Liu, K. Gong, and G. Xiao, “SiamFT: An RGB-infrared fusion tracking method via fully convolutional Siamese networks,” IEEE Access, vol. 7, pp. 122122–122133, 2019.

[20] C. Li, X. Wu, N. Zhao, X. Cao, and J. Tang, “Fusing two-stream convolutional neural networks for RGB-T object tracking,” Neurocomputing, vol. 281, pp. 78–85, Mar. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217318271

[21] C. Li, A. Lu, A. Zheng, Z. Tu, and J. Tang, “Multi-adapter RGBT tracking,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, Oct. 2019, pp. 2262–2270.

[22] P. Zhang, J. Zhao, C. Bo, D. Wang, H. Lu, and X. Yang, “Jointly modeling motion and appearance cues for robust RGB-T tracking,” IEEE Trans. Image Process., vol. 30, pp. 3335–3347, 2021.

[23] X. Yun, Z. Jing, G. Xiao, B. Jin, and C. Zhang, “A compressive tracking based on time-space Kalman fusion model,” Sci. China Inf. Sci., vol. 59, no. 1, pp. 1–15, Jan. 2016.

[24] N. Cvejie et al., “The effect of pixel-level fusion on object tracking in multi-sensor surveillance video,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2007, pp. 1–7.

[25] F. Bunyak, K. Palaniappan, S. Nath, and G. Seetharaman, “Geodesic active contour based fusion of visible and infrared video for persistent object tracking,” in Proc. IEEE Workshop Appl. Comput. Vis. (WACV), Feb. 2007, p. 35.

[26] S. Mangale and M. Khambete, “Camouflaged target detection and tracking using thermal infrared and visible spectrum imaging,” in Proc. Int. Symp. Intell. Syst. Technol. Appl., 2016, pp. 193–207.

[27] X. Lan, M. Ye, S. Zhang, H. Zhou, and P. C. Yuen, “Modality-correlation-aware sparse representation for RGB-infrared object tracking,” Pattern Recognit., Lett., vol. 130, pp. 12–20, Jul. 2020.

[28] C. Li, C. Zhu, S. Zheng, B. Luo, and J. Tang, “Two-stage modality graphs regularized manifold ranking for RGB-T tracking,” Image Comput. Sci. China, vol. 38, no. 12, pp. 6993–7016, Dec. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22577

[29] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2790–2798.
[30] G. Chen et al., “A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 2, pp. 936–953, Feb. 2022.

[31] J. Peng, H. Zhao, Z. Hu, Y. Zhuang, and B. Wang, “Siamese infrared and visible light fusion network for RGB-T tracking,” 2021, arXiv:2103.07302.

[32] C. Li, L. Li, A. Lu, Q. Ji, and J. Tang, “Challenge-aware RGBT tracking,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 222–237.

[33] X. Zhang, P. Ye, S. Peng, J. Liu, and G. Xiao, “DSiamMFT: An RGB-T fusion tracking method via dynamic Siamese networks using multi-layer feature fusion,” Signal Process., Image Commun., vol. 84, May 2020, Art. no. 115736. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1057931720301354.

[34] Y. Zhu, C. Li, J. Tang, and B. Luo, “Quality-aware feature aggregation network for robust RGBT tracking,” IEEE Trans. Intell. Vehicles, vol. 6, no. 1, pp. 121–130, Mar. 2021.

[35] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang, “Studying very low resolution recognition using deep networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 4792–4800.

[36] Y. Li, H. Zhao, Z. Hu, Q. Wang, and Y. Chen, “IVFuseNet: Fusion of infrared and visible light images for depth prediction,” Inf. Fusion, vol. 58, pp. 1–12, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519301034.

[37] J. Pei, M. Zou, and L. Wang, “A local adaptive threshold noise detection linear interpolation filter (LALIF) for stripe noise removal in infrared images,” in Proc. IEEE 13th Int. Conf. Signal Process. (ICSP), Nov. 2016, pp. 681–686.

[38] W. Hu, Y. Zhuang, J. Bao, and Q. Zhao, “Effects of radiation-induced changes in low-frequency noise of infrared detectors,” in Proc. Int. Conf. Noise Fluctuations (ICNF), Jun. 2015, pp. 1–3.

[39] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, “CondConv: Conditionally parameterized convolutions for efficient inference,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–12.

[40] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, “Dynamic convolution: Attention over convolution kernels,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11027–11036.

[41] N. Ma, X. Zhang, J. Huang, and J. Sun, “WeightNet: Revisiting the design space of weight networks,” in Proc. Eur. Conf. Comput. Vis., A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 776–792.

[42] C. Yu, Y. Shao, C. Gao, and N. Sang, “CondNet: Conditional classifier for scene segmentation,” IEEE Signal Process. Lett., vol. 28, pp. 758–762, 2021.

[43] H. Jiang et al., “Synthesizing indoor scene layouts in complicated architecture using dynamic convolution networks,” Proc. ACM Comput. Graph. Interact. Techn., vol. 4, no. 1, pp. 1–16, Apr. 2021.

[44] X. Ma, X. Zhou, H. Huang, Z. Chai, X. Wei, and R. He, “Free-form image inpainting via contrastive attention network,” in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 9242–9249.

[45] Y. Li, G. Cao, and W. Cao, “A dynamic group equivariant convolutional networks for medical image analysis,” in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), Dec. 2020, pp. 1056–1062.

[46] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “RepVGG: Making VGG-style ConvNets great again,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 13733–13742.

[47] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, “Acquisition of localization confidence for accurate object detection,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 816–832.

[48] J. Jung, J. Son, M. Baek, and B. Han, “Real-time MDNet,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 89–104.

[49] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in Proc. Brit. Mach. Vis. Conf., 2014, pp. 1–11.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2009, pp. 248–255.

[51] C. Li, H. Cheng, S. Hu, X. Liu, J. Tang, and L. Lin, “Learning collaborative sparse representation for grayscale-thermal tracking,” IEEE Trans. Image Process., vol. 25, no. 12, pp. 5743–5756, Dec. 2016.

[52] C. Li, X. Liang, Y. Lu, N. Zhao, and J. Tang, “RGB-T object tracking: Benchmark and baseline,” Pattern Recognit., vol. 96, Dec. 2019, Art. no. 106977.

[53] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 4293–4302.

[54] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional Siamese networks for object tracking,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 850–865.

[55] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with Siamese region proposal network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jan. 2018, pp. 8971–8980.

Jingchao Peng received the B.S. degree from the East China University of Science and Technology, Shanghai, China, in 2019, where he is currently pursuing the Ph.D. degree. His major research interests include target detection and tracking, information fusion, and big data analysis.

Haitao Zhao received the Ph.D. degree in pattern recognition and intelligent system from the Nanjing University of Science and Technology, Nanjing, China, in 2003. He is currently a Professor with the East China University of Science and Technology, Shanghai, China. His current research interests are in the areas of pattern recognition, machine learning, and computer vision.

Zhengwei Hu is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China. His major research interests include big data analysis, machine learning, and pattern recognition.