BiCnet-TKS: Learning Efficient Spatial-Temporal Representation for Video Person Re-Identification

Ruibing Hou1,2, Hong Chang1,2, Bingpeng Ma2, Rui Huang3, Shiguang Shan1,2,4
1Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
2University of Chinese Academy of Sciences, Beijing, 100049, China
3Shenzhen Institute of Artificial Intelligence and Robotics for Society, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, China
4CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, 200031, China
ruibing.hou@vipl.ict.ac.cn, bpma@ucas.ac.cn, ruihuang@cuhk.edu.cn, {changhong, sgshan}@ict.ac.cn

Abstract

In this paper, we present an efficient spatial-temporal representation for video person re-identification (reID). Firstly, we propose a Bilateral Complementary Network (BiCnet) for spatial complementarity modeling. Specifically, BiCnet contains two branches. Detail Branch processes frames at original resolution to preserve the detailed visual clues, and Context Branch with a down-sampling strategy is employed to capture long-range contexts. On each branch, BiCnet appends multiple parallel and diverse attention modules to discover divergent body parts for consecutive frames, so as to obtain an integral characteristic of target identity. Furthermore, a Temporal Kernel Selection (TKS) block is designed to capture short-term as well as long-term temporal relations by an adaptive mode. TKS can be inserted into BiCnet at any depth to construct BiCnet-TKS for spatial-temporal modeling. Experimental results on multiple benchmarks show that BiCnet-TKS outperforms state-of-the-arts with about 50% less computations. The source code is available at https://github.com/blue-blue272/BiCnet-TKS.

1. Introduction

Person re-identification (reID) [34, 50, 11] aims at retrieving a particular person across multiple non-overlapped cameras. Recently, with the emergence of large video benchmarks [50, 20] and the growth of computational resource, video person reID has been attracting a lot of attention. The video data contain richer spatial and temporal clues, which can be utilized to reduce visual ambiguities for more robust reID.

Despite the significant progress in video reID, most existing methods only model

Figure 1: An example of class activation maps [53] of a pair of input video sequences of existing method [50] and our method.

existing methods do not take full advantage of the rich spatial-temporal clues in videos. For spatial clues, most methods [28, 26, 12] conduct the same operation on each frame at same input resolution, resulting in highly redundant spatial features for consecutive frames. The redundant features easily focus on the same most representative local region [14], which may be indistinguishable for the two persons with seemingly similar local body parts. For example, as shown in Fig. 1 (a), the green T-shirt of the sequence pair attracts the most attention, but is difficult to distinguish the two pedestrians. Therefore, it is desirable to automatically capture the diverse spatial clues across consecutive frames to form a full characteristic of each identify.

For temporal clues, most existing methods only model
Figure 2: Short and long-term temporal relations have varying importance for different sequences. (a) A sequence with partial occlusion. The long-term temporal clues are desired to alleviate occlusion. (b) A sequence of a fast-moving pedestrian. The short-term temporal clues are desired to model detailed motion patterns.

either short-term [28, 44, 9] or long-term temporal relations [39, 46, 13]. To enhance the temporal modeling ability, a few works [20, 21] attempt to jointly capture short and long-term temporal relations and fuse the two relations with equal weights. However, the two temporal relations have varying importance for different sequences. For example, as shown in Fig. 2, for a sequence with partial occlusion, the long-term temporal relations are more important to alleviate occlusion. For a fast-moving pedestrian sequence, the short-term temporal relations play a greater role to model the detailed motion patterns. So it is necessary to adaptively capture short and long-term temporal relations of videos.

To explicitly fulfill above goals, we present an efficient spatial-temporal representation for video reID. We first propose a Bilateral Complementary Network (BiCnet) to extract complementary spatial features across consecutive frames. Firstly, BiCnet contains two scale-specific branches, Detail Branch operating on frames at original resolution to retain spatial details, and Context Branch processing frames at down-sampled resolution to enlarge receptive field for long-range contexts. As shown in Fig. 1 (b), with larger receptive field, the third-frame feature of the first sequence can capture broader visual clues of a green T-shirt with a backpack strap on it, which can help differentiate the two similar pedestrians. Then on each branch, BiCnet appends multiple parallel spatial attention modules. By enforcing the diversity of individual attention modules, the attention modules can focus on different regions for consecutive frames. As shown in Fig. 1 (b), with the diverse attention modules, the consecutive-frame features from same branch can focus on complementary body regions, covering the whole body of the target identity. Finally, BiCnet aggregates the complementary features from the two branches to a comprehensive spatial representation.

Furthermore, we develop a Temporal Kernel Selection (TKS) block to adaptively model the short and long-term temporal relations. Utilizing both small kernel and large kernel along the temporal dimension can capture the short and long-term temporal relations simultaneously. So TKS is designed to contain several parallel temporal convolution paths with various kernel sizes. More importantly, TKS selects a dominant temporal scale according to the global information from the multiple paths. With the selection strategy, TKS can adaptively vary the scale of temporal modeling depending on the properties of input videos, thereby exhibiting stronger temporal representational capability. TKS is computationally lightweight and imposes a slight increase in model complexity. It can be readily inserted into BiCnet, called “BiCnet-TKS”, to progressively learn spatial-temporal patterns.

We evaluate our approach on multiple challenging video reID benchmarks. The evaluations show that our approach outperforms state-of-the-arts. Moreover, by down-sampling some frames to low-resolution, BiCnet-TKS greatly reduces the computations, requiring about 50% less computation cost than state-of-the-arts.

2. Related Work

Person ReID. Existing video reID methods mainly focus on exploiting rich spatial-temporal clues in videos. For spatial clues, most works [50, 43, 26, 49] apply temporal average pooling or a weighting strategy to fuse frame features. For temporal clues, existing methods use optical flow [28, 54, 43], recurrent neural network [28, 44, 37, 3], 3D convolution [24, 9] or non-local block [39, 12, 13] to model the temporal relations. Recently, the works [21, 20] propose to jointly capture short and long-term temporal relations. However, these methods fuse the two temporal relations with equal weights. In contrast, our TKS adaptively selects a dominate temporal relation based on the input video, exhibiting stronger temporal modeling capability.

The most similar work to ours BiCnet is TCLNet [14], which also extracts complementary features for consecutive frames. BiCnet has several advantages over it. First, TCLNet only considers one spatial scale to focus on local details, while our method is built on a two-branch architecture, which can capture both detailed features as well as long-range contexts. Second, TCLNet uses hard erasing to drop the salient features which may deteriorate the representation capacity, while our method adopts soft attention to flexibly determine the regions that should be attended to. Third, TCLNet uses multiple expensive CNNs to mine diverse parts. Our method uses diverse and lightweight attention modules with sharing CNNs, which is more computational efficient and parametric friendly.

Multi-branch Architecture. Multi-branch architecture has exhibited great success in image based vision tasks. For example, M3DNet [17] and HR-Nets [33] propose the networks that contain multiple branches and each branch has its own spatial resolution, respectively for image classification and pose estimation. The works [5, 25] propose a pyramidal feature learning network that consists of multiple scale-specific feature learning branches for image reID. However, above methods process each image at multiple resolutions, incurring additional computations. On the con-
Our approach uses an individual resolution for each frame which largely reduces the computational cost. Moreover, very few methods explore the multi-branch architecture for efficient video understanding. SlowFast Networks [8] rely on a similar two-branch structure, but each branch encodes different frame rates, while our method processes frames with different spatial resolutions.

**Attention Model.** Attention mechanism has proven to be a potential way to enhance CNNs. SENet [16] proposes an efficient channel attention module. CBAM [40] and BAM [29] further introduce spatial attention blocks. SKNet [23] brings the feature attention across two spatial convolutions. Recent methods [38, 7, 47, 15] further improve the channel attention block. However, existing methods are usually designed to enhance spatial representational capability. In contrast, our TKS adopts attention over different temporal kernels, which can boost the temporal representational power of video networks. Also, our BiCNet is the first work to use diverse attention modules across consecutive frames to enhance the video representation.

### 3. Our Approach

We aim at developing an efficient spatial-temporal representation for video reID. Our method includes two novel components, i.e., BiCNet for complementary spatial representations across consecutive frames, and TKS for adaptively modeling the short and long-term temporal relations.

#### 3.1. Bilateral Complementary Network

As shown in Fig. 1, most existing methods extract highly redundant features for consecutive frames that only highlight a local body part [14]. To this end, we design a Bilateral Complementary Network to mine complementary visual clues from consecutive frames. As shown in Fig. 3, BiCNet is built on a two-branch architecture and adds a Diverse Attentions Operation (DAO) on each branch. The two-branch architecture is used to model complementary scales for different video sub-segments, and DAO is utilized to mine complementary body parts for consecutive frames. By adding DAO on each branch, BiCNet can obtain an integral characteristic of the target person, producing a comprehensive spatial representation.

**Two-branch Architecture.** As shown in Fig. 3 (a), BiCNet contains two CNN branches, a Detail Branch processing former several frames of given video segment at original resolution and a Context Branch operating on remaining frames at half of original resolution. By down-sampling input frames to small size, Context Branch provides larger receptive field to encode long-range spatial contexts, which can complement the detailed features extracted by Detail Branch. Concretely, suppose a video segment $I = \{I_n\}_{n=1}^N$ contains $N$ consecutive frames and $n$ is the index of the video frame. We firstly divide $I$ into two sub-segments, namely big frames $I_B = \{I_n\}_{n=1}^M$ at original resolution, and small frames $I_S = \{I_n\}_{n=M+1}^N$ at half of the original resolution, where $M$ is a hyper-parameter that determines the ratio of the small frames to big frames $\alpha$. Then $I_B$ and $I_S$ are fed into Detail Branch (CNND) and Context Branch (CNNC) separately, to obtain the corresponding feature vectors $f_d$ and $f_c$ as follows:

$$f_d = \frac{1 + \alpha}{N} \sum_k \text{CNN}_D(I_k), I_k \in \{I_n\}_{n=1}^N$$

$$f_c = \frac{1 + \alpha}{\alpha N} \sum_k \text{CNN}_C(I_k), I_k \in \{I_n\}_{n=M+1}^N$$

(1)
Finally, we simply average \( f_d \) and \( f_c \) to obtain the video feature for recognizing.

**Cross-Scale Paths.** Further, we add Cross-Scale Paths (CSP) that propagate the intermediate information of Detail Branch to Context Branch. CSP enables Context Branch to aware the features extracted by Detail Branch, such that Context Branch can focus on exploiting long-range visual clues less activated by the other branch.

The structure of CSP is illustrated in Fig. 3 (a). Formally, let \( F_D \in \mathbb{R}^{M \times C \times H \times W} \) and \( F_C \in \mathbb{R}^{\alpha M \times C \times \frac{H}{2} \times \frac{W}{2}} \) be the intermediate video feature map extracted by the same stage of Detail Branch and Context Branch respectively, where \( C, H \) and \( W \) denote the number of channels, the height and the width of feature map of big frames respectively. \( F_D \) and \( F_C \) have different spatial and temporal dimensions, so CSP first performs transformation on \( F_D \) to \( \overline{F_D} \in \mathbb{R}^{\alpha M \times C \times \frac{H}{2} \times \frac{W}{2}} \) to match the size as:

\[
\overline{F_D} = R (W_c \ast P(F_D)).
\]

Here \( P \) is the pooling operation that performs max pooling with stride 2 to match the spatial dimension, \( \ast \) is the convolution operation, \( W_c \in \mathbb{R}^{1 \times 1 \times C \times \alpha C} \) is the parameter of the convolution operation, and \( R \) is the reshape operation reshaping the convolutional result with size \( M \times \alpha C \times \frac{H}{2} \times \frac{W}{2} \) to \( \alpha M \times C \times \frac{H}{2} \times \frac{W}{2} \) to match the temporal dimension. At last, \( \overline{F_D} \) is fused into \( F_C \) by element-wise summation.

**Diverse Attentions Operation.** As shown in Fig. 3 (a), although the big frames and small frames can provide some complementary clues (e.g., detailed T-shirt/additional long-distance knapsack strap feature), the frames on each branch still easily focus on around the most representational region (e.g., upper-clothes). To this end, we design Diverse Attentions Operation to mine discriminative features for consecutive frames. By adding DAO on each branch, BiCnet can discover abundant discriminative parts and produce an integral complementary characteristic of each identity.

As shown in Fig. 3 (b), DAO contains several parallel attention modules and uses a specific attention module for each frame. By encouraging diversity among the generated attention maps, the attention modules can attend to complementary parts, so as to acquire diverse discriminative features for the consecutive frames.

In particular, DAO takes \( F_D \) (or \( F_C \)) as input, and uses a specific attention module for each frame feature map \( (F_D)_k \in \mathbb{R}^{C \times H \times W} \). We take \( F_D \) as an example, and denote \( (F_D)_k \) as \( D_k \) for simplicity. Firstly, as pointed out by [6], the intensity of each pixel in high-level feature map is proportional to the discriminative power. So we compress \( D_1 \) by channel-wise average pooling to locate the region activated by \( D_1 \), producing a self-attention map \( A_1 \in \mathbb{R}^{H \times W} \):

\[
A_k = \text{softmax} \left( \frac{1}{C} \sum_{c=1}^{C} (D_k)_c \right), \quad k = 1,
\]

Then we introduce parallel attention modules to learn to mine different and non-activated regions. Specifically, given \( D_k \) (\( k \geq 1 \)), the corresponding attention module first takes a convolutional layer to compress the channel dimension and reshapes the result to \( \mathbb{R}^{H \times W} \). After that a fully-connected layer is applied to embed the global spatial context. Finally, the result is reshaped to \( \mathbb{R}^{H \times W} \) followed by a softmax layer to produce corresponding attention map \( A_k \in \mathbb{R}^{H \times W} \) (\( k > 1 \)).

In order to guide different attention modules to activate diverse regions, the corresponding spatial attention maps should be different. To achieve this, a divergence regularization term is introduced to measure the diversity of two attention maps \( A_k \) and \( A_l \), which is defined as:

\[
L_{\text{div}}(A_k | A_l) = 1 - \text{sim}(A_k, A_l),
\]

where \( \text{sim}(A_k, A_l) \) computes the similarity of \( A_k \) and \( A_l \). Any distance measure is applicable, and we use the dot-product similarity [39] since dot-product is more implementation-friendly in modern deep learning platforms. Then the divergence loss is calculated as:

\[
L = \frac{-1}{M - 1} \sum_{k=2}^{M} \left( \frac{1}{k - 1} \sum_{l=1}^{k-1} L_{\text{div}}(A_k | A_l) \right).
\]

\( L \) is used to guide the optimization of parallel attention modules. When any two attention modules focus on similar person region, the generated attention maps would have a low diversity value, producing a high loss value \( L \). So optimizing with \( L \) can drive the different attention modules to focus on different person regions. Next, we encode the diverse attention information into input feature maps by a residual operation.

At last, the updated feature maps are fed into the subsequent convolutional layers to generate feature vectors embedded with complementary visual clues.

**3.2. Temporal Kernel Selection Block**

Following [30, 42], we factor the video network to treat spatial clues and temporal relations separately. With the efficient BiCnet to fully mine the spatial clues, we build a Temporal Kernel Selection block to jointly model the short-term and long-term temporal relations. Since the temporal relations with different scales have varying importance for different sequences (as illustrated in Fig. 2), TKS combines the multi-scale temporal relations in a dynamic way, i.e., different weights are assigned to different temporal scales according to input sequences.

In particular, TKS takes a sequence of consecutive-frame feature maps \( F = \{ F_t \}_{t=1}^{T} \) as input, where \( F_t \) is the feature map of the \( t \)-th frame, and conducts a triple of operations, Partition, Select and Excite on \( F \).
Temporal Kernel $3 \times 1 \times 1$

Temporal Kernel $5 \times 1 \times 1$

Temporal Kernel $2K + 1 \times 1 \times 1$

Weighted Sum

GAP

⋮

⋮

$X$

$u$

$Z$

$g_1$

$g_2$

$g_K$

⋮

Reshape

Figure 4: The architecture of Temporal Kernel Selection block.

**Partition Operation.** Due to imperfect person detection algorithm, the adjacent frames of a video are not well aligned, which might make the temporal convolution ineffective on video reID [9]. Following [34], we use the partition strategy to alleviate the spatial misalignment issue. Specifically, given video feature map $\{F_i\}_{i=1}^T$, we divide each frame feature map into $h \times w$ spatial regions uniformly, and perform average pooling on each divided region to construct a region-level video feature map $X \in \mathbb{R}^{T \times C \times h \times w}$.

**Select Operation.** As shown in Figure 4, given $X$, we conduct $K$ parallel paths $\{\mathcal{F}(i) : X \rightarrow Y^{(i)} \in \mathbb{R}^{T \times C \times h \times w}\}_{i=1}^K$, where $\mathcal{F}(i)$ is 1D temporal convolution [30] with kernel size $2i + 1$. For further efficiency, the temporal convolution with a $(2i + 1) \times 1 \times 1$ kernel is replaced with dilated convolution with a $3 \times 1 \times 1$ kernel and dilation size $i$. The basic idea of select operation is to use global information from all temporal paths to determine the assigned weights to each path. In particular, we first fuse the outputs of all paths by element-wise summation, then perform global average pooling to obtain a global feature

$$u = \text{GAP}_{T,h,w}\left(\sum_{i=1}^{K} Y^{(i)}\right),$$

(6)

where $\text{GAP}_{T,h,w}$ denotes global average pooling along the temporal and spatial dimension. After that the channel selection weights $\{g_i \in \mathbb{R}^{C \times 1}\}_{i=1}^K$ are obtained according to the global embedding $u$,

$$g_i = \frac{\exp (W_iu)}{\sum_{j=1}^{K} \exp (W_ju)} \quad i \in \{1, \ldots, K\},$$

(7)

where $W_i \in \mathbb{R}^{C \times C}$ is the transformed parameters to generate $g_i$ for $Y^{(i)}$. The aggregated feature map $Z \in \mathbb{R}^{T \times C \times h \times w}$ is then obtained through the selection weights on various temporal kernels,

$$Z = \sum_{i=1}^{K} \mathcal{R}(g_i) \odot Y^{(i)},$$

(8)

where $\mathcal{R}$ is the reshape operation reshaping $g_i \in \mathbb{R}^{C \times 1}$ to $\mathbb{R}^{1 \times C \times 1 \times 1}$ to be compatible with the size of $Y^{(i)}$.

It is worth pointing out that, in contrast to using scale-wise weight to provide coarse fusion, we choose to use channel-wise weights (Eq. 7) for fusing. This design results in more fine-grained fusion that tunes each feature channel. In addition, the weights are dynamically computed conditioned on input videos. This is crucial for reID where different sequences may have different dominate temporal scales.

**Excite Operation.** The excite operation modulates the input feature map by conditioning on $Z$ with a residual scheme. The final feature map $E = \{E_i\}_{i=1}^T$ is obtained as:

$$E_i = U(Z_i) + F_i.$$ Here $U$ is the nearest neighbor upsampler that performs upsampling on $Z_i$ to match the spatial resolution of $F_i$. TKS block maintains the input size, thus can be inserted at any depth of BiCnet to extract efficient spatial-temporal feature.

### 3.3 Overall Architecture

Our idea of BiCnet is generic, and it can be instantiated with different backbones [36, 35, 10]. Following recent works [9, 20, 32], we use ResNet-50 [10] pretrained on ImageNet [19] with last down-sampling operation removed as the backbone. The branches of BiCnet are built on ResNet-50 that consists of four consecutive stages, i.e., stage$1$ ~ stage$4$. Diverse Attentions Operation is added after stage$3$ since the high-level feature maps contain more semantic information. TKS block can be inserted into BiCnet to any stage to construct BiCnet-TKS for spatial-temporal modeling.

**Structure and Weight Sharing between Branches.** An immediate problem of multi-branch architecture [5] is that it introduces several times parameters and incurs a higher risk of overfitting. So we use the same structure and share the parameters for the two branches of BiCnet. It reduces the number of parameters and makes BiCnet need no extra parameters over single-branch reID network.

**Computation Cost Analysis.** To illustrate the computation cost of BiCnet-TKS, we consider a common video reID Baseline [50] that uses ResNet-50 to extract feature for each frame at original resolution. Assume that the FLOPs for Baseline to extract one-frame feature is $p$. Baseline requires $Np$ FLOPs to process a video with $N$ frames. BiCnet-TKS splits the video frames to big frames at original resolution and small frames at half of original resolution by a ratio $1 : \alpha$ (Eq. 1). So BiCnet-TKS requires about $\frac{N}{1+\alpha} + \left(\frac{\alpha N}{1+\alpha}\right)3$ FLOPs$^3$, corresponding to about $\frac{3}{4} - \frac{3}{4\alpha+1}$ relative decrease over Baseline.

We can see that the the computation cost decreases as $\alpha$ increases. However, when $\alpha$ is too large, the small frames would dominate the network optimization, causing a severe performance drop. We experimentally observe that setting $\alpha$ to 3 offers the best trade-off between computation

---

1The computations of CSP, DAO and TKS are negligible compared to the feature extraction of ResNet-50.
Table 1: Comparison with state-of-the-arts on MARS, DukeMTMC-VideoReID and LS-VID datasets. The methods are separated into three groups, mainly for spatial (S), temporal (T) and spatial-temporal (ST) modeling.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MARS mAP top-1</th>
<th>Duke-Video mAP top-1</th>
<th>LS-VID mAP top-1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COSAM* [32]</td>
<td>79.9 84.9</td>
<td>94.1 95.4</td>
<td>-</td>
</tr>
<tr>
<td>MGRAFA [48]</td>
<td>85.9 88.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Two-stream [31]</td>
<td>-</td>
<td>-</td>
<td>32.1 48.2</td>
</tr>
<tr>
<td>STMP [27]</td>
<td>72.7 84.4</td>
<td>-</td>
<td>39.1 56.8</td>
</tr>
<tr>
<td>M3D [21]</td>
<td>74.1 84.4</td>
<td>-</td>
<td>40.1 57.7</td>
</tr>
<tr>
<td>GLTP [20]</td>
<td>78.5 87.0</td>
<td>93.7 96.3</td>
<td>44.3 63.1</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRSA [22]</td>
<td>65.8 82.3</td>
<td>-</td>
<td>37.8 55.8</td>
</tr>
<tr>
<td>VRSTC [12]</td>
<td>82.3 88.5</td>
<td>93.5 95.0</td>
<td>-</td>
</tr>
<tr>
<td>I3D [2]</td>
<td>83.0 88.6</td>
<td>-</td>
<td>33.9 51.0</td>
</tr>
<tr>
<td>P3D [30]</td>
<td>83.2 88.9</td>
<td>-</td>
<td>35.0 53.4</td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STGCN [46]</td>
<td>83.7 89.9</td>
<td>95.7 97.3</td>
<td>-</td>
</tr>
<tr>
<td>IANet [13]</td>
<td>85.0 90.2</td>
<td>96.1 96.9</td>
<td>-</td>
</tr>
<tr>
<td>TCLNet [14]</td>
<td>85.1 89.8</td>
<td>96.2 96.9</td>
<td>70.3 81.5</td>
</tr>
<tr>
<td>AP3D [9]</td>
<td>85.1 90.1</td>
<td>95.6 96.3</td>
<td>73.2 84.5</td>
</tr>
<tr>
<td>MGH [45]</td>
<td>85.8 90.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiCnet-TKS</td>
<td>86.0 90.2</td>
<td>96.1 96.3</td>
<td>75.1 84.6</td>
</tr>
</tbody>
</table>

cost and accuracy. In this case, BiCnet-TKS only requires ~44% computation costs over Baseline, which is more efficient to extract the spatial-temporal feature.

4. Experiment

4.1. Dataset and Settings

Datasets. We evaluate the proposed method on multiple video reID datasets, i.e., MARS [50], DukeMTMC-VideoReID [41] and LS-VID [20].

Evaluation Metric. We adopt mean Average Precision (mAP) [51] and Cumulative Matching Characteristics (CMC) [1] as evaluation metrics.

Implementation Details. During training, for each video sequence, we randomly sample 8 frames with a stride of four frames to form a video segment. Each batch contains 16 persons, each person with 4 video segments. We resize the split big frames to $256 \times 128$ and small frames to $128 \times 64$. The horizontal flip and random erasing [52] are adopted for data augmentation. As for the optimizer, Adam [18] with weight decay 0.0005 is adopted to update the parameters. We train the model for 150 epochs in total. The learning rate is initialized to $3.5 \times 10^{-4}$ with a decay factor 0.1 at every 40 epochs. In BiCnet, the ratio of small frames to big frames is set to 3. In TKS, the number of temporal kernels is set to 2, and the divided regions is $4 \times 2$.

During testing, for each video sequence, we first split it into several 8-frame video segments. Then we extract the feature for each video segment by BiCnet-TKS and the final video feature is the averaged representation of all segments. After feature extraction, the cosine distances between the query and gallery features are computed for retrieval.

4.2. Comparison with State-of-the-arts

In Tab. 1, we compare our method with state-of-the-arts on MARS and DukeMTMC-VideoReID and LS-VID datasets. Our method achieves the best performance. It is noted that: (1) The spatial-based methods [32, 4, 48] process each frame by same operation and resolution, so they do not fully consider the spatial redundancy between frames. On the contrary, our BiCnet ensures different frames to focus on divergent regions to form an integral person representation and achieves better performance. (2) Our method outperforms TCLNet [14], with an improvement up to 4.8% mAP on LS-VID dataset. The significant improvements can be attributed to the use of two-branch architecture and flexible soft attention modules. (3) The temporal-based methods [3, 24, 9] lack the ability of modeling both short and long-term temporal relations. Our method outperforms these methods with an 1% mAP improvement on MARS. (4) The methods [20, 21, 45] aggregate the multi-scale temporal relations with equal weights. Our method achieves better performance by an adaptive selection mechanism. (5) All existing methods add computations over Baseline. In contrast, our method greatly reduces the computation cost by processing some frames at low-resolution. Overall, our method outperforms state-of-the-arts with about 50% computation budgets.

4.3. Ablation Study

In this section, we respectively investigate the effectiveness of BiCnet and TKS block by conducting a series of ablation studies on MARS dataset.
4.3.1 The components of BiCnet.

To validate the effectiveness of BiCnet, we introduce a baseline that adopts ResNet-50 with temporal average pooling to generate the video feature. The baseline processes all frames at the same resolution and is trained with cross entropy and triplet loss. We consider two baseline models, i.e., Base-B processing frames at original resolution (256×128), and Base-S processing frames at half of original resolution (128×64). The comparisons are shown in Tab. 2.

The influence of branch number. BiCnet is built on a two-branch architecture. It is easy to extend to multiple branches case which splits the video frames into multiple groups and uses an individual resolution for each group. In this part, we conduct an uniform split for fair comparison. The results are shown in Tab. 3. From Tab. 3, we have following observations: (1) Training ResNet-50 on frames at 128×64 resolution still offers reasonable accuracy, while saving 75% computations (measured by floating point operations). (2) Too small input resolution (64×32) causes severe performance degradation, with a drop up to 21% mAP. We argue that too small input size leads to serious loss of spatial details, which is difficult to distinguish pedestrians with small inter-class variations. (3) The three-branch architecture performs worse than two-branch structure. It is likely that the branch with 64×32 input resolution would disturb the optimization of network parameters. So we use a two-branch architecture, which can achieve comparable performance to Base-B with less computations.

Two-branch architecture w.r.t split ratio. We then investigate the influence of the split ratio α (in Eq. 1), i.e., the ratio of small frames (128×64) to big frames (256×128), to the two-branch architecture (TB). The results are shown in Tab. 4. We can observe that with α increases, TB greatly reduces the average computations of processing one frame. But the mAP of TB decreases as α increases. We argue that it is due to the lack of interaction between the two branches. In particular, the two branches of TB independently extract features, so it is difficult for one branch to learn to capture the clues ignored by the other branch. Moreover, the feature discriminative power of low-resolution frames is lower than that of high-resolution frames. So directly using low-resolution frames inevitably weakens the discrimination of final features. In addition, we observe that α=3 only brings slight drop compared to α=2. Considering computational complexity, we set α to 3 in this work.

Effectiveness of Cross-Scale Paths. We evaluate the effect of CSP by adding it after each stage of above two-branch architecture. As shown in Tab. 2, compared with TB, employing CSP brings 0.7% mAP gains with small computational overhead. We argue that with the information propagation from Detail Branch to Context Branch, Context Branch can enhance its representational power. In addition, the two branches can learn to work collaboratively to mine complementary clues, i.e., Detail Branch extracts the detailed feature of local body parts, and Context Branch focuses more on the long-distance contexts, to further enhance the feature representation.

Effectiveness of Diverse Attentions Operation. Finally, we investigate the individual effect of the attention modules and divergence constraint on DAO. The results are presented in Tab. 2. The difference between TB+CSP+AO and TB+CSP+DAO is that TB+CSP+AO appends parallel attention modules without L1 to guide optimization. As shown in Tab. 2, TB+CSP+AO achieves negligible gains over TB+CSP, which indicates that the visual features captured by different attention modules are almost the same. TB+CSP+DAO achieves 0.6% mAP improve-
It can be seen that the improvements by placing one TKS block in stage2 and stage3 are similar. However, placing TKS block in stage1 and stage4 leads to performance degradation. It is likely that the low-level features in stage1 are insufficient to provide precise semantic information, thus TKS can not model temporal relations between body parts very well. And since BiCnet learns to focus on different regions for consecutive frames on stage3, the frame features on stage4 lack of coherent temporal relations, so TKS is not capable to extract an effective temporal feature on stage4. We also observe that adding more TKS blocks does not bring gain, indicating that a TKS block is usually enough for temporal modeling.

Time Overhead. The running times are positively correlated with computation cost of models. In Tab. 2, Base-B takes 11ms to extract feature for a 8-frames sequence. While BiCnet-TKS only takes 6ms, corresponding to a 45.4% relative decrease over Base-B (both timings are performed on one NVIDIA 2080Ti GPU).

5. Conclusions

In this work, we present a computation-friendly spatial-temporal representation for video reID. Firstly, we introduce Bilateral Complementary Network. BiCnet contains two branches, Detail Branch preserving the spatial detail clues from original resolution, and Context Branch utilizing down-sampling operation to enlarge receptive field for longer-range contexts modeling. On each branch, BiCnet performs on one NVIDIA 2080Ti GPU).
References


