Supplementary Material: Channel Recurrent Attention Networks for Video Pedestrian Retrieval

Pengfei Fang$^{1,2[0000-0001-8939-0460]}$, Pan Ji$^{\star 3[0000-0001-6213-554X]}$, Jieming Zhou$^{1,2[0000-0002-3880-6160]}$, Lars Petersson$^{2[0000-0002-0103-1904]}$, and Mehrtash Harandi$^{4[0000-0002-6937-6300]}$

[1] The Australian National University
Pengfei.Fang@anu.edu.au
[2] DATA61-CSIRO
[3] OPPO US Research Center
[4] Monash University

1 Channel Recurrent Attention Module Analysis

In the channel recurrent attention module, we use an LSTM to jointly capture spatial and channel information. In the implementation, we feed the spatial vectors to the LSTM sequentially, such that the recurrent operation of the LSTM captures the channel pattern while the FC layer in the LSTM has a global receptive field of each spatial slice. Since the LSTM is a temporal model and its output depends on the order of the input sequences, we analyze how the order of the input spatial vectors affects the attention performance and whether the LSTM has the capacity to learn the pattern in the channel dimension of the feature maps.

In the main paper, we feed the spatial vectors to the LSTM sequentially in a "**forward**" manner (*i.e.*, from $\hat{x}_1$ to $\hat{x}_{\frac{c}{d}}$). We continue to define other configurations to verify how the sequence order affects the attention performance. The "**reverse**" configuration: feeding the spatial vectors from $\hat{x}_{\frac{c}{d}}$ to $\hat{x}_1$ to the LSTM, whose direction is opposite to that in the "forward" configuration. In the "**random shuffle**" configuration, we first randomly shuffle the order of spatial vectors in $\hat{\boldsymbol{x}}$, and then feed them to the LSTM sequentially. Then we recover the produced $\hat{\boldsymbol{h}}$ and generate the attention maps. This "random shuffle" is operated in each iteration during training. The last configuration, we term "**fixed permutation**". In this configuration, we randomly generate a permutation matrix (*i.e.*, $\boldsymbol{p}_1$) and apply to $\hat{\boldsymbol{x}}$, to produce $\hat{\boldsymbol{x}}^p$ (*i.e.*, $\hat{\boldsymbol{x}}^p = \boldsymbol{p}_1\hat{\boldsymbol{x}}$). Then we feed each row of $\hat{\boldsymbol{x}}^p$ to the LSTM and obtain $\hat{\boldsymbol{h}}^p$ and apply another permutation matrix $\boldsymbol{p}_2$, as $\hat{\boldsymbol{h}} = \boldsymbol{p}_2\hat{\boldsymbol{h}}^p$. Here, $\boldsymbol{p}_2 = \boldsymbol{p}_1^\top$ and $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ are fixed during training. For this configuration, we perform the experiments twice with two different permutation matrices.

We empirically compare the aforementioned four configurations on the iLIDS-VID and the MARS datasets, shown in Table 1. From Table 1, we observe that

[⋆] Work done while at NEC Laboratories America the LSTM does indeed learn useful information along the channel dimension via the recurrent operation (*i.e.*, row (ii), (iv), (v) and (vi)) when the order of the spatial vectors is fixed during training. However, if we randomly shuffle the order of the spatial vectors before feeding to them to the LSTM in each iteration (*i.e.*, row (iii)), the LSTM fails to capture useful information in the channel, and the attention mechanism even degrades below the performance of the baseline network on the MARS dataset (*i.e.*, row (i)).

In this analysis, we can draw the conclusion that the order of the spatial vectors has a minor influence on the attention performance when the order of spatial vectors is fixed. However, it is still difficult to figure out the optimal order of spatial vectors. In all experiments, we empirically use the "forward" configuration in our attention mechanism.

Table 1. Channel recurrent attention module analysis on the iLIDS-VID [1] and the MARS [2] datasets.

		iLIDS-VID		MARS	
	Sequences order	R-1	mAP	R-1	mAP
(i)	No Attention	80.0	87.1	82.3	76.2
(ii)	Forward	87.0	90.6	86.8	81.6
(iii)	Reverse	86.8	90.7	86.3	81.2
(iv)	Random Shuffle	82.7	88.8	79.2	72.4
(v)	Fixed Permutation 1	86.4	89.3	85.8	80.4
(vi)	Fixed Permutation 2	86.7	90.3	86.1	80.9

Intuitively, we further use Bi-LSTM to replace the LSTM in the channel recurrent attention module, to verify whether the sophisticated recurrent network is able to learn more complex information in the channel dimension. Table 2 compares the difference of LSTM and Bi-LSTM in channel recurrent attention module. This study shows that the attention w/ Bi-LSTM cannot brings more performance gain than the that w/ LSTM. However, the Bi-LSTM almost doubles the computation complexities and parameters. Thus we choose regular LSTM in our attention module.

2 Set Aggregation Cell Analysis

In this section, we show the analysis of modeling the video clip as a set and the set aggregation cell acting as a valid set function.

In our channel recurrent attention network, we sample t frames in a video sequence *randomly*, to construct a video clip (*i.e.*, $[T^1, \dots, T^t], T^j \in \mathbb{R}^{C \times H \times W}$) with its person identity as label (*i.e.*, y). In such a video clip, the frames are order-less and the order of frames does not affect the identity prediction by

Table 2. Comparison of LSTM and Bi-LSTM in channel recurrent attention module on the iLIDS-VID [1] and the MARS [2] datasets. FLOPs: the number of FLoating-point OPerations, PNs: Parameter Numbers.

		iLIDS-VID		MARS		Comparison	
	Model	R-1	mAP	R-1	mAP	FLOPs	PNs
(i)	No Attention	80.0	87.1	82.3	76.2	3.8×10^9	25.4×10^6
(ii)	CRA w/LSTM	87.0	90.6	86.8	81.6	0.18×10^9	2.14×10^6
(ii)	CRA w/ Bi-LSTM	87.2	90.2	85.4	81.0	0.32×10^9	4.25×10^6

the network during training. The video frames are fed to the deep network and encoded to a set of frame feature vectors (*i.e.*, $\boldsymbol{F} = [\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t], \boldsymbol{f}^j \in \mathbb{R}^c$), then the frame features are fused to a discriminative clip representation (*i.e.*, $\boldsymbol{g}$) by the aggregation layer (*i.e.*, set aggregation cell).

The set aggregation cell realizes a permutation invariant mapping, $g_\kappa : \mathcal{F} \to \mathcal{G}$ from a set of vector spaces onto a vector space, such that the frame features (*i.e.*, $\boldsymbol{F} = [\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t], \boldsymbol{f}^j \in \mathbb{R}^c$) are fused to a compact clip representation (*e.g.*, $\boldsymbol{g} \in \mathbb{R}^c$). If in this permutation invariant function (*i.e.*, g_κ), the input is a set, then the response of the function is invariant to the ordering of the elements of its input. This property is described as:

Property 1. [3] A function $g_\kappa : \mathcal{F} \to \mathcal{G}$ acting on sets must be **invariant** to the order of objects in the set, *i.e.*, for any permutation $\mathit{\Pi} : g_\kappa\big([\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t]\big) = g_\kappa\big([\boldsymbol{f}^{\mathit{\Pi}(1)}, \ldots, \boldsymbol{f}^{\mathit{\Pi}(t)}]\big)$.

In our supervised video pedestrian retrieval task, it is given t frame samples of $T^1, \ldots, T^t$ as well as the person identity y. Since the frame features are fused using average pooling, shown in Fig. 1, thus it is obvious that the pedestrian identity predictor is permutation invariant to the order of frames in a clip (*i.e.*, $f_\theta([T^1, \ldots, T^t]) = f_\theta([T^{\mathit{\Pi}(1)}, \ldots, T^{\mathit{\Pi}(t)}])$ for any permutation $\mathit{\Pi}$). We continue to study the structure of the set function on *countable sets* and show that our set aggregation cell satisfies the structure of the set function.

Theorem 1. *[3] Assume the elements are countable,* i.e., $|\mathfrak{X}| < \mathfrak{N}_0$*. A function* $g_\kappa : 2^{\mathfrak{X}} \to \mathbb{R}^c$*, operating on a set* $\boldsymbol{F} = [\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t]$ *can be a valid set function,* i.e. *it is permutation invariant to the elements in* $\boldsymbol{F}$*, if and only if it can be decomposed in the form* $\beta\big(\sum_{\boldsymbol{f} \in \boldsymbol{F}} \gamma(\boldsymbol{f})\big)$*, for suitable transformations* β *and* γ*.*

In our deep architecture, we use an aggregation layer (*i.e.*, set aggregation cell) to fuse frame features in a countable set (*i.e.*, $|F| = t$), and this aggregation layer is a permutation invariant function. We use a simple case as an example, shown in Fig. 2(a). In this architecture, the γ function is a mapping: $\mathbb{R}^{c \times t} \to \mathbb{R}^{c \times t}$, formulated as:

$$\boldsymbol{G} = \gamma(\boldsymbol{F}) = \sigma\Big(\varpi\big(\mathrm{Avg}(\boldsymbol{F})\big)\Big) \odot \boldsymbol{F}, \tag{1}$$

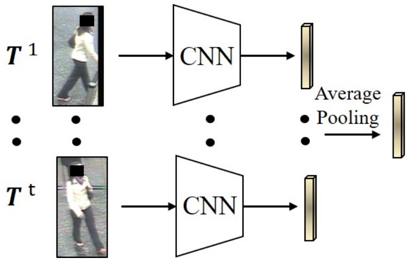
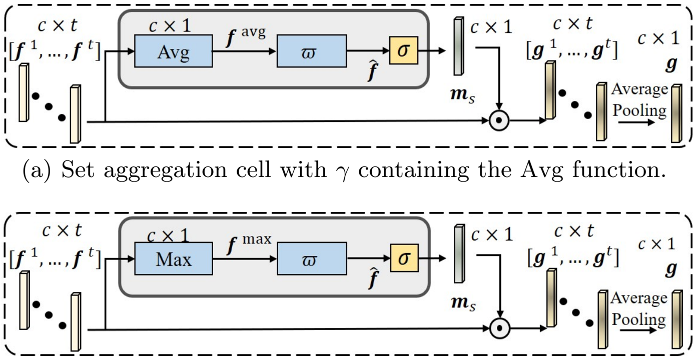


Fig. 1. The pipeline of fusing frame features. The frame features are fused by using average pooling; thus the pedestrian identity predictor is permutation invariant to the order of frames in a clip.

where $\boldsymbol{G} = [\boldsymbol{g}^1, \ldots, \boldsymbol{g}^t]$ and $\boldsymbol{F} = [\boldsymbol{f}^1, \ldots, \boldsymbol{f}^t]$. Thereafter, average pooling operates on the feature set, to realize the summation and β function. Since the γ and β functions are all permutation invariant, the set aggregation cell is a valid set function. Similarly, in Fig. 2(b), the γ function is realized as:

$$\boldsymbol{G} = \gamma(\boldsymbol{F}) = \sigma\Big(\varpi\big(\mathrm{Max}(\boldsymbol{F})\big)\Big) \odot \boldsymbol{F}, \tag{2}$$

which also satisfies the condition of permutation invariance of its input. In the main paper, we evaluated the performance of two vanilla aggregation cells empirically and we observed that the aggregation cell with Avg function is superior to that with the Max function.



(a) Set aggregation cell with γ containing the Avg function.

(b) Set aggregation cell with γ containing the Max function.

Fig. 2. Two set aggregation cells following from $\beta\big(\sum_{\boldsymbol{f} \in \boldsymbol{F}} \gamma(\boldsymbol{f})\big)$.

Since the Avg and the Max operations are permutation invariant, their summation is also permutation invariant; thus we continue to develop our set aggregation in the main paper, shown in Fig. 3. The γ function is formulated as:

$$\boldsymbol{G} = \gamma(\boldsymbol{F}) = \sigma\Big(\varpi\big(\mathrm{Avg}(\boldsymbol{F})\big) \oplus \psi\big(\mathrm{Max}(\boldsymbol{F})\big)\Big) \odot \boldsymbol{F}. \tag{3}$$

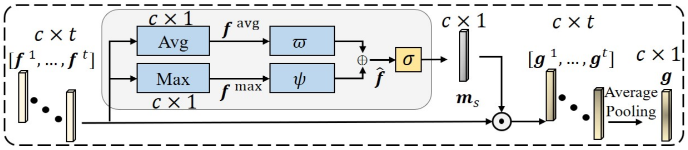


Fig. 3. The architecture of the proposed set aggregation cell in the main paper.

The above analysis shows the necessity to model the frame features in a clip as a set and that the set aggregation cell is a valid set function. In the main paper, we also verify the effectiveness of the set aggregation cell.

3 Loss Function

Triplet Loss. To take into account the between-class variance, we use the triplet loss [4], denoted $\mathcal{L}_{\mathrm{tri}}$, to encode the relative similarity information in a triplet. In a mini-batch, a triplet is formed as $\{\boldsymbol{T}_i, \boldsymbol{T}_i^+, \boldsymbol{T}_i^-\}$, such that the anchor clip $\boldsymbol{T}_i$ and the positive clip $\boldsymbol{T}_i^+$ have the same identity, while the negative clip $\boldsymbol{T}_i^-$ has a different identity. With the clip feature embedding, the triplet loss is formulated as: $\mathcal{L}_{\mathrm{tri}} = \frac{1}{PK}\sum_{i=1}^{PK}\Big[\|\boldsymbol{F}_i - \boldsymbol{F}_i^+\| - \|\boldsymbol{F}_i - \boldsymbol{F}_i^-\| + \xi\Big]_+$, where ξ is a margin and $[\cdot]_+ = \max(\cdot, 0)$. A mini-batch is constructed by randomly sampling P identities and K video clips for each identity. We employ a hard mining strategy [5] for triplet mining.

Cross-entropy Loss. The cross-entropy loss realizes the classification task in training a deep network. It is expressed as: $\mathcal{L}_{\mathrm{sof}} = \frac{1}{PK}\sum_{i=1}^{PK} -\log\big(p(y_i|\boldsymbol{F}_i)\big)$, where p is the predicted probability that $\boldsymbol{F}_i$ belongs to identity y_i. The classification loss encodes the class specific information, which minimizes the within-class variance. The total loss function is formulated as: $\mathcal{L}_{\mathrm{tot}} = \mathcal{L}_{\mathrm{sof}} + \mathcal{L}_{\mathrm{tri}}$.

4 Description of Image Pedestrian Datasets

In the main paper, we evaluate our attention module on image person re-ID tasks on the CUHK01 [6] and the DukeMTMC-reID [7] datasets. The description of the two datasets is as follows:

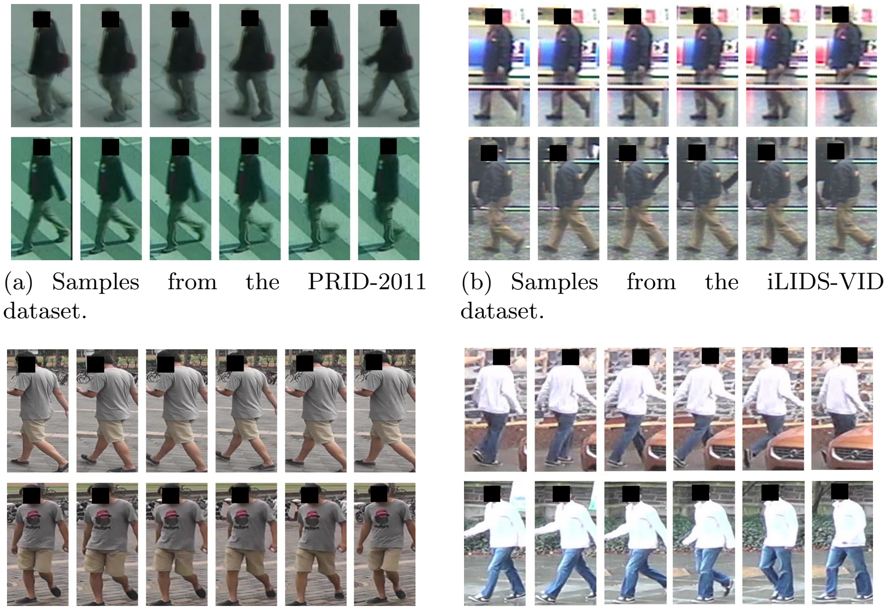
CHUK01 contains $3,884$ images of 971 identities. The person images are collected by two cameras with each person having two images per camera view (*i.e.*, four images per person in total). The person bounding boxes are labelled manually. We adopt the 485/486 training protocol to evaluate our network.

DukeMTMC-reID is the image version of DukeMTMC-VideoReID dataset for the re-ID purpose. It has $1,404$ identities and includes $16,522$ training images of 702 identities, $2,228$ query and $17,661$ gallery images of 702 identities. The pedestrian bounding boxes are labeled manually.

We use a single query (SQ) setting for both datasets when calculating the network prediction accuracy.

5 Pedestrian Samples of Datasets

In the main paper, we have evaluated our attention mechanism across four video person re-ID datasets and two image person re-ID datasets. Here, we show some samples from the aforementioned datasets, in Fig. 4 and Fig. 5. In each pedestrian bounding box, we use a black region to cover the face parts for the sake of privacy.



(a) Samples from the PRID-2011 dataset.

(b) Samples from the iLIDS-VID dataset.

(c) Samples from the MARS dataset.

(d) Samples from the DukeMTMC-VideoReID

Fig. 4. Samples from: (a) PRID-2011 dataset [8], (b) iLIDS-VID dataset [1], (c) MARS dataset [2] and (d) DukeMTMC-VideoReID dataset [9]. In each dataset, we sample two video sequences from one person, and the video sequences are captured by disjoint cameras. For the sake of privacy, we use a black region to cover the face in each frame.

(a) Samples from the CUHK01 dataset.

(b) Samples from the DukeMTMC-reID dataset.

Fig. 5. Samples from: (a) CUHK01 [6], and (b) DukeMTMC-reID [7]. In each dataset, we sample two images from each person, and the video sequences are captured by disjoint cameras. For the sake of privacy, we use a black region to cover the face in each frame.

6 Ablation Study for the Baseline Network

In this section, extensive experiments are performed to choose a proper setting for the baseline network, including the number frames to use from a video clip, dimensionality of the video feature embedding and the training strategies (*e.g.*, pre-training and random erasing [10]). This ablation studies are performed on the iLIDS-VID [1] and the MARS [2] datasets.

Number of Frames in Video Clip. First, we perform experiments with a different number of frames (*i.e.*, t) in a video clip. When $t = 1$, it is reduced to the single image-based model. From Table 3, we observe that $t = 4$ achieves the highest accuracy in both R-1 and mAP values. Thus we use $t = 4$ in our work.

Table 3. Effect of the number of frames in a video clip on the iLIDS-VID [1] and the MARS [2] datasets.

		iLIDS-VID		MARS	
Num of Frames		R-1	mAP	R-1	mAP
(i)	$t = 1$	76.3	84.2	79.2	74.3
(ii)	$t = 2$	79.3	86.1	81.5	75.6
(iii)	$t = 4$	**80.0**	**87.1**	**82.3**	**76.2**
(iv)	$t = 8$	79.6	86.4	82.1	76.0

Dimensionality of Video Feature Embedding. The dimension, *i.e.*, D_v, of the video feature embedding is evaluated and illustrated in Table 4 on both the iLIDS-VID [1] and the MARS [2] datasets. On iLIDS-VID, it is clear that the video feature embedding with $\mathrm{D}_v = 1024$ performs better for both R-1 and mAP accuracy. Therefore, we choose $D_v = 1024$ as the dimension of the feature embedding across all datasets. On the MARS dataset, we observe that R-1 has the peak value when $D_v = 512$, while mAP achieves the peak value when $D_v = 1024$. However, the mAP value in $D_v = 512$ is much lower than that in $D_v = 1024$. Thus we also choose $D_v = 1024$ for MARS.

Table 4. Effect of the dimensionality of video feature embedding on the iLIDS-VID [1] and the MARS [2] datasets.

		iLIDS-VID		MARS	
Dim of Embedding		R-1	mAP	R-1	mAP
(i)	$D_v = 128$	72.0	81.0	82.0	75.1
(ii)	$D_v = 256$	73.3	82.5	82.4	76.3
(iii)	$D_v = 512$	76.6	85.5	**82.6**	75.2
(iv)	$D_v = 1024$	**80.0**	**87.1**	82.3	**76.2**
(v)	$D_v = 2048$	79.6	86.5	82.0	75.6

Training Strategies. We further analyze the effect of different training strategies of the deep network (*e.g.*, random erasing, pre-training model) in Table 5 on both the iLIDS-VID and the MARS datasets. Here, f_θ denotes the backbone network (see Fig. 3 in the main paper). Pre-T and RE denote pre-training on imageNet [11] and random erasing data augmentation, respectively. This table reveals that both training components of pre-training (*i.e.*, Num (ii)) and random erasing (*i.e.*, Num (iii)) improve the R-1 and mAP values, compared to the baseline (*i.e.*, Num (i)). In addition, the network continues to improve its performance when both training strategies are employed, showing that those two training strategies work in a complementary fashion. Thus we choose the network with the pre-trained model and random erasing as our baseline network.

Table 5. Effect of the different training strategies on the iLIDS-VID [1] and the MARS [2] datasets. f_θ, Pre-T and RE denote backbone network, pre-training and random erasing, respectively

		iLIDS-VID		MARS	
Model		R-1	mAP	R-1	mAP
(i)	f_θ	60.8	67.6	76.4	71.8
(ii)	f_θ + Pre-T	70.8	81.6	81.1	75.4
(iii)	f_θ + RE	65.3	74.6	78.8	74.5
(iv)	f_θ + Pre-T + RE	**80.0**	**87.1**	**82.3**	**76.2**

7 Ablation Study for Set Aggregation Cell

In this section, we perform experiments for parameter selection in the set aggregation cell.

Effectiveness of Dimension Reduction in the Self-gating Layers. The dimension of the hidden layer in the self-gating layers (*i.e.*, ϖ and ψ) of the set aggregation block is studied. The dimension of the hidden layer is reduced by a factor of r (*i.e.*, $\mathrm{D}_{\mathrm{hid}} = 2048/r$). Table 6 reveals that setting $r = 16$ achieves good performance in both datasets; thus we use this value for the set aggregation cell.

Table 6. Effect of dimension reduction in the self-gating layers on the iLIDS-VID [1] and the MARS [2] datasets.

		iLIDS-VID		MARS	
Reduction Ratio		R-1	mAP	R-1	mAP
(i)	Only CRA	87.0	90.6	86.8	81.6
(ii)	$r = 2$	88.2	91.2	87.2	82.1
(iii)	$r = 4$	88.4	91.6	87.6	82.4
(iv)	$r = 8$	88.5	91.7	**87.9**	82.8
(v)	$r = 16$	**88.7**	**91.9**	**87.9**	**83.0**
(vi)	$r = 32$	87.9	90.9	87.6	82.2

8 Visualization

Fig. 6 shows additional visualizations of feature maps for qualitative study. In the visualizations, we can clearly observe that our attention module has the capacity to focus more on the foreground areas and ignore some background areas, which boosts the baseline network to achieve the state-of-the-art performance on the video pedestrian task.

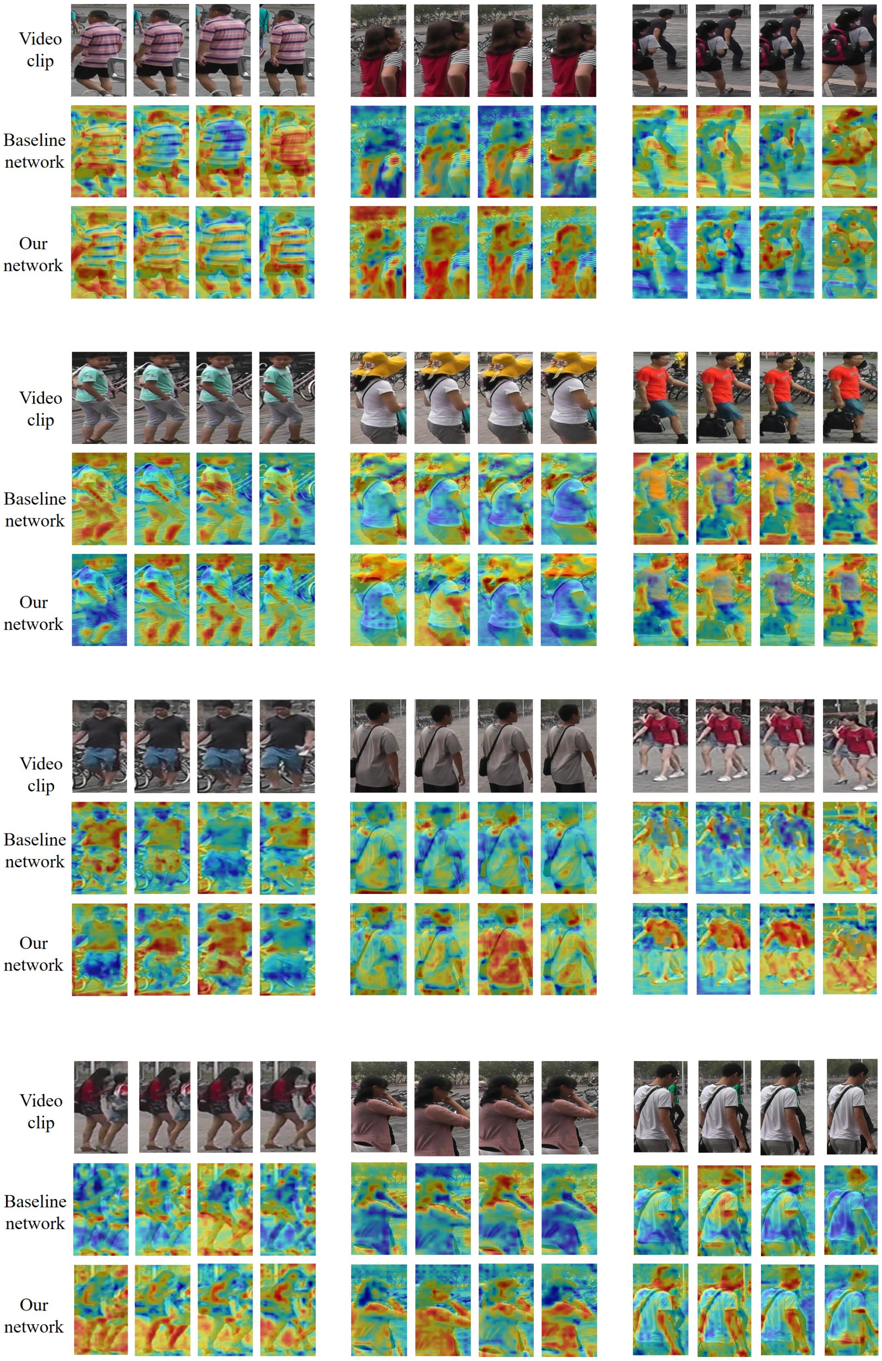


Fig. 6. Visualization of feature maps. We sample video clips from different pedestrians and visualize the feature maps.

References

1. Wang, T., Gong, S., Zhu, X., Wang, S.: Person re-identification by discriminative selection in video ranking. TPAMI (2016)
2. Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., Tian, Q.: Mars: A video benchmark for large-scale person re-identification. In: ECCV. (2016)
3. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: NeurIPS. (2017)
4. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR. (2015)
5. Hermans, A., Beyer, B., Leibe, B.: In Defense of the Triplet Loss for Person Re-Identification (2017) arXiv:1703.07737 [cs.CV], [cs.NE].
6. Li, W., Zhao, R., Wang, X.: Human reidentification with transferred metric learning. In: ACCV. (2012)
7. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCVworkshop on Benchmarking Multi-Target Tracking. (2016)
8. Hirzer, M., Beleznai, C., Roth, P.M., Bischof, H.: Person re-identification by descriptive and discriminative classification. In: Image Analysis. (2011)
9. Wu, Y., Lin, Y., Dong, X., Yan, Y., Ouyang, W., Yang, Y.: Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In: CVPR. (2018)
10. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. arXiv:1708.04896 (2017)
11. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)