Combine Coarse and Fine Cues: Multi-grained Fusion Network for Video-Based Person Re-identification

Chao Li[1,2], Lei Liu[1], Kai Lv[1], Hao Sheng[1,2(✉)], and Wei Ke[3]

[1] State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing, China
{licc,leiliu,lvkai,shenghao}@buaa.edu.cn
[2] Shenzhen Key Laboratory of Data Vitalization, Research Institute in Shenzhen, Beihang University, Shenzhen, People's Republic of China
[3] Macao Polytechnic Institute, Macao, People's Republic of China
wke@ipm.edu.mo

Abstract. Video-based person re-identification aims to precisely match video sequences of pedestrian across non-overlapped cameras. Existing methods deal with this task by encoding each frame and aggregating them along time. In order to increase the discriminative ability of video features, we propose an end-to-end framework called Multi-grained Fusion Network (MGFN) which aims to keep both global and local information by combining frame-level representations with different granularities. The final video features are generated by aggregating multi-grained representations on both spatial and temporal. Experiments indicate our method achieves excellent performance on three widely used datasets named PRID-2011, iLIDS-VID, and MARS. Especially on MARS, MGFN surpass state-of-the-art result by 11.5%.

Keywords: Video-based person re-identification
Multi-grained fusion network · Part-based model
Multi-grained feature

1 Introduction

Person re-identification is a significant task for social security and video surveillance. It aims to retrieve all pedestrians in probe set from a large gallery set in different camera viewpoints. At present, person re-identification is mainly divided into two parts, image-based and video-based person re-identification, and the latter one is closer to realistic scenarios. Both of them should face challenging problems, which include pose variations, complex illumination, multi-camera viewpoints, background clutter and occlusion.

In this paper, we focus on video-based person re-identification, and aim to generate discriminative features from video data. Generating frame-level representations and aggregating them is an intuitive way to encode video date into

W. Liu et al. (Eds.): KSEM 2018, LNAI 11061, pp. 177–184, 2018.
https://doi.org/10.1007/978-3-319-99365-2_16

discriminative features. There are many models show their effectiveness for generating robust frame-level features [8,10]. Wei et al. [10] based on pose estimator to extract keypoints, and separates person into three parts. Different from coarsely divide into three parts, they get more precise partitions. Sun et al. [8] proposed a powerful baseline network. They do not directly separate part on original image, instead, they make partition on activation maps. They think the integrating part information can increase the discriminative ability of feature.

After generating features of each frame, a superb aggregation method can promote the performance of original model. Recently, aggregation method for video-based person re-identification is separated into two groups. The first group use temporal pooling to aggregate frame-level features along time. Mclaughlin et al. [6] use Recurrent Neural Network to learn temporal information, and then adopt average pooling to aggregate frame-level features. Another group use weighted average method. There are two ways to generate weights for each frame, attention mechanism and learning by network itself. Zhou et al. [12] adopt RNN hidden state to generate attention map on each frame-level feature. Liu et al. [5] use an independent branch to learn weight of each frame by the network itself. In order to get final video representation, they use temporal average pooling to aggregate weighted frame-level features.

In this paper, simple horizontal partition is adopted to generate fine representation of each person. During this process, fine-grained representations is generated, however, global information is lost. In order to preserve both global and local information, we construct an end-to-end model which combines multi-grained features. In order to preserve both global and local information, we construct an end-to-end model which combines multi-grained features. Totally, our contributions are summarized as two parts:

First, we propose a simple network named Regular Partition Network (RPN). RPN generates representations of each frame which is divided into numbers of partitions firstly, then aggregates them by using temporal pooling along time dimension. Second, we combine multi-grained features of each person, and construct a framework called Multi-grained Fusion Network (MGFN). MGFN combines differently grained features which are generated by three independent branches on spatial, and uses temporal pooling to generate final video features.

2 Proposed Method

Given an input video V contains N frames, $V = \{I_1, I_2, \cdots, I_N\}$, and I_n represents the *n-th* frame in this video sequence. Because ResNet-50 [2] has relatively concise structure and good performance, we use it as a baseline model in this paper. When a video data pass through ResNet-50 before its fully connection layer, the feature of *n-th* frame is generated as f_n, where $f_n \in \mathbb{R}^D$. After that, we use temporal pooling function TP to aggregate representation of each frame.

$$feat_{baseline} = TP(f_1, f_2, \cdots, f_N) \tag{1}$$

$feat_{baseline}$ is a feature of video, $feat_{baseline} \in \mathbb{R}^D$. The subscript of $feat_{baseline}$ shows this feature is extracted by baseline model.

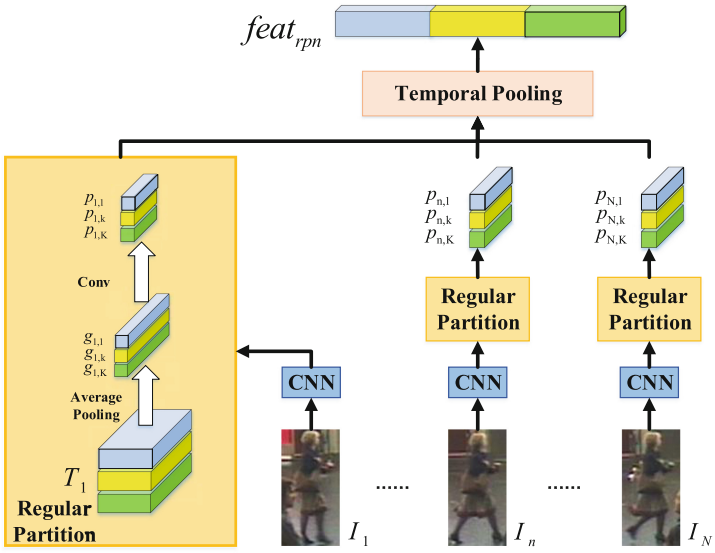

Fig. 1. Regular partition network structure.

2.1 Regular Partition Network

Regular Partition Network. Our regular partition method (Fig. 1) directly makes partitions on activation maps which is generated by ResNet-50. For a better illustration, we compact activation maps extractor and use CNN-block to represent it in Fig. 1. When a video data V pass through RPN, CNN-block is adopted to transform each frame into 3-D tensor and get $T_n \in \mathbb{R}^{H_T \times W_T \times C_T}$ for I_n. Then tensor T_n is separated into K non-overlapped partitions, the size of each partition is $\lfloor H_T/K \rfloor \times W_T$. In next step, T_n is transformed into $g_n \in \mathbb{R}^{K \times C_T}$, where $g_{n,k} \in \mathbb{R}^{1 \times C_T}$ represents the transformed result of $T_{n,k}$ by using average pooling. Especially, the kernel size of average pooling is as same as the size of $T_{n,k}$, where $T_{n,k}$ is the *k-th* part of T_n. After that, 1×1 2D-convolution is used to reduce the dimension of $g_{n,k}$. Then we get K low dimension vector $p_{n,k} \in \mathbb{R}^{1 \times d}$. Finally, $feat_{rpn} = \theta(P_1, P_2, \cdots, P_K)$, where $P_k = TP(p_{1,k}, p_{2,k}, \cdots, p_{N,k})$, and θ is concatenation operation.

Training and Testing. During training procedure, we transfer identification task into classification problem. To our empirical practice, the feature of each part should be separated to do classification. The classification loss of P_k is formulated as:

$$loss_k = -\sum_{m=1}^{M} log \frac{e^{(W_{k,y_m})^T P_k + b_{k,y_m}}}{\sum_{j=1}^{C} e^{(W_{k,j})^T P_k + b_{k,j}}} \tag{2}$$

where M is size of a mini-batch in training and C is the class number of classification task. In Eq. 2, $W_k \in \mathbb{R}^{d \times C}$ and $b_k \in \mathbb{R}^C$ is the weights and bias of

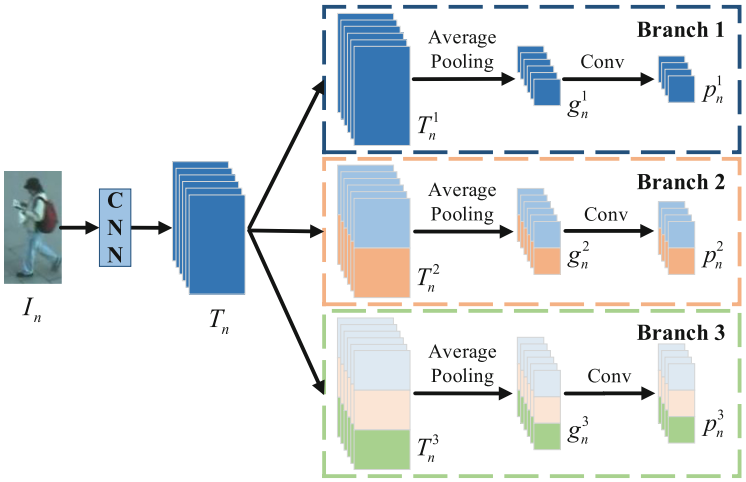

Fig. 2. Multi-grained fusion network structure for single frame.

classifier, and subscript y_m means the ground truth label of *i-th* sample in this mini-batch. As for the whole RPN model, loss function is defined as:

$$loss_{rpn} = \frac{1}{M \times K} \sum_{k=1}^{K} loss_k \tag{3}$$

where K is the number of partition. During testing period, $feat_{rpn}$ is used as whole video representation. Using Euclidean distance as evaluation method, the distance between two person is closer, the probability of being same is higher.

2.2 Multi-grained Fusion Network

Information Complementary. In our experiments on Regular Partition Network, we divide middle representation of each frame into horizontal stripes. To our empirical practice the more stripes are divided, the finer features are extracted. Because of the increasing computation and decreasing relevance of data, the stripe should not be too thin. We consider that different partition numbers mean differently grained representation, and combining diversely grained features can keep more information. For achieving our motivation, we construct a framework which fuses global and local cues, and we call it Multi-Grained Fusion Network (MGFN). MGFN combines features with different granularities, and it makes a complementary between local and global information.

Structure of Multi-grained Fusion Network. Multi-grained Fusion Network has multiple branches, and different branch generate a feature with different granularity. In our model, we set branch number to 3 as Fig. 2 shows, and for each independent branch we set the number of partition to $1, 3, 6$

(The number of partitions $1, 2, 3$ in Fig. 2 is just for better illustration) separately. As Fig. 2 shows, the number of parts K_1 in the top branch is set to 1, and aims to keep global information. $K_2 = 3$ in the middle branch proposes to keep finer-grained information, and $K_3 = 6$ in the bottom branch is expected to keep the finest-grained cues. For easy explanation, we describe the process of MGFN for each frame. Before partition, an input frame I_n is transfered into T_n by using shared CNN-block. Then in *i-th* branch, T_n is divided into K_i parts, and the partition rules are as same as RPN. We donate *k-th* part of T_n^i as $T_{n,k}^i$, where $k \in [1, K_i]$. After going through average pooling and 1×1 2D-convolution, p_n^i is generated for *i-th* branch, where $p_n^i \in \mathbb{R}^{K_i \times C_T}$. Finally, we concatenate p_n^i of each independent branch, and obtain final feature f_n for I_n, where $f_n \in \mathbb{R}^{TK \times d}$, $TK = \sum_{i=1}^{3} K_i$ and d is the value of reduced dimension. As for video feature, we also use temporal pool function TP to generate $feat_{mgfn} = TP(f_1, f_2, \cdots, f_N)$.

Training and Testing. During training step, we do not combine part feature together as same as the training process of RPN. Firstly, we use temporal pool function TP to aggregate part features in the same location along time. $P_{k_i}^i = TP(P_{1,k_i}^i, P_{2,k_i}^i, \cdots, P_{N,k_i}^i)$, where i represent the branch ID, $i \in \{1, 2, 3\}$, k_i is part location in *i-th* branch, $k_i \in [1, K_i]$. We also regard identification task as classification problem, so fully connection layer is used to change the dimension of $P_{k_i}^i$ to satisfy classification jobs. The $loss_{k_i}^i$ is defined as

$$loss_{k_i}^i = -\frac{1}{M} \sum_{m=1}^{M} log \frac{e^{(W_{k_i,y_m}^i)^T P_{k_i}^i + b_{k_i,y_m}^i}}{\sum_{j=1}^{C} e^{(W_{k_i,j}^i)^T P_{k_i}^i + b_{k_i,j}^i}} \tag{4}$$

where $loss_{k_i}^i$ represent k_i partition loss value of *i-th* branch, $W_{k_i}^i$ and $b_{k_i}^i$ is weights and bias of classifier for k_i parts in *i-th* branch. And M is size of mini-batch, C is the class number, y_m is ground truth label of the *m-th* sample.

$$loss_{mgfn} = \frac{1}{TK} \sum_{i=1}^{3} \sum_{k=1}^{K_i} loss_k^i \tag{5}$$

TK is the total number of partition, $TK = \sum_{i=1}^{3} K_i$. In testing period, we extract $feat_{mgfn}$ for each person firstly. Then using the same evaluation method and protocol to compute the similarity between identities as RPN.

3 Experiments

3.1 Implementation Details

We evaluate our proposed methods on three widely used video-based person re-identification dataset: PRID-2011 [3], iLIDS-VID [9] and MARS [11]. For PRID-2011 and iLIDS-VID, we use the same evaluation protocol as Wang et al. [6]. As for MARS, we follow the evaluation protocol from Zheng et al. [11]. CMC rank-1 is computed on all the three datasets, mean average precision (mAP) is adopted on MARS at the same time. We sample $N = 16$ consecutive frames as

input from each image sequence, and each adjacent input has 50% overlapped frames for generating more data on training process. Image preprocessing and augmentation are also used to enlarge training set. We first pretrain the baseline model and RPN on DukeMTMC-reID [7] without temporal pooling, then fine-tune them on PRID-2011, iLIDS-VID with temporal pooling. During training process for MGFN, we use pretrained weights on RPN to initialize each branch. For RPN, we set $K = 6$ as Sun et al. [8] has proved in their work. Different from Sun et al. [8], we use 1×1 convolution to reduce the dimension of g_n from 2048 to 256. Except baseline model image size is 256×128, we resize the frame to 384×128. Our proposed model uses batched stochastic gradient descent. And we set learning rate to 0.1 at beginning, then drop it to 10% after each 20 epochs.

3.2 Experiments Analyses

Max or Average pooling. In our methods, we utilize temporal pool function to aggregate features of each frame for generating final video representation. Widely used temporal pooling methods include temporal max pooling and temporal average pooling. Temporal max pooling aims to find out the salient value in feature vectors, while temporal average pooling attempts to dilute each value. We compare the performance of these two temporal pooling methods, and the results are summarized in the top group of Table 1. We find that temporal average pooling performs better on both PRID-2011 and iLIDS-VID. One possible explanation is that temporal average pooling method take all information into consideration, so it has a superb ability of anti-interference. So our following experiments use temporal average pooling as default, unless special notification.

Table 1. Comparison of temporal pooling method and classifier parameters shared or not. ResNet: baseline model ResNet-50. MAX: temporal max pooling. AVG: temporal average pooling. Share: classifier parameters are shared. NotShare: classifier parameters are not shared. CMC Rank-1, Rank-5, Rank-10 accuracy (%) are shown.

Models	PRID-2011			iLIDS-VID		
	Rank-1	Rank-5	Rank-10	Rank-1	Rank-5	Rank-10
ResNet+MAX	61.8	81.3	86.7	38.3	63.4	73.7
ResNet+AVG	72.1	94.0	97.8	47.8	75.1	85.5
RPN+Share	86.5	96.7	98.1	66.5	87.5	93.9
RPN+NotShare	88.5	98.1	99.4	69.2	90.1	95.0

Share Parameters or Not Share. In Regular Partition Network, for each partition feature we utilize a classifier to determine this part belongs to which identities. There are two parameters in the classifier, W and b. The intuitive consideration is whether use shared parameters for all classifier. For clearly explanation, we donate the structure of not share parameters as *NSP* and share parameters as *SP*. We compare these two structures, and record the performance

in the bottom group of Table 1. The results of experiments show *SP* is more inferior than *NSP*. For different parts, *NSP* uses specialized classifier which more pertinent with each part, however, *SP* only uses a general classifier. In intuition, specialized one must be more superior than general classifier. Even though, *NSP* introduces more parameters, and the increasing computation can be bared.

Table 2. Comparison of our method with state-of-the-art methods. CMC Rank-1 accuracy (%) and mAP (%) are shown. R-1: CMC Rank-1 accuracy (%)

Methods	PRID-2011	iLIDS-VID	MARS	
			R-1	mAP
mvRMLLC+Alignment [1]	66.8	69.1	-	-
CNN+RNN [6]	70.0	58.0	-	-
QAN [5]	90.3	60.8	-	-
Mars [11]	77.3	53.0	68.3	49.3
SeeForest [12]	79.4	55.2	70.6	50.7
end-to-end AMOC+EpicFlow [4]	83.7	68.7	68.3	52.9
ResNet-50	72.1	47.8	-	-
RPN (ours)	88.5	69.2	-	-
MGFN (ours)	**90.9**	**72.8**	**82.1**	**56.8**

Comparison with State-of-the-Art. Table 2 shows the performance of our methods and other state-of-the-art methods. In Table 2, using ResNet-50 as a feature extractor for each frame is able to get a competitive result. Based on this powerful feature extractor, RPN makes improvements 16.4% and 21.4% on PRID-2011 and iLIDS-VID separately. Future more, MGFN improves the results through combining diversely grained features generated by RPN with the different number of partitions. Especially on MARS, MGFN surpasses the method of Zhou et al. [12] by 11.5%. Zhou et al. proposed a complex model based on six spatial RNNs and temporal attention. In contrast, MGFN is conciser on structure and easier for training. MARS is the most challenging dataset, because of distractor tracklets. Finer performance suggests that our Multi-grained Fusion Network is effective for video-based person re-identification in complex scenarios.

4 Conclusion

In this paper, we propose two methods for video-based person re-identification. One is Regular Partition Network (RPN), the other is Multi-grained Fusion Network (MGFN). RPN adopts partition cues to keep local information. Our experiments indicate RPN shows competitive performance on each video-based dataset. Based on RPN, we construct MGFN to combine differently grained information together, and aim to keep both global and local cues. According to our experiments, MGFN makes remarkable performance although in challenging scenarios.

Acknowledgement. This study is partially supported by the National Key R&D Program of China (No. 2017YFB1002000), the National Natural Science Foundation of China (No. 61472019), the Macao Science and Technology Development Fund (No. 138/2016/A3), the Program of Introducing Talents of Discipline to Universities and the Open Fund of the State Key Laboratory of Software Development Environment under grant SKLSDE-2017ZX-09, the Project of Experimental Verification of the Basic Commonness and Key Technical Standards of the Industrial Internet network architecture. Thank you for the support from HAWKEYE Group.

References

1. Chen, J., Wang, Y., Tang, Y.Y.: Person re-identification by exploiting spatio-temporal cues and multi-view metric learning. IEEE Sig. Process. Lett. **23**(7), 998–1002 (2016)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition, pp. 770–778 (2016)
3. Hirzer, M., Beleznai, C., Roth, P.M., Bischof, H.: Person re-identification by descriptive and discriminative classification. In: Scandinavian Conference on Image Analysis, pp. 91–102 (2011)
4. Liu, H., Jie, Z., Jayashree, K., Qi, M., Jiang, J., Yan, S., Feng, J.: Video-based person re-identification with accumulative motion context. IEEE Trans. Circ. Syst. Video Technol. **PP**(99), 1–1 (2017)
5. Liu, Y., Yan, J., Ouyang, W.: Quality aware network for set to set recognition. In: Computer Vision and Pattern Recognition, pp. 5790–5799 (2017)
6. Mclaughlin, N., Rincon, J.M.D., Miller, P.: Recurrent convolutional network for video-based person re-identification. In: Computer Vision and Pattern Recognition, pp. 1325–1334. IEEE (2016)
7. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision Workshop on Benchmarking Multi-target Tracking, pp. 17–35 (2016)
8. Sun, Y., Zheng, L., Yang, Y., Tian, Q., Wang, S.: Beyond part models: person retrieval with refined part pooling. arXiv preprint arXiv:1711.09349 (2017)
9. Wang, T., Gong, S., Zhu, X., Wang, S.: Person re-identification by video ranking. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 688–703. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_45
10. Wei, L., Zhang, S., Yao, H., Gao, W., Tian, Q.: Glad: Global-local-alignment descriptor for pedestrian retrieval. In: Proceedings of the 2017 ACM on Multimedia Conference, pp. 420–428. ACM (2017)
11. 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: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 868–884. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_52
12. Zhou, Z., Huang, Y., Wang, W., Wang, L., Tan, T.: See the forest for the trees: joint spatial and temporal recurrent neural networks for video-based person re-identification. In: Computer Vision and Pattern Recognition, pp. 6776–6785 (2017)