Revisiting Temporal Modeling for Video-based Person ReID

Jiyang Gao
jiyangga@usc.edu
University of Southern California
Los Angeles, CA, USA

Ram Nevatia
nevatia@usc.edu
University of Southern California
Los Angeles, CA, USA

Abstract

Video-based person reID is an important task, which has received much attention in recent years due to the increasing demand in surveillance and camera networks. A typical video-based person reID system consists of three parts: an image-level feature extractor (*e.g.* CNN), a temporal modeling method to aggregate temporal features and a loss function. Although many methods on temporal modeling have been proposed, it is hard to directly compare these methods, because the choice of feature extractor and loss function also have a large impact on the final performance. We comprehensively study and compare four different temporal modeling methods (temporal pooling, temporal attention, RNN and 3D convnets) for video-based person reID. We also propose a new attention generation network which adopts temporal convolution to extract temporal information among frames. The evaluation is done on the MARS dataset, and our methods outperform state-of-the-art methods by a large margin. Our source codes are released at https://github.com/jiyanggao/Video-Person-ReID.

1 Introduction

Person re-Identification (re-ID) tackles the problem of retrieving a specific person (*i.e.* query) in different images or videos, possibly taken from different cameras in different environments. It has received increased attentions in recent years due to the increasing demand of public safety and rapidly growing surveillance camera networks. Specifically, we focus on video-based person re-ID, that is, given a query video of one person, the system tries to identify this person in a set of gallery videos.

Most of the recent existing video-based person reID methods are based on deep neural networks [12, 13, 24]. Typically, three important parts have large impacts on a video-based person reID system: an image-level feature extractor (typically a Convolutional Neural Network, CNN), a temporal modeling module to aggregate image-level features and a loss function to train the network. During test, the query video and gallery videos are encoded to feature vectors using the aforementioned system, and then differences between them (usually L2 distances) are calculated to retrieve the top-N results. Recent work [12, 13, 24] on video-based person reID mostly focuses on the temporal modeling part, *i.e.*, how to aggregate a sequence of image-level features into a clip-level feature.

Previous work on temporal modeling methods on video-based person reID falls into two categories: Recurrent Neural Network (RNN) based and temporal attention based. In RNN-based methods, McLanghlin *et al.* [13] proposed to use an RNN to model the temporal information between frames; Yan *et al.* [20] also used an RNN to encode sequence features, where the final hidden state is used as video representation. In temporal attention based methods, Liu *et al.* [12] designed a Quality Aware Network (QAN), which is actually an attention weighted average, to aggregate temporal features; Zhou *et al.* [24] proposed to encode the video with temporal RNN and attention. Besides, Hermans *et al.* [7] adopted a triplet loss function and a simple temporal pooling method, and achieved state-of-the-art performance on the MARS [17] dataset.

Although extensive experiments have been reported on the aforementioned methods, it's hard to directly compare the influence of temporal modeling methods, as they used different image level feature extractors and different loss functions, these variations can affect the performance significantly. For example, [13] adopted a 3-layer CNN to encode the images; [20] used hand-crafted features; QAN [12] extracted VGG [16] features as image representations.

In this paper, we explore the effectiveness of different temporal modeling methods on video-based person re-ID by fixing the image-level feature extractor (ResNet-50 [6]) and the loss function (triplet loss and softmax cross-entropy loss) to be the same. Specifically, we test four commonly used temporal modeling architectures: temporal pooling, temporal attention [12, 24], Recurrent Neural Network (RNN) [13, 20] and 3D convolution neural networks [5]. A 3D convolution neural network [5] directly encodes an image sequence as a feature vector; we keep the network depth the same as for the 2D CNN for fair comparison. We also propose a new attention generation network which adopts temporal convolution to extract temporal information. We perform experiments on the MARS [17] dataset, which is the largest video-based person reID dataset available to date. The experimental results show that our method outperforms state-of-the-art models by a large margin.

In summary, our contributions are two-fold: First, we comprehensively study four commonly used temporal modeling (temporal pooling, temporal attention, RNN and 3D conv) methods for video-based person reID on MARS. We will release the source codes. Second, we propose a novel temporal-conv based attention generation network, which achieves the best performance among all temporal modeling methods; with the help of strong feature extractor and effective loss functions, our system outperforms state-of-the-art methods by a large margin.

In the following, we first discuss related work in Sec 2, then present the overall person reID system architecture in Sec 3, and describe the temporal modeling methods in detail. In Sec 4, we show the experiments and discuss the results.

2 Related Work

In this section, we discuss related work, including video-based and image-based person reid and video temporal analysis.

Video-based person reID. Previous work on temporal modeling methods on video-based person reID fall into two categories: Recurrent Neural Network (RNN) based and temporal attention based. McLanghlin *et al.* [13] first proposed to model the temporal information between frames by RNN, the average of RNN cell outputs is used as the clip level representation. Similar to [13], Yan *et al.* [20]also used RNN to encode sequence features, the final hidden state is used as video representation. Liu *et al.* [12] designed a Quality Aware

Network (QAN), which is essentially an attention weighted average, to aggregate temporal features; the attention scores are generated from frame-level feature maps. Zhou *et al.* [24] and Xu *et al.* [15] proposed to encode encode the video with temporal RNN and attention. Chung *et al.* [1] presented a two-stream network which models both RGB images and optical flows, simple temporal pooling is used to aggregate the features. Recently, Zheng *et al.* [17] built a new dataset MARS for video-based person reID, which becomes the standard benchmark on this task.

Image-based person reID. Recent work on image-based person reID improves the performance by mainly two directions: image spatial modeling and loss function for metric learning. In the direction of spatial feature modeling, Su *et al.* [18] and Zhao *et al.* [21] used human joints to parse the image and fuse the spatial features. Zhao *et al.* [22] proposed a part-aligned representation for handling the body part misalignment problem. As for loss functions, typically, hinge loss in a Siamese network and identity softmax cross-entropy loss function are used. To learn an effective metric embedding, Hermans *et al.* [7] proposed a modified triplet loss function, which selects the hardest positive and negative for each anchor sample, and they achieved state-of-the-art performance.

Video temporal analysis. Besides person reID work, temporal modeling methods in other fields, such as video classification [8], temporal action detection [3, 14], are also related. Karpathy *et al.* [8] designed a CNN network to extract frame-level features and used temporal pooling method to aggregate features. Tran *et al.* [19] proposed a 3D CNN network to extract spatio-temporal features from video clips. Hara *et al.* [5] explored the ResNet [6] architecture with 3D convolution. Gao *et al.* [2, 4] proposed a temporal boundary regression network to localize actions in long videos.

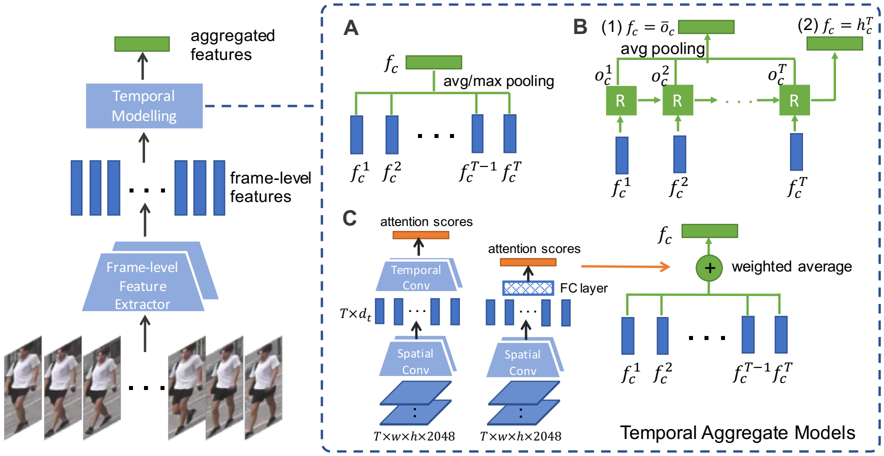

Figure 1: Three temporal modeling architectures (A: temporal pooling, B: RNN and C: temporal attention) based on an image-level feature extractor (typically a 2D CNN). For RNN, final hidden state or average of cell outputs is used as the clip-level representation; For temporal attention, two types of attention generation network are shown: “spatial conv + FC [12]" and “spatial conv + temporal conv".

3 Methods

In this section, we introduce the overall system pipeline and detailed configurations of the temporal modeling methods. The whole system could be divided into two parts: a video encoder which extract visual representations from video clips, and a loss function to optimize the video encoder and a method to match the query video with the gallery videos. A video is first cut into consecutive non-overlap clips $\{c_k\}$, each clip contains T frames. The clip encoder takes the clips as inputs and outputs a D-dimensional feature vector f_c for each clip. The video level feature is the average of all clip level features.

3.1 Video Clip Encoder

To build a video clip encoder, we consider two types of convolutional neural network (CNN): (1) 3D CNN and (2) 2D CNN with temporal aggregation method. 3D CNN directly takes a video clip c which contains n frames as input and output a feature vector f_c, while 2D CNN first extracts a sequence of image-level features $\{f_c^t\}, t \in [1,n]$, and then $\{f_c^t\}$ are aggregated into a single vector f_c by a temporal modeling method.

3D CNN. For 3D CNN, we adopt 3D ResNet [5] model, which adopts 3D convolution kernels with ResNet architecture [6] and is designed for action classification. We replace the final classification layer with person identity outputs and use the pre-trained parameters (on Kinetics [9]). The model takes T consecutive frames (*i.e.* a video clip) as input, and the layer before final classification layer is used as the representation of the person.

For 2D CNN, we adopt a standard ResNet-50 [6] model as an image level feature extractor. Given an image sequence (*i.e.* a video clip), we input each image into the feature extractor and output a sequence of image level features $\{f_c^t\}, t \in [1,T]$, which is a $T \times D$ matrix, n is clip sequence length, D is image level feature dimension. Then we apply a temporal aggregation method to aggregate the features into a single clip level feature f_c, which is a D-dimensional vector. Specifically, we test three different temporal modeling methods: (1) temporal pooling, (2) temporal attention, (3) RNN; the architectures of these methods are shown in Figure 1.

Temporal pooling. In temporal pooling model, we consider max pooling and average pooling. For max pooling, $f_c = max_t(f_c^t)$; for average pooling, $f_c = \frac{1}{T}\sum_{t=1}^{n} f_c^t$.

Temporal attention. In temporal attention model, we apply an attention weighted average on the sequence of image features. Given that the attention for clip c is $a_c^t, t \in [1,T]$, then

$$f_c = \frac{1}{T}\sum_{t=1}^{n} a_c^t f_c^t \tag{1}$$

The tensor size of last convolutional layer in Resnet-50 is $[w,h,2048]$, w and h depend on input image size. The attention generation network takes a sequence of image-level features $[T,w,h,2048]$ as inputs, and outputs T attention scores. We design two types of attention networks. (1) “spatial conv + FC [12]": we apply a spatial conv layer (kernel width = w, kernel height = h, input channle number =2048, output channel number = d_t, short for $\{w,h,2048,d_t\}$) and a Fully-Connected (FC) (input channel = d_t, output channel = 1) layer on the aforementioned output tensor; the output of the conv layer is a scalar vector $s_c^t, t \in [1,T]$, which is used as the score for the frame t of clip c. (2) “spatial + temporal conv": first a conv layer with shape $\{w,h,2048,d_t\}$ is applied, then we get a d_t-dimensional feature for each frame of the clip, we apply a temporal conv layer $\{3,d_t,1\}$ on these frame-level features to generate temporal attentions s_c^t. The two networks are shown in Figure 1 (C).

Once we have s_c^t, there are two ways of calculating the final attention scores a_c^t: (1) Softmax function [24],

$$a_c^t = \frac{e^{s_c^t}}{\sum_{i=1}^{T} e^{s_c^i}} \tag{2}$$

and (2) a Sigmoid function + L1 normalization [12],

$$a_c^t = \frac{\sigma(s_c^t)}{\sum_{i=1}^{T} \sigma(s_c^i)} \tag{3}$$

where σ means the Sigmoid function.

RNN. An RNN cell encodes one image feature in a sequence at one time step t and passes the hidden state h_t into the next time step. We consider two methods of aggregating a sequence of image features into a single clip feature f_c. The first method directly takes the hidden state h^T at the last time step, $f_c = h_c^T$, as shown in Figure 1 (B). The second method calculates the average of the RNN outputs $\{o^t\}, t \in [1,n]$, that is $f_c = \frac{1}{T}\sum_{t=1}^{T} o_c^t$. We test two types of RNN cell: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Other settings can be found in Section 4.1.

3.2 Loss Functions

We use a triplet loss function and a Softmax cross-entropy loss function to train the networks. The triplet loss function we use was originally proposed in [7], and named as Batch Hard triplet loss function. To form a batch, we randomly sample P identities and randomly sample K clips for each identity (each clip contains T frames); totally there are PK clips in a batch. For each sample a in the batch, the hardest positive and the hardest negative samples within the batch are selected when forming the triplets for computing the loss $L_{triplet}$.

$$L_{triplet} = \overbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}^{\text{all anchors}} \Big[m + \overbrace{\max_{p=1\ldots K} D\left(f_{i,a}), f_p^i\right))}^{\text{hardest positive}} \\ \underbrace{- \min_{\substack{j=1\ldots P \\ n=1\ldots K \\ j\neq i}} D\left(f_{i,a}), f_{j,n}\right))}_{\text{hardest negative}} \Big]_+ , \tag{4}$$

The Softmax cross-entropy loss function $L_{softmax}$ encourages the network to classify the PK clips to the correct identities.

$$L_{softmax} = -\frac{1}{PK}\sum_{i=1}^{P}\sum_{a=1}^{K} p_{i,a} \log q_{i,a} \tag{5}$$

where $p_{i,a}$ and $q_{i,a}$ are the groundtruth identity and prediction of sample $\{i,a\}$. The total loss L is the combination of these two losses.

$$L = L_{softmax} + L_{triplet} \tag{6}$$

3.3 Similarity Calculation for Testing

As mentioned before, a video is cut into consecutive non-overlapping clips $\{c_k\}$, each clip containing T frames. During testing, we extract clip level representation for each clip in a video, the video level representation is the average of all clip level representations. L2 distance is used to measure the similarity among videos.

4 Evaluation

In this section, we list the evaluation settings and discuss experimental results.

4.1 Evaluation Settings

We introduce evaluation metric, dataset and image baseline models. Implementation details are also shown.

Metric. We use the standard evaluation metrics: mean average precision score (mAP) and the cumulative matching curve (CMC) at rank-1, rank-5, rank-10 and rank-20.

Dataset. We test all models on the MARS dataset [17], which is the largest video-based person reID dataset to date. MARS consist of "tracklets" which have been grouped into person IDs. It contains 1261 IDs and around 20000 tracklets. The train and test are split evenly.

Implementations. Standard ResNet-50 [6] pretrained on ImageNet is used as the 2D CNN and 3D ResNet-50 [5] pretrained on Kinectics [9] is used as the 3D CNN video encoder. The video frames are resized to 224 × 112. Adam [10] is used to optimize the networks. Batch size is set to 32; if the total memory usage exceeds GPU memory limit, we accordingly reduce the batch size to the maximum possible size. In a batch, we select $P = 4$ samples for each identity. We test the learning rate with 0.0001 and 0.0003 for different models to achieve the best performance.

Image-based baseline models. We provide an image-based baseline model to test the effectiveness of temporal modeling. This model is similar to [7], but use an additional Softmax cross-entropy loss. Specifically, the sequence length of the clip is set to $T = 1$, no temporal modeling method is used. The same loss function (triplet loss and cross-entropy loss) and ResNet-50 network is adopted. During testing, the same similarity calculation procedure is used as described in Sec 3.3.

4.2 Experiments on MARS

In this part, we report the performance of 3D CNN, temporal pooling, temporal attention and RNN separately and then discuss the experimental results.

3D CNN. ResNet3D-50 [5] is used as the test architecture, which has the same number of layer as ResNet-50. ResNet3D-50 is a fully convolutional network, so we can change the input image sequence length. The input image height and width are kept as 224 and 112. Due to limited GPU memory, we only test sequence length for $T = 4$ and $T = 8$. We set the learning rate as 0.0001. The results are shown in Table 1. We can see that $T = 4$ performs better $T = 8$.

Temporal pooling. The average pooling and max pooling model are tested with different sequence lengths. The learning rate is set to 0.0003, as we found this rate achieves the best

Table 1: Comparison of different sequence length $T = 4,8$ with 3D CNN.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
T=4	**70.5**	**78.5**	**90.9**	**93.9**	**95.9**
T=8	69.1	78.8	89.8	93.2	95.1

performance. First, we compare average pooling and max pooling with the same sequence length $T = 8$, as shown in Table 2.

Table 2: Comparison between average pooling and max pooling with sequence length $T = 8$.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
avg pool	**76.2**	**82.9**	**93.7**	**95.4**	**96.8**
max pool	74.5	83.1	93.3	95.6	96.7

We can see that average pooling consistently works better than max pooling. Next, we test average pooling with different sequence length $T = 1,2,4,8$. $T = 1$ is image-based baseline model and does need to use a temporal aggregation method. The results are shown in Table 3, we can see that $T = 4$ achieves the best performance.

Table 3: Comparison of different sequence length $T = 1,2,4,8$ with average pooling.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
T=1	74.1	81.3	92.6	94.8	96.7
T=2	76.1	83.3	93.2	95.5	96.8
T=4	76.5	83.5	93.0	95.3	96.8
T=8	76.2	82.9	93.7	95.4	96.8

Temporal attention. We evaluate two temporal attention models mentioned in Sec 3: “spatial conv + FC" and “spatial conv + temporal conv" and two attention generation functions (Softmax and Sigmoid). The sequence length is set to $T = 4$ (according to the experiments above, $T = 4$ achieves the best results), learning rates are both set to 0.0003, and d_t is set to 256. The comparison between attention generation functions are shown in Table 4, “spatial conv + FC" is used. We can see that “softmax" and “sigmoid" performs similarly. The comparison between attention generation network is shown in Table 5, “softmax" is used as attention generation function. It can be seen that “spatial conv + temporal conv" performs better than “spatial conv + FC", which shows the effectiveness of using temporal convolution to capture information amongst frames.

RNN. We first test the RNN model with different output selections: final hidden state $f_c = h_c^T$ and cell output average pooling $f_c = \overline{o_c^T}$, as described in Sec 3. Sequence length is set to 8, and learning rate is set to 0.0001. LSTM is used as the basic RNN cell. Different hidden state size (512,1024,2048) are tested. The results for final hidden state and cell output average pooling are shown in Table 6 and Table 7 respectively. We can see that “cell outputs average" works generally better than “final hidden state". $D_h = 512$ achieves the best performance in both two models.

To test different types of RNN cells (LSTM and GRU), we fix hidden state size to be $D_h = 512$, sequence length to be $T = 8$ and use “cell outputs average" as the clip representation. The results are shown in Table 8, it can be seen that LSTM outperforms GRU consistently.

To test RNN performance with different test sequence lengths $T = 2,4,8,16$, we fix hidden state size to be $D_h = 512$, use LSTM cell and “cell outputs average" as the clip representation. The results are shown in Table 9, we can see that $T = 4$ gives slightly accurate

Table 4: Comparison between the Softmax and Sigmoid attention generation function.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
softmax	75.8	82.7	92.8	95.4	97
sigmoid	76.1	82.8	93.6	95.4	96.8

Table 5: Comparison between the “spatial conv + FC" and “spatial conv + temporal conv", Softmax is used for attention generation.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
spatial conv + FC	75.8	82.7	93.0	95.1	96.5
spatial + temporal conv	**76.7**	**83.3**	**93.8**	**96.0**	**97.4**

results than $T = 2$ and $T = 8$.

Comparison with state-of-the-art methods. We select the best setting in each model (temporal pooling, temporal attention and RNN), and compare their performance; results are shown in Table 10. For temporal pooling, we select mean pooling and set the sequence length to $T = 4$.; for temporal attention, we select “Softmax" + “spatial conv + temporal conv" and set $T = 4$; for RNN, we set $D_h = 512, T = 4$ and use “cell outputs average". We also list the image-based baseline model in Table 10. We can see that the performance of RNN is even lower than that of image-based baseline model, which implies that using RNN for temporal aggregation is less effective. Temporal attention gives slightly better performance than mean pooling, and outperforms the image-based baseline model by 3%, which shows the effectiveness of temporal modeling. State-of-the-art methods are also listed in Table 10. It can be seen that our image-based baseline model (mAP=74.1%) already outperforms state-of-the-art model [7] (mAP=67.7%) by a large margin; we believe that the improvement mainly comes from the use of Softmax cross-entropy loss. We also list the performance after re-ranking [23] in Table 10, which brings another 7.8% improvement in mAP and 1.7% in CMC-1.

Discussion. Based on the experiments, it can be seen that mean pooling gives 3% improvement over the image baseline model, which indicates that modeling temporal information on clip level is effective for video-based person reID. Comparing RNN with mean pooling and image baseline model, we can see that RNN's performance is inferior, even worse than the image-based baseline model, which indicates that RNN is either ineffective way to capture temporal information, or too hard to train on this dataset. The reason that previous work [13, 20] shows improvement by using RNNs may be that image feature extractor they use are from a shallow CNN, which is an inferior model itself compared to dedicated designed pre-trained CNNs [6, 16], and RNN performs like an additional layer to extract more representative features. Our temporal attention gives slightly better performance than mean pooling, the reason may be that mean pooling is already good enough to capture information in a clip, as a clip only only contains a few frames, which is 1/4 to 1/2 second, it's difficult to observe any image quality change in such a short period. However, the quality difference among clips can be very large, as the whole video can be very long, thus, a possible future direction is that how to aggregate clip level information (our current solution averages all clips).

Table 6: Comparison of different hidden state size $D_h = 512, 1024, 2048$ with using “final hidden state" as the clip level representation.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
512	**69.8**	**79.3**	**91.7**	**93.3**	**96.2**
1024	66.7	76.4	89.0	92.2	94.7
2048	67.0	77.7	89.6	92.8	94.9

Table 7: Comparison of different hidden state size $D_h = 512, 1024, 2048$ with using the “cell outputs average" as the clip level representation.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
512	72.0	**80.4**	**92.7**	**94.9**	**96.9**
1024	**72.1**	81	92.5	94.8	96.5
2048	70.1	79.5	91	93.8	95.5

Table 8: Comparison between LSTM and GRU with using the “cell outputs average" as the clip level representation and $D_h = 512, T = 8$.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
LSTM	**72.0**	**80.4**	**92.7**	**94.9**	**96.9**
GRU	70.5	79.7	91.5	93.8	95.3

Table 9: Comparison of different sequence length $T = 2, 4, 8, 16$ of the RNN model. LSTM cell is used and “cell outputs average" is used as the clip representation.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
T=2	72.4	81.4	92.1	94.9	96.4
T=4	**73.9**	**81.6**	**92.8**	94.7	96.3
T=8	72.0	80.4	92.7	**94.9**	**96.9**
T=16	60.3	71.2	86.5	90.8	93.7

Table 10: Comparison of temporal pooling, temporal attention and RNN with the image-based baseline model.

	mAP	CMC-1	CMC-5	CMC-10	CMC-20
Zheng *et al.* [17]	45.6	65.0	81.1	-	88.9
Li *et al.* [11]	56.1	71.8	86.6	-	93.1
Liu *et al.* [12]	51.7	73.7	84.9	-	91.6
Zhou *et al.* [24]	50.7	70.6	90.0	-	97.6
Hermans *et al.* [7]	67.7	79.8	91.4	-	-
Ours (image)	74.1	81.3	92.6	94.8	96.7
Ours (3Dconv)	70.5	78.5	90.9	93.9	95.9
Ours (pool)	76.5	83.3	93.0	95.3	96.8
Ours (att)	**76.7**	**83.3**	**93.8**	**96.0**	**97.4**
Ours (RNN)	73.9	81.6	92.8	94.7	96.3
Hermans*et al.* (re-rank) [7]	77.4	81.2	90.7	-	-
Ours (re-rank)	**84.5**	**85.0**	**94.7**	**96.6**	**97.7**

5 Conclusion

Video-based person reID is an important task, which has received much attention in recent years. We comprehensively study and compare four different temporal modeling methods for video-based person reID, including temporal pooling, temporal attention, RNN and 3D convnets. To directly compare these methods, we fix the base network architecture (ResNet-50) and the loss function (triplet loss and softmax cross-entropy loss). We also propose a new attention generation network which adopts temporal convolutional layers to extract temporal information among frames. The evaluation is done on the MARS dataset. Experimental results show that RNN's performance is inferior, even lower than the image baseline model; temporal pooling can bring 2%-3% improvement over the baseline; our temporal-conv-based attention model achieves the best performance among all temporal modeling methods.

References

[1] Dahjung Chung, Khalid Tahboub, and Edward J Delp. A two stream siamese convolutional neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1991, 2017.

[2] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017.

[3] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *BMVC*, 2017.

[4] Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017.

[5] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *arXiv preprint*, arXiv:1711.09577, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[8] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[9] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*, 2017.

[12] Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4694–4703. IEEE, 2017.

[13] Niall McLaughlin, Jesus Martinez del Rincon, and Paul Miller. Recurrent convolutional network for video-based person re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1325–1334. IEEE, 2016.

[14] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[15] Kang Gu Yang Yang Shiyu Chang Shuangjie Xu, Yu Cheng and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *ICCV*, 2017.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] *MARS: A Video Benchmark for Large-Scale Person Re-identification*, 2016. Springer.

[18] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3980–3989. IEEE, 2017.

[19] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[20] Yichao Yan, Bingbing Ni, Zhichao Song, Chao Ma, Yan Yan, and Xiaokang Yang. Person re-identification via recurrent feature aggregation. In *European Conference on Computer Vision*, pages 701–716. Springer, 2016.

[21] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017.

[22] Liming Zhao, Xi Li, Yueting Zhuang, and Jingdong Wang. Deeply-learned part-aligned representations for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3219–3228, 2017.

[23] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3652–3661. IEEE, 2017.

[24] Zhen Zhou, Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. 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 (CVPR), 2017 IEEE Conference on*, pages 6776–6785. IEEE, 2017.