Deep Spatio-temporal Network for Accurate Person Re-identification

Quan Nguyen Hong[*†], Nghia Nguyen Tuan[*], Trung Tran Quang[*], Dung Nguyen Tien[*], Cuong Vo Le[*]
[*]School of Electronics and Telecommunications
Hanoi University of Science and Technology
Hanoi, Vietnam
[†] Faculty of Information Technology
Viet-Hung Industrial University
Hanoi, Vietnam

***Abstract*—Feature extraction is one of two core tasks of a person re-identification besides metric learning. Building an effective feature extractor is the common goal of any research in the field. In this work, we propose a deep spatio-temporal network model which consists of a VGG-16 as a spatial feature extractor and a GRU network as an image sequence descriptor. Two temporal pooling techniques are investigated to produce compact yet discriminative sequence-level representation from a sequence of arbitrary length. To highlight the effectiveness of the final sequence-level feature set, we use a cosine distance metric learning to find an accurate probe-gallery pair. Experimental results on the ilIDS-VID and PRID 2011 dataset show that our method is slightly better on one dataset and significantly better on the other than state-of-the-art ones.**

Keywords—Person Re-identification, Multi-shot, GRU, VGG-16

I. Introduction

Person re-identification (re-id) has been popular in Computer Vision for years. Essentially, it is the process of recognizing a person in images captured from a non-overlapping camera network. The results of the re-id can be applied to automated surveillance applications, like person tracking or retrieval of all videos containing the interested persons. Based on the number of images used to build a feature set for each person, re-id works can be divided into two approaches: single-shot and multi-shot. Recent works have been focusing on the multi-shot approach because of some reasons. Firstly, in practical surveillance, persons usually appear in a video, rather than a still image. In addition, heavy occlusion and viewpoint variations cause the appearances of a person in multiple cameras to be either unclear or inconsistent, thus making it harder to recognize the person by using the single-shot methods. With the multi-shot approach, it is essential to make use of sequential data, which is usually called sequence representation. A few methods have been proposed to solve this task, most of which use recurrent neural networks [1]–[3].

Yan et al. [1] successfully built a recurrent feature aggregation network (RFA-Net) with one long short-term memory (LSTM) layer. This network takes a set of frame-level features extracted from each image in a sequence as the input, and produces sequence-level features. However, the authors only used simple hand-crafted features, LBP&Color [5] as frame-level features. That may be the cause of not fully promoting the effectiveness of the network. Additionally, in some cases, the gated recurrent unit (GRU) [6] architecture is proved to be more effective than the LSTM architecture [7].

In this paper, we present a novel method based on the RFA-Net that uses frame-level features extracted from a VGG-16 convolutional neural network [8] as inputs, and uses the GRU architecture instead of the LSTM one. Under the same circumstance as the one mentioned in [1], the experimental results show that our method outperforms the original method on both iLIDS-VID [9] and PRID 2011 [10] dataset. Moreover, the performance of our proposed method is slightly better than the state-of-the-art one on the challenging dataset, iLIDS-VID. On PRID 2011, it is significantly better than the others.

II. Related Work

Deep learning-based image representation has become a research field of interested due to its wide range of applications, especially for person re-id. In the past, many hand-crafted features such as LBP [11], color texture, and HOG [12] were investigated. Although they are simple and straight-forward to think of, these features are not descriptive enough for sequence matching. In order to enhance the matching accuracy in the re-id task, state-of-the-art methods focus on building learnable features. Most of those proposals used a convolutional neural network (CNN) which is a popular type of neural network for image processing. Many proposed CNN models achieve increasingly better performance on the renowned ImageNet dataset for object detection. For instance, in the ImageNet large-scale visual recognition challenge contest (ILSVRC) 2014, the winner is GoogLeNet [13]. K.Simonyan et al. [8] constructed a famous CNN model named VGG-16 to extract features from images, audiovisual and textual data. There are some other architectures which are worth mentioning such as AlexNet from [14], OverFeat from [15], and a recent model by He et al. [16]. As a result, due to their great generalizing ability, those named models are manipulated for different tasks including person re-id. They can be either directly used as fix feature extractors or fine-tuned to fit the target problem. In general, they are the advanced replacement of hand-crafted methods.

Recently, person re-id has been focusing on the multi-shot approach. Sharing the same goal with single-shot approach, multi-shot one aims to describe a person in an image sequence

978-1-5386-0962-0/17/$31.00 ©2017 IEEE

instead of one single image. Most state-of-the art multi-shot methods inherit the feature extracting algorithm in image processing. However, to make full use of sequential data, a temporal information extractor must be investigated besides the existing spatial ones. Towards solving this problem, most of the proposed methods used a recurrent neural network (RNN), a powerful tool to learn sequential data. Yan et al. [1] constructed a model with a long short-term memory (LSTM) network to build a sequence-level representation from simple features. McLaughlin et al. [2] successfully used a convolutional neural network with a recurrent one to boost the accuracy of video matching. Besides RNN and LSTM, gated recurrent unit (GRU) [6] is a popular recurrent architecture that can be used as sequence learner in person re-id.

Inspired by the above ideas, in this paper, we propose a deep spatio-temporal network model to solve multi-shot person re-id problem. Our model has a VGG-16 spatial feature extractor and a GRU network to learn temporal information. Image sequence flowing into the model are well represented with the help of an additional temporal pooling technique, thus, improving the matching accuracy of the system.

III. PROPOSED METHOD

We propose a deep spatio-temporal network model to address the video-based person re-identification problem. We first introduce the overall architecture of the proposed model in subsection III-A. Then, we discuss each component of the model in detail in the three last subsections.

A. Architecture Overview

A diagram of our proposed model is illustrated in Fig 1. At first, each image frame is fed through a convolutional neural network to create a feature set that describes the appearance of the person. This process is repeated a number of times that equals to the number of frames in the input sequence, resulting in multiple feature sets to be fed into a recurrent layer constructed of GRU unit. The sequential information of all output nodes is then combined using temporal pooling methods. Finally, the output of our network is a sequence-level representation of the person. The proposed model allows descriptive information to be selectively maintained while the rest is eliminated, thus making the final feature set compact and discriminative.

We only train the recurrent layer of the whole network. During the training phase, input to the recurrent layer is the spatial feature sets of the persons in the training set. A dropout layer and a loss function are concurrently adopted after the recurrent layer. The network is trained as a classification problem. During the test phase, the output nodes of the recurrent layer is connected to a temporal pooling layer as described.

In the following subsections, we will explain in greater detail the architecture of each main block, including the spatial and temporal feature extractors and the temporal pooling layer.

B. Spatial Feature Extractor

The VGG-16 [8] is selected to use as a spatial feature extractor in our network. Proposed by Simonyan et al., VGG-16 is a deep feed-forward CNN model that won the ILSVRC 2014 [17] in the single-object localization, and took the second place in image classification. The model includes 16 learnable weight layers, 13 of which are convolutional layers and the others are fully connected layers. VGG-16 is proved too well generalize on other datasets, where it achieves state-of-the-art results [8].

Due to the fact that the datasets for the multi-shot person re-identification are poor while generalizing ability of VGG-16 is high, we choose to use the pre-trained VGG-16 model as fixed feature extractor. The input to the network is a single image of a person. The raw input can be of any size. It is then resized to 224× 224× 3 before being fed into the model. Spatial feature set is extracted from “pool5” pooling layer as illustrated in Fig. 2. This set is originally a 3-dimensional array of 512 × 7 × 7 until it is reshaped to a vector of 25088 elements, which is a preparation before flowing into recurrent layer.

In summary, we propose using all 13 convolutional layers plus 5 pooling layers of pre-trained VGG-16 model and extract features from the last pooling layer of the original VGG-16 architecture. This is repeated N times that depends on the length of the input video or image sequence. All N 25088-element feature vectors are the input of a temporal feature extractor, which will be discussed in the following subsection.

C. Sequence-level Representation

Traditional neural networks lack the ability to create a link between frames since they process each image disjointedly. To make use of temporal information, we deploy a recurrent network to process spatial feature sets of multiple images in the sequence. Recurrent networks are networks with loops in them. An unrolled recurrent network is similar to one containing a series of identical layer, in which one layer connects and transfers its output to another layer sequentially. The role of the loops is to connect the information from previous nodes with the current one and to make use of parameter sharing. In other words, a single output of the network concurrently depends on the current and a number of previous inputs.

There are some popular variants of the original recurrent neural network (RNN). We choose to use a gated recurrent unit (GRU) [6] in our proposal. One reason for this selection is that the GRUs are able to overcome the long-term dependency problem of the original RNNs, where the recurrent network does not effectively learn from a long image sequence. In addition, the GRUs have been proved to work slightly better than others like long short-term memory (LSTM) [4] in some tasks.

The structure of a GRU is described as follows:

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \tag{2}$$

$$g_t = \tanh\left[W_{gx}x_t + W_{ghr}(h_{t-1} \cdot r_t) + b_g\right] \tag{3}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot g_t \tag{4}$$

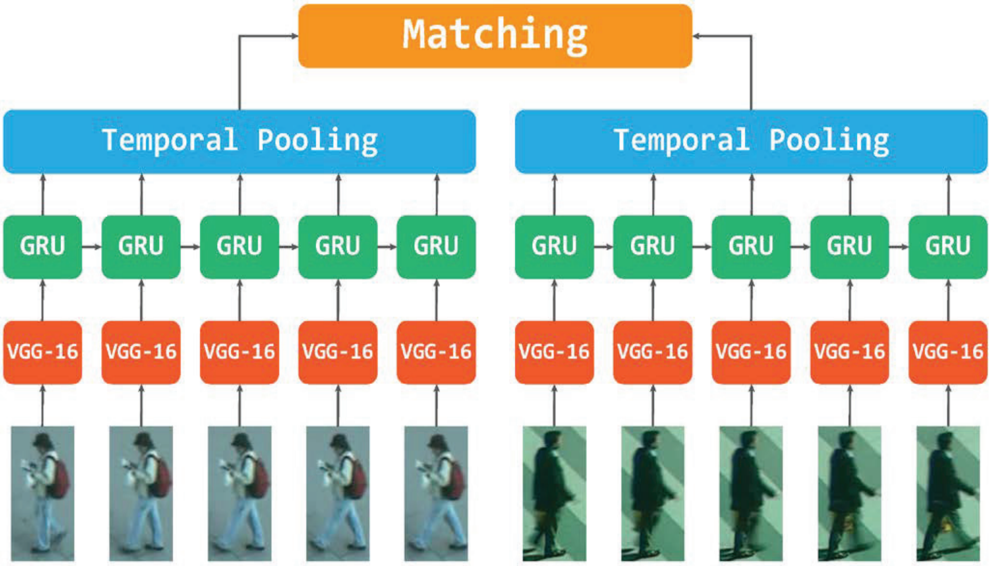

Fig. 1. Deep spatio-temporal network for person re-identification. An image sequence of a person is fed through a pre-trained VGG-16 model to extract spatial features. Then, the feature sets of multiple images flow into a gated recurrent unit, and finally through a temporal pooling layer to create sequence-level representation. The same procedure is applied to other image sequences for matching.

where t and $t-1$ denotes the current and the previous time step, respectively. In the above equations, σ is the sigmoid function and "·" denotes an element-wise multiplication operator. The inputs to a node of the GRU consist of one data vector x_t and one output vector of the previous time-step h_{t-1}. In detail, x_t is a spatial feature set of the current time-step which is extracted by VGG-16 as explained in the previous subsection. The GRU has six learnable weight vectors W_{zx}, W_{zh}, W_{rx}, W_{rh}, W_{gx}, W_{ghr} and three biases b_z, b_r, b_g. During training, these weights and biases are tuned. Since parameter sharing is applied, they will not alter as time-step t varies.

The output of the GRU at time-step t is h_t, called hidden output. Commonly, the dimension of h_t is many times smaller than one of the input data x_t. At each time-step, the current hidden output has information of all the previous inputs up to the current one. To summarize all outputs and form a complete sequence-level feature set, the hidden outputs of multiple nodes are fed through a temporal pooling layer. The detail of this layer is given in the following subsection.

D. Temporal Pooling

Temporal pooling is a useful technique to summarize information over a long image sequence. Sharing the same purpose with other techniques, like fusion and concatenation, however, temporal pooling keeps the final feature set compact yet still descriptive. This technique helps avoid bias towards later time steps by looking at all input feature sets and capturing the relevant information that presents in the sequence. In our proposal, we apply two methods: max pooling and mean pooling.

Max pooling over the temporal dimension is used to select the maximum elements of the input feature vectors as follows:

$$s^i = \max(h_1^i, h_2^i, \dots, h_T^i) \tag{5}$$

where T is the length of the sequence, s^i denotes the i-th element of the final sequence-level feature vector s, and h_1^i, h_2^i, …, h_T^i are equivalent i-th elements in the hidden output vectors of T time-steps of the recurrent layer. Meanwhile, the mean pooling is as follows:

$$s = \frac{1}{T}\sum_{t=1}^{T} h_t \tag{6}$$

Mean pooling tends to produce a smooth sequence-level representation by averaging all noisy feature vectors. This is different from the max pooling method, which seeks for the outstanding values. The performance of these two methods will be compared in detail later.

In summary, we aim to create a sequence-level representation from a sequence of arbitrary length. The result feature vector has the same dimension as of the input. This is an advantage over other methods like feature concatenation which increases the dimension as the number of inputs increases. However, compactness must follow with high performance. In the next section, we show step-by-step how the experiments were carried out and discuss the results in detail.

IV. EXPERIMENTS

A. Datasets

Two datasets (iLIDS-VID [9] and PRID 2011 [10]), which are common for video-based person re-identification with public benchmarks available, are used to evaluate the performance of our proposed model.

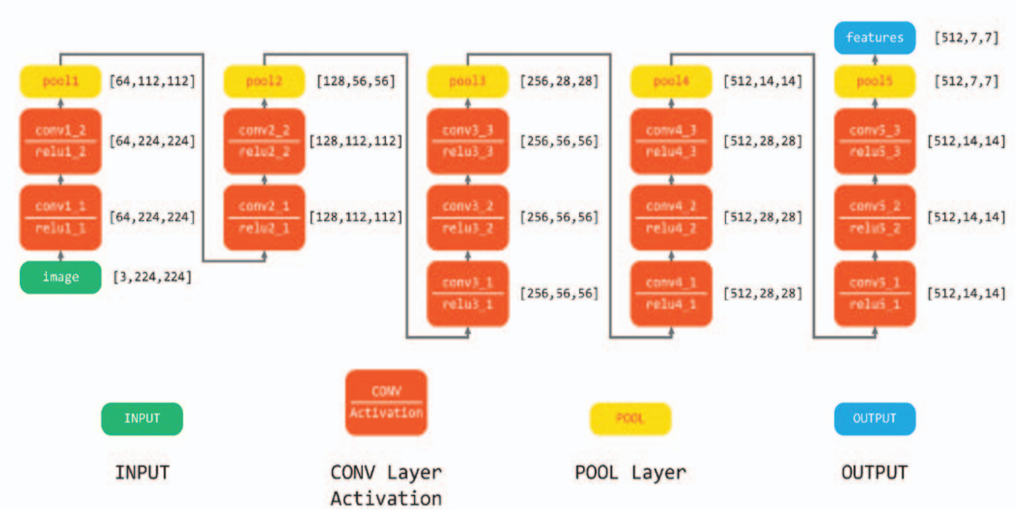
features [512,7,7]
pool1 [64,112,112]
conv1_2 relu1_2 [64,224,224]
conv1_1 relu1_1 [64,224,224]
image [3,224,224]
pool2 [128,56,56]
conv2_2 relu2_2 [128,112,112]
conv2_1 relu2_1 [128,112,112]
pool3 [256,28,28]
conv3_3 relu3_3 [256,56,56]
conv3_2 relu3_2 [256,56,56]
conv3_1 relu3_1 [256,56,56]
pool4 [512,14,14]
conv4_3 relu4_3 [512,28,28]
conv4_2 relu4_2 [512,28,28]
conv4_1 relu4_1 [512,28,28]
pool5 [512,7,7]
conv5_3 relu5_3 [512,14,14]
conv5_2 relu5_2 [512,14,14]
conv5_1 relu5_1 [512,14,14]
INPUT
INPUT
CONV Activation
CONV Layer Activation
POOL
POOL Layer
OUTPUT
OUTPUT

Fig. 2. The architecture of VGG-16 model as fixed feature extractor. All 13 convolutional layers and 5 pooling layers are used. The spatial features are extracted from "pool5" layer with a dimension of 512 × 7 × 7.

iLIDS-VID dataset. This dataset consists of 600 image sequences for 300 persons. Each person has one pair of image sequences, and each image sequence has variable lengths of 23 to 192 frames, with an average of 73 frames. This dataset was created based on two non-overlapping cameras of a CCTV network at an airport arrival hall. The variations of lighting and viewpoint, cluttered background, and occlusions make this dataset a challenging one.

PRID 2011 dataset. This dataset includes 400 images sequences of 200 persons. Each person has one pair of image sequences, each of which consists of 5 to 675 frames, with an average of 100 frames. The images were captured from two cameras in public outdoor environment with clean background and rare occlusion, which make it simpler than the iLIDS-VID dataset.

Like Wang et al. [9], we filter out from each dataset the persons whose sequences have fewer than 21 frames. As a result, 600 image sequences of 300 persons in iLIDS-VID and 356 image sequences of 178 persons in PRID 2011 were used for evaluation. Each dataset was randomly split into two subsets, one for training the model and other for testing. During the test phase, image sequences from the first and second camera were regarded as probe and gallery, respectively.

B. Layer Settings

Before training and testing, we set up each layer in the network. The detail settings are as follows: **VGG-16**. For the spatial feature extractor, we used a pre-trained VGG-16 model. The settings for all the convolutional and pooling layers were maintained as the original. The fully connected layers were discarded and spatial features were extracted from "pool5" layer. In short, the input to the extractor is a 224 × 224 × 3 image and the output is a 25088-element feature vector.

GRU. Inspired by the work of Yan et al. [1], we set the dimension of the hidden output to 512. During training, the output of the GRU was fed to a dropout layer, then a fully connected layer and finally a Softmax layer to calculate loss. Networks were trained as a classification task with the number of classes, N, equal to the number of persons in the training set, i.e. N = 89 and N = 150 for the PRID 2011 and iLIDSVID dataset, respectively. As suggested by Yan et al. [1], we also chose 10 random subsequences to create averaged final sequence-level representation to avoid walking-cycle and pose changing problem in multi-shot person re-identification.

Temporal Pooling. We deployed two temporal pooling methods as discussed in the previous section (III-D). In our experiments, we compared re-identification performance of the proposed network when applying max pooling and mean pooling technique to performance when applying concatenation. The detail will be shown later in this section.

Cosine Distance Metric Learning. A cosine distance metric learning method is used to find probe-gallery pair. Let *d* be the distance; then, it is computed as follows:

$$d_{ij} = \frac{s_i^a . s_j^b}{\|s_i^a\| \|s_j^b\|} \tag{7}$$

where s_i^a and s_j^b are final sequence level representation of person i from camera a and person j from camera b respectively, and $\|v\|$ denotes L_2 norm of the vector v. For one probe, we compute the distances to all instances in the gallery and list them in ascending order. If the true match s_i^a and s_j^b is expected in top k, the distance d_{ij} have to appear within ascending list of k distances.

C. Evaluation protocols

To obtain stable statistical results, we conduct 10 trials for each architecture with different training/testing splits of each dataset and report the averaged results. A cumulative match characteristic (CMC) curve is adopted to show accuracy of each model with ranking.

Since VGG-16 was deployed as fixed feature extractor, we did not fine-tune the pre-trained model. During training phase, only the recurrent layer is tuned. We tried different set of hyper-parameters to find out the most suitable one. In detail, we trained the model for at least 30,000 iterations with a batch size of 8 on PRID 2011 dataset. On iLIDS-VID, we train for 60,000 iterations with the same batch size. Learning rate was initially set to 0.001 and dropped to 0.0001 after 20,000 and 40,000 iterations with the PRID 2011 and iLIDS-VID dataset respectively. Early-stopping strategy was applied to avoid overfitting. The final models were carefully selected from a few snapshots to ensure that they were the most optimized ones. All implementation and training was based on the Caffe deep learning framework [18]. It took an average of 30 minutes to train a model on a machine with an Intel Xeon CPU E3-1245 v5 and an NVIDIA Titan X GPU card.

D. Results

We show the experimental results in Table I and II.

Comparison of temporal pooling methods. In Table I, TP_Mean and TP_Max denote mean and max temporal pooling respectively. From the table, it is clear that applying temporal pooling is more appropriate than concatenation. The quality of the sequence-level feature vector does not change despite that its dimension is 10 times smaller than concatenated vector in our experiment (512 and 5120). We also notice a slight improvement over concatenation method when applying mean pooling. Mean pooling gives a better result with 0.8% higher than concatenation method at rank-1. This is not a great difference, however with the small-size advantage, it is considerable improvement for the person re-identification task.

Comparison with state-of-the-art. The matching accuracy of the proposed model is shown in Table II. For the challenging dataset iLIDS-VID, the matching accuracy at rank-1 increases by 0.5% comparing to the nearest neighbor. However, in detail, our proposed method has two advantages over the one proposed in [1]. The first advantage is that our sequence-level feature vectors are 10 times smaller than ones in [1]. On the other hand, we used simple cosine distance metric learning to match two feature sets while the work by Yan et al. proposed using RankSVM [22] which is a complex matching algorithm. Under the same circumstance in which both methods used cosine distance, our method outperformed the others. The performance gap is enlarged in the simpler, PRID 2011, dataset. The proposed method stands at the first place as it achieved 75.1% at rank-1, which is 11% higher than the second one. As a result, our method also outperformed other state-of-the-art ones on two datasets.

V. CONCLUSION

In this work, we propose a novel method to increase the quality of features for video person re-identification. This is achieved by building a deep spatio-temporal network model with a VGG-16 as spatial feature extractor and a GRU as sequence descriptor. We apply temporal pooling technique to produce compact yet discriminative sequence-level representations. The experimental results showed that mean temporal pooling is appropriate for our proposal. It helps increase the matching accuracy at rank-1 to 49.8% and 75.1% on the iLIDS-VID and PRID 2011 dataset respectively. The proposed method is slightly bettern than other state-of-the-art ones on the challenging dataset and outperforms most of them on the other.

VI. ACKNOWLEDGMENT

We would like to thank the Computer Architecture and Parallel Processing Lab, Department of Electrical and

Computer Engineering, Seoul National University, Seoul, South Korea for supporting the machines which are used to conduct our experiments.

REFERENCES

[1] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang, "Person reidentification via recurrent feature aggregation," in European Conference on Computer Vision. Springer, 2016, pp. 701–716.

[2] N. McLaughlin, J. Martinez del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1325–1334.

[3] K. Liu, B. Ma, W. Zhang, and R. Huang, "A spatio-temporal appearance representation for viceo-based pedestrian re-identification," in Proceedings of the IEEE International Conference on Computer Vision, 2015,pp. 3810–3818.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] M. Hirzer, P. Roth, M. Kostinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," Computer Vision–ECCV 2012, pp. 780–793, 2012.

[6] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[7] D. Steckelmacher and P. Vrancx, "An empirical comparison of neural architectures for reinforcement learning in partially observable environments," arXiv preprint arXiv:1512.05509, 2015.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *European Conference on Computer Vision*. Springer, 2014, pp. 688–703.

[10] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person reidentification by descriptive and discriminative classification," in *Scandinavian conference on Image analysis*. Springer, 2011, pp. 91–102.

[11] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[19] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by discriminative selection in video ranking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 12, pp. 2501–2514, 2016.

[20] S. Karanam, Y. Li, and R. J. Radke, "Person re-identification with discriminatively trained viewpoint invariant dictionaries," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4516–4524.

[21] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2288–2295.

[22] O. Chapelle and S. S. Keerthi, "Efficient algorithms for ranking with svms," *Information Retrieval*, vol. 13, no. 3, pp. 201–215, 2010.

TABLE I. PERFORMANCE COMPARISON OF THE PROPOSED MODEL WHEN APPLYING DIFFERENT TEMPORAL POOLING METHODS

Dataset	**iLIDS-VID**				**PRID 2011**			
CMC Rank	**1**	**5**	**10**	**20**	**1**	**5**	**10**	**20**
VGG-GRU+TP_Mean	49.8	77.4	86.5	93.5	75.1	93.7	97.5	99.5
VGG-GRU+TP_Max	49.1	76.8	86.3	93.4	74.6	93.5	97.7	99.5
VGG-GRU+Concat	49.8	77.4	86.5	93.4	74.3	93.5	97.5	99.5

TABLE II. PERFORMANCE COMPARISON OF THE PROPOSED MODEL AND THE STATE-OF-THE-ART

Dataset	**iLIDS-VID**				**PRID 2011**			
CMC Rank	**1**	**5**	**10**	**20**	**1**	**5**	**10**	**20**
VGG-GRU+TP_Mean	**49.8**	**77.4**	**86.5**	**93.5**	**75.1**	**93.7**	**97.5**	**99.5**
LBP&Color+RFA-Net+RankSVM [1]	49.3	76.8	85.3	90.0	58.2	85.8	93.4	97.9
LBP&Color+RFA-Net+Cosine [1]	44.5	71.9	82.0	90.1	54.9	84.2	93.7	98.4
3D HOG&Color + DVR [19]	39.5	61.1	71.7	81.0	40.0	71.7	84.5	92.2
DVDL [20]	25.9	48.2	57.3	68.9	40.6	69.7	77.8	85.6
STFV3D [3]	37.0	64.3	77.0	86.9	21.6	46.4	58.3	73.8
STFV3D+KISSME [21]	44.3	71.7	83.7	91.7	64.1	87.3	89.9	92.0