Learning Effective Representations from Global and Local Features for Cross-View Gait Recognition

Beibei Lin[1], Shunli Zhang[1], Xin Yu[2], Chuihan Kong[1], and Chenwei Wan[1]

[1] Beijing Jiaotong University
{18126289, slzhang, 19121723, 20126339}@bjtu.edu.cn
[2] University of Technology Sydney
xin.yu@uts.edu.au

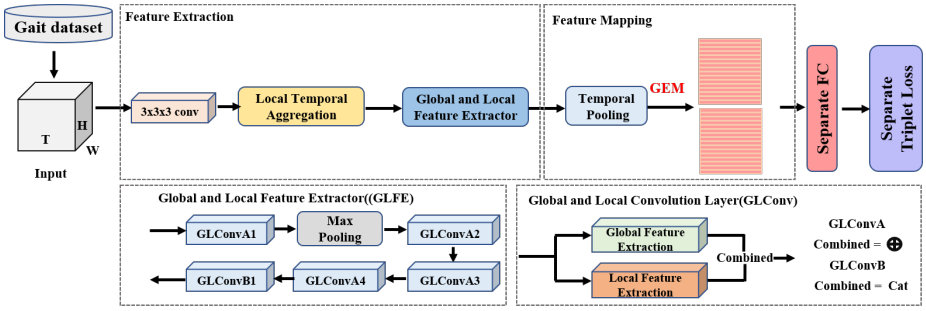

Fig. 1. Overview of the whole gait recognition framework

1 Proposed Method

In this section, we first overview the framework of the proposed method. Then we describe the key components of the proposed method, including Local Temporal Aggregation (LTA), Global and Local Feature Extractor (GLFE) and Generalized-Mean (GeM) pooling layer [1]. Finally, the details of training and testing are presented.

1.1 Overview

The overview of the proposed method is shown in Fig. 1, which aims to extract more comprehensive feature representation for gait recognition, includes three key components. First, the Local Temporal Aggregation (LTA) operation is designed to aggregate the temporal information and preserve more spatial information for trade off. After that, the global and local feature extractor (GLFE) is implemented to extract the combined feature ensembling both global and local information. Then, we leverage temporal pooling and GeM pooling layer to implement feature mapping. Finally, we choose the separate triplet loss to train the proposed model [2, 3].

1.2 Local Temporal Aggregation

We present the LTA operation to replace the first spatial pooling layer, which can integrate temporal information of local clips and maintain more spatial information. Assume that $X_{in} \in \mathbb{R}^{C_1 \times T_1 \times H_1 \times W_1}$ is the input gait sequence, where C_1 is the number of input channels, T_1 is the length of the gait sequence and (H_1,W_1) is the image size of each frame. The process can be formulated as follows

$$X_{LTA} = f^{b\times1\times1}_{a\times a\times a}(f^{1\times1\times1}_{a\times a\times a}(X_{in})) \tag{1}$$

where $f^{b\times1\times1}_{a\times a\times a}(\cdot)$ denotes the 3D convolution operation with kernel size a and temporal stride b. $X_{LTA} \in \mathbb{R}^{C_2 \times T_2 \times H_1 \times W_1}$ is the output of LTA operation, in which $T_2 = \lfloor \frac{T_1 - a}{b} \rfloor + 1$. After LTA operation, the channel number of input sequences is C_2 and the length of input sequences becomes T_2, while the image size of input sequences remains the same as before.

1.3 Global and Local Feature Extractor

We propose a novel GLFE module to extract features, which can take advantage of both global and local information. The GLFE module is implemented by the GLConv layer, which contains global and local feature extractors. The global feature extractor can extract the whole gait information, while the local feature extractor is used to extract more details from local feature maps. The GLConv has two different structures due to different combinations, e.g. GLConvA and GLConvB. The GLFE module includes six layers, "GLConvA-SP-GLConvA-GLConvA-GLConvA-GLConvB as shown in Fig.1.

The GLConv layer is shown in Fig.2. Assume that its input is $X_{global} \in \mathbb{R}^{c_1 \times t \times h \times w}$, where c_1 is the number of channels, t is the length of feature maps and (h,w) is the image size of each frame. We first partition the global feature map into n-parts as local feature maps $X_{local} = \left\{X^i_{local} | i = 1, ..., n\right\}$, where n is the number of partitions and $X^i_{local} \in \mathbb{R}^{c_1 \times t \times \frac{h}{n} \times w}$ corresponds to the i-th local gait part. Then, we use 3D convolutions to extract global and local gait features, respectively. Note that all local feature maps share the same convolutional weights. There are two ways to combine the global and local feature maps, i.e. by element-wise addition (GLconvA) or by concatenation (GLconvB). The GLconvA and GLconvB layers can be representated as

$$Y_{GLConvA} = Y_{global} + Y_{local} \in \mathbb{R}^{c_2 \times t \times h \times w} \tag{2}$$

$$Y_{GLConvB} = cat \begin{Bmatrix} Y_{global} \\ Y_{local} \end{Bmatrix} \in \mathbb{R}^{c_2 \times t \times 2h \times w} \tag{3}$$

where *cat* means concatenating operation. Y_{global} and Y_{local} can be represented as

$$Y_{global} = F_{3\times3\times3}(X_{global}) \in \mathbb{R}^{c_2 \times t \times h \times w} \tag{4}$$

$$Y_{local} = cat \begin{Bmatrix} F'_{3\times3\times3}(Y^1_{local}) \\ F'_{3\times3\times3}(Y^2_{local}) \\ \cdots \\ F'_{3\times3\times3}(Y^n_{local}) \end{Bmatrix} \in \mathbb{R}^{c_2\times t\times h\times w} \tag{5}$$

where $F_{3\times3\times3}(\cdot)$ and $F'_{3\times3\times3}(\cdot)$ denote 3D convolutions with kernel size 3.

Based on the above two forms of GLGonv layer, the GLFE module can be built to extract gait features after LTA operation. Experimentally, GLConvA is used to implement the first two GLGonv blocks and GLConvB is ulitized to realize the last one in the GLFE module.

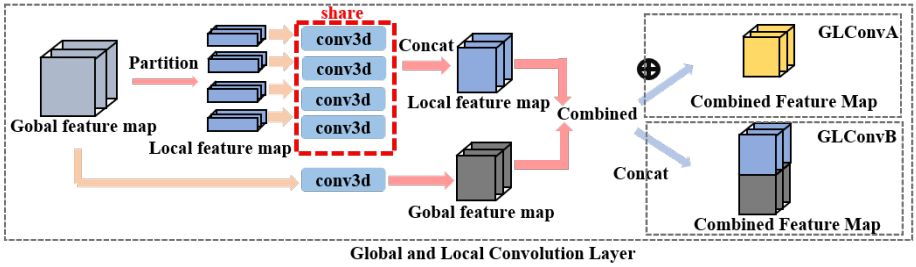

Fig. 2. Architecture of Global and Local Convolution Layer. $\oplus$ means element-wise addition, and 'Concat' means concatenating the feature maps of different parts horizontally.

1.4 Feature Mapping

Since the length of the input gait sequences may be different, we introduce Generalized-Mean pooling mehod to aggregate the temporal information of the whole sequence [2, 3]. Assume that $X_{GLFE} \in \mathbb{R}^{C_3\times T_2\times H_2\times W_2}$ is the output of GLFE module, where C_3 is the number of input channels, T_2 is the length of feature maps and (H_2,W_2) is the spatial size of the feature in each frame. Because of the spatial pooling layer in the GLFE module, the spatial size becomes (H_2,W_2), while the length of feature maps remains unchanged. The temporal pooling $TP(\cdot)$ can be realized by

$$Y_{TP} = Max^{T_2\times1\times1}(X_{GLFE}) \tag{6}$$

where $Y_{TP} \in \mathbb{R}^{C_3\times1\times H_2\times W_2}$ is the output of temporal pooling.

To improve feature representation ability, researchers develop the spatial feature mapping operation with weighted sum [2, 3]. After temporal pooling, gait feature maps are split into strips and two statistical functions, max and average, are used to aggregate each strip's information. The spatial feature mapping can be represented as

$$Y_{MA} = \alpha Max^{1\times1\times W_2}(Y_{TP}) + \beta Avg^{1\times1\times W_2}(Y_{TP}) \tag{7}$$

where $Y_{MA} \in \mathbb{R}^{C_3 \times 1 \times H_2 \times 1}$ is the output of spatial feature mapping. However, the weighted sum strategy is inflexible because trade-off parameters are predefined manully.

Hereby, we introduce the Generalized-Mean pooling to integrate the spatial information adaptively. The GeM pooling layer $GeM(\cdot)$ can be defined as

$$Y_{GeM} = (Avg^{1 \times 1 \times W_2}((Y_{TP})^p))^{\frac{1}{p}} \tag{8}$$

where $Y_{GeM} \in \mathbb{R}^{C_3 \times 1 \times H_2 \times 1}$ is the output of GeM operation. p is the parameter which can be learned by network training. Specifically, if $p = 1$, Y_{GeM} is equal to $Avg^{1 \times 1 \times W_2}(Y_{TP})$, and if $p \to \infty$, Y_{GeM} is equal to $Max^{1 \times 1 \times W_2}(Y_{TP})$. Then, we use multiple separate fully connected layers to further aggregate the information from channels of Y_{GeM}. The feature mapping can be represented as

$$Y_{out} = Separate_{fc}(Y_{GeM}) \in \mathbb{R}^{C_4 \times 1 \times H_2 \times 1} \tag{9}$$

Y_{out} is the output of feature mapping with H_2 horizontal features, each of which has C_4 channels.

1.5 Loss Function

To effectively train the proposed gait recognition framework, we introduce the triplet loss function to calculate the loss[4, 5], which can improve the inter-class distance and reduce the intra-class distance. In the training stage, each horizontal feature of Y_{out} is fed into the triplet loss function to calculate the loss independently. The triplet loss is defined as

$$L_{triplet} = [D(F(A_1), F(B_1)) - D(F(A_1), F(A_2)) + m]_+ \tag{10}$$

where A_1 and A_2 are the samples from the same class A, while B_1 represents the sample from class B. $F(\cdot)$ denotes the feature extraction and mapping operation of the proposed method. $D(d_1, d_2)$ is the Euclidean distance between d_1 and d_2. m is the margin of the triplet loss. The operation $[\gamma]_+$ is equal to $max(\gamma, 0)$.

1.6 Training and Test

Training Stage. During the training stage, we first feed the input gait sequences into the proposed network to generate gait feature representation Y_{out}. Then, the triplet loss function is used to compute the loss and Batch ALL (BA) is adopted as the sampling strategy, which is the same as [4, 2, 3]. Specifically, each batch contains P subject IDs, and K samples are selected from each subject ID. Correspondingly, the batch size is $P \times K$.

Test Stage. During the test stage, the whole gait sequences are put into the proposed network to obtain gait features Y_{out}. Then, the $Y_{out} \in \mathbb{R}^{C_4 \times 1 \times H_2 \times 1}$ can be flattened to a feature vector with dimension $C_4 \times H_2$ and then taken as a sample. To calculate Rank-1 accuracy, the test dataset is divided into two sets, i.e. the gallery set and the probe set. The gallery set is regarded as the

standard view to be retrieved, while the feature vectors of the probe are used to match the feature vectors from the gallery view. Multiple metric strategies, such as Euclidean distance and cosine distance, can be used to calculate similarity between the samples from gallery and probe set. Specifically, Euclidean distance is selected as the metric strategy.

2 Experiments

2.1 Datasets

OUMVLP [6] is one of the largest gait recognition databases, which contains 10,307 subjects in total. Each subject contains two groups of videos, Seq#00 and Seq#01. Each group of sequences are captured from 14 angles: 0°-90° and 180°-270° and the sampling interval is 15°. We adopt the same protocol (5,153 subjects are taken as training data and 5,154 subjects are used as test data) as [2] and [3] to evaluate the proposed method. In the test stage, the sequences Seq#01 are taken as the gallery set, while the sequences Seq#00 are regarded as the probe set to evaluate the performance.

2.2 Implementation Details

We adopt the same preprocessing approach as [2] to obtain gait silhouettes for CASIA-E and OUMVLP datasets. The image of each frame is normalized to the size 64×44. The network parameters are shown in Table.1. m in Equ.10 is set to 0.2. p in Equ.8 is initialized to 6.5. The batch size parameters P and K are set to 12 and 8 in the CASIA-E dataset, respectively. Since the OUMVLP dataset is much larger than CASIA-E, the batch size P$\times$ K is set to $32\times8 = 256$. During the training stage, the length of input gait sequences of the CASIA-E and OUMVLP datasets are set to 64 and 30, respectively. During the test stage, the whole gait sequences are put into the proposed model to extract gait features. All experiments take Adam as the optimizer, and the learning rate is 1e-4. For the OUMVLP dataset, the epoch number is set to 250K. The learning rate is first set to 1e-4 and reset to 1e-5 after 150K. For the CASIA-E dataset, the epoch number is set to 15K. The learning rate is first set to 1e-4 and reset to 1e-5 after 10K. Specifically, The model parameters pre-trained on the OUMVLP dataset are used to initialize our model when training the model parameters of the CASIA-E dataset.

Layer Name	In_C	Out_C	Kernel	Global	N-part
First Conv	1	32	(3,3,3)	✓	×
LTA	32	32	(3,1,1)	–	–
GLConvA1	32	64	(3,3,3)	✓	2
Max Pooling, kernel size =(1, 2, 2), stride=(1, 2, 2)					
GLConvA2	64	128	(3,3,3)	✓	2
GLConvA3	128	128	(3,3,3)	✓	2
GLConvA4	128	128	(3,3,3)	✓	2
GLConvB1	128	128	(3,3,3)	✓	2

Table 1. Network parameters of the proposed method

3 Results

The accuracy of the CASIA-E dataset is 63.0%.

References

1. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE transactions on pattern analysis and machine intelligence **41** (2018) 1655–1668
2. Chao, H., He, Y., Zhang, J., Feng, J.: Gaitset: Regarding gait as a set for cross-view gait recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8126–8133
3. Fan, C., Peng, Y., Cao, C., Liu, X., Hou, S., Chi, J., Huang, Y., Li, Q., He, Z.: Gaitpart: Temporal part-based model for gait recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 14225–14233
4. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
5. Luo, H., Gu, Y., Liao, X., Lai, S., Jiang, W.: Bag of tricks and a strong baseline for deep person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0
6. Takemura, N., Makihara, Y., Muramatsu, D., Echigo, T., Yagi, Y.: Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition. IPSJ Transactions on Computer Vision and Applications **10** (2018) 4