Set Residual Network for Silhouette-Based Gait Recognition

Saihui Hou, Xu Liu, Chunshui Cao, and Yongzhen Huang

Abstract—**Recently gait recognition receives increasing attention since it can be conducted at a long distance without the cooperation of subjects and suit for the cases of changing clothes. A key challenge is to learn gait features from the silhouettes that are invariant to the external factors such as clothing, carrying conditions and camera viewpoints. In this work, we propose a Set Residual Network for gait recognition which tries to learn more discriminative features from the silhouettes. Specifically, the silhouettes of each gait sequence are regarded as an unordered set and we propose a Set Residual Block to extract the silhouette-level and set-level features in a parallel way. Particularly, a residual connection is adopted to connect the two-level features inside each block, which enables more silhouette-set interaction and effectively coordinates the silhouette-level and set-level information for set-based feature learning from the silhouettes. Besides, we propose an efficient strategy to exploit the features from the shallow layers to learn more robust part representations for gait recognition, where the upsampling or lateral connections are unnecessary and only marginal memory cost is required. The experiments on CASIA-B and OUMVLP show that our approach can bring consistent improvement over the baselines for all walking conditions.**

Index Terms—**Gait recognition, silhouette set, set residual learning, dual feature pyramid.**

I. Introduction

Compared to other biometrics such as face, fingerprint and iris, gait recognition can be conducted at a long distance without the cooperation of subjects and suit for the cases of changing clothes, which contribute to its broad applications in crime prevention, forensic identification and social security [1], [2]. However, gait recognition suffers from a lot of external factors such as clothing, carrying conditions and camera viewpoints [3], [4]. A key challenge is to learn gait features that are invariant to the factors mentioned above.

The existing methods for gait recognition mostly take the *silhouettes* of gait sequences as input, which can be roughly divided into three categories. The first category [5], [6], [7], [8], [9] aggregates the silhouettes of a gait sequence into a template, e.g., Gait Energy Image (GEI) [10], which is simple but ignores the temporal information. The second [11], [12], [13] treats the silhouettes of a gait sequence as a video to extract the spatial and temporal information, however, the model is relatively hard to train. The third [14], [15], [16] is recently proposed which regards the silhouettes of a gait sequence as an unordered set and achieves significant improvement. The set-based gait recognition is motivated by the observation that it is feasible to rearrange the shuffled silhouettes of a gait sequence according to their appearances, indicating that the appearance of a silhouette also encodes some temporal information.

In this work, we deal with the silhouette-based gait recognition also regarding the silhouettes of each gait sequence as an unordered set [14], [15], [16]. Differently, we propose a Set Residual Network (SRN) which is motivated to effectively coordinate the silhouette-level and set-level information to learn more discriminative features from the silhouette sets.

In the previous literatures, the unordered set is first introduced to the visual tasks by [17] to tackle the point cloud, where the instance-level[1] and set-level features are extracted in a cascaded way. However, as far as we can see, the instance-level and set-level information are not fully coordinated in this way which are only connected in the middle of the network. Differently, in this work, we propose a Set Residual Block (SRBlock) for set-based gait recognition where the silhouette-level and set-level features are extracted in a parallel way and connected inside each block. Specifically, a basic SRBlock consists of two parallel branches which is denoted as silhouette-branch and set-branch respectively. The silhouette-branch learns the features from each silhouette separately while the set-branch learns the features regarding all silhouettes as a whole. Particularly, the features extracted by silhouette-branch are added to those extracted by set-branch as *residual* signals before the last Leaky ReLU, which is the key in SRBlock and aims to make the two branches cooperate well. It is worth noting that, the structure of SRBlock is similar to the regular residual block [18] where the residual connection is taken to alleviate the gradient vanishing/exploding. While in SRBlock, we adopt the residual connection to facilitate the feature learning from the silhouette sets. In SRN, SRBlock is taken as a basic block to construct the backbone for set-based gait recognition.

Manuscript received December 11, 2020; revised February 26, 2021 and April 13, 2021; accepted April 18, 2021. Date of publication April 22, 2021; date of current version June 29, 2021. This article was recommended for publication by Associate Editor Y. Yagi upon evaluation of the reviewers' comments. *(Corresponding author: Yongzhen Huang.)*

Saihui Hou is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with Watrix Technology Limited Company Ltd., Beijing 100088, China (e-mail: saihui.hou@ia.ac.cn).

Xu Liu and Chunshui Cao are with Watrix Technology Limited Comapny Ltd., Beijing 100088, China.

Yongzhen Huang is with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China, and also with Watrix Technology Limited Co. Ltd., Beijing 100088, China (e-mail: huangyongzhen@bnu.edu.cn).

Digital Object Identifier 10.1109/TBIOM.2021.3074963

[1] Each instance in a set represents a point in [17] and a silhouette in this work. We will use *instance* and *silhouette* interchangeably in the following.

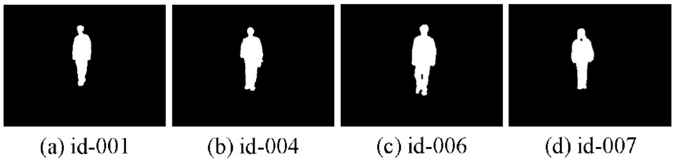

(a) id-001 (b) id-004 (c) id-006 (d) id-007

Fig. 1. Example silhouettes from CASIA-B [3]. The silhouettes belong to different subjects.

Besides, as shown in Fig. 1, the silhouettes for different subjects only have subtle differences in many cases which are usually encoded in the shallow convolutional layers [19], [20]. However, the features extracted by the shallow layers have not been fully exploited for gait recognition. And we notice that slicing the features horizontally to learn part representations has proven to be useful for person re-identification [21], [22], [23]. Inspired by these observations, we propose a novel technique named Dual Feature Pyramid which exploits the silhouette-level and set-level features from the shallow layers to learn more robust part representations for gait recognition. Particularly, we adopt an *efficient* strategy where the upsampling [19] or lateral connections [20] are unnecessary and only marginal memory cost is required.

In summary, the contributions of this work mainly lie in three folds: (1) We propose SRBlock as a basic block to construct the backbone for set-based gait recognition, which can effectively coordinate the silhouette-level and set-level information to learn more discriminative features from the silhouettes. (2) We propose an efficient Dual Feature Pyramid to exploit the shallow-layer features to learn more robust part representations for gait recognition. (3) Extensive experiments on the popular CASIA-B [3] and OUMVLP [4] show that SRN can bring consistent improvement over the baselines for all walking conditions. Under the most challenging condition of walking in different clothes on CAISA-B, SRN exceeds the baseline [15] by 3.1%.

II. Related Work

This work is built on the insights of multiple earlier works, not only for gait recognition but also for other visual tasks. In this section, we summarize the most related ones to our work which are comprised of the following three parts.

A. Gait Recognition

A majority of works for gait recognition take the silhouettes of gait sequences as input to suit for the low-resolution conditions and the cases of changing clothes. As aforementioned, these works can be roughly divided into three categories which regard the silhouettes of a gait sequence as a template [5], [6], [7], [8], [9], a video [11], [12], [13] or an unordered set [14], [15], [16]. Here we present some representative methods for each category respectively. (a) *Template-based.* A comprehensive study on the GEI-based gait recognition with the vanilla CNNs is conducted in [7]. A quintuplet loss based on the GEI pairs is proposed to integrate the cross-gait and unique-gait supervision in [9]. And a Conditional GAN [24] is introduced in [8] to learn gait features from the GEIs that are invariant to camera viewpoints. (b) *Video-based.* A basic 3D-CNN is leveraged to learn the spatial-temporal features from the silhouettes in [11]. And MT3D [13] applies 3D convolution in both small and large temporal scales to extract the spatial-temporal information. (c) *Set-based.* GaitSet [14] first proposes to treat the silhouettes of each gait sequence as an unordered set and extract part-level representations for gait recognition. GaitPart [15] improves GaitSet using a Focal Convolutional Layer and Micro-Motion Capture Module to enhance the part representations. And GLN [16] takes the lateral connections to merge multi-layer features for gait recognition and proposes a Compact Block to significantly reduce the representation dimension.

Besides, there are some works taking other types of input for gait recognition such as GaitNet [25] (RGB frames), GaitMotion [26] (optical flow), SM-Prod [27] (gray images and optical flow). And the previous works [28], [29], [30] also explore to explicitly model the human body parameters from the surveillance videos for gait recognition. For example, the recent End2EndGait [30] first extracts pose and shape features by fitting SMPL [31] and subsequently feeds the pose and shape features to a recognition network. The method in [30] reports promising results among the model-based methods, which yet relies on the pretraining of HMR network [32] on the six datasets with different properties. Besides, the model-based methods [28], [29], [30] are difficult to adapt to the low-resolution conditions where it is hard to estimate the body parameters accurately.

B. Residual Learning

The residual learning is first proposed in [18] and widely used to build the deep networks for many visual tasks [20], [23], [33]. However, the backbone for gait recognition is relatively shallow [7], [9], [34] and the regular residual block is usually not involved. In this work, we propose SRBlock as a basic block to construct the backbone for set-based gait recognition which differs from the regular residual block in three aspects. First, the regular residual block is motivated to alleviate the gradient vanishing/exploding in the deep networks. While in SRBlock, we propose to adopt the residual connection to effectively coordinate the silhouette-level and set-level information for set-based feature learning from the silhouettes. Second, Conv+BN in the residual connection of the regular residual block is optional while the two branches in SRBlock are both essential which extract the silhouette-level and set-level features in a parallel way. Finally, Leaky ReLU [35] is chosen as the activation function in SRBlock instead of ReLU [36].

C. Feature Pyramid

The features extracted by different layers of deep CNNs have been explored in many visual tasks, which, however, are not fully exploited for gait recognition. For example, SSD [37] predicts the objects using the features from different layers separately without fusing features or scores. FCN [19] aggregates the multi-layer features by upsampling to progressively refine the predictions for semantic segmentation. FPN [20] takes the lateral connections to merge the multi-layer features in a top-down manner for object detection. Besides, slicing the features horizontally to learn part representations is widely used for person re-identification [21], [22], [23], which, however, is only applied to the features output by the last layer. While in SRN, we exploit the silhouette-level and set-level features extracted by the shallow layers to learn more robust part representations for gait recognition in an efficient way, where the upsampling or lateral connections are unnecessary and only marginal memory cost is required.

Besides, the concept to exploit the low-level features for gait recognition is mentioned in the early GEI-based works [6], [7] where the low-level features from different gait sequences are fused to perform the contrastive learning [6] or binary classification [7] (the same subject or not). Differently, in SRN, the features from different layers are extracted from the same gait sequence and we efficiently integrate the multi-layer features to enhance the discriminability of the final representation.

III. Our Approach

In this work, we deal with the silhouette-based gait recognition and propose a Set Residual Network (SRN) with the aim of effectively coordinating the silhouette-level and set-level information to learn more discriminative features from the silhouettes. The composition of Set Residual Block (SRBlock) is shown in Fig. 2 where the regular residual block is also displayed for comparison. And the overall network structure is illustrated in Fig. 3 which is simple but nontrivial. Specifically, in SRN, the input silhouettes of each gait sequence are regarded as an unordered set. SRBlock is taken as a basic block to construct the backbone, and an efficient strategy named Dual Feature Pyramid is proposed to exploit the silhouette-level and set-level features extracted by the shallow layers to learn more robust part representations. In what follows, we will first introduce the composition of SRBlock and how it constructs the backbone, and then we will elaborate the strategy to exploit the shallow-layer features for each part in an efficient way.

A. Set Residual Learning

The structure of SRBlock is illustrated in Fig. 2(b) consisting of two parallel branches which are denoted as silhouette-branch and set-branch respectively. Specifically, the silhouette-branch learns the features from each silhouette separately while the set-branch regards all silhouettes as a whole. Set Function is taken to aggregate the features in a silhouette set which transforms the silhouette-level features into the set-level features. In Fig. 2(b), the second Set Function aggregates the features from silhouette-branch as *residual* signals which are added to those from set-branch before the last Leaky ReLU. Different from the regular residual connection in Fig. 2(a) to alleviate the gradient vanishing/exploding, the residual connection in Fig. 2(b) is introduced to connect the silhouette-level and set-level features inside each block which enables more silhouette-set interaction. It is the key of SRBlock and effectively coordinates the silhouette-level and set-level information for set-based feature learning from the silhouettes.

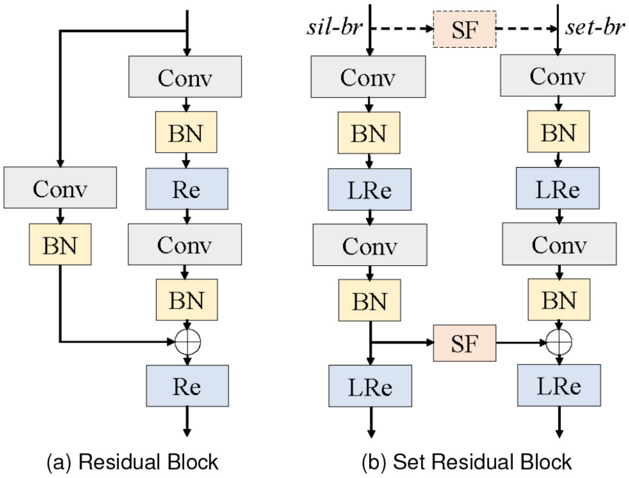


Fig. 2. Illustration of Residual Block and Set Residual Block, Re for ReLU, LRe for Leaky ReLU, SF for Set Function, sil-br for silhouette-branch, set-br for set-branch. In Set Residual Block, Set Function is taken to aggregate the features in a silhouette set and the residual connection is adopted to effectively coordinate the silhouette-level and set-level information for set-based feature learning from the silhouettes.

Formally, let X_{sil} and X_{set} denote the input for silhouette-branch and set-branch respectively, Y_{sil} and Y_{set} denote the corresponding output,[2]

$$Y_{sil} = \Theta_{sil}(X_{sil}) \tag{1}$$

$$Y_{set} = \Theta_{set}(X_{set}) + \phi(Y_{sil}) \tag{2}$$

$$Y_{sil} = \sigma(Y_{sil}) \tag{3}$$

$$Y_{set} = \sigma(Y_{set}) \tag{4}$$

where Θ_{sil} and Θ_{set} denote the layers in two branches before the last Leaky ReLU, ϕ denotes Set Function, σ denotes the last Leaky ReLU, the reshape operations are omitted for simplicity. Note that, the first SRBlock consists of an additional Set Function to process the input which is annotated using the dashed line in Fig. 2(b), i.e., there is $X_{set} = \phi(X_{sil})$ in the first SRBlock.

As shown in Fig. 1, each silhouette is a binary image and the spatial structure is relatively simple. Therefore, we stack two SRBlocks after several initial layers as the backbone for set-based gait recognition which is illustrated in Fig. 3. Specifically, the input silhouettes are first processed separately in the initial layers. The first SRBlock takes the silhouette-level features output by the initial layers as the input for silhouette-branch and consists of an additional Set Function to generates the input for set-branch. And the subsequent SRBlock takes the silhouette-level and set-level features output by the last SRBlock as the input for the two branches. More Specifically, Max Pooling in the backbone provides basic translation invariance to the features, and reduces the spatial size of the features, since the training of set-based gait recognition consumes a vast amount of GPU memory. For example, in our experiments, the batch size is set to 15360 for the training on OUMVLP [4]. The initial layers consist of two convolutional

[2] X_{sil} and Y_{sil} have the shape of [*batch, set, channel, height, width*] where *set* denotes the number of silhouettes in a set. X_{set} and Y_{set} have the shape of [*batch, channel, height, width*].

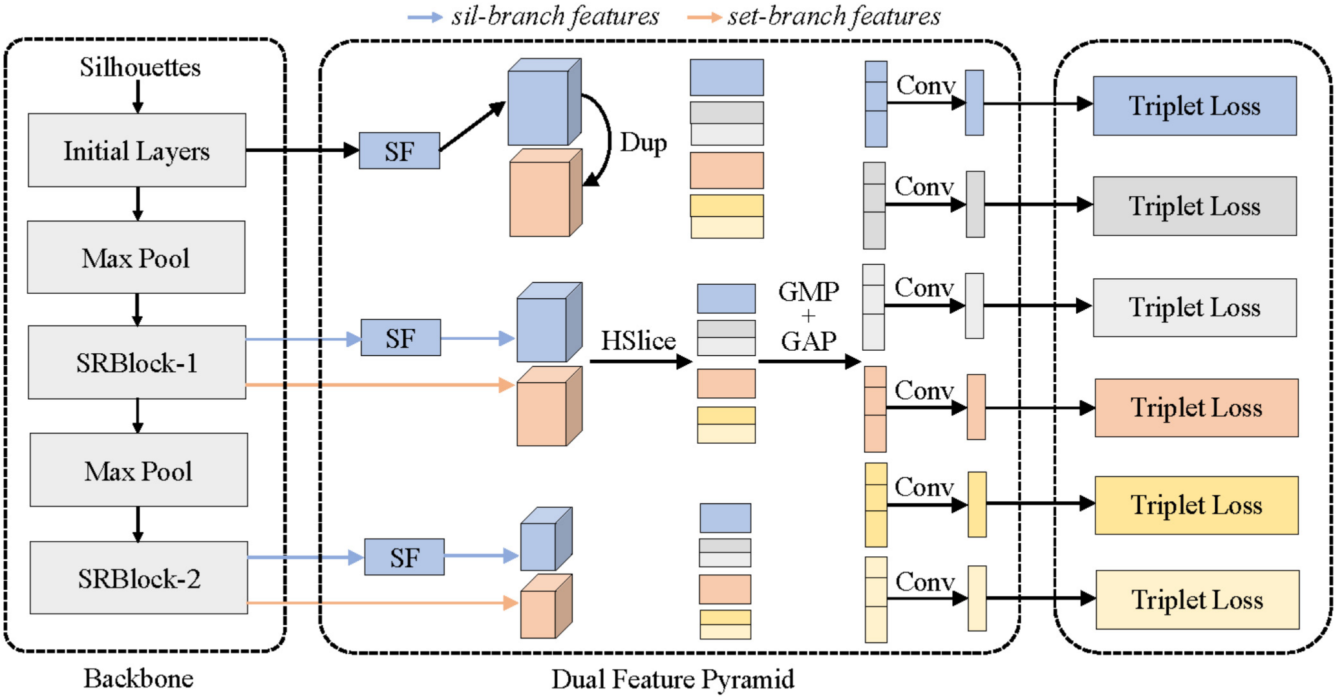


Fig. 3. Illustration of Set Residual Network (SRN), SF for Set Function, Dup for Duplicate, HSlice for Horizontally Slice, GMP for Global Max Pooling, GAP for Global Average Pooling. In SRN, Set Residual Block (SRBlock) is taken as a basic block to construct the backbone for set-based gait recognition and we propose an efficient strategy to exploit the features from the shallow layers to learn more robust part representations. The features from different layers are sliced horizontally using the scales $S = \{1, 2\}$ for simplicity. The triplet loss is added to the features of each part in the training and the features of all parts are taken as a unified representation for evaluation. Best viewed in color.

layers followed by Batch Normalization [38] and Leaky ReLU, which transform each silhouette into the internal features separately. And we adopt Batch Normalization of the synchronized version [39] for our implementation.

As aforementioned, Set Function is taken to aggregate the features in a silhouette set, which should be permutation invariant to the order of silhouettes and is implemented by Max Pooling for simplicity. Different from Max Pooling in the backbone operating along the *spatial* dimensions (including height and width), Max Pooling for Set Function operates along the *set* dimension. A theoretical analysis provided in [17] indicates that Max Pooling is a reasonable choice to aggregate the features in an unordered set, and we also conduct the ablation study to compare with the alternative methods as Set Function, such as Mean Pooling or Max+Mean Pooling. The experimental results will be provided in Section IV-C3.

B. Dual Feature Pyramid

1) Layer Feature Pyramid: The features from different layers of deep CNNs have different receptive fields and capture various visual details of the input [19], [20]. In general, the features from the shallow layers encode the local spatial structural information, while those from the deep layers encode the global context-aware features [40]. The upsampling [19] or lateral connections [20] are usually required to combine the multi-scale features. As shown in Fig. 1, the silhouettes for different subjects only have subtle differences in many cases, which makes it vital to exploit the features from the shallow layers for gait recognition.

2) Horizontal Feature Pyramid: Slicing the features horizontally and equally to learn part representations is widely used in person re-identification [21], [22], which is first adapted to gait recognition by [14]. As far as we can see, Horizontal Pyramid Pooling (HPP) [22] and Horizontal Pyramid Matching (HPM) [14] are equivalent, both of which use the pyramid scales (e.g., $S = \{1, 2, 4, 8, 16\}$) to slice the features horizontally. The difference is that the 1×1 convolutional layers in [22] are replaced by the fully connected layers in [14]. We observe that both HPP and HPM are only deployed at the end of the backbone, which are applied to the features output by the last layer.

3) Dual Feature Pyramid: In SRN, we propose a technique named Dual Feature Pyramid to exploit the shallow-layer features to learn more robust part representations. A natural way is to combine the multi-scale features from different layers through the upsampling [19] or lateral connections [16], [20], which are then sliced horizontally for part representation learning. However, the upsampling is likely to introduce some noise and the lateral connections require additional parameters. Besides, the GPU memory consumption in the training will be further increased.

To address the issue, we propose an *efficient* strategy where the upsampling or lateral connections are unnecessary and only marginal memory cost is required. Specifically, as shown in Fig. 3, instead of directly merging the multi-scale features, we first slice the silhouette-level (processed by Set Function) and set-level features extracted by different layers horizontally and separately. Then we adopt the global max and average pooling to generate the part features at each layer. After that, the part features from different layers *corresponding to the same regions in the input* are concatenated to form more robust part representations. Finally, a 1×1 convolutional layer (or a

TABLE I
THE DATASET STATISTICS, NM FOR NORMAL WALKING, BG FOR WALKING WITH BAGS, CL FOR WALKING IN DIFFERENT CLOTHES

Dataset	Subject		Walking Condition			View
	Train	Test	NM	BG	CL	
CASIA-B [3]	74	50	6	2	2	11
OUMVLP [4]	5153	5154	2	-	-	14

fully-connected layer) is utilized for each part to alleviate the aliasing effect caused by the semantic gaps of different layers. Formally, let F^i denote the features extracted by the i-th layer,

$$P^i_{s,t} = MaxPool\big(F^i_{s,t}\big) + AvgPool\big(F^i_{s,t}\big) \tag{5}$$

$$P_{s,t} = Concat\big(\big\{P^i_{s,t}, i = 1, 2, \ldots, \big\}\big) \tag{6}$$

$$\widehat{P}_{s,t} = Conv\big(P_{s,t}\big) \tag{7}$$

where $F^i_{s,t}$ is obtained by slicing F^i horizontally and equally using the pyramid scales, $s \in S = \{1, 2, 4, 8, 16\}$ here, $t \in \{1, 2, \ldots, s\}$. The global max and average pooling operate along the *spatial* dimensions, and the 1×1 convolutional layers have independent weights for different parts. Compared to $P^i_{s,t}$, $P_{s,t}$ integrates the visual details extracted by the shallow layers and thus can capture the subtle differences of the silhouettes for gait recognition. Besides, $\{P^i_{s,t}, \forall(i, s, t)\}$ is a group of one-dimensional vectors and thus only incurs marginal GPU memory consumption.

C. Summary

In this work, we propose a Set Residual Network (SRN) to learn more discriminative features from the silhouettes for gait recognition. As shown in Fig. 3, Set Residual Block (SRBlock) is proposed as a basic block to construct the backbone which effectively coordinates the silhouette-level and set-level information to facilitate set-based feature learning from the silhouettes. And an efficient strategy is proposed to exploit the features extracted by the shallow layers to learn more robust part representations, where the upsampling or lateral connections are unnecessary and only marginal memory cost is required.

In the training, the loss is added to the features of each part and we adopt the batch all version of triplet loss [41]. Formally,

$$L_{tp} = \frac{1}{N_{tp_+}} \overbrace{\sum_{s\in S}\sum_{t=1}^{s}}^{bins} \overbrace{\sum_{u=1}^{U}\sum_{v=1}^{V}}^{anchor} \overbrace{\sum_{\substack{a=1\\a\neq v}}^{V}}^{pos.} \overbrace{\sum_{\substack{b=1\\b\neq u}}^{U}\sum_{c=1}^{V}}^{negative} [m + dist]_+ \tag{8}$$

$$dist = \mathcal{D}\big(f\big(si^{s,t}_{u,v}\big), f\big(si^{s,t}_{u,a}\big)\big) - \mathcal{D}\Big(f\big(si^{s,t}_{u,v}\big), f\Big(si^{s,t}_{b,c}\Big)\Big) \tag{9}$$

where (U, V) are the number of subjects and the number of sequences for each subject in a mini-batch, N_{tp_+} is the number of triplets resulting in the non-zero loss terms,[3] S is the pyramid scales to slice the features horizontally, m is the margin threshold, si denotes the silhouette set, f denotes the feature extraction, $\mathcal{D}$ measures the similarity between two features, e.g., Euclidean distance. For evaluation, the features of all parts are taken as a unified representation for each silhouette set.

IV. EXPERIMENT

A. Settings

1) Datasets: The experiments are conducted on two popular gait datasets, i.e., CASIA-B [3] and OUMVLP [4]. The dataset statistics are summarized in Table I.

CASIA-B: It is a typical gait dataset consisting of 124 subjects and the videos of normal walking (NM-1,2,3,4,5,6), walking with bags (BG-1,2) and walking in different clothes (CL-1,2) for each subject. The 11 views for each walking condition are uniformly distributed in [0°, 180°]. In our experiments, we take the first 74 subjects as the training set and the rest 50 subjects as the test set. For evaluation, the sequences of NM-1,2,3,4 for each subject are taken as the gallery. The probe contains the sequences of normal walking (NM-5,6), walking in bags (BG-1,2) and walking in different clothes (CL-1,2).

OUMVLP: It is the largest gait dataset in public which consists of 10307 subjects. However, the sequences for each subject only cover normal walking (NM-00,01). The 14 views are uniformly distributed between [0°, 90°] and [180°, 270°]. In our experiments, we take the 5153 subjects as the training set and the rest 5154 subjects as the test set. For evaluation, the sequences of NM-01 for each subject are taken as the gallery and the sequences of NM-00 are taken as the probe.

2) Implementation Details: Our model is implemented with PyTorch library [42] and trained on TITAN-XP GPUs.

Input: The silhouettes are pre-processed using the method in [6]. The input size of each silhouette is set to 128×88 for CASIA-B and 64×44 for OUMVLP. In the training phase, we randomly select 30 silhouettes from each gait sequence as the input. The number of subjects and the sequences for each subject in a mini-batch are set to $(8, 16)$ for CASIA-B and $(32, 16)$ for OUMVLP. In the test phase, all silhouettes of each gait sequence are taken to obtain the representations. Each probe sequence is assigned the label of gallery sequence with the highest similarity measured by Euclidean distance averaged on all parts.

Network Structure: The convolutional channels in the initial layers and the two SRBlocks are set to $(32, 64, 128)$ for CASIA-B and $(64, 128, 256)$ for OUMVLP. The output dimension of the 1×1 convolutional layers for each part is set to 256. We adopt the pyramid scales $S = \{1, 2, 4, 8, 16\}$ to slice the features horizontally. The margin threshold m for L_{tp} in Eq. (8) is set to 0.2.

Optimization: SGD with momentum is taken as the optimizer for SRN. The initial learning rate is set to 0.1 and scaled to its 1/10 three times until convergence. The step size to decrease the learning rate is 10000 iterations for CASIA-B and 50000 iterations for OUMVLP. We use the momentum 0.9 and the weight decay 0.0005 for the optimization. Besides, the warmup strategy [43] is adopted at the start of training.

3) Baselines: In order to validate the effectiveness of SRN, we conduct the experiments in comparison to several baselines which are listed as follows:

(a) GEINet [34] is a representative method taking Gait Energy Images (GEIs) as the input. It customizes a

[3] In each batch, there are $UV(UV - V)(V - 1)$ combinations of the triplets and only the non-zero loss terms are averaged.

TABLE II

THE RANK-1 ACCURACY (%) ON CASIA-B FOR DIFFERENT PROBE VIEWS EXCLUDING THE IDENTICAL-VIEW CASES. FOR EVALUATION, THE SEQUENCES OF NM-1,2,3,4 FOR EACH SUBJECT ARE TAKEN AS THE GALLERY AND THE PROBE SEQUENCES OF THREE WALKING CONDITIONS, I.E., NM, BG AND CL, ARE RESPECTIVELY EVALUATED. CBLOCK FOR COMPACT BLOCK [16]

	Method	Probe View											Average
		0°	18°	36°	54°	72°	90°	108°	126°	134°	162°	180°	
NM	GEINet [34]	40.2	38.9	42.9	45.6	51.2	42.0	53.5	57.6	57.8	51.8	47.7	48.1
	CNN-LB [7]	82.6	90.3	96.1	94.3	90.1	87.4	89.9	94.0	94.7	91.3	78.5	89.9
	GaitSet [14]	93.4	98.1	98.5	97.8	92.6	90.9	94.2	97.3	98.4	97.0	89.1	95.2
	GaitPart [15]	94.1	98.6	99.3	98.5	94.0	92.3	95.9	98.4	99.2	97.8	90.4	96.2
	GLN-Backbone [16]	93.0	98.7	98.6	97.8	94.0	94.0	95.0	97.4	98.7	96.1	87.7	95.5
	GLN [16]	93.2	99.3	**99.5**	**98.7**	96.1	95.6	97.2	98.1	99.3	98.6	90.1	96.9
	SRN(**ours**)	**94.7**	**99.4**	99.4	98.4	96.5	94.8	96.0	98.2	99.3	98.4	**92.9**	97.1
	SRN+CBlock(**ours**)	94.4	99.3	99.4	**98.7**	**96.8**	**96.8**	**97.5**	**98.5**	**99.5**	**98.8**	92.3	**97.5**
BG	GEINet [34]	34.2	29.3	31.2	35.2	35.2	27.6	35.9	43.5	45.0	39.0	36.8	35.7
	CNN-LB [7]	64.2	80.6	82.7	76.9	64.8	63.1	68.0	76.9	82.2	75.4	61.3	72.4
	GaitSet [14]	85.9	92.1	93.9	90.4	86.4	78.7	85.0	91.6	93.1	91.0	80.7	88.1
	GaitPart [15]	89.1	94.8	96.7	95.1	88.3	84.9	89.0	93.5	96.1	93.8	85.8	91.6
	GLN-Backbone [16]	89.6	96.5	97.1	94.5	89.5	87.2	90.3	93.8	95.8	92.5	85.0	92.0
	GLN [16]	91.1	**97.7**	97.8	95.2	**92.5**	**91.2**	92.4	96.0	97.5	95.0	**88.1**	94.0
	SRN(**ours**)	**92.0**	97.4	97.6	95.8	91.8	90.4	**93.2**	95.3	**97.6**	95.3	87.8	94.0
	SRN+CBlock(**ours**)	91.5	97.4	**98.4**	**97.1**	92.2	89.7	93.1	**96.2**	97.5	**96.5**	88.0	**94.3**
CL	GEINet [34]	19.9	20.3	22.5	23.5	26.7	21.3	27.4	28.2	24.2	22.5	21.6	23.5
	CNN-LB [7]	37.7	57.2	66.6	61.1	55.2	54.6	55.2	59.1	58.9	48.8	39.4	54.0
	GaitSet [14]	63.7	75.6	80.7	77.5	69.1	67.8	69.7	74.6	76.1	71.1	55.7	71.1
	GaitPart [15]	70.7	85.5	86.9	83.3	77.1	72.5	76.9	82.2	83.8	80.2	66.5	78.7
	GLN-Backbone [16]	72.7	83.5	86.1	83.0	75.3	73.0	73.4	78.6	79.8	78.1	65.9	77.2
	GLN [16]	70.6	82.4	85.2	82.7	79.2	76.4	76.2	78.9	77.9	78.7	64.3	77.5
	SRN(**ours**)	**75.1**	**88.2**	**89.9**	**86.3**	**81.2**	**78.8**	**80.0**	**84.0**	**86.3**	**80.7**	**68.8**	**81.8**
	SRN+CBlock(**ours**)	69.2	82.5	84.0	81.0	78.6	76.3	78.6	82.8	80.5	76.8	64.7	77.7

network for gait recognition and treats each subject as a separate class in the training.

(b) CNN-LB [7] is another GEI-based method. It concatenates two GEIs as the input and performs the binary classification, i.e., the same subject or not.

(c) GaitSet [14] is the first set-based method for gait recognition and splits the features output by the last layer horizontally to learn part representations.

(d) GaitPart [15] is based on GaitSet and propose a Focal Convolutional Layer and Micro-Motion Capture Module to enhance the part representations.

(e) GLN [16] takes the lateral connections to aggregate the multi-layer features for gait representations and proposes a Compact Block to reduce the representation dimension. It is orthogonal to this work which will be separately discussed and compared.

Besides, we provide the performance comparison with more baselines such as J-CNN [44], MT3D [13], GaitNet [25] and End2EndGait [30] in Section IV-D1.

B. Performance Comparison

1) CASIA-B: The performance comparison on CASIA-B is shown in Table II. The probe can be divided into three subsets according to the walking conditions, i.e., NM, BG, CL, which are respectively evaluated. The accuracy for each probe view is averaged on all gallery views excluding the identical-view cases [7].

From the results in Table II, it can be observed that SRN brings consistent improvement over the baselines for all walking conditions. Among the three walking conditions, walking in different clothes (CL) greatly changes the shape of the silhouettes and is the most challenging for gait recognition, which yet occurs frequently in the real-world applications [16]. Before this work, GaitPart reports the competitive accuracy for CL on CASIA-B among the silhouette-based methods, i.e., 78.7%, while SRN achieves state-of-the-art performance for the challenging CL (81.8%) and outperforms GaitPart by a large margin (+3.1%). Furthermore, the performance under the other two conditions (NM-97.1%, BG-94.0%) achieved by SRN also consistently outperform the baselines.

More specifically, as the first set-based method for gait recognition, GaitSet achieves significant improvement compared to the GEI-based methods such as GEINet and CNN-LB. SRN mainly differs from GaitSet in the way to coordinate the silhouette-level and set-level information for set-based feature learning from the silhouettes. And we further propose an efficient strategy in SRN to exploit the shallow-layer features to learn more robust part representations. The performance comparison between GaitSet and SRN validates the effectiveness of our approach. Besides, compared to GaitPart which adds much complexity to GaitSet, SRN is much more computationally efficient. The comparison of time statistics will be provided in Section IV-C4.

2) OUMVLP: Table III shows the performance comparison on OUMVLP where our method also works reasonably well. Given that the gait data for some subjects in OUMVLP is incomplete, we respectively conduct the evaluation *including* and *ignoring* the probe sequences which have no corresponding ones in the gallery. The training and evaluation of CNN-LB is too time-consuming for the large-scale dataset which is thus not listed [7], [16].

It is worth noting that, though OUMVLP contains much more subjects than CASIA-B, the lack of walking with bags (BG) and walking in different clothes (CL) makes this dataset

TABLE III

THE RANK-1 ACCURACY (%) ON OUMVLP FOR DIFFERENT PROBE VIEWS EXCLUDING THE IDENTICAL-VIEW CASES. WE TAKE THE SEQUENCES OF NM-01 FOR EACH SUBJECT AS THE GALLERY, AND WE RESPECTIVELY CONDUCT THE EVALUATION *Including* AND *Ignoring* THE PROBE SEQUENCES WHICH HAVE NO CORRESPONDING ONES IN THE GALLERY. CBLOCK FOR COMPACT BLOCK [16]

Method	Probe View														Average
	0°	15°	30°	45°	60°	75°	90°	180°	195°	210°	225°	240°	255°	270°	
GEINet [34]	23.2	38.1	48.0	51.8	47.5	48.1	43.8	27.2	37.9	46.8	49.9	45.9	45.6	41.0	42.5
GaitSet [14]	79.3	87.6	90.0	90.1	88.0	88.7	87.7	81.8	86.5	89.0	89.2	87.2	87.6	86.2	87.0
GaitPart [15]	82.6	88.9	90.8	91.0	89.8	89.9	89.5	85.2	88.1	90.0	90.2	89.0	89.1	88.2	88.7
GLN-Backbone [16]	81.0	88.5	90.3	90.4	89.0	89.4	88.7	83.4	87.4	89.4	89.5	88.3	88.3	87.2	87.9
GLN [16]	83.8	90.0	91.0	91.2	90.2	90.0	89.4	85.3	89.1	90.5	90.6	89.6	89.3	88.5	89.2
SRN(**ours**)	83.8	89.7	90.9	91.2	89.9	90.2	89.6	85.8	88.8	90.1	90.4	89.0	89.4	88.5	89.1
SRN+CBlock(**ours**)	**85.6**	**90.7**	**91.5**	**91.7**	**90.6**	**90.6**	**90.1**	**86.8**	**90.0**	**90.9**	**91.1**	**89.9**	**90.0**	**89.3**	**89.9**
GEINet [34]	24.9	40.6	51.5	55.1	49.8	51.0	46.4	29.2	40.7	50.5	53.3	48.4	48.6	43.5	45.3
GaitSet [14]	84.5	93.3	96.7	96.6	93.5	95.3	94.2	87.0	92.5	96.0	96.0	93.0	94.3	92.7	93.2
GaitPart [15]	88.0	94.7	97.7	97.6	95.5	96.6	96.2	90.6	94.2	97.2	97.1	95.1	96.0	95.0	95.1
GLN-Backbone [16]	86.3	94.2	97.1	97.0	94.6	96.0	95.2	88.7	93.5	96.5	96.3	94.2	95.2	93.9	94.2
GLN [16]	89.3	95.8	97.9	97.8	96.0	96.7	96.1	90.7	95.3	97.7	97.5	95.7	96.2	95.3	95.6
SRN(**ours**)	89.2	95.5	97.8	97.8	95.6	97.0	96.3	91.2	95.0	97.3	97.3	95.1	96.3	95.3	95.5
SRN+CBlock(**ours**)	**91.2**	**96.5**	**98.3**	**98.4**	**96.3**	**97.3**	**96.8**	**92.3**	**96.3**	**98.1**	**98.1**	**96.0**	**97.0**	**96.2**	**96.4**

less challenging. GaitSet and GaitPart are taken as two important baselines which achieve the rank-1 accuracy of 93.2% and 95.1% respectively. As aforementioned, SRN and GaitSet constitute a more fair comparison and the accuracy improvement (95.6% *vs.* 93.2%) validates the effectiveness of our approach. Although the average accuracy improvement is not that significant compared to GaitPart, SRN can bring consistent improvement for all 14 views. Moreover, SRN can be seamlessly integrated with Compact Block [16] and achieve state-of-the-art performance for OUMVLP (96.4%), which will be elaborated in the next.

3) Comparison With GLN: In Table II and Table III, we also provides the performance comparison with GLN [16] which is orthogonal to this work. Specifically, in the backbone of GLN (denoted as GLN-Backbone), the lateral connections are adopted to aggregate multi-layer features for each part. While in SRN, an efficient strategy is adopted to exploit the shallow-layer features to learn more robust part representations where the upsampling or lateral connections are unnecessary and only marginal memory cost is required. As shown in Table II and Table III, SRN achieves superior performance to GLN-Backbone for all walking conditions.

Besides, SRN can be seamlessly integrated with Compact Block [16] (denoted as SRN+CBlock) where the features of all parts are concatenated as the input for feature reduction.[4] As shown in Table III, SRN+CBlock achieves the rank-1 accuracy of 96.4% on OUMVLP which is state-of-the-art. While on CASIA-B, the performance of SRN+CBlock for normal walking (NM) and walking with bags (BG) are improved compared to SRN while the performance for walking in different clothes (CL) is degraded. A possible reason is that the sequences of walking in different clothes (CL) largely rely on the subtle differences for the recognition and the features encoding the subtle details are likely to be overwhelmed when concatenating all part features for feature reduction. It is worth noting that, with or without Compact Block, the performance of SRN under all walking conditions are superior to GLN on CASIA-B as shown in Table II.

C. Ablation Study

In this section we provide more experimental results to analyze the behaviors of SRN. The experiments are mainly conducted on CASIA-B which covers different walking conditions, i.e., NM, BG, CL. For simplicity, we provide the rank-1 accuracy for each walking condition averaged on all probe and gallery views excluding the identical-view cases.

1) Impact of Dual Feature Pyramid: In SRN, we propose an efficient strategy called Dual Feature Pyramid to exploit the silhouette-level and set-level features from the shallow layers to learn more robust part representations for gait recognition. The resulting part presentations aggregate the visual details extracted by the shallow layers and thus can capture the subtle differences of the silhouettes. Here we provide the experimental results only using the features output by the last layer to obtain the part representations: NM-97.1%, BG-93.5%, CL-80.6%. The performance indicates that the shallow-layer features is mainly beneficial for walking in different clothes (CL-81.8%). It makes sense since walking in different clothes cause lots of shape variance for the silhouettes belonging to the same subject, which is the most challenging and largely relies on the subtle differences for the recognition [16]. Besides, the performance under all walking conditions without exploiting the shallow-layer features are still superior to those of GaitSet and GaitPart, which further validates the effectiveness of Set Residual Learning.

2) Impact of Activation Function: Leaky ReLU [35] is adopted as the activation function in SRBlock and here we conduct the experiment in comparison to the classical ReLU [36] which is adopted in the regular residual block. And we have also tried other types of activation functions in our experiments including Sigmoid [45], PReLU [46] and ELU [47]. The results are shown in Table IV which indicates that Leaky ReLU is a reasonable choice.

3) Impact of Set Function: Set Function is taken to aggregate the features in a silhouette set which should be permutation invariant to the order of silhouettes. Max Pooling is

[4] We set the output dimension to 256 for the experiments with Compact Block.

TABLE IV
THE ABLATION STUDY OF ACTIVATION FUNCTION AND SET FUNCTION. THE RESULTS ARE REPORTED IN THE RANK-1 ACCURACY ON CASIA-B

Activation	Set Function	NM	BG	CL
Leaky ReLU [35]	Max Pool	**97.1**	**94.0**	**81.8**
ReLU [36]	Max Pool	96.8	93.6	81.0
Sigmoid [45]	Max Pool	93.0	86.4	70.3
PReLU [46]	Max Pool	96.7	93.6	79.3
ELU [47]	Max Pool	96.8	93.8	80.1
Leaky ReLU [35]	Mean Pool	96.0	90.2	71.2
Leaky ReLU [35]	Max+Mean Pool	96.1	92.2	76.3

TABLE V
THE TIME STATISTICS OBTAINED ON CASIA-B. IN THE TRAINING PHASE, THE TIME IS AVERAGED ON THE FIRST 100 ITERATIONS. AND IN THE TEST PHASE, THE TIME IS AVERAGED ON ALL SEQUENCES FOR EVALUATION

Method	Train (per iteration)	Test (per sequence)
GaitSet [14]	0.96s	0.021s
GaitPart [15]	1.16s	0.031s
SRN(**ours**)	0.97s	0.022s

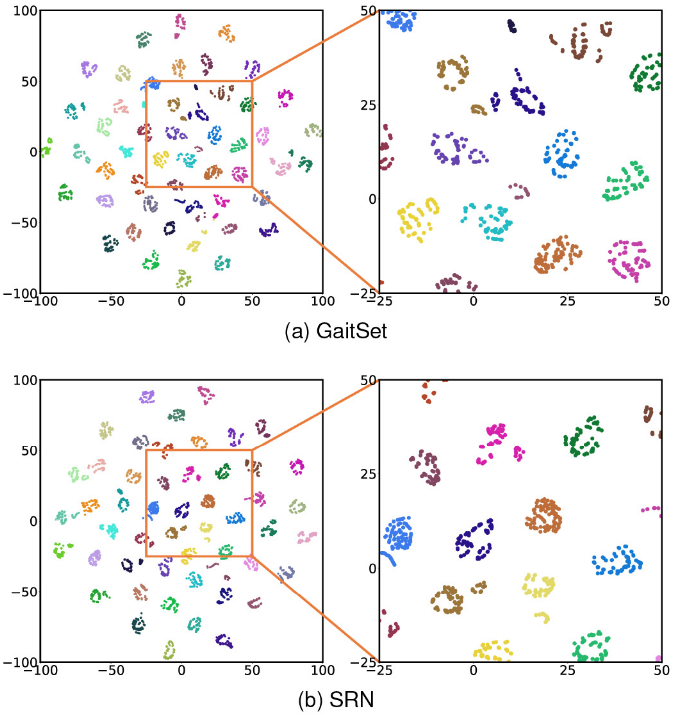
adopted as Set Function in the implementation of SRN for simplicity which operates along the *set* dimension. Here we provide the experiments using two alternative methods as Set Function, i.e., Mean Pooling and Max+Mean Pooling. The results are shown Table IV. More complicated methods for Set Function need further exploration.

4) Time Statistics: In Table V we provide the time statistics of SRN compared to GaitSet [14] and GaitPart [15] in the training and test phase respectively. As aforementioned, SRN and GaitSet constitute a fair comparison. SRN mainly differs from GaitSet in the way to coordinate the silhouette-level and set-level information for set-based feature learning from the silhouettes. In SRN, the silhouette-level and set-level features are extracted in a parallel way and connected inside each block before the last Leaky ReLU. The results in Table V show that the running time of SRN is only increased marginally compared to GaitSet, while SRN can bring significant performance improvement compared to GaitSet as shown in Table II and Table III.

5) Feature Visualization: In Fig. 4, we further provide the visualization comparison of the features extracted by GaitSet and SRN. The experiments are conducted on the test set of CASIA-B consisting of 50 subjects. There are 110 sequences for each subject (11 views and 10 variants of walking conditions) which are all taken for the visualization. From the results in Fig. 4, we observe that the variance of the sequences for each subject is more effectively addressed in SRN, which extracts more discriminative features from the silhouette sets and outperforms GaitSet by a large margin under all walking conditions.

D. Discussion

1) Comparison With More Baselines: In Table VI, we provide the performance comparison with more baselines on CASIA-B and OUMVLP including the methods taking other types of input for gait recognition. In the silhouette-based methods, MT3D [13] utilizes 3D convolution to design the backbone and applies 3D convolution in both the small and large temporal scales to extract the spatial-temporal information, which reports a little inferior results but introduces more computation cost than SRN. And End2EndGait [30] is a recent model-based method and holds state-of-the-art performance among the model-based methods, which yet relies on the pretraining of HMR network [32] on the six datasets with different properties. Besides, we notice that SM-Prod [27] report the highest performance for normal walking (NM) or walking with bags (BG) on CASIA-B. However, the optical flow needs a lot of computation cost and the accuracy for the most challenging condition of walking in different clothes (CL) is much inferior to SRN.



(a) GaitSet

(b) SRN

Fig. 4. Visualization comparison using t-SNE [48] on the test set of CASIA-B. In each row, the first figure shows the visualization results of 50 subjects and the second figure shows the enlarged partial area. Best viewed in color.

TABLE VI
THE PERFORMANCE COMPARISON WITH MORE BASELINES. THE RESULTS ARE REPORTED IN THE RANK-1 ACCURACY. OF FOR OPTICAL FLOW, CBLOCK FOR COMPACT BLOCK [16]

Dataset	Method	Input	NM	BG	CL
CASIA-B	J-CNN [44]	Silhouettes	91.2	75.0	54.0
	GaitSet-L [16]	Silhouettes	95.6	91.5	75.3
	MT3D [13]	Silhouettes	96.7	93.0	81.5
	GaitNet [25]	RGB Frames	92.3	88.9	62.3
	End2EndGait [30]	RGB Frames	97.9	93.1	77.6
	GaitMotion [26]	OF	97.5	83.6	48.8
	SM-Prod [27]	Gray+OF	**99.8**	**96.1**	67.0
	SRN(**ours**)	Silhouettes	97.1	94.0	**81.8**
	SRN+CBlock(**ours**)	Silhouettes	97.5	94.3	77.7
OUMVLP	End2EndGait [30]	RGB Frames	95.8	-	-
	SRN(**ours**)	Silhouettes	95.5	-	-
	SRN+CBlock(**ours**)	Silhouettes	**96.4**	-	-

2) Analysis on Set Residual Learning: For set-based feature learning, how to effectively coordinate the instance-level and set-level information is a key problem. In this work, we deal with set-based gait recognition and address the issue by resorting to residual learning. In SRN, the silhouette-level and set-level features extracted in two parallel branches are complementary to each other, since the set-branch takes the output of Set Function as the input and Set Function cannot preserve all the important details of each silhouette in a set. The set residual connection is adopted to aggregate the features from silhouette-branch as *residual* signals added to those from set-branch, which can thus form more robust set representations. From another perspective, the residual connection enables more silhouette-set interaction during set-based feature learning from the silhouettes, which is beneficial to learn more discriminative features for gait recognition.

For a comprehensive study, we implement a cascaded benchmark in comparison to SRN, i.e., the gait features are first extracted from each silhouette in a sequence separately without the silhouette-set interaction and then the features for the sequence are obtained at the end of the network. Specifically, we remove the residual connection and the set-branch in each SRBlock of SRN, and the other settings remain unchanged to enable a fair comparison. The benchmark finally achieves NM-96.6%/BG-93.3%/CL-79.4% on CASIA-B. The comparison with SRN (NM-97.1%/BG-94.1%/CL-81.8%) further validates the effectiveness of our approach.

V. Conclusion

In this work, we propose a Set Residual Network (SRN) for silhouette-based gait recognition. Set Residual Block (SRBlock) is adopted as a basic block to construct the backbone which can effectively coordinate the silhouette-level and set-level information for set-based feature learning from the silhouettes. And we propose an efficient strategy to exploit the silhouette-level and set-level features extracted by the shallow layers to learn more robust part representations for gait recognition, where the upsampling or lateral connections are unnecessary and only marginal memory cost is required. The experiments on CASIA-B and OUMVLP demonstrate that SRN can bring consistent improvement under all walking conditions.

Acknowledgment

The authors would like to thank Prof. Dongbin Zhao for his support to this work.

References

[1] I. Bouchrika, M. Goffredo, J. Carter, and M. Nixon, “On using gait in forensic biometrics,” *J. Forensic Sci.*, vol. 56, no. 4, pp. 882–889, 2011.

[2] P. K. Larsen, E. B. Simonsen, and N. Lynnerup, “Gait analysis in forensic medicine,” *J. Forensic Sci.*, vol. 53, no. 5, pp. 1149–1153, 2008.

[3] S. Yu, D. Tan, and T. Tan, “A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition,” in *Proc. Int. Conf. Pattern Recognit.*, vol. 4, 2006, pp. 441–444.

[4] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, “Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition,” *IPSJ Trans. Comput. Vis. Appl.*, vol. 10, no. 1, p. 4, 2018.

[5] Y. He, J. Zhang, H. Shan, and L. Wang, “Multi-task GANs for view-specific feature learning in gait recognition,” *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 1, pp. 102–113, Jan. 2019.

[6] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, “On input/output architectures for convolutional neural network-based cross-view gait recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2708–2719, Sep. 2019.

[7] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, “A comprehensive study on cross-view gait based human identification with deep CNNs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 209–226, Feb. 2017.

[8] B. Hu, Y. Gao, Y. Guan, Y. Long, N. Lane, and T. Ploetz, “Robust cross-view gait identification with evidence: A discriminant gait GAN (DIGGAN) approach on 10000 people,” 2018. [Online]. Available: arXiv:1811.10493.

[9] K. Zhang, W. Luo, L. Ma, W. Liu, and H. Li, “Learning joint gait representation via quintuplet loss minimization,” in *Proc. CVPR*, 2019, pp. 4700–4709.

[10] J. Han and B. Bhanu, “Individual recognition using gait energy image,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.

[11] W. Liu, C. Zhang, H. Ma, and S. Li, “Learning efficient spatial–temporal gait features with deep learning for human identification,” *Neuroinformatics*, vol. 16, nos. 3–4, pp. 457–471, 2018.

[12] S. Tong, Y. Fu, X. Yue, and H. Ling, “Multi-view gait recognition based on a spatial–temporal deep neural network,” *IEEE Access*, vol. 6, pp. 57583–57596, 2018.

[13] B. Lin, S. Zhang, and F. Bao, “Gait recognition with multiple-temporal-scale 3D convolutional neural network,” in *Proc. ACM Multimedia*, 2020, pp. 3054–3062.

[14] H. Chao, Y. He, J. Zhang, and J. Feng, “GaitSet: Regarding gait as a set for cross-view gait recognition,” in *Proc. AAAI*, vol. 33, 2019, pp. 8126–8133.

[15] C. Fan *et al.*, “GaitPart: Temporal part-based model for gait recognition,” in *Proc. CVPR*, 2020, pp. 14225–14233.

[16] S. Hou, C. Cao, X. Liu, and Y. Huang, “Gait lateral network: Learning discriminative and compact representations for gait recognition,” in *Proc. ECCV*, 2020, pp. 382–398.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. CVPR*, 2017, pp. 652–660.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016, pp. 770–778.

[19] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. CVPR*, 2015, pp. 3431–3440.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. CVPR*, 2017, pp. 2117–2125.

[21] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, “Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline),” in *Proc. ECCV*, 2018, pp. 480–496.

[22] Y. Fu *et al.*, “Horizontal pyramid matching for person re-identification,” in *Proc. AAAI*, vol. 33, 2019, pp. 8295–8302.

[23] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou, “Learning discriminative features with multiple granularities for person re-identification,” in *Proc. ACM MM*, 2018, pp. 274–282.

[24] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014. [Online]. Available: arxiv.abs/1411.1784.

[25] Z. Zhang *et al.*, “Gait recognition via disentangled representation learning,” in *Proc. CVPR*, 2019, pp. 4710–4719.

[26] K. Bashir, T. Xiang, S. Gong, and Q. Mary, “Gait representation using flow fields,” in *Proc. BMVC*, 2009, pp. 1–11.

[27] F. M. Castro, M. J. Marín-Jiménez, N. Guil, and N. P. de la Blanca, “Multimodal feature fusion for CNN-based gait recognition: An empirical comparison,” *Neural Comput. Appl.*, vol. 32, no. 17, pp. 14173–14193, 2020.

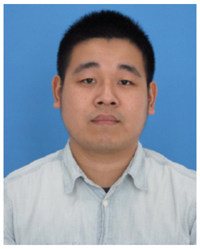
[28] G. Ariyanto and M. S. Nixon, “Model-based 3D gait biometrics,” in *Proc. Int. Joint Conf. Biometrics*, 2011, pp. 1–7.

[29] R. Liao, S. Yu, W. An, and Y. Huang, “A model-based gait recognition method with body pose and human prior knowledge,” *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107069.

[30] X. Li, Y. Makihara, C. Xu, Y. Yagi, S. Yu, and M. Ren, “End-to-end model-based gait recognition,” in *Proc. ACCV*, 2020, pp. 3–20.

[31] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, 2015.
[32] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. CVPR*, 2018, pp. 7122–7131.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, 2016, pp. 630–645.
[34] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "GEINet: View-invariant gait recognition using a convolutional neural network," in *Proc. Int. Conf. Biometrics*, 2016, pp. 1–8.
[35] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, pp. 1–6.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.
[37] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.
[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, vol. 37, 2015, pp. 448–456.
[39] H. Zhang *et al.*, "Context encoding for semantic segmentation," in *Proc. CVPR*, 2018, pp. 7151–7160.
[40] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. CVPR*, 2019, pp. 3085–3094.
[41] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017. [Online]. Available: arXiv:1703.07737.
[42] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.
[43] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *Proc. CVPR Workshops*, 2019, pp. 1487–1495.
[44] Y. Zhang, Y. Huang, L. Wang, and S. Yu, "A comprehensive study on gait biometrics using a joint CNN-based method," *Pattern Recognit.*, vol. 93, pp. 228–236, Sep. 2019.
[45] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. Int. Workshop Artif. Neural Netw.*, 1995, pp. 195–201.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.
[47] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. ICLR*, 2016, pp. 1–5.
[48] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Saihui Hou received the B.E. and Ph.D. degrees from the University of Science and Technology of China in 2014 and 2019, respectively. He is currently a Postdoctoral Fellow with the Institute of Automation, Chinese Academy of Sciences and a Research Scientist with the Watrix Technology Limited Company Ltd. His research interests include computer vision and machine learning. He focuses on gait recognition which aims to identify different people according to the walking patterns.



Xu Liu received the B.E. and Ph.D. degrees from the University of Science and Technology of China in 2013 and 2018, respectively. He is currently a Research Scientist with Watrix Technology Limited Company Ltd. His research interests include gait recognition, object detection, and image segmentation.

Chunshui Cao received the B.E. and Ph.D. degrees from the University of Science and Technology of China in 2013 and 2018, respectively. During his Ph.D. study, he joined Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. From 2018 to 2020, he worked as a Postdoctoral Fellow with PBC School of Finance, Tsinghua University. He is currently a Research Scientist with Watrix Technology Limited Company Ltd. His research interests include pattern recognition, computer vision, and machine learning.

Yongzhen Huang received the B.E. degree from the Huazhong University of Science and Technology in 2006, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences in 2011. He is currently an Associate Professor with the School of Artificial Intelligence, Beijing Normal University and a Research Scientist with the Watrix Technology Limited Company Ltd. He has published one book and more than 80 papers at international journals and conferences, such as IEEE TPAMI, IJCV, IEEE TIP, IEEE TSMCB, IEEE TMM, IEEE TCSVT, CVPR, ICCV, ECCV, NIPS, and AAAI. His research interests include pattern recognition, computer vision, and machine learning.