3D Local Convolutional Neural Networks for Gait Recognition

Zhen Huang[1, 2*], Dixiu Xue[2], Xu Shen[2], Xinmei Tian[1†],
Houqiang Li[1], Jianqiang Huang[2], Xian-Sheng Hua[2†]
[1]University of Science and Technology of China, [2]Alibaba Group
hz13@mail.ustc.edu.cn, {xinmei,lihq}@ustc.edu.cn,
{dixiu.xdx,shenxu.sx,jianqiang.hjq,xiansheng.hxs}@alibaba-inc.com

Abstract

The goal of gait recognition is to learn the unique spatio-temporal pattern about the human body shape from its temporal changing characteristics. As different body parts behave differently during walking, it is intuitive to model the spatio-temporal patterns of each part separately. However, existing part-based methods equally divide the feature maps of each frame into fixed horizontal stripes to get local parts. It is obvious that these stripe partition-based methods cannot accurately locate the body parts. First, different body parts can appear at the same stripe (e.g., arms and the torso), and one part can appear at different stripes in different frames (e.g., hands). Second, different body parts possess different scales, and even the same part in different frames can appear at different locations and scales. Third, different parts also exhibit distinct movement patterns (e.g., at which frame the movement starts, the position change frequency, how long it lasts). To overcome these issues, we propose novel 3D local operations as a generic family of building blocks for 3D gait recognition backbones. The proposed 3D local operations support the extraction of local 3D volumes of body parts in a sequence with adaptive spatial and temporal scales, locations and lengths. In this way, the spatio-temporal patterns of the body parts are well learned from the 3D local neighborhood in part-specific scales, locations, frequencies and lengths. Experiments demonstrate that our 3D local convolutional neural networks achieve state-of-the-art performance on popular gait datasets. Code is available at: https://github.com/yellowtownhz/3DLocalCNN.

1. Introduction

Gait is one of the most important and effective biometric patterns since it can be authenticated at a distance from a camera without subject's cooperation. Gait recognition has broad usage in crime prevention, forensic identification and social security insurance [2, 14]. In real-world scenarios, beyond the change of body shape caused by walking movement, variations such as bag-carrying, coat-wearing, and camera viewpoints switch, also lead to dramatic changes in body appearance, resulting in significant challenges to gait recognition.

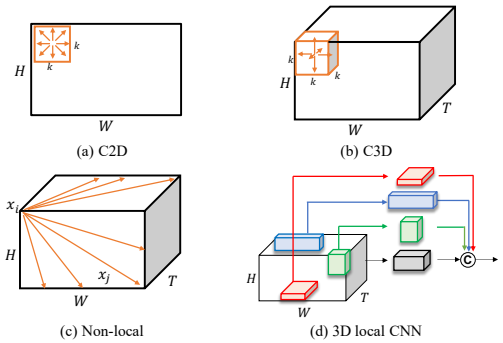


Figure 1. Blocks in backbone CNNs. All these blocks extract features from a local neighborhood. In C2D and C3D, the local neighborhood is a **fixed** 2D patch ($k \times k$) or 3D volume ($k \times k \times k$). Non-local networks learn adaptive long-range dependency with all positions ($H \times W \times T$). Our 3D local CNN is designed to localize **adaptive** 3D volumes, instead of a fixed local neighborhood, for **multiple** local paths and extract corresponding local features.

The essential goal of gait recognition is to learn the unique and invariant representations from the temporal changing characteristics of human body shapes. Early works in gait recognition focused on extracting global features using convolutional neural networks (CNNs) [35, 20, 29, 19]. GaitNet [41, 40] proposed an auto-encoder framework to extract the gait-related features from raw RGB images and then used LSTMs to model the temporal changes of gait sequences. Thomas *et al*. [33] directly applied 3D-CNNs to extract the sequential information using a model pretrained on natural image classification tasks. However, global features do not consider the spatial structure and lo-

*This work was done when the author was visiting Alibaba as a research intern.
†Corresponding author.

cal details of the body shape, thus are not discriminative enough when faced with viewpoint variations. A natural choice is to learn the detailed part-based local features complementary to the global features or learn features embedding for both of them.

Since human body consists of well-defined parts, *i.e.*, head, arms, legs and torso, part-based models have the potential to solve the variations in gait recognition. Previous part-based models extracted part features by equally dividing the feature maps into fixed horizontal stripes. In GaitPart [7], 2D appearance features were firstly extracted by applying pre-defined horizontal partition to the output CNN feature maps of each input frame. Then, the corresponding features of the same stripe from all frames were aggregated by temporal concatenation of local short-range 2D part features. In GaitSet [3] and GLN [11], frame-level feature maps of the last 2D convolutions were firstly split into uniform stripes, then max-pooling along the set dimension was applied to them to extract set-level part features. In MT3D [18], multiple temporal-scale 3D CNNs were used to explore the temporal relations in sequences. Then, the output feature maps were partitioned into multiple stripes too. However, two issues are neglected by these partition-based gait recognition methods. First, different parts of the human body appear at different scales, and even the same part can appear at different locations and scales in different frames [3]. Second, different parts exhibit distinct movement patterns, *e.g.*, at which frame the movement starts, the frequency of position changes, and how long it lasts. Thus, visual appearance and temporal movement changes are mutually dependent in a gait period and the characteristics of different natural human body parts are distinct from each other. It suggests that the gait recognition model should support the extraction and processing of adaptive 3D local volumes for each specific human body part.

To overcome the aforementioned issues in gait recognition, we propose novel 3D local operations as a generic family of building blocks for 3D gait recognition backbones. Our 3D local operations support the extraction of local 3D volumes in a sequence with adaptive spatial and temporal scales, locations and lengths. In this way, the 3D local neighborhoods of different body parts are processed in specific part scales, locations and movement locations, frequencies, lengths, as shown in Fig. 1. 2D local operation is already proved to be valid in image recognition [10, 36], where a differentiable 2D attention mechanism is utilized to yield 2D image/feature patches of smoothly varying locations and scales. However, due to the different mechanism of temporal foveation [21], it is very challenging to adapt this idea to 3D local operations. The reason is in two-fold. 1) Spatial sampling of pixels follows the foveation of the human eye, while temporal sampling of frames is different in following the distribution of optical flow. 2) Spatial sampling processes 2D patches, temporal sampling deals with 1D sequences, and spatio-temporal sampling processes 3D video volumes. Therefore, a new strategy for 2D and 1D joint sampling is required.

Our local operation consists of four modules: localization, sampling, feature extraction, and fusion. The localization module is designed to learn the adaptive spatial and temporal scales, locations and temporal lengths of six body parts: head, torso, left arm, right arm, left leg and right leg. The sampling module samples local volumes of smoothly varying locations, scales and temporal lengths. The feature extraction module consists of several convolution and ReLU [22] layers as in general convolutional blocks. The fusion module is formed as a concatenation layer of global and local outputs followed by a $1 \times 1 \times 1$ convolutional layer. In practice, any building block of existing 3D backbone CNNs can be viewed as a global path, and the proposed local path can be easily inserted into these blocks without any change in the training scheme. Furthermore, the architecture of each component in the local operation is quite flexible for different configurations.

The main contributions of this work are summarized as follows:

- Compared with C3D [30], P3D [24] and Non-local networks [31], we design a new building block for backbone 3D CNNs that incorporates part-specific sequential information, termed 3D local convolutional neural networks.
- We implement a simple but effective form of 3D local CNNs for gait recognition. This model outperforms state-of-the-art gait recognition methods on two of the most popular datasets, CASIA-B and OU-MVLP.
- To the best of our knowledge, we are the first to present a framework that enables the interaction/boosting of global and local 3D volume information in any layer of 3D CNNs.

2. Related Works

Gait Recognition. Many studies on gait recognition have focused on spatial feature extraction and temporal modeling [11, 7, 3, 41, 40, 33, 34, 18, 4]. To obtain spatial representations, most CNN-based studies have employed regular 2D [3, 41] or 3D [33, 34, 18, 35, 20, 29, 19] convolutions operations on entire feature maps along spatial dimensions. While it is natural to equally scan over all the feature maps, these methods ignore the significant differences among human body parts in a gait task. GaitSet[3, 4], GaitPart [7], GLN[11], MT3D[18] all tried to obtain part-level spatial features by equally dividing the output feature maps of backbone into *m* stripes horizontally. However, it is

neither flexible nor fine-grained for the well-defined human body parts.

Furthermore, to obtain spatio-temporal representations of gait sequences, many studies directly compress the whole sequences into one frame [16, 39], or extract frame-level features from each silhouette independently and simply aggregate frame-level features using Max Pooling along the temporal dimensions [3, 11], which ignore the temporal correlations between consecutive frames. Another method explicitly captures the temporal changes using a LSTM to aggregate pose features in time series to generate the final gait feature [41, 25, 17], retaining unnecessary sequential constrains for the periodic gait sequence. All these methods extract spatial features and temporal features separately, neglecting the spatio-temporal dependency of different positions of different frames, which is crucial for recognizing the spatio-temporal movement patterns of human gait.

Local-based model. The local-based model has been exploited in many visual tasks. In fine-grained image classification, many works [37, 26, 5, 42, 8, 32] have automatically located informative regions to capture subtle discriminative details that make the subordinate classes different from each other. Sun *et al*. [26] leveraged multiple channel attentions to learn several relevant regions. Wang *et al*. [32] used a bank of convolutional filters to capture discriminative regions in the feature maps. Zheng *et al*. [42] proposed trilinear attention sampling network to learn features from different details.

In person ReID, Li *et al*. [15] equally divided the output feature maps of the first convolution layer into m local regions horizontally and learned local/global separately. Cheng *et al*. [6] divided the low-level feature map into four equal parts horizontally and concatenated them with a global stream before the last full connection layer. Yang *et al*. [36] proposed a set of operations to locate key positions of human body in a static image. All these previous local-based models are designed to extract patterns of a spatial local region in a static image. For gait recognition, it is natural to extend this insight to the spacetime dimensions of gait sequences, and extract the spatio-temporal movement pattern of a specific human body part within a specific time interval.

Backbone CNNs. Generally used backbone CNNs [13, 30, 24, 31] show that extracting local features from a local neighborhood is helpful to improve vision models. As shown in Fig. 1, C2D [13] and C3D [30] capture short-rage dependencies within the local neighborhood. Their local neighborhoods are a fixed 2D patch ($k \times k$) or 3D volume ($k \times k \times k$). P3D [24] splits $3 \times 3 \times 3$ convolutions into $1 \times 3 \times 3$ convolutional filters on spatial domain and $3 \times 1 \times 1$ convolutions on temporal domain. In Non-local neural networks [31], the non-local operation is designed to capture long-range dependencies between all possible positions in the input feature maps, where the entire input can be regarded as a fixed global neighborhood. Our 3D local CNN is proposed to localize an adaptive 3D local volume, instead of a fixed local neighborhood, for different local paths.

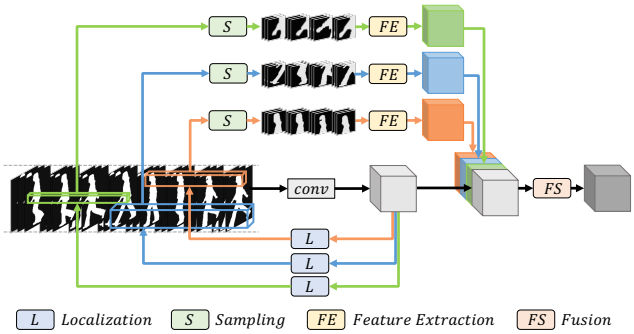


Figure 2. The building block of 3D local CNNs. There are four components: localization module, sampling module, feature extraction module and fusion module. The localization module is designed to locate the positions of each body parts. The sampling module is formulated as specific filters (Gaussian or Trilinear or Mixture) applied to the input. The feature extraction module consists of several convolution and ReLU [22] layers. The fusion module is formed as a concatenation layer of global and local outputs followed by a $1 \times 1 \times 1$ convolutional layer. For simplicity, here we only illustrate three local paths (the head, left-hand and right-leg).

3. Method

In this section, we firstly define a general formulation of 3D local convolution (Sec. 3.1). Then we present an instantiation of 3D convolutional local block (Sec. 3.2), followed by the detailed definitions of corresponding components (Sec. 3.2.1, 3.2.2 and 3.2.3). Finally, the specific 3D Local CNN model for gait recognition is presented (Sec. 3.3).

3.1. Formulation

3D local convolution can be viewed as a special form of the generic convolutional operations in neural networks. Considering a convolutional block with 3D input $\mathbf{x} \in \mathbb{R}^{H \times W \times T}$ and the corresponding output $\mathbf{y}$, the 3D local convolution is defined as:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{j \in \Omega(\mathbf{x}_i)} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j). \tag{1}$$

Here i is the index of an output position (in $\mathbb{R}^{H \times W \times T}$) whose response is to be computed and j is the index of one possible positions in the neighborhood of $\mathbf{x}$, $\Omega(\mathbf{x})$. f computes the correlation coefficient between $\mathbf{x}_i$ and $\mathbf{x}_j$. g computes a representation of the input signal at the position j. The response is normalized by a factor $\mathcal{C}$.

Different forms of convolutional operations in neural networks lie in the definition of the neighborhood $\Omega(\mathbf{x})$. As

shown in Fig. 1, 2D convolution and 3D convolution aggregate features from a fixed local neighborhood patch or volume. In the non-local neural networks [31], the neighborhood is defined as all the feature maps. Different from these operations, 3D local convolution define the neighborhood as a 3D local volume with adaptive spatial and temporal scales, locations and temporal lengths:

$$\Omega(\mathbf{x}) = \{(p, q, r)|\forall(p, q, r) \in \mathcal{V}, \mathcal{V} \subset \mathbf{x}\}, \tag{2}$$

where $\mathcal{V}$ is the adaptive 3D local volume for a specific local part. The essential goal of our 3D local convolutional operation is to sample adaptive 3D volumes $\mathcal{V}$ from a given input $\mathbf{x}$ and extract corresponding local features from these volumes.

3.2. Instantiation

An instantiation of the building block in our 3D Local CNNs is shown in Fig. 2. This block consists of a global path, as in other 3D backbone building blocks, and several local paths. There are four components in our local operations: localization module (**L**), sampling module (**S**), feature extraction module (**FE**) and fusion module (**FS**). The localization module generates the position/scale of the local volume for the corresponding local part based on the global features. Then, the sampling module samples specific local 3D volumes with the given position/scale. The feature extraction module is designed to extract the features from the sampled local volume. The feature fusion module is designed to synthesize the generated global and local features.

3.2.1 Localization

Inspired by the differentiable attention mechanisms used in [36, 10, 9], we specify our localization module by seven independent parameters: $(\Delta x, \Delta y, \Delta t, \delta_x, \delta_y, \delta_t, \sigma^2, \gamma)$. They are dynamically determined for each input. $(\Delta x, \Delta y)$ are the real-valued height and width offsets of the sampling grid center to the predefined center of the corresponding part in each frame, while Δt is the frame offset of the whole sequence. $(\delta_x, \delta_y, \delta_t)$ are the real-valued stride of the sampling grid. σ^2 is the isotropic variance applied to the input if Gaussian filters are applied. The combination of δ and σ^2 controls the "zoom" of the local part. γ acts as a confidence score that multiplies the filter response. Ideally, γ indicates the presence of the focused part, *i.e.*, it should be close to 0 when faced with occlusions.

Given $H \times W \times T$ output feature maps of the global path, we utilize a convolutional block with convolution, ReLU, batch normalization, max pooling and fully connected layers to infer the following parameters:

$$\begin{aligned}(\tanh^{-1}(\Delta x), \tanh^{-1}(\Delta y), \tanh^{-1}(\Delta t),\\ \log \delta_x, \log \delta_y, \log \delta_t, \log \sigma^2, \sigma^{-1}(\gamma)) = \mathbf{L}(\mathbf{G}(\mathcal{I})),\end{aligned} \tag{3}$$

where $\mathbf{G}$ is the global module, $\mathcal{I}$ is the input, $\mathbf{L}$ is the localization module, and $\sigma(\gamma) = \frac{1}{1+exp(-\gamma)}$. $(\Delta x, \Delta y, \Delta t)$ are normalized and scaled to $(-1, 1)$ to ensure the grid center is within the sampling input. The variance and stride are emitted in the log scale to ensure positivity. The confidence score γ is scaled to $(0, 1)$.

The architecture of our localization module is detailed in the Table 1 in the supplementary details. Batch Normalization and ReLU non-linearity are adopted after input and each convolutional layer. Notably, to ensure that the global path focuses on representation only, gradients of this module are not propagated to the global path.

3.2.2 Sampling

To sample local 3D volumes from a given input, we consider an explicitly three-dimensional form of attention. An array of 3D filters is applied to the input sequence, yielding a sequence of local patches with smoothly varying location and zoom. Given the expected local output size $M \times N \times L$, the $M \times N \times L$ grid of sampling filters is applied to the input based on the coordinates of the grid center and the stride distance between adjacent filters. The larger the stride is, the larger the area of the input that will be visible in the attention volume, but the lower the effective resolution of the volume will be. The larger the isotropic variance is, the smoother the output volume is, but the less clear of the details of local volume will be. Based on the normalized prior volume center location (c_x, c_y, c_t), the volume center offset $(\Delta x, \Delta y, \Delta t)$ and stride $(\delta_x, \delta_y, \delta_t)$ provided by the localization module (all of them are real-valued), the grid location (μ_X, μ_Y, μ_T) at row i, column j and frame k in the volume is:

$$\mu_X^i = c_x W + \frac{\Delta x}{2} W + (i - M/2 - 0.5)\delta_x, \tag{4}$$

$$\mu_Y^j = c_y H + \frac{\Delta y}{2} H + (j - N/2 - 0.5)\delta_y, \tag{5}$$

$$\mu_T^k = c_t T + \frac{\Delta t}{2} T + (k - L/2 - 0.5)\delta_t, \tag{6}$$

where $H \times W \times T$ is the size of block input $\mathcal{I}$.

Spatial filtering. Inspired by the techniques for differentiable attention in [36, 10, 9] that mimics the foveation of the human eye [10], we adopt Gaussian filters for spatial filtering. For Gaussian filters, the coordinate of the sampling grid is also the mean location of the filter. Given the isotropic variance σ^2 output by the localization module, the horizontal and vertical filterbank weight matrices $\mathcal{G}_X$ and $\mathcal{G}_Y$ (dimensions $M \times W$ and $N \times H$ respectively) are defined as follows:

$$\mathcal{G}_X[i, p] = \frac{1}{Z_X} exp(-\frac{(\mu_X^i - p)^2}{2\sigma^2}), \tag{7}$$

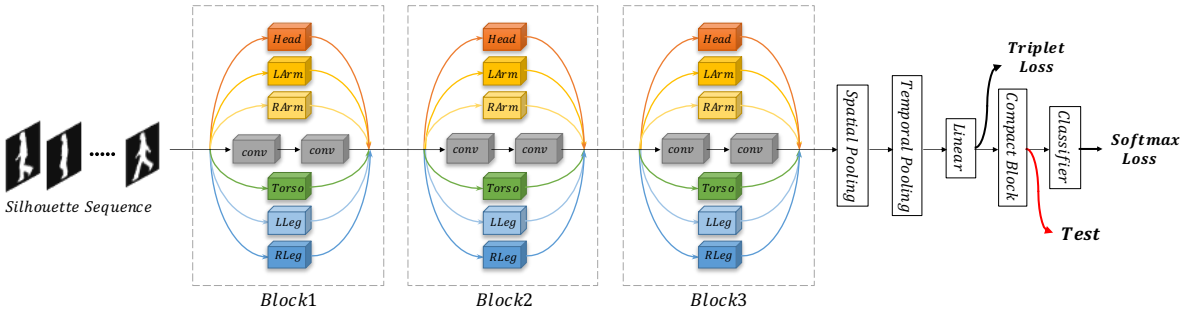


Figure 3. The framework of 3D local CNN for gait recognition (better viewed in color). The backbone path in each of the three blocks is the same as GaitPart. The six colorful paths in each block correspond to the head, left-arm, right-arm, torso, left-leg and right-leg paths.

$$\mathcal{G}_Y[j,q] = \frac{1}{Z_Y} exp(-\frac{(\mu_Y^j - q)^2}{2\sigma^2}), \tag{8}$$

where (i, j) is the spatial index of a point in the attention 3D volume and (p, q) is the spatial index of a point in the input. Z_X and Z_Y are the normalization constants that ensure $\sum_p \mathcal{G}_X[i,p] = 1$ and $\sum_k \mathcal{G}_Y[j,k] = 1$.

Temporal filtering. Inspired by techniques for differentiable motion layers in CNNs [12] and video interpolation [21], a natural choice for the temporal sampling of a volume is trilinear interpolation. Similar to Gaussian filters, here we formulate temporal trilinear interpolation functions as weight matrices $\mathcal{T}_T$ (dimensions $L \times T$). The interpolated value at the target location is computed as a linear combination of values at the ceil and floor integer locations:

$$\mathcal{T}_T[k,r] = \begin{cases} \lceil \mu_T^k \rceil - \mu_T^k, & \text{if } r = \lfloor \mu_T^k \rfloor \\ \mu_T^k - \lfloor \mu_T^k \rfloor, & \text{if } r = \lceil \mu_T^k \rceil \\ 0, & \text{else} \end{cases} \tag{9}$$

where $\lfloor \cdot \rfloor$ is the floor function and $\lceil \cdot \rceil$ is the ceil function. k is the frame index in the attention 3D volume and r is the frame index in the input.

Finally, the overall mixture sampling operation is formulated as three one-dimensional sampling, which combines spatial Gaussian filters that mimics the foveation of the human eye [10] and temporal linear filters assuming that optical flow between consecutive frames is locally linear. Based on $(\mathcal{G}_X, \mathcal{G}_Y, \mathcal{T}_T)$ and the confidence score γ privided by the localization module, the output 3D local volume $\mathcal{V}$ from input $\mathcal{I}$ is sampled as:

$$\mathcal{V} = \gamma \mathcal{G}_X \mathcal{G}_Y \mathcal{T}_T \mathcal{I}. \tag{10}$$

Interestingly, we will show by experiments (Table 4) that our 3D local models are not sensitive to the choice of sampling filters. Only using Gaussian filters or linear filters shows comparable performance with the aforementioned combined filters. This results indicates that the generic local behavior is the main reason for the observed improvements.

3.2.3 Feature Extraction

As illustrated in Fig. 2, the feature extraction module is used to extract the features of the local path. All types of convolutional blocks, such as C3D[30], P3D[24] and MT3D[18], are candidates. The current incarnation of the feature extraction module in this paper is restricted to one convolutional layer of filter size $3 \times 3 \times 3$ followed by ReLU, and this design is made based more on convenience rather than necessity. More sophisticated architecture in feature extraction module may bring in larger performance gains, but that is not the priority of this paper. The number of output feature maps of this module is set to be half of that in the global path. The output and the input of the feature extraction module have the same height, width and length.

3.2.4 Feature Fusion

The feature fusion module is designed to produce more robust and discriminative representations by synthesizing on given global and local outputs. In this paper, feature fusion module is formed as a concatenation layer of global and local outputs along the channel dimension, followed by a $1 \times 1 \times 1$ convolutional layer with ReLU, which refines representations based on the synthesis of both local and global information and ensure that the cardinality remains unchanged. More sophisticated mechanism like attention may bring in more performance gains, but it is not the priority of this paper. The number of output feature maps of this module is set to be the same as the global path.

3.3. 3D Local CNN for Gait Recognition

To insert our 3D local CNN block into backbone CNNs, we need to define the following settings based on prior knowledge: 1) the number of local paths, 2) the prior position of the center of the sampling grid (c_x, c_y, c_t) of each path, and 3) the expected dimension of the local sampling output (M, N, L) for each path.

For feature learning of gait recognition, it is quite natural to define six local paths corresponding to the head, left-arm, right-arm, torso, left-leg and right-leg. (as shown in Fig. 3). Following [1] and common sense knowledge, the general (height, width, length) proportions (p_H, p_W, p_L) of the head, left-arm, right-arm, torso, left-leg right-leg of the human body are summarized in Table 3 in the supplementary

Table 1. Averaged rank-1 accuracy on **CASIA-B**, identical views cases excluded, of GaitSet [3], GaitPart [7], GLN [11] and MT3D [18]. The probe sequences are divided into three subsets (NM, BG and CL) according to the walking conditions. 64×44 and 128×88 denote the input size of each silhouette. Our 3D local CNN shows a significant improvement, especially in the most challenging CL scenario where the temporal changing characteristics dominate the recognition, revealing the superiority of our adaptive local volume sampling and processing mechanism.

Gallery NM #1-4			0° – 180°											Mean
Probe			0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	
NM #5-6	GaitSet	64 × 44	90.8	97.9	99.4	96.9	93.6	91.7	95.0	97.8	98.9	96.8	85.8	95.0
	GaitPart	64 × 44	94.1	98.6	99.3	98.5	94.0	92.3	95.9	98.4	**99.2**	97.8	90.4	96.2
	MT3D	64 × 44	95.7	98.2	99.0	97.5	95.1	93.9	96.1	98.6	**99.2**	98.2	92.0	96.7
	3DLocal	64 × 44	**96.0**	**99.0**	**99.5**	**98.9**	**97.1**	**94.2**	**96.3**	**99.0**	98.8	**98.5**	**95.2**	**97.5**
	GaitSet	128 × 88	91.4	98.5	98.8	97.2	94.8	92.9	95.4	97.9	98.8	96.5	89.1	95.6
	GLN	128 × 88	93.2	99.3	99.5	98.7	96.1	95.6	97.2	98.1	99.3	98.6	90.1	96.9
	3DLocal	128 × 88	**97.8**	**99.4**	**99.7**	**99.3**	**97.5**	**96.0**	**98.3**	**99.1**	**99.9**	**99.2**	**94.6**	**98.3**
BG #1-2	GaitSet	64 × 44	83.8	91.2	91.8	88.8	83.3	81.0	84.1	90.0	92.2	94.4	79.0	87.2
	GaitPart	64 × 44	89.1	94.8	96.7	95.1	88.3	84.9	89.0	93.5	96.1	93.8	85.8	91.5
	MT3D	64 × 44	91.0	95.4	97.5	94.2	92.3	86.9	91.2	95.6	97.3	96.4	86.6	93.0
	3DLocal	64 × 44	**92.9**	**95.9**	**97.8**	**96.2**	**93.0**	**87.8**	**92.7**	**96.3**	**97.9**	**98.0**	**88.5**	**94.3**
	GaitSet	128 × 88	89.0	95.3	95.6	94.0	89.7	86.7	89.7	94.3	95.4	92.7	84.4	91.5
	GLN	128 × 88	91.1	97.7	97.8	95.2	92.5	91.2	92.4	96.0	97.5	95.0	88.1	94.0
	3DLocal	128 × 88	**94.7**	**98.7**	**98.8**	**97.5**	**93.3**	**91.7**	**92.8**	**96.5**	**98.1**	**97.3**	**90.7**	**95.5**
CL #1-2	GaitSet	64 × 44	61.4	75.4	80.7	77.3	72.1	70.1	71.5	73.5	73.5	68.4	50.0	70.4
	GaitPart	64 × 44	70.7	85.5	86.9	83.3	77.1	72.5	76.9	82.2	83.8	80.2	66.5	78.7
	MT3D	64 × 44	76.0	87.6	89.8	85.0	81.2	75.7	81.0	84.5	85.4	82.2	68.1	81.5
	3DLocal	64 × 44	**78.2**	**90.2**	**92.0**	**87.1**	**83.0**	**76.8**	**83.1**	**86.6**	**86.8**	**84.1**	**70.9**	**83.7**
	GaitSet	128 × 88	66.3	79.4	84.5	80.7	74.6	73.2	74.1	80.3	79.7	72.3	62.9	75.3
	GLN	128 × 88	70.6	82.4	85.2	82.7	79.2	76.4	76.2	78.9	77.9	78.7	64.3	77.5
	3DLocal	128 × 88	**78.5**	**88.9**	**91.0**	**89.2**	**83.7**	**80.5**	**83.2**	**84.3**	**87.9**	**87.1**	**74.7**	**84.5**

details. To validate the effectiveness of 3D local CNNs, we insert the proposed local operations after every two layers of the backbone networks. As in [7, 3, 11], the backbone network consists of three building blocks. Each block is composed of two convolutional layers, followed by ReLU layers. We adopt the spatial pooling and temporal pooling from GaitPart [7], the compact block and the linear modules from GLN [11].

4. Experiments

4.1. Datasets and Evaluation Protocols

CASIA-B [38] is the most popular gait datasets for evaluation and contains 124 subjects. There are three walking conditions: normal walking (NM, 6 variants per subject), walking with bags (BG, 2 variants per subject), and walking in different clothes (CL, 2 variants per subject). In each condition, the subjects are sampled in 11 view angles (0°-180° with interval 18°). To evaluate the performance, we use the same protocol as [3, 7]. We take 74 subjects for training and the rest 50 subjects for test. During the test, the first 4 sequences of NM condition (NM #1-4) are regarded as gallery, and the remaining 6 sequences (NM #5-6, BG #1-2 and CL #1-2) are regarded as probe.

Table 2. Averaged rank-1 accuracy on **OU-MVLP**, identical views cases excluded of GaitSet [3], GaitPart [7] and GLN [11]. For evaluation, the first variant of normal walking (NM) for each subject is taken as the gallery with the rest as the probe.

Probe	Gallery all 14 views			
	GaitSet	GaitPart	GLN	3DLocal
0°	79.5	82.6	83.8	**86.1**
15°	87.9	88.9	90.0	**91.2**
30°	89.9	90.8	91.0	**92.6**
45°	90.2	91.0	91.2	**92.9**
60°	88.1	89.7	90.3	**92.2**
75°	88.7	89.9	90.0	**91.3**
90°	87.8	89.5	89.4	**91.1**
180°	81.7	85.2	85.3	**86.9**
195°	86.7	88.1	89.1	**90.8**
210°	89.0	90.0	90.5	**92.2**
225°	89.3	90.1	90.6	**92.3**
240°	87.2	89.0	89.6	**91.3**
255°	87.8	89.1	89.3	**91.1**
270°	86.2	88.2	88.5	**90.2**
Mean	87.1	88.7	89.2	**90.9**

OU-MVLP [28] is the largest public gait dataset. It is composed of 10307 subjects (5153 subjects for training,

5154 subjects for testing). However, only the sequences of normal walking (NM, 2 variants per subject) are available for each subject. Each subject has 14 views, and the 14 views are uniformly distributed between $[0°, 90°]$ and $[180°, 270°]$ at an interval of $15°$. At the test phase, the sequences with index $\#01$ are grouped into the galleries while the sequences with index $\#02$ are grouped into the probes.

4.2. Implementation Details

All models are implemented in PyTorch [23] and are randomly initialized. The sampling module adopts Mixture mechanism. To make sure the spatial centers of adjacent frames are consistent, the prior spatial ceter location parameters (c_x, c_y) of a part are smoothed using the center offset $(\Delta x, \Delta y)$ of previous two frames, $\hat{c}_x^t = c_x^t + (\Delta x^{t-1} + \Delta x^{t-2})/2, \hat{c}_y^t = c_y^t + (\Delta y^{t-1} + \Delta y^{t-2})/2$. The silhouettes are pre-processed using methods proposed in [27]. In a mini-batch, the number of subjects and the number of sequences for each subject are set to $(8, 16)$ for CASIA-B and $(32, 16)$ for OU-MVLP. At the training phase, the sequences are sampled according to [7], with random horizontal flipping. Adam optimizer is used with a learning rate of $1e$-4. In CASIA-B, the model is trained for 120k iterations. In OU-MVLP, the iterations is set to 250k, and the learning rate is reduced to $1e$-5 at 150k iteration. For evaluation, all silhouettes of a gait sequences are taken to obtain the final representation. Since OU-MVLP has 20 times more sequences than CASIA-B, we double the number of channels in the convolutional layers ($C1$=$C2$=64, $C3$=$C4$=128, $C5$=$C6$=256). To make a fair comparison with GLN [11], we adopt the same compact block and the cross-entropy loss. The details can be found in [11]. And the experiments on CASIA-B are conducted on two input sizes respectively (64×44 and 128×88).

4.3. Comparison with State-of-the-Art Methods

CASIA-B. Table 1 demonstrates the superiority of 3D local CNN over all state-of-the-art models. Compared with GaitSet [3] and GLN [11], 3D local CNN clearly presents better performance with both two input sizes. Under the most challenging condition of walking in different clothes (CL), 3D local CNN exceeds GaitSet by 13.0% and GLN by 6.8%. Both GaitSet and GLN consider silhouettes as a set, instead of a sequence. This result reveals the superiority of processing local movement patterns in a sequence rather than in a set without order information.

Compared with GaitPart[7], our method also outperforms it with a great margin. This result shows that our 3D local volume operations with adaptive spatio-temporal locations, scales and lengths capture more effective local part information than horizontally splitting feature maps into stripes as parts.

More importantly, we find that our method surpasses other methods with a great performance margin in CL scenario. In CL scenario, large appearance changes makes temporal changing characteristics (the core concept of gait) dominate the recognition compared with visual appearances. Therefore, our 3D local CNN is much better in learning the core gait representations than SOTA methods.

Table 3. Performance on CASIA-B with 3D local CNN inserted into different blocks of the backbone model.

Setting	Block1	Block2	Block3	NM	BG	CL
a				94.3	90.4	76.7
b	✓			95.0	91.9	77.7
c		✓		95.7	92.6	79.5
d			✓	96.4	93.1	81.6
e		✓	✓	97.1	93.8	82.7
f	✓	✓	✓	**97.5**	**94.3**	**83.7**

Table 4. Performance on CASIA-B with different sample mechanisms in 3D local blocks.

Type	NM	BG	CL
Gaussian	97.4	94.0	83.2
Trilinear	96.2	93.7	82.9
Mixture	**97.5**	**94.3**	**83.7**

OU-MVLP. As shown in Table 2, 3D local CNN achieves the best performance under all cross-view conditions. For some probe sequences, there are no corresponding sequences in the gallery. If the subjects without corresponding samples in probe are discarded, the average rank-1 accuracy of all probe views are 96.5%, while GaitSet is 93.3%, GaitPart is 95.1% and GLN is 95.6%.

4.4. Ablation Experiments

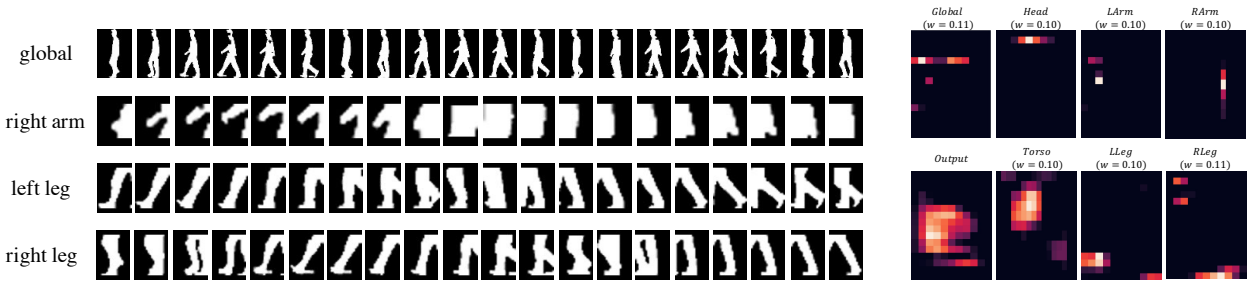
Architecture. We first investigate the performance when the proposed local operations are inserted into different blocks of the backbone model. The results are summarised in Table 3. Inserting our 3D local block into any block of the backbone model can bring significant performance gain. This validates our design of synthesizing global and 3D local information in building blocks is rational. Inserting 3D local blocks into higher layers tends to perform better than inserting them into lower layers. This is because in higher layers the expressive power of learned representations are strong enough to convey the semantic concepts of well-defined human body parts.

Sampling. In Sec. 3.2.2, we describe three variants of the sampling module: Gaussian, Trilinear and Mixture. Table 4 shows the results of applying these three sampling mechanisms respectively. We can see that these three settings have comparative performance, indicating that our 3D local operation is general and our model is not sensitive to specific implementations of the sampling module.

Local Path and Feature Extraction. To validate the

Table 5. Performance on CASIA-B with different settings of local operations. "Head", "LArm", "RArm", "Torso", "LLeg" and "RLeg" denote six different local paths. "FE" denotes the feature extraction module for all local paths. Size: 64×44.

Setting	Global	Head	LArm	RArm	Torso	LLeg	RLeg	FE	NM	BG	CL
a	✓								94.3	90.4	76.7
b	✓	✓							94.7	91.0	77.9
c	✓	✓	✓	✓					95.5	91.7	79.8
d	✓	✓	✓	✓	✓				96.5	92.6	81.2
e	✓	✓	✓	✓	✓	✓	✓		96.9	93.2	81.9
f	✓	✓	✓	✓	✓	✓	✓	✓	**97.5**	**94.3**	**83.7**



(a) Sampled part sequences (b) Extracted features

Figure 4. Visualizations of different local paths. **(a):** The sampled sequences of the right-arm,left-leg and right-leg paths in 3D local CNN block. Different local parts have different spatial positions, scales and temporal lengths. **(b):** Global and local feature maps with the max coefficients, and the output feature map. Different branches are complementary to each other since they focus on different regions. The averaged fusion weights are attached above each feature map. Every path contributes to the output (all coefficients are non-zeros).

effectiveness of our local operations, we present the performance of models with and without local paths or feature extraction module on CASIA-B. In Table 5, (a) means there are no local operations. From (b) to (e), local paths are gradually added. (f) denotes using feature extraction module. We can see that models with local operations consistently outperform models without local operations,indicating that our design of local operations is reasonable. The model with FE achieves better performance than that without FE. Currently, our implementation of FE module is very simple. We believe that more sophisticated architecture of FE will bring in more performance gain.

Sampled Sequences. Except the prior positions of part volume centers, our localization module is learned in a totally unsupervised manner. Fig. 4 (a) shows the results of sampling module of different local paths. Due to the limited space, we choose three most discriminative branches, right-arm, left-leg and right-leg. The left-leg path is able to track the movement of the left leg for a raw gait period. The sampled sequences of left and right legs are from different temporal segments. These show that even with little supervision, 3D local CNN succeeds in learning adaptive spatial positions, scales and temporal lengths for different parts.

Fusion of Global and Local Paths. To inspect how this module synthesizes global and local features, we visualize the input feature maps, output feature maps and the weights of the convolutional layer in the feature fusion module. Fig. 4 (b) shows the output feature maps and the input feature maps, indicating that different branches are complementary to each other since they focus on different regions. Synthesizing features of different branches makes the output features more informative and discriminative. The averaged fusion weights are attached above each corresponding feature map, showing that every path contributes to the output (all coefficients are non-zeros).

5. Conclusion

We present a new building block for 3D CNNs with local information incorporated, termed as 3D local convolutional neural networks. Our local operations can be combined with any existing architectures. We demonstrate the superiority of local operations on the task of gait recognition where 3D local CNN consistently outperforms state-of-the-art models. We hope this work will shed light on more research on introducing simple but effective local operations as submodules of existing convolutional building blocks.

Acknowledgements

This work was supported in part by Alibaba Innovative Research (AIR) program, Major Scientific Research Project of Zhejiang Lab (No. 2019DB0ZX01), NSFC No. 61872329, the National Key R&D Program of China under contract No. 2017YFB1002203 and the Fundamental Research Funds for the Central Universities under contract WK3490000005.

References

[1] Barry Bogin and Maria Inês Varela-Silva. Leg length, body proportion, and health: a review with a note on beauty. *International journal of environmental research and public health*, 7(3):1047–1075, 2010. 5

[2] Imed Bouchrika, Michaela Goffredo, John Carter, and Mark Nixon. On using gait in forensic biometrics. *Journal of forensic sciences*, 56(4):882–889, 2011. 1

[3] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: Regarding gait as a set for cross-view gait recognition. In *AAAI*, volume 33, pages 8126–8133, 2019. 2, 3, 6, 7

[4] Hanqing Chao, Kun Wang, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: Cross-view gait recognition through utilizing gait as a deep set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[5] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *CVPR*, pages 5157–5166, 2019. 3

[6] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, pages 1335–1344, 2016. 3

[7] Chao Fan, Yunjie Peng, Chunshui Cao, Xu Liu, Saihui Hou, Jiannan Chi, Yongzhen Huang, Qing Li, and Zhiqiang He. Gaitpart: Temporal part-based model for gait recognition. In *CVPR*, pages 14225–14233, 2020. 2, 6, 7

[8] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *CVPR*, pages 3034–3043, 2019. 3

[9] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 4

[10] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 2, 4, 5

[11] Saihui Hou, Chunshui Cao, Xu Liu, and Yongzhen Huang. Gait lateral network: Learning discriminative and compact representations for gait recognition. In *ECCV*, pages 382–398. Springer International Publishing, 2020. 2, 3, 6, 7

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015. 5

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 3

[14] Peter K Larsen, Erik B Simonsen, and Niels Lynnerup. Gait analysis in forensic medicine. *Journal of forensic sciences*, 53(5):1149–1153, 2008. 1

[15] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. In *IJCAI*, pages 2194–2200, 2017. 3

[16] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, and Mingwu Ren. Gait recognition via semi-supervised disentangled representation learning to identity and covariate features. In *CVPR*, pages 13309–13319, 2020. 3

[17] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, Shiqi Yu, and Mingwu Ren. End-to-end model-based gait recognition. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3

[18] Beibei Lin, Shunli Zhang, and Feng Bao. Gait recognition with multiple-temporal-scale 3d convolutional neural network. In *ACM MM*, pages 3054–3062, 2020. 2, 5, 6

[19] Beibei Lin, Shunli Zhang, Xin Yu, Zedong Chu, and Haikun Zhang. Learning effective representations from global and local features for cross-view gait recognition. *arXiv preprint arXiv:2011.01461*, 2020. 1, 2

[20] Wu Liu, Cheng Zhang, Huadong Ma, and Shuangqun Li. Learning efficient spatial-temporal gait features with deep learning for human identification. *Neuroinformatics*, 16(3):457–471, 2018. 1, 2

[21] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4463–4471, 2017. 2, 5

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 2, 3

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019. 7

[24] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017. 2, 3, 5

[25] Alireza Sepas-Moghaddam and Ali Etemad. View-invariant gait recognition with attentive recurrent learning of partial representations. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2020. 3

[26] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. In *ECCV*, pages 805–821, 2018. 3

[27] Noriko Takemura, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. On input/output architectures for convolutional neural network-based cross-view gait recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. 7

[28] Noriko Takemura, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition. *IPSJ Transactions on Computer Vision and Applications*, 10(1):4, 2018. 6

[29] Daksh Thapar, Aditya Nigam, Divyansh Aggarwal, and Punjal Agarwal. Vgr-net: A view invariant gait recognition network. In *2018 IEEE 4th international conference on identity, security, and behavior analysis (ISBA)*, pages 1–8. IEEE, 2018. 1, 2

[30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 2, 3, 5

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2, 3, 4

[32] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *CVPR*, pages 4148–4157, 2018. 3

[33] Thomas Wolf, Mohammadreza Babaee, and Gerhard Rigoll. Multi-view gait recognition using 3d convolutional neural networks. In *ICIP*, pages 4165–4169. IEEE, 2016. 1, 2

[34] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep cnns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(2):209–226, 2016. 2

[35] Weiwei Xing, Ying Li, and Shunli Zhang. View-invariant gait recognition method by three-dimensional convolutional neural network. *Journal of Electronic Imaging*, 27(1):013010, 2018. 1, 2

[36] Jiwei Yang, Xu Shen, Xinmei Tian, Houqiang Li, Jianqiang Huang, and Xian-Sheng Hua. Local convolutional neural networks for person re-identification. In *ACM MM*, pages 1074–1082, 2018. 2, 3, 4

[37] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *ECCV*, pages 420–435, 2018. 3

[38] Shiqi Yu, Daoliang Tan, and Tieniu Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *ICPR*, volume 4, pages 441–444. IEEE, 2006. 6

[39] Kaihao Zhang, Wenhan Luo, Lin Ma, Wei Liu, and Hongdong Li. Learning joint gait representation via quintuplet loss minimization. In *CVPR*, pages 4700–4709, 2019. 3

[40] Ziyuan Zhang, Luan Tran, Feng Liu, and Xiaoming Liu. On learning disentangled representations for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[41] Ziyuan Zhang, Luan Tran, Xi Yin, Yousef Atoum, Xiaoming Liu, Jian Wan, and Nanxin Wang. Gait recognition via disentangled representation learning. In *CVPR*, pages 4710–4719, 2019. 1, 2, 3

[42] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *CVPR*, pages 5012–5021, 2019. 3