Knowledge-Based Systems 206 (2020) 106273

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

SpiderNet: A spiderweb graph neural network for multi-view gait recognition

Aite Zhao *, Jianbo Li, Manzoor Ahmed

College of Computer Science and Technology, Qingdao University, Qingdao, China

ARTICLE INFO

Article history:
Received 9 March 2020
Received in revised form 13 July 2020
Accepted 15 July 2020
Available online 27 July 2020

Keywords:
Multi-view
Gait knowledge
Spiderweb graph convolutional network
Feature fusion
Capsule network
Long-short term memory
Spatio-temporal information

ABSTRACT

Human gait is a proven biometric trait with applications in security for authentication and disease diagnosis. However, it is one-sided to express and interpret gait data from a single point of view, which cannot reflect multi-dimensional characteristics of gait changes. Moreover, if the gait pattern observed from other views has pathological or abnormal behavior, or has micro movement, it is not easy to be detected and thus affects the recognition rate of gait. In addition, the multi-view fusion of gait knowledge can be challenging due to the close correlation between various visual angles. Owing to the above facts, we propose a spiderweb graph neural network (SpiderNet) to solve the multi-view gait recognition problem, which connects the gait data of single view with that of other views concurrently and constructs an active graph convolutional neural network. The gait trajectory of each view is analyzed by the combination of a memory module and a capsule module, which accomplishes the multi-view feature fusion, as well as the spatio-temporal feature extraction of single view. The experimental results show that the SpiderNet is superior to fifteen state-of-the-art methods, such as random forest, long-short term memory and convolutional neural network, and achieves 98.54%, 98.77%, and 96.91% of the results on three challenging gait datasets: SDUgait, CASIA-B, and OU-MVLP.

1. Introduction

1.1. Background

Data's diverse development has an positive influence on computer vision and intelligent algorithm. Benefit by the full understanding of multi-modal data, some gait recognition studies can achieve high accuracy and are also suitable for the changeable and complex environment. However, different dimensions of single-modal data cannot be deeply considered due to the acquisition tools' limitation. In this situation, multiple cameras can provide information of the gait and scene at diverse observation points, which help in solving the problem of multi-view feature fusion.

Gait recognition is one of the active fields in pattern recognition, such as face recognition and emotion recognition [1], and has been explored extensively. The single-view recognition of human gait mainly depends on the way and data collection's type, because the input of single view can only represent the gait representation of one direction, and the data of other directions cannot be observed and understood. In this way, it can only be explored from the multi-modal identification of data. At present, some visual sensors, such as depth camera, infrared camera, and binocular camera [2–5], acquire different types of gait images, which are further analyzed and recognized using fusion algorithms. But it can only express features from a single angle, ignoring the gait view observed from other angles.

The key of multi-view gait recognition is to build a model that can capture and combine the static structure and dynamic changes of gait in each view. There are some studies to achieve gait recognition by generating 3D gait structure or developing multi-view fusion approaches [6,7]. The establishment of gait 3D model is the most straightforward multi-view fusion method, the authors in [6] put forward, a 3D object to share common view surfaces in significantly different views. In addition, canonical correlation analysis (CCA) is a well-known multivariate analysis and a data fusion method for quantifying the correlations between two groups of multi-dimensional variables. In the study of [7], CCA is improved to achieve multi-view gait recognition by using typical gait image data. Although there were some works focusing on the multi-view fusion task towards this direction [8–10], analyzing temporal gait sequences in each view and spatial feature expression of one frame simultaneously is an under considered difficulty. Moreover, the fusion process is also relatively rigid and has no dynamic close relationships between various views.

* Corresponding author.
E-mail addresses: zhaoaite@gmail.com (A. Zhao), lijianbo@188.com (J. Li), manzoor.achakzai@gmail.com (M. Ahmed).

https://doi.org/10.1016/j.knosys.2020.106273
0950-7051/

Feature extraction and pre-process methods of gait inputs also achieve impressive performance. The gait inputs can be preprocessed into various forms, such as gait energy images (GEI), 3D skeleton joint coordinates, optical flow maps, or gait silhouettes. After the pre-process of the original data, a majority of hand-crafted feature definition and extraction methods, and model-based feature extraction methods are proposed for the analysis of secondary gait data [11–14]. Among these studies, the ability of convolutional neural network (CNN) to learn the spatial feature is superior in comparison with other methods in a wide range of fields [15–17]. While the long-short term memory (LSTM) is also representative for temporal feature computation and contributes to model the gait dynamics. The major limitation of these methods applied in gait recognition is not only the absence of information about the structure of human body parts (i.e., arms, legs, trunk, etc.), but also the relationship between them.

1.2. Motivations and contributions

Our goal in this paper is the analysis of the multi-view inputs for identifying human gaits. Having information of multiple visual angles, our proposed spiderweb graph neural network (SpiderNet) can link the observed data from each visual angle, and make unified training, which can be more comprehensive than any of the information from single view. Our proposed approach introduces memory and capsule modules to extract spatio-temporal features, gait context, and overall structure information. Simultaneously, we construct an adjacency matrix for the data corresponding to each view, and leverage all the views' data for training the SpiderNet to achieve the effect of dynamic recognition. Moreover, we use a joint loss function to train internal network's parameters, which is beneficial for the components' close relationship.

Our motivation for selecting and building these models is as follows:

(1) For the memory module, LSTM is applied for distilling contextual information of gait input frames with novel expandable memory units. Heretofore, LSTM has been widely utilized in gait recognition, activity recognition, and other recognition fields [18,19]. It is a gated neural network specially designed for time series data, which is continuously collected and well fits sequences from distinct views. Moreover, LSTM has the advantage of linking the observation with its context information and explicitly describes the features at the adjacent time and reveals the internal relationship of the entire sequence.
(2) Apart from temporal features, we also consider the spatial and structural relationship between parts of the human body, such as the relationship between feet and legs, head and body. Taking into account the information loss of the max-pooling layer in CNN, capsule neural network (CapsNet) [20] do not ignore information about the precise position of body parts within human body. Body parts are "place-coded" by active capsule. By using dynamic routing, each active capsule will choose a capsule in the layer above to be its parent in a tree. For gait recognition, this iterative process enables to solve the problem of locating body parts to the wholes in one image.

The main contributions of this paper are summarized as follows:

- We study the problem of multi-view gait recognition in continuous image streams by modeling each view and fusing multiple views, which makes up for the limitations of incomplete gait information in single view.
- The proposed spiderweb graph convolutional neural network (SpiderNet) connects all the view frames with angular correlation to avoid losing the associated features, and then trains the parameters until the best classification result is obtained.
- The SpiderNet is better in describing the spatio-temporal and structural knowledge of gait data via the communication between the novel memory unit in memory module and the capsule module. SpiderNet can integrate the high and low frequency features in gait data by utilizing the octave convolution in the capsule module.
- The proposed method achieves superior performance on three challenging datasets for multi-view gait recognition. It has portability for other multi-view gait datasets with image or signal data.

The control provided by our approach is both simple, intuitive, and computationally efficient. After introducing the related work in Section 2, we present the proposed SpiderNet in Section 3, and experiments that include the comparison of various similarity criteria, several advanced approaches on three multi-view gait datasets are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Related work

Feature extraction in different views plays a vital role in gait recognition and its precision directly affects the accuracy of gait recognition. This section introduces the related work from single-view and multi-view gait recognition.

2.1. Single-view gait recognition

Single-view gait recognition schemes discover the underlying mathematical construct characteristics of gait sequences by analyzing data collected in one view. The gait data collected from one viewpoint has many types. For example, conventional or non-machine learning (non-ML) methods include distance and correlation calculation, as well as some machine learning (ML) methods are random forest (RF) [21], hidden Markov model (HMM) [22]. Moreover, some deep learning models such as LSTM [23,24] and CNN [25,26], have been used for gait feature extraction.

Several ML methods can capture the dynamic changes of gait cycle, as well as highlight human spatial information from single frame. Combined with the manual calculation of the relative distance between the skeleton joints and other variables, the feature of strong separability was generated as the evidence of classification. For example, Kim et al. in [21] proposed to combine the discrete cosine transform and random forest (RF) for detecting frequency feature and classifying human gait based on wearable shoe sensors. The authors in [22] adopted a population HMM stands for gait dynamics. By summing the shape distance between the two standardized gait feature points, the biometric similarity between was calculated. Moreover, Choi et al. proposed a robust frame-level matching method for gait recognition by using 180 degree uni-directional sequences, which tracked position and distance for modeling the temporal and spatial information of each frame, and designed a two-level linear matching method to distinguish different gaits [27]. However, in terms of feature extraction there was a slight gap between these methods and deep learning methods.

On the other hand, for spatial feature analysis of single view on gait data, a great deal of deep learning methods relied on the temporality and dynamics of the gait to develop a variety of LSTM models to learn the temporal features of gait data [23,24].

Although these methods were sensitive to the representation of temporal information, they were inferior to CNN in the ability to analyzing human spatial features. Because the CNN was suitable for the spatial structure of single-view gait data, some varieties of CNN with different number of hidden layers and pooling layers were produced [25,26]. Besides, LSTM and CNN were also improved and applied to gait recognition field. For instance, the combination of a 3D-CNN and a stacked LSTM has been confirmed to be effective in representing spatio-temporal features and learning the long and short relationship between inter gait-cycle-segment [17].

Although the aforementioned methods achieved great results in the field of single-view gait analysis and recognition, they are inevitably affected by the lack of multi-view information, which is the reason for the evolution of multi-view gait recognition.

2.2. Multi-view gait recognition

Multi-view gait recognition approaches construct a gait modal by integrating features extracted from different views. Due to the complexity of multi-view gait recognition, a large number of fusion methods emerge instead of using single model for extracting features and identify gait.

First of all, the most common feature extraction method was to calculate hand-crafted features. Manjunatha et al. [28] regarded four directional variations of gradient gait energy image generated by neighborhood gradient computation as the discriminative gait features, which utilized support vector machine (SVM) as the classifier to identify human gait. While the authors in [29] studied the static features, such as a distance between joint 1 and joint 2, and dynamic features such as speed, strip length, and variation of barycenter were manually calculated based on the skeleton coordinate points collected by Kinect V2. Even though the hand-crafted features can successfully represent gait changes and relative positions of limbs from different views, they were unable to represent spatial information in detail.

For spatial feature fusion of multiple views, several methods have made great progress. Yan et al. [30] proposed CNN and multi-task learning based model to identify human gait and learn rich spatial features simultaneously. Worapan et al. [31] created a view transformation model based on spatial-domain gait energy image (GEI) by fusing different ML approaches, such as single value decomposition (SVD) and linear discriminant analysis (LDA). Additionally, a stacked sparse auto-encoders (SSAE) extracted the view-invariance feature for identity recognition [32], which included three shallow auto-encoder networks that were stacked in turn. It also takes advantages of convolutional layers for spatial information learning.

A multitude of methods showed their advantages for the extraction of spatio-temporal features. For instance, a novel spatio-temporal deep neural network (STDNN) was proposed for multi-view gait recognition based on gait energy image [33]. They used two streams for spatio-temporal feature extraction, and generated fusion feature by all the views. Liu et al. combined CNN and LSTM to learn the temporal and spatial deep feature information from gait energy image (GEI) and relative distance and angle (RDA) features [34]. Furthermore, Thien et al. [35] focused on first capturing the statistic and dynamic gait information, and then learning the fully gait information via a compact deep convolutional neural network. Different from the above method [33–35], the temporal feature of this model was obtained by calculating the Euclidean distance between the skeleton joints instead of using the LSTM.

However, these methods only build spatio-temporal models from different views, and then merge the outputs or directly integrate features of all the views and then build suitable models, which is easy to cause feature confusion, rather than flexible training according to the fixed relationship. In order to solve this problem, and also inspired by the above methods to extract spatio-temporal features of multi-view gait recognition, as well as data fusion, method fusion, and other technologies, we develop a hybrid model SpiderNet to determine which significant spatio-temporal features can be able to uniquely mark gait patterns, and closely integrate the features from each view as the final identification evidence.

3. The spiderweb graph neural network

In this section, we give a detailed description of our spiderweb graph neural network (SpiderNet). Specifically, we first introduce the overall framework of SpiderNet and formulate the structure of this model. And then, the single-view sequence modeling approach is demonstrated. Afterwards, we present the active multi-view graph modeling method. Finally, we illustrate the joint loss function of the whole model.

3.1. Framework and formulation

We aim to design an effective image-level framework, coined as spiderweb graph neural network (SpiderNet), by using a new strategy of graph based modeling. Instead of working on frames from single view, SpiderNet operates on sequences that are collected by multiple cameras from various views. For combining each corresponding image frame under each view and show the correlation between them, we search the key frame representation of the current view in other views to build multi-view model. On the other hand, we also build the single-view model for single image feature extraction and single-view temporal modeling, which can help to understand the dynamic gait changes and unique gait attributes.

The overall schema of this method is illustrated in Fig. 1. In the beginning, we input the image frame into the memory module to extract the temporal feature O_{vw} of the sequence, then feed the output feature O_{vt} into the capsule network for spatial and logical information analysis. The temporal feature O_{vw} will be used as the final evaluation basis for gait recognition. In the training process, the loss of capsule module and memory module should be reduced and optimized together generate more representative features. Finally, the spiderweb graph is trained according to the output feature of single-view model O'_{vw} and the optimal recognition result is generated.

Formally, we reshape the image sequence I_v of each view v into the corresponding M frame, and preprocess each frame to eliminate the background, and cut into $w \times h$ size pictures. The sequences are defined as $I_v = \{I_v \in \mathbb{R}^M, v = 1, 2, \ldots, V\}$ with corresponding label sequences $L_v = \{L_v \in \mathbb{R}^M, v = 1, 2, \ldots, V\}$. For the task of feature extraction, our goal is to address two problems at the same time: (1) successfully obtaining temporal feature O_{vw} from the inputs, and (2) analyzing the logical and spatial structure of the output O_{vt} of each memory node to output s_j. For the classification's task, a spiderweb graph is utilized to learn and separate the output feature of memory module and then produce the final classification results y.

3.2. Single-view sequence modeling

In this section, we aim to model the motion frame in single-view, which is shown in Fig. 2. We can see that the input image sequence I_v is fed into the memory module based on the expendable nodes. The picture sequence is trained in batches. The number of expandable nodes is the width of the input image. That is to say, time step is set to w, the input batch of the current node

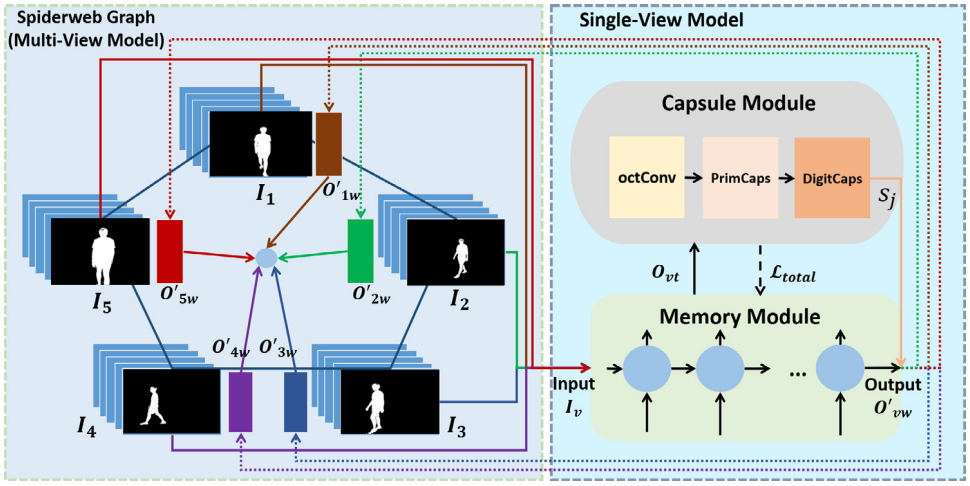


Fig. 1. The framework of SpiderNet. It includes two models: the multi-view model and the single-view model. We first input the image sequences I_v into the memory module, after obtaining the temporal feature O_{vt} of each memory node, a capsule module is used to do structural and spatial analysis of O_{vt} with three layers including octconv layer, primary capsule layer, and digit capsule layer, whose parameters are collaboratively trained with memory module to achieve stable loss ($\mathcal{L}_{total}$) reduction. The output feature s_j of the capsule module is concatenated to the temporal feature O_{vw} of the last memory node to get O'_{vw} as the input of the spiderweb graph.

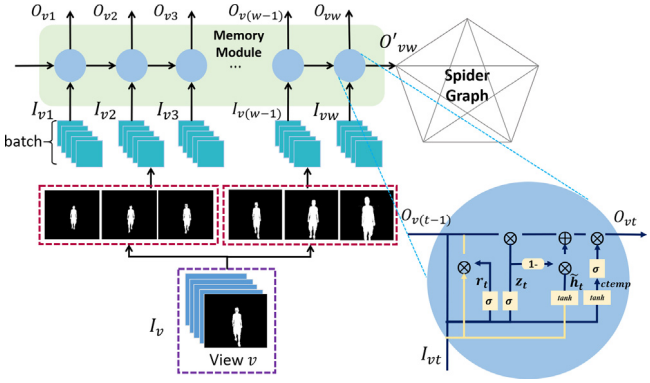


Fig. 2. The memory module. We input the image sequence I_v of a view into the memory module in batches to calculate the gait change feature. The input I_{vt} of each internal expandable node is the image matrix of a batch. The output of all nodes (O_{v1}- O_{vw}) is imported into the capsule module to obtain the vector s_j, and the memory module takes the fusion feature O'_{vw} (O_{vw} and s_j) of the last internal node as the input of the spiderweb graph.

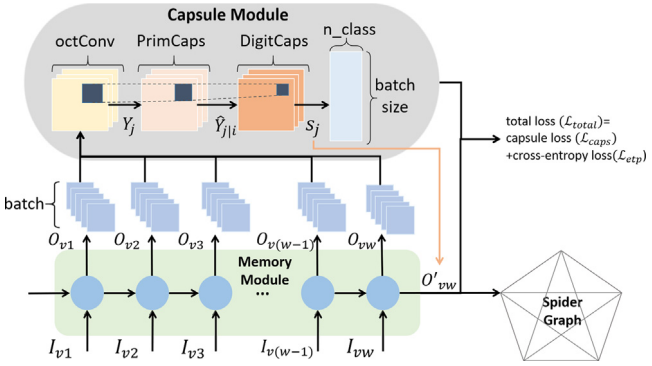


Fig. 3. The capsule module. After obtaining the temporal feature O_{vt} from the memory module, a three-layer capsule module can be utilized to train features O_{vt} from all the nodes. First, the octave convolutional layer is able to learn the high and low frequency features Y_j, the primary capsule (PrimCaps) and digit capsule (DigitCaps) extract features $\hat{Y}_{j|i}$ and s_j through affine transformation and dynamic routing, and integrate them with the output of memory module O_{vw} as the input of spiderweb graph network O'_{vw}.

at t is $I_{vt}, t \in \{1, 2, \ldots, w\}$, the input of each node is a matrix of $h \times$ *batch size*, and the hidden output of each node is sent to the node at the next time. We only take the output features of the last node to represent the temporal features of this batch. The internal structure and calculation operation of each node are shown in Fig. 2 and Eq. (1).

$$\begin{cases} z_t = \sigma(W_z \cdot [O_{v(t-1)}, I_{vt}]), \\ r_t = \sigma(W_r \cdot [O_{v(t-1)}, I_{vt}]), \\ \tilde{h}_t = \tanh(W \cdot [r_t \odot O_{v(t-1)}, I_{vt}]), \\ ctemp = \tanh(W_{ctemp} \cdot [O_{v(t-1)}, I_{vt}]), \\ c_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot O_{v(t-1)}, \\ O_{vt} = c_t \odot \sigma(ctemp), \end{cases} \tag{1}$$

where I_{vt} and $O_{v(t-1)}$ are the current input and previous output of the memory node at time $t \in \{1, 2, 3, \ldots, w\}$. σ and tanh are the logistic functions. z_t is the output of the update gate at time step t. r_t denotes the reset gate. The update gate z determines whether or not the hidden state will be updated with hidden state $\tilde{h}$. Unlike the LSTM node, a temporary state *ctemp* is added to determine I_{vt} and $O_{v(t-1)}$ while c_t denotes the active state of the node. We then extract features from c_t, an O_{vw} is the final output of the proposed node.

In order to deeply extract the structural and spatial feature, we introduce CapsNet to train output features from all the memory nodes, instead of only employing the last node of the memory module.

3.2.1. Octave convolutional layer

Unlike the original structure of the CapsNet, the octave convolutional layer is the first layer in our capsule module, which is to store and process feature maps that vary spatially slower at a lower spatial resolution reducing both memory and computation cost.

Octave convolution (Octconv) is a kind of plug and play convolution unit, which can directly replace the traditional convolution without any adjustment to the network architecture. In natural images, information is transmitted at different frequencies, the higher frequency is usually encoded with fine details, and the lower frequency is usually encoded with global structure [36].

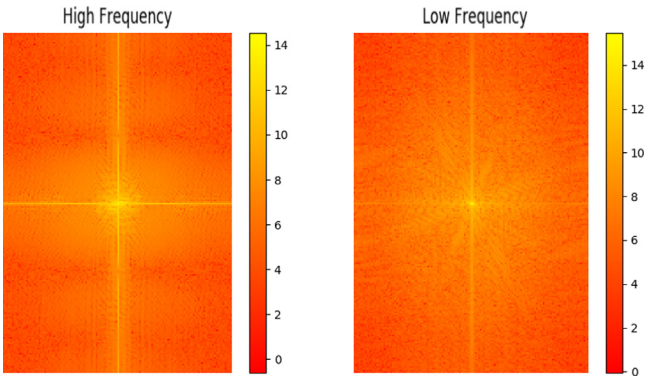


Fig. 4. The high and low frequency features of the gait data.

The operation process is shown in Eq. (2). In the process of convolution, the input feature of octave revolution consists of two parts, i.e., high frequency feature Y^H, and low frequency feature Y^L, both of which constitute the input part $Y = \{Y^H, Y^L\}$. Correspondingly, the input consists of two parts: high frequency feature output and low frequency feature output. Among them, the high frequency output feature Y^H is the sum of the feature $Y^{H \to H}$ obtained by convolution of the high frequency input feature itself and a mutual feature $Y^{L \to H}$ of the low frequency, and the low frequency output feature is similar. The high frequency and low frequency output features are derived as follows:

$$\begin{cases} Y^H = f(O_{vt}^H; W^{H \to H}) + upsample(f(O_{vt}^L; W^{L \to H}), 2), \\ Y^L = f(O_{vt}^L; W^{L \to L}) + f(pool(O_{vt}^H, 2); W^{H \to L}), \end{cases} \tag{2}$$

where $f(O_{vt}, W)$ denotes a convolution operation with parameters W, $pool(O_{vt}, k)$ is an average pooling operation with kernel size $k \times k$ and stride k. $upsample(O_{vt}, k)$ is an up-sampling operation by a factor of k via nearest interpolation.

The output of the Octconv is demonstrated in Fig. 4, we use the fast Fourier transform to represent the spectrum low frequency and high frequency features. The high-frequency features represent the places with fast gray changes, while the low-frequency features represent the places with slow changes. Fig. 3 shows the main process of the capsule module.

3.2.2. Capsule layer

As is shown in Fig. 3, subsequently, the processing of output Y can be divided into two stages: linear combination and dynamic routing. This process can be expressed by the following formula:

$$\hat{Y}_{j|i} = W_{ij} Y_i, \tag{3}$$

$$s_j = \sum_i c_{ij} \hat{Y}_{j|i}, \tag{4}$$

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}, \tag{5}$$

where $\hat{Y}_{j|i}$ is a linear combination of Y_i, and represents the output vector of the first capsule of the previous layer and the corresponding weight vector multiplied. $\hat{Y}_{j|i}$ is the connection to jth capsule of the next layer while the capsule i in former layer (Eq. (3)).

The dynamic routing is utilized to allocate the second stage for the calculation of the output s_j by updating c_{ij} iteratively. And then, the non-linear function "Squashing" can activate s_j (Eq. (4)).

A set of coupling coefficients c_{ij} can be updated and determined iteratively by dynamic routing process, which can generate predictions $\hat{Y}_{j|i}$. b_{ij} depends on the location and type of two capsules (Eq. (5)).

Finally, we leverage capsule module's advantages in feature analysis and extraction to synchronously train the memory module's parameters, which makes the output features O_{vw} of memory module more significant. We combine the output of digit capsule of capsule module with the output of memory module to get fusion feature O'_{vw}.

3.3. Multi-view graph modeling

To the best of our knowledge, CNN has been widely used in the field of gait recognition [17,26], but its research object is still limited to the data of Euclidean domains. The most significant feature is the regular spatial structure, for example, the voice is a regular one-dimensional sequence. The structure of these data can be represented by one-dimensional or two-dimensional matrix, which enables CNN to work efficiently.

However, a large amount of data in our real life do not have regular spatial structure, which is called non Euclidean data. For example, recommendation system, brain signal, molecular structure, and other abstract graphs, each node of these graph structures has different connections, some nodes have three or two connections, which are irregular data structures. The multi-view gait data can also be abstracted into a spiderweb graph convolutional network (spiderweb graph), which is the inspiration of the overall design.

The gait movement in different views does not have a complete matrix structure, but forms a graph together with certain connection relations, which is our motivation to construct spiderweb graph. In this section, we provide a specific spiderweb graph-based neural network model $f(H, A)$, which involves the theory of graph convolutional network (GCN) [37]. The output feature of the single-view model $O'_{vw} \in \mathbb{R}^D$ is fed into the spiderweb graph model for the collaborative training. We use 20% of the labeled data for semi-supervised classification with adjacency matrix. First we provide a V by V adjacency matrix (V is the number of nodes, namely, the number of views), a V by D feature matrix (D is the number of features per node), and a V by E binary label matrix (E is the number of classes) for the preprocessing of graph input.

3.3.1. Node calculation of spiderweb graph

Spiderweb graph convolutional network is a kind of deep learning method for graph-based data. Each node transfers its own feature information to the neighbor node after transformation. After extracting and transforming the node's feature information, each node gathers the feature information of the neighbor node, fuses the local structure information, and makes nonlinear transformation after obtaining the previous information to increase the expression ability of the model.

Fig. 5 illustrates the relationship of all view nodes and the framework of our spiderweb graph network, which is composed of an input layer, five hidden layers, and an output layer. The input layer is a spiderweb graph $H^0 \in \mathbb{R}^{V \times D}$ that takes the output O'_{vw} of single-view model as the node. The hidden layer consists of two graph convolutional layers (GCN), two ReLU activation functions, and an MLP full-connected layer. Finally, the gait label y is determined by a *softmax* function.

The GCN operator can represent the features of each node in the input graph, and expressed as:

$$h_j^{l+1} = \sigma(\sum_{j \in N_i} \frac{1}{c_{ij}} h_j^l w_{R_j}^l), \tag{6}$$

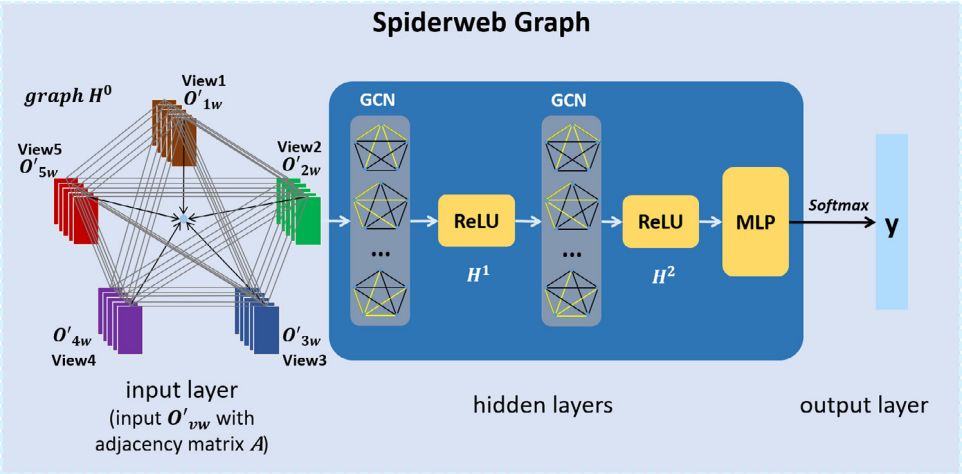


Fig. 5. The multi-view graph model. We take the five-view graph as an instance. After obtaining the feature O'_{vw} from single-view model, spiderweb graph convolutional network is constructed to train features in all views synchronously. It includes two GCN layers, two RelU activation function and an MLP Layer. The GCN layer includes the operation of graph nodes and the overall operation of graph. $H^0 - H^2$ represents the output of each layer. After extracting the eigenvector of each node, the corresponding label is obtained by matrix and probability operation.

where h_j^{l+1} denotes the feature of node i in layer l, c_{ij} is the normalization factor, N_i represents the neighbor node of node i, R_i shows the type of the node i, and w_{R_j} represents the transform weight parameter of node i in type R_j.

The receptive domain of GCN is proportional to the number of layers. Starting from the first layer, each node in the graph contains the information of direct neighbors. When calculating output of the second layer, the information of neighbors can be included so that the information involved in the operation is increasingly sufficient.

In addition to the operation of spiderweb on nodes, we will also introduce the training process of the inputting GCN layer in the form of the graph.

3.3.2. Spatial graph convolutions of spiderweb graph

We consider the spatial convolution for this task, which is to spread the features of each node and its neighbors to the next layer after weighted average. With the layers' deepening, the farther each node can aggregate the features, that is, the larger the receptive field. The weights are shared and will not be specific to each node, which is the same as traditional CNN. The number of adjacent nodes of each vertex may be different, which results in more significant eigenvalues of vertices with more adjacent nodes. The adjacency matrix cannot include the feature of node itself into the aggregate feature value. If two nodes are adjacent, the corresponding position in the matrix is 1, otherwise 0.

The propagation mode of features between layers can be expressed as follows:

$$H^{l+1} = f(H^l, A) = \sigma(AH^lW^l), H^{l+1} \in \mathbb{R}^{V\times D}, \tag{7}$$

when $l = 0$, H^0 represents the feature matrix of the input graph, which consists of the outputs O'_{vw} of the single-view model. $A \in \mathbb{R}^{V\times V}$ is the input graph's adjacency matrix, which is built before training, and $W^l \in \mathbb{R}^{D\times D}$ is the weight matrix of the lth layer. σ is the activation function ReLU. The aggregation of features can be realized by multiplying the feature matrix by the adjacency matrix to the left, and then weighting operation can be realized by multiplying the weight matrix to the right. Weight matrix W^l and the feature matrix H^l are the key research objects when using graph convolution for gait recognition. In the SpiderNet's training process, W^l and H^l are learned to reduce the loss until they obtain the best classification results.

We consider a five-view gait graph, the adjacency matrix A is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{8}$$

After the input graph H^0 is processed by GCN layer and the activation function ReLU, a $V \times E$ (E is the class number) - size predictive label matrix is generated in the full-connected layer (MLP), and the final label y is obtained by *softmax*.

3.4. Loss function

We calculate the sum of the loss of capsule module, memory module, and spiderweb graph as the joint loss to train the whole network.

In capsule module, margin loss is reduced to optimize the model, which is illustrated as follows:

$$\mathcal{L}_{cap} = T_c max(0, m^+ - ||v_c||)^2 + \lambda(1 - T_c)max(0, ||v_c|| - m^-)^2, \tag{9}$$

where c is the class label of the sample, T_c is the indicate function (1 if c exists, else 0), m^- is the top margin of $\|v_c\|$, m^+ is the bottom margin of $\|v_c\|$.

To compute the loss of memory module, we utilize the softmax cross-entropy function as shown in Eq. (10).

$$\mathcal{L}_{etp} = -\sum_j y_j \log(\frac{e^{y'_j}}{\sum_{i=1}^n e^{y'_i}}), \tag{10}$$

where $\mathcal{L}_{etp}$ represents the cross-entropy loss of model. In the optimization process, we apply Adam optimizer to consider the first and second moments of gradient in order to reduce the loss quickly. Here, y_j denotes the jth true label of a training batch while y'_j represents the predicted label.

Combining cross-entropy loss $\mathcal{L}_{etp}$ from the memory module, and the margin loss $\mathcal{L}_{cap}$ from the capsule module, the final joint loss function is formed as:

$$\mathcal{L}_{total} = \mathcal{L}_{etp} + \mathcal{L}_{cap}. \tag{11}$$

Mean square error (MSE) is the expectation of the square of the difference between the estimated value and the true value of the parameter. The spiderweb graph adopts the MSE to evaluate the classification independently.

$$\mathcal{L}_{mse} = \frac{1}{N}\sum_{i=1}^{N}(y_j - y_j')^2, \tag{12}$$

where N represents the number of samples, y_j is the true label, and y_j' is the predicted label.

4. Experiment

In the experiment, we test and evaluate the classification performance of the proposed model on three multi-view gait datasets. By comparing the current mainstream methods and the various components of SpiderNet, the results are superior to most of the existing methods.

The comparison methods are listed as follows:

- LSTM is a model with expandable nodes and is suitable for temporal data.
- CNN is a kind of feed-forward neural network with deep structure and convolution calculation.
- RF (Random Forest) is a classifier with multiple decision trees.
- HMM (Hidden Markov Model) is a statistical model used to describe a Markov process with unknown parameters, which is suitable for time series data.
- GRU (Gate Recurrent Unit) is a lightweight variant of LSTM with fewer variables
- Two-level linear matching [27] is a robust frame-level matching method for gait recognition by tracking position and distance for modeling the temporal and spatial information of each frame.
- NGC-SVM [28] is a method using neighborhood gradient computation and SVM (support vector machine) to identify human gait.
- Hand-crafted Method [29] computes static and dynamic features of the gait.
- CNN-MT [30] combines CNN and multi-task learning model for gait recognition.
- SVD-LDA [31] adopts various ML approaches like SVD and LDA for gait recognition.
- SSAE [32] is a stacked sparse auto-encoder to extract the view-invariance feature for identity identification based on gait data.
- STDNN [33] is a novel spatio-temporal deep neural network is proposed for multi-view gait recognition based on gait energy image.
- CNN-LSTM [34] is a fusion model of LSTM and CNN.
- 3D-STCNN [35] is designed for identifying statistic and dynamic gait information through a deep convolutional neural network.
- CNN-3DCNN [38] and CNN-CGI [38] are two models designed for gait recognition based on CNN.

4.1. Datasets

In the experiment, we utilized three gait datasets, the CASIA-B gait dataset [39], the SDUgait dataset [40], and the OU-MVLP gait dataset [41] including multi-view image sequences.

The SDUgait dataset: The dataset includes 52 subjects, which comprise of 28 males and 24 females with the average age of 22. Each subject had 20 sequences with at least 6 fixed walking directions and 2 arbitrary directions, and totally 1040 sequences; The dataset is based on the second generation Kinect (Kinect V2). Each view includes gait silhouettes of video frames and 3D skeleton coordinate points of joints.

The CASIA-B gait dataset: Dataset B is a large multi-view gait database including 124 subjects, and the gait data is captured from 11 views. Three variations, namely view angle, clothing, and carrying condition changes, are separately considered. We utilize the provided human silhouettes extracted from video files, and select five views from the 11 views for gait recognition. Because the number of silhouettes in each view is different, although, they are collected uniformly, they also need to be filtered and preprocessed accordingly.

The OU-MVLP gait dataset: Multi-View Large Population Dataset (OU-MVLP) is collected in conjunction with an experience-based long-run exhibition of video-based gait analysis at a science museum [42]. The approved consent is obtained from all the subjects in this dataset. The dataset consists of 10307 subjects (i.e., 5114 males and 5193 females with various ages, ranging from 2 to 87 years) from 14 view angles, ranging $0^\circ - 90^\circ$, $180^\circ - 270^\circ$. Gait images of 1280×980 pixels at 25 fps are captured by seven network cameras (Cam. 1–7) placed at intervals of 15-deg azimuth angles along a quarter of a circle whose center coincides with the center of the walking course. Its radius is approximately 8 m and the height is approximately 5 m.

4.2. Experimental set-up

The experiments are implemented by Tensorflow [43] and Python [44] libraries. We will introduce the parameter settings for the three datasets. First, we cut the pictures of these three datasets and resize them to 50 * 50 to unify some variables for preprocessing.

For SDUgait dataset, first of all, we take 93 corresponding gait silhouettes from the sequence of each view, and then cut the input image of 512 * 424 to 50 * 50 according to the height of the human body. After preprocessing, there are 24180 pictures in the dataset, 93 gait silhouettes in each view of each category. We use 3 * 3 convolutional kernels in octConv with the stride 1, α is taken as 0.5 and padding method is 'same'. In primary capsule layer, 3 * 3 kernels are used with the stride 2. In memory module, time step is set to 50 with the batch size of 128 and dimension of the input is 50 (the size of the image is 50 * 50), the dropout is set to 0.5 to avoid over-fitting. In spiderweb graph, we establish an adjacency matrix to connect the image indexes of each view, the dropout is 0.1 and the learning rate is 0.001. The whole model is iterated 10^7 times.

For CASIA-B dataset, we apply four views of the silhouette sequence with 83 frames in each view. After each frame with the size of 320*240 is cropped to 50 * 50, we send the sequence to memory module in order. The settings of memory unit and capsule network are the same as those of SDUgait except for the number of data classes. In spiderweb graph, we build the adjacency matrix of four views based on the relationship between the index of each frame and other indexes. Dropout is set to 0.2, and the number of iterations is 10^6 times to reach the optimal value of parameters.

For OU-MVLP dataset, we select 20 subjects in the dataset to carry out the experiment, and collect the gait data of 6 views to conduct joint training. Differently, the time step and the dimension of the input feature in the memory module are set to 50 and 50. The size of convolutional kernels in convolutional layer and primary capsule layer of the capsule module are designed to 9*9 and 5*5. For the spiderweb graph network, an adjacency matrix of six views is built as the input of the spiderweb network. With the dropout of 0.5 and the learning rate of 0.001, the joint training is implemented 10^6 iterations for multi-view gait recognition.

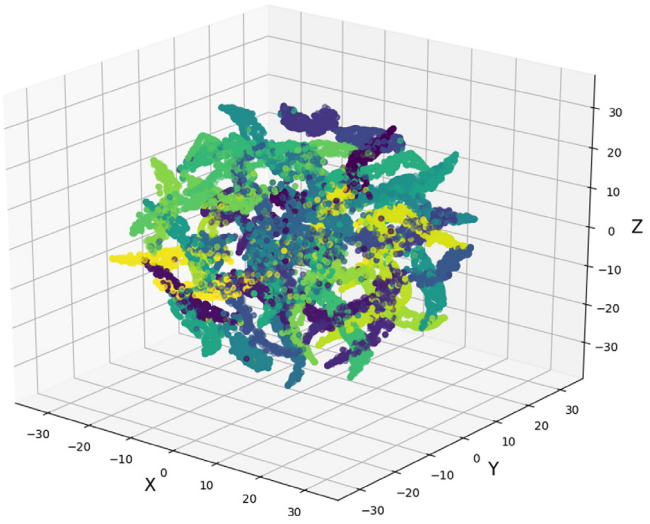
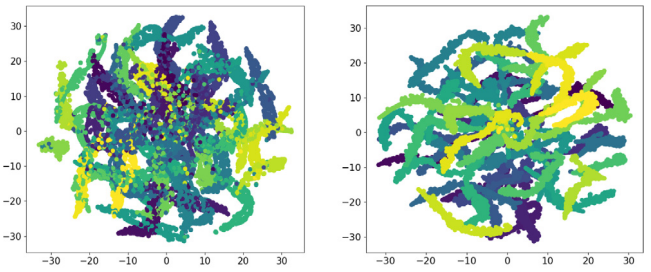


Fig. 6. The output features of the memory module on SDUgait dataset. We can see that the features of different colors represent different classes. The feature clustering near the center of coordinate axis is poor, and the classification near the edge is clearer.



(a) Memory Module (b) Single-View Model

Fig. 7. The output features of the memory module and the single-view model (memory module and capsule module) on SDUgait dataset. In (a) the feature points are more crowded, and there are chaotic scattered points in the center, and the boundary of each class in (b) is more obvious.

In addition, the testing time for the best experimental results of the three modules of the proposed model are 2.65 s (memory module), 2.34 min (capsule module), 1.23 s (spiderweb graph) which run on the system of GTX1050Ti GPU, Inter Cure i7 CPU and 8G RAM.

4.3. Results on SDUgait dataset

In this section, we mainly explain the recognition results of each split model on SDUgait dataset, and compare the whole model with methods in other literature.

We select 0°, 180°, 225°, 270° and an arbitrary direction for gait recognition.

4.3.1. Performance of the memory module

We use the improved memory node to analyze the input image frame. The distributions of the extracted 3D feature are shown in Fig. 6. We utilize t-SNE (t-distributed stochastic neighbor embedding) for nonlinear dimension reduction visualization. The distance between similar data in high-dimensional space and low-dimensional space is also similar. In general, the Euclidean distance is used to describe the similarity, while SNE transforms the distance relationship into a conditional probability to express the similarity. Although the t-SNE takes long time and occupies a large amount of memory, it performs very well as an unsupervised dimension reduction method [45].

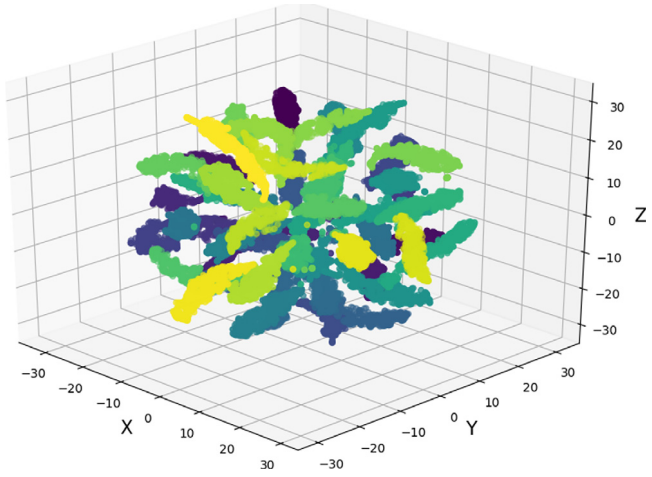


Fig. 8. The output features of the spiderweb graph network on SDUgait dataset.

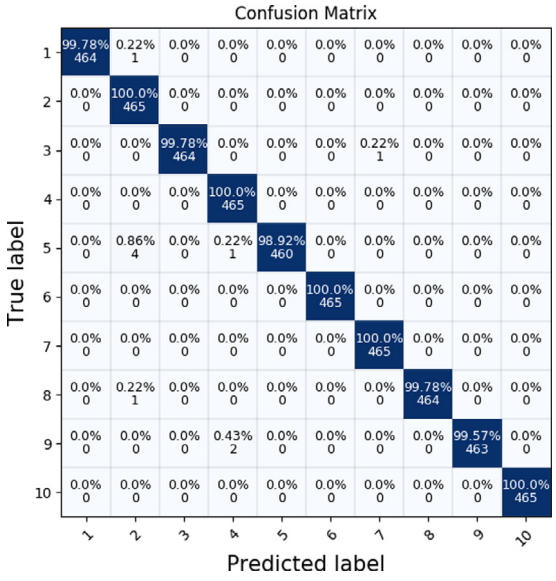
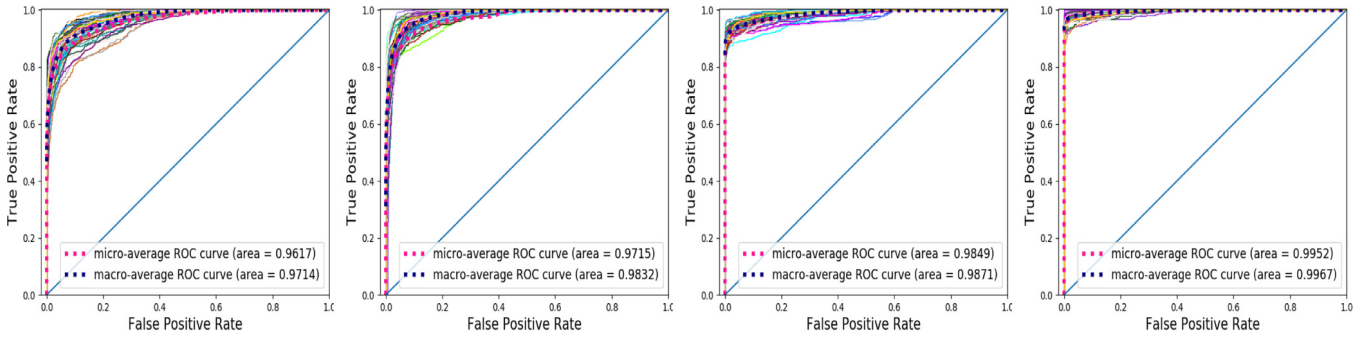


Fig. 9. The confusion matrix of SpiderNet on SDUgait dataset. Since the recognition results of 52 subjects are difficult to display completely, we select the subjects labeled 1–10 to show the recognition results and achieve 98.54% of the classification result.

After the memory module's feature extraction, the capsule network is used to train the parameters of the memory module to enhance its classification performance. The feature output of the expansion node in the memory module is imported into the capsule network. The loss function in the training process controls the parameter estimation of the two modules and interfaces with the feature output of the last node in the memory module.

4.3.2. Performance of the capsule module

Although the feature extraction effect of capsule module is limited, after the intervention of memory module, the performance of capsule module is improved significantly, which also makes the output features of memory module more separable. The visual result of feature output is shown in Fig. 7. According to the extracted feature distribution, the performance of memory module and capsule module is higher.



(a) Memory Module (b) Single-View Model (c) Spiderweb Graph (d) SpiderNet

Fig. 10. The ROC curve of different components in SpiderNet on SDUgait dataset. The best performance of the components is spiderweb graph, which has a slightly higher classification accuracy than memory module and capsule module. SpiderNet has a high recognition rate for each class and the curve is concentrated in the upper left corner. Besides, The micro-average and macro-average AUC values are compared in the four models, which shows the strong learning ability of SpiderNet.

Table 1
Performance of classifying 52 subjects compared with advanced models on SDUgait dataset.

SDUgait-52					
Advanced models				The proposed methods	
Two-level linear matching [27]	98.08%	LSTM	88.36%	Memory module	90.22%
Hand-crafted method [29]	94.23%	RF	81.38%	Capsule module	83.34%
CNN-LSTM [34]	88.11%	HMM	85.29%	Single-View model	91.23%
3D-STCNN [35]	91.04%	GRU	88.97%	Spiderweb graph	92.69%
		CNN	87.55%	SpiderNet	***98.54%***

4.3.3. Performance of the spiderweb graph network

Different from the former two modules, the graph network links every frame of the five views together to train gait data, so the result is slightly higher than the former two modules. In Fig. 8, the generated features of the same class are closely aggregated, and the boundary between different classes is noticeable.

It is proved that the graph network successfully represents the similarity of the same type of gait from different views, and their commonness is within a certain range.

4.3.4. Performance of the whole model

In this section, we use ROC curve to evaluate the classification effect of each module of the proposed model in Fig. 10, which can easily find out the performance recognition ability of any boundary value. The closer the ROC curve is to the upper left corner, the more accurate the result is. The point closest to the ROC curve in the upper left corner is the best threshold with least errors, and the total number of false positives and false negatives is the least.

We also calculate the average area under curve (AUC, area under ROC curve) of all the classes. The value of AUC is between 0.5 and 1, the closer the value is to 1.0, the higher the effect of the classifier is. When it is equal to 0.5, it has no application value. The AUC of SpiderNet and its components is more than 95%, which proves the high accuracy of the SpiderNet.

The confusion matrix is used to explain the classification results of SpiderNet in Fig. 9. The probability of each classification of 52 classes is more than 90%, which indicates the validity of the model.

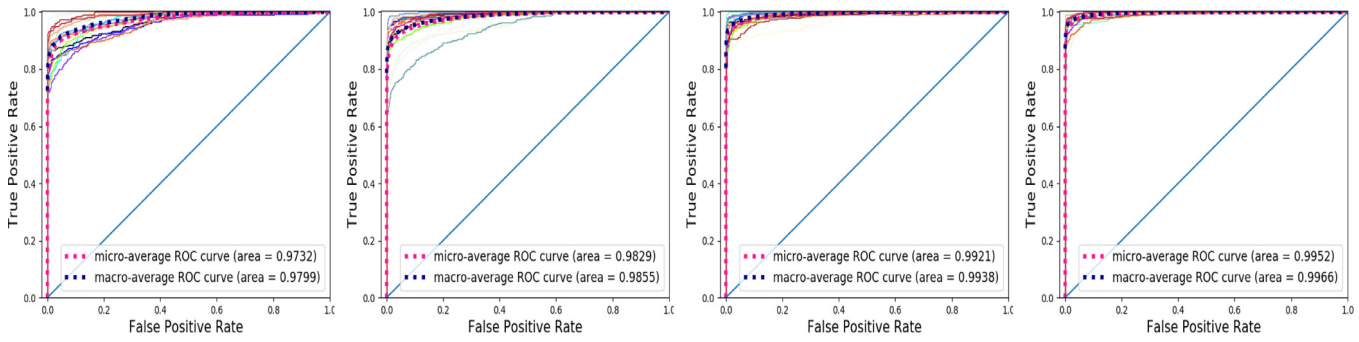
In addition, we also compare the performance of recognition methods in other literature on the SDUgait dataset, which is shown in Table 1. There are lots of advanced models achieving excellent results. As mentioned above, the CNN-LSTM [34] proposed for 3D gait recognition based on the fusion of gait energy image's relative distance and angle features. A 3D spatio-temporal CNN (3D-STCNN) presented by Thien et al. [35], achieves 88.11% and 91.04% of the accuracy respectively, which is inferior than some complex non deep learning approaches studied by Wang et al. [29] and Choi et al. [27]. It is noticed that a robust frame-level matching method [27] shows distinctive effect in gait recognition through view-invariant modeling and kinematic gait analysis.

Table 2
Performance of classifying different subjects compared with advanced models on CASIA-B dataset.

CASIA-B-124		CASIA-B-20	
NGC-SVM [28]	90.40%	CNN	85.23%
CNN-MT [30]	95.88%(avg)	LSTM	89.66%
SVD-LDA [31]	90.00%	GRU	89.97%
SSAE [32]	93.67%	HMM	87.35%
STDNN [33]	95.67%	RF	86.55%
CNN-3DCNN [38]	92.11%	–	–
CNN-CGI [38]	89.83%	–	–
Memory module	90.76%	Memory module	90.78%
Capsule module	88.45%	Capsule module	89.34%
Single-View model	91.95%	Single-View model	92.33%
Spiderweb graph	93.12%	Spiderweb graph	93.65%
SpiderNet	***97.23%***	SpiderNet	***98.77%***

Among the popular algorithms, the deep learning method is better than the traditional ML method (Random Forest, RF) in general, the temporal feature extraction method LSTM outperforms spatial feature extraction method CNN, GRU is superior to LSTM because of simpler internal structure and fewer parameters.

The result of memory module is better than that of original LSTM because of adding state control channel. The effect of capsule module is a little poor, which is added to the memory module for collaborative training to enhance the feature extraction ability of the memory module. The performance of spiderweb graph achieves remarkable result because it is associated with the original gait feature from different views, but the fusion model uses the separable output of memory module as the input to classify the features more effectively to get the best recognition accuracy.



(a) Memory Module (b) Single-View Model (c) Spiderweb Graph. (d) SpiderNet

Fig. 11. The ROC curve of different components of SpiderNet on CASIA-B dataset. We can see that the micro-average and macro-average AUC values of spiderweb graph and SpiderNet, which illustrates the outstanding performance of classification.

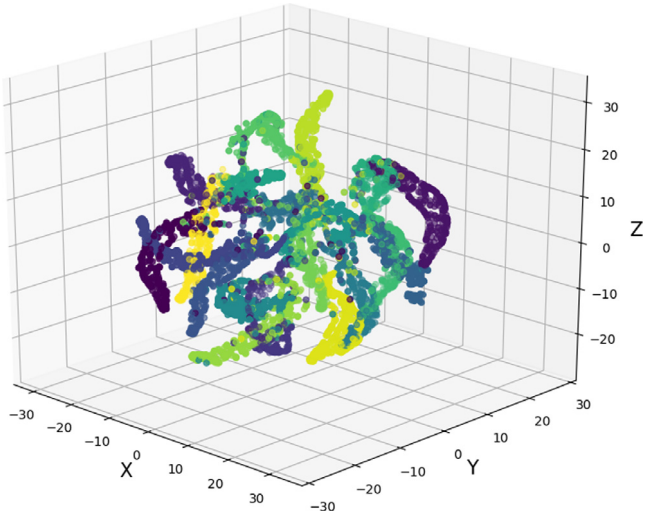
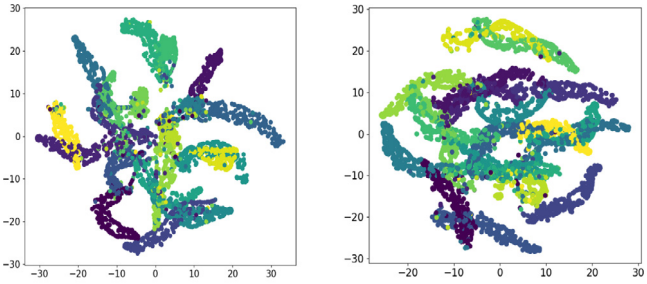


Fig. 12. The output features of the memory module on CASIA-B dataset.



(a) Memory Module (b) Single-View Model

Fig. 13. The output features of the single-view model (memory module and capsule module) on CASIA-B dataset.

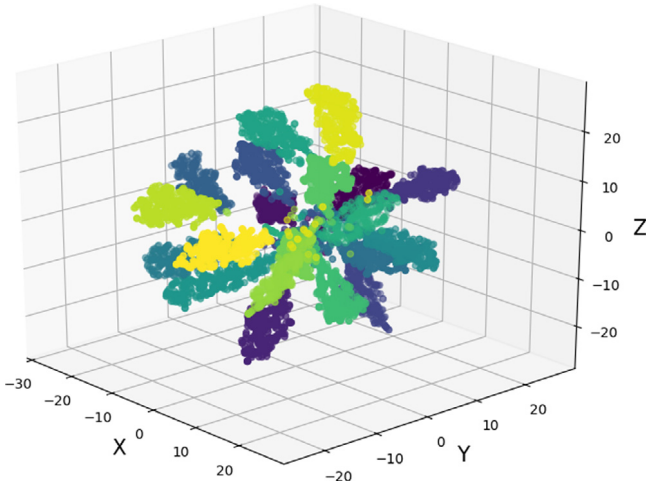


Fig. 14. The output features of the spiderweb graph on CASIA-B dataset.

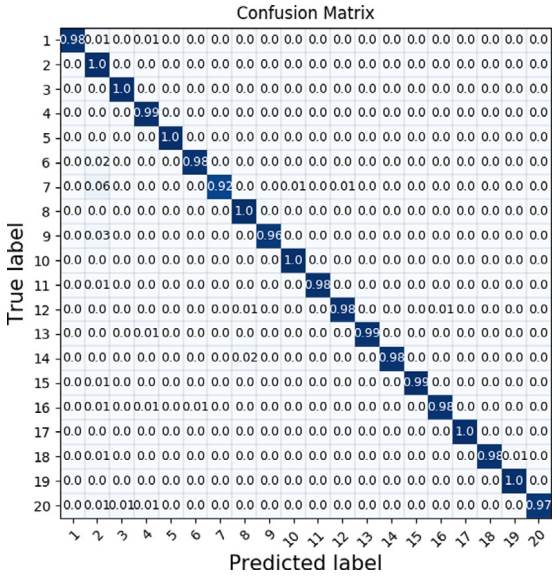


Fig. 15. The confusion matrix of SpiderNet on CASIA-B dataset. We achieve 98.77% of the classification result.

4.4. Results on CASIA-B dataset

According to the processing method on SDUgait dataset, we compare the classification accuracy of different split models and other recognition methods on CASIA-B dataset. We take silhouette samples of 20 subjects in the dataset from the dataset for experiment. There are four directions under normal condition 0°, 18°, 54°, 180°, which are selected for gait recognition, because more than 80 frames of images from these views are conducive to model training.

4.4.1. Performance of the memory module

In the memory module, the gait data in CASIA-B is input to the expandable memory nodes in the order of context. We take the

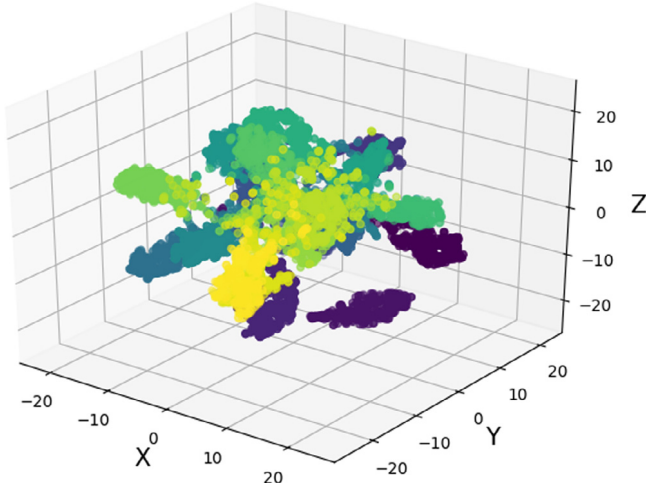


Fig. 16. The output features of the memory module on OU-MVLP dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

output feature of the last node for display. The result of dimension reduction to 3D is shown in Fig. 12. It can be seen that the number of cross samples between different classes is large and the misclassification rate is relatively high.

4.4.2. Performance of the capsule module

With the assistance of capsule module, the output features of the last node in memory module are illustrated in Fig. 13. For 20 subjects, the capsule module improves the classification performance of the memory module, features extracted by the memory module are separable and serve as the input of spiderweb graph.

4.4.3. Performance of the spiderweb graph network

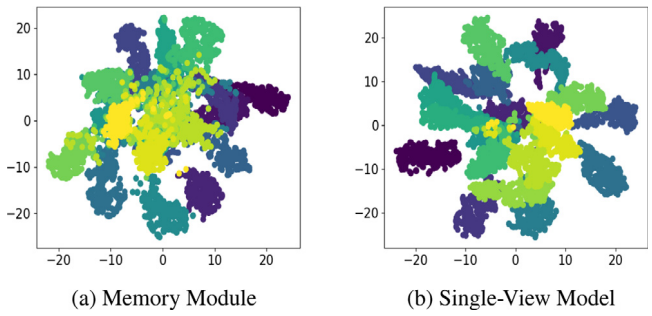
The output features of the spiderweb graph are demonstrated in Fig. 14. The feature distribution of 20 classes is clearer than that of 52 classes in SDUgait dataset. The performance of spiderweb graph still stays ahead of the other submodels.

4.4.4. Performance of the whole model

Fig. 11 demonstrates the ROC curve of the split models. We can see that the average AUC values increase according to the complexity of the four models. Our SpiderNet is the model with the highest complexity and the best performance, the AUC values that achieve the recognition rate of 99.52% and 99.66%. The confusion matrix is indicated in Fig. 15.

Additionally, the recognition accuracy of the fusion model SpiderNet is shown in Table 2. We also evaluate the recognition rate of 124 subjects in the dataset, as shown in Table 2. The prevalent models are compared with our methods. It can be seen that the performances of CNN or LSTM based on deep learning are all concentrated in 89%–93%, with little difference. Slightly better than these methods are STDNN [33] and the method in [30], which have advantage of extracting temporal and spatial features of gait data and dealing with the data of frame-level and sequence-level, respectively. In addition to these merits, our SpiderNet adds the overall structural features, low and high frequency fusion features and multi-view fusion features, of the gait to enable the promising results.

The performance ranking of other models and components on SDUgait and CASIA-B datasets is basically the same. The results of our SpiderNet on 124-class and 20-class are 97.23% and 98.77%, respectively.



(a) Memory Module (b) Single-View Model

Fig. 17. The output features of the single-view model (memory module and capsule module) on OU-MVLP dataset.

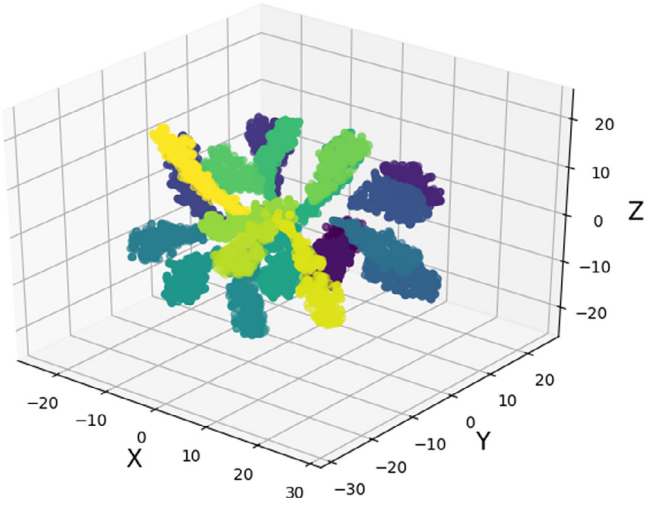


Fig. 18. The output features of the spiderweb graph on OU-MVLP dataset.

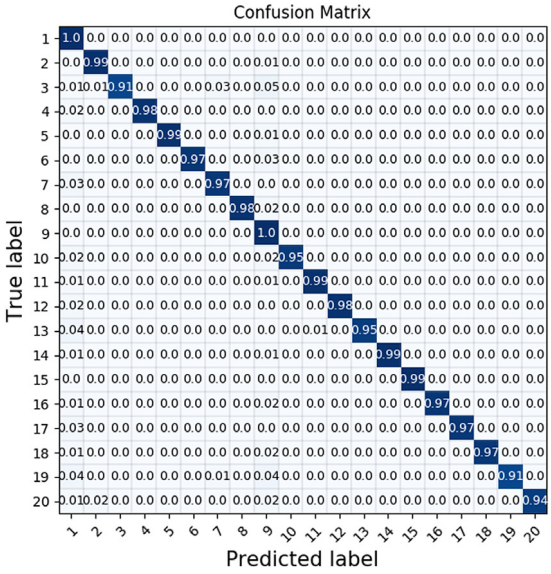
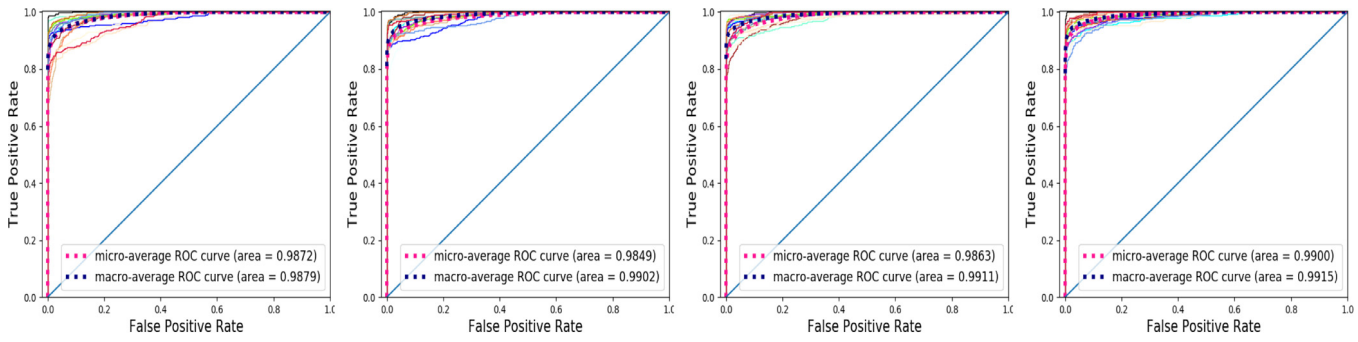


Fig. 19. The confusion matrix of SpiderNet on OU-MVLP dataset. We achieve 96.91% of the classification result.

4.5. Results OU-MVLP dataset

This section demonstrates the recognition results of split models on OU-MVLP dataset, we also compare the whole model with methods in other literature. 20 subjects are randomly selected to



(a) Memory Module (b) Single-View Model (c) Spiderweb Graph. (d) SpiderNet

Fig. 20. The ROC curve of different components of SpiderNet on OU-MVLP dataset. We can see that the micro-average and macro-average AUC values of spiderweb graph and SpiderNet are slightly larger than those of other models.

Table 3
Performance of advanced models on OU-MVLP (20 subjects) dataset.

OU-MVLP-20							
Advanced models				The proposed methods			
LSTM	90.85%	HMM	91.73%	Memory module	93.67%	Spiderweb graph	95.45%
RF	93.18%	GRU	92.97%	Capsule module	91.33%	SpiderNet	***96.91%***
CNN	89.41%			Single-View model	94.58%		

participate in the multi-view joint training of 6 views: 0°, 45°, 90°, 210°, 225°, 255°. We use 20% of the gait data for testing and verify the classification performance of the model.

4.5.1. Performance of the memory module

First of all, we evaluate the memory module with the enhanced memory unit. The extracted 3D temporal feature from the gait sequences after dimension reduction is shown in Fig. 16. The t-SNE figure illustrates that the closer the 3D feature points are to the coordinate center, the more chaotic the features are. The yellow and green sample points are overlapped, which have a certain error rate. The dark points are close to the edge of the coordinate axis, and the aggregation effect indicates that the classification accuracy is high, which also shows that the features of these subjects are quite different from those of the light color ones.

4.5.2. Performance of the capsule module

After all, with the improvement of the capsule module, the classification ability of the memory module has been enhanced a lot. This section shows the advantages of the whole single-view modeling (memory module+capsule module) compared with the independent memory module. The visual result of 2D feature output is shown in Fig. 17. Various colors represent different categories, and the yellow category in subfigure (a) can hardly see the boundary, which indicates that the misclassification rate of this category is very high. In subfigure (b), although the sample points in the coordinate center are still very concentrated, they can be roughly separated from the boundary, and the aggregation degree of dark sample points is also higher than that of memory module, highlighting the advantages of single-view modeling.

4.5.3. Performance of the spiderweb graph network

Because of the joint training of multiple views and the information enhancement of the GCN layer, the classification ability of spiderweb graph network is stronger. As shown in Fig. 18, we can observe the boundary of each sample category, as well as the aggregation density of the same kind is higher. In the coordinate center, we can clearly identify the corresponding classification indicating the validity of the spiderweb graph.

4.5.4. Performance of the whole model

As in the previous two experiments, the confusion matrix and ROC curve are also introduced to show the classification performance of SpiderNet, and the results are explained in Figs. 19 and 20. The recognition rate of 20 subjects in Fig. 19 is above 90%, which shows the stability of the SpiderNet model. The subjects with the highest and the lowest error rates are No.19 and No.1, respectively. The probability of identifying samples No.1 and No.9 is 100%, which demonstrates the significance of their gait patterns.

For the ROC curve, the macro-average AUC value of the four models is higher than the micro-average AUC value, which increases with the complexity of the model. The worst performance model is memory module, achieves 98% of the AUC. The trend of the 20-class curves can also be seen that the false positive and true positive samples are later fitted to 1, for which SpiderNet performs the best, and its macro-average and micro-average AUC values are higher than 99%, with the best performance.

In addition, we also compare some studies on the OU-MVLP dataset, which are shown in Table 3. In terms of feature extraction methods, the temporal feature extraction method is better than the spatial feature extraction method, indicating the significance and uniqueness of sequence features. However, the classification effect of RF outperforms that of some deep learning methods such as LSTM, CNN and capsule module, which shows the advantage of the discriminant model. RF combines the prediction of multiple decision trees into one model when distinguishing multi-dimensional features, which composes of many mediocre models is still better than a single model, and it is also not easy to produce the phenomenon of over-fitting.

Our multi-model fusion method is still superior to other state-of-the-art methods. Experiments show that the single-view and multi-view modeling methods of SpiderNet proposed are higher than the classification results of RF, verifying its superiority in multi-view gait recognition.

5. Conclusion

In this paper, we presented a deep framework for multi-view gait recognition, and formulate an active learning technique that

utilized the structural, contextual, and spatio-temporal information of the gait data from different views. A spiderweb graph neural network (SpiderNet) was built to train the sequences of all views and account for the interrelationships between them. We also showed the technical procedure to cooperatively extract features under various views. Finally, experimental results have demonstrated the effectiveness of our model, which indicated that the fusion of multiple features allows to boost the recognition accuracy of the system in numerous cases, or it is superior than the best results achieved by using single modality.

In the light of the obtained results, the SpiderNet has achieved 98.54%, 98.77%, and 96.91% of the results on SDUgait, CASIA-B and OU-MVLP datasets, respectively, which are better than other main-stream methods. Regarding to the fusion method, we only considered the view-level fusion method, not involving the multi-modal data fusion method, such as the integration of image, coordinate, and force sensitive data. For the future work, we will extensively explore the fields of abnormal gait detection, gait segmentation, and real-time gait tracking.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Sincere thanks are due to the following institutions for their assistance in the gait data collection of the study.

Portions of the research in this paper use the CASIA Gait Database collected by Institute of Automation, Chinese Academy of Sciences, SDUgait Gait Database collected by Shandong University (SDU), China, and OU-MVLP Gait Database collected by the Institute of Scientific and Industrial Research (ISIR), Osaka University (OU).

References

[1] L. Fiorini, G. Mancioppi, F. Semeraro, H. Fujita, F. Cavallo, Unsupervised emotional state classification through physiological parameters for social robotics applications, Knowl.-Based Syst. 190 (2020) 1–10.
[2] E. Auvinet, F. Multon, V. Manning, J. Meunier, J. Cobb, Validity and sensitivity of the longitudinal asymmetry index to detect gait asymmetry using microsoft kinect data, Gait Posture 51 (2017) 162–168.
[3] Z. Xue, D. Ming, W. Song, B. Wan, S. Jin, Infrared gait recognition based on wavelet transform and support vector machine, Pattern Recognit. 43 (8) (2010) 2904–2910.
[4] D. Kastaniotis, I. Theodorakopoulos, G. Economou, S. Fotopoulos, Gait-based gender recognition using pose information for real time applications, in: 2013 18th International Conference on Digital Signal Processing (DSP), 2013, pp. 1–6.
[5] L. Liu, Y. Peng, M. Liu, Z. Huang, Sensor-based human activity recognition system with a multilayered model using time series shapelets, Knowl.-Based Syst. 90 (2015) 138–152.
[6] J. Tang, J. Luo, T. Tjahjadi, F. Guo, Robust arbitrary-view gait recognition based on 3D partial similarity matching, IEEE Trans. Image Process. 26 (1) (2017) 7–22.
[7] X. Xing, K. Wang, T. Yan, Z. Lv, Complete canonical correlation analysis with application to multi-view gait recognition, Pattern Recognit. 50 (2016) 107–117.
[8] W. Kusakunniran, Q. Wu, J. Zhang, H. Li, Support vector regression for multi-view gait recognition based on local motion feature selection, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 974–981, http://dx.doi.org/10.1109/CVPR.2010.5540113.
[9] T. Wolf, M. Babaee, G. Rigoll, Multi-view gait recognition using 3D convolutional neural networks, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 4165–4169, http://dx.doi.org/10.1109/ICIP.2016.7533144.
[10] F.M. Castro, M.J. Marín-Jiménez, N.G. Mata, R. Muñoz-Salinas, Fisher motion descriptor for multiview gait recognition, 2016, CoRR abs/1601.06931.
[11] T.T. Verlekar, P.L. Correia, L.D. Soares, View-invariant gait recognition system using a gait energy image decomposition method, IET Biometrics 6 (4) (2017) 299–306.
[12] A.S.M.H. Bari, M.L. Gavrilova, Artificial neural network based gait recognition using kinect sensor, IEEE Access 7 (2019) 162708–162722, http://dx.doi.org/10.1109/ACCESS.2019.2952065.
[13] F.M. Castro, M.J. Marin-Jimenez, N. Guil, S. Lopez-Tapia, N. Perez de la Blanca, Evaluation of cnn architectures for gait recognition based on optical flow maps, in: 2017 International Conference of the Biometrics Special Interest Group (BIOSIG), 2017, pp. 1–5, http://dx.doi.org/10.23919/BIOSIG.2017.8053503.
[14] Q. Chen, Y. Wang, Z. Liu, Q. Liu, D. Huang, Feature map pooling for cross-view gait recognition based on silhouette sequence images, in: 2017 IEEE International Joint Conference on Biometrics (IJCB), 2017, pp. 54–61, http://dx.doi.org/10.1109/BTAS.2017.8272682.
[15] S. Wen, W. Liu, Y. Yang, T. Huang, Z. Zeng, Generating realistic videos from keyframes with concatenated GANs, IEEE Trans. Circuits Syst. Video Technol. PP (2018) 1, http://dx.doi.org/10.1109/TCSVT.2018.2867934.
[16] S. Wen, H. Wei, Z. Yan, Z. Guo, Y. Yang, T. Huang, Y. Chen, Memristor-based design of sparse compact convolutional neural network, IEEE Trans. Netw. Sci. Eng. PP (2019) 1, http://dx.doi.org/10.1109/TNSE.2019.2934357.
[17] D. Thapar, G. Jaswal, A. Nigam, C. Arora, Gait metric learning siamese network exploiting dual of spatio-temporal 3D-CNN intra and LSTM based inter gait-cycle-segment features, Pattern Recognit. Lett. 125 (2019) 646–653.
[18] M. Pei, X. Wu, Y. Guo, H. Fujita, Small bowel motility assessment based on fully convolutional networks and long short-term memory, Knowl.-Based Syst. 121 (2017) 163–172.
[19] C. Dai, X. Liu, J. Lai, Human action recognition using two-stream attention based LSTM networks, Appl. Soft Comput. 86 (2020) 105820.
[20] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: 31st Conference on Neural Information Processing Systems (NIPS 2017), 2017.
[21] J. Kim, K.B. Lee, S.G. Hong, Random forest based-biometric identification using smart shoes, in: 2017 Eleventh International Conference on Sensing Technology (ICST), 2017, pp. 1–4.
[22] Z. Liu, S. Sarkar, Improved gait recognition by gait dynamics normalization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 863–876.
[23] A. Zhao, L. Qi, J. Dong, H. Yu, Dual channel LSTM based multi-feature extraction in gait for diagnosis of neurodegenerative diseases, Knowl.-Based Syst. 145 (2018) 91–97.
[24] M. Khokhlova, C. Migniot, A. Morozov, O. Sushkova, A. Dipanda, Normal and pathological gait classification LSTM model, Artif. Intell. Med. 94 (2019) 54–66.
[25] Y. Zhang, Y. Huang, L. Wang, S. Yu, A comprehensive study on gait biometrics using a joint CNN-based method, Pattern Recognit. 93 (2019) 228–236.
[26] L. Yao, W. Kusakunniran, Q. Wu, J. Zhang, Z. Tang, Robust CNN-based gait verification and identification using skeleton gait energy image, in: 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1–7.
[27] S. Choi, J. Kim, W. Kim, C. Kim, Skeleton-based gait recognition via robust frame-level matching, IEEE Trans. Inf. Forensics Secur. 14 (10) (2019) 2577–2592, http://dx.doi.org/10.1109/TIFS.2019.2901823.
[28] V.G.M. Guru, V.N. Kamalesh, R. Dinesh, Human gait recognition using four directional variations of gradient gait energy image, in: International Conference on Computing, Communication and Automation (ICCCA2016), 2016, pp. 1368–1371.
[29] Y. Wang, J. Sun, J. Li, D. Zhao, Gait recognition based on 3D skeleton joints captured by kinect, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3151–3155, http://dx.doi.org/10.1109/ICIP.2016.7532940.
[30] C. Yan, F. Coenen, B. Zhang, Multi-attributes gait identification by convolutional neural networks, in: 8th International Congress on Image and Signal Processing (CISP 2015), 2015, pp. 642–647, http://dx.doi.org/10.1109/CISP.2015.7407957.
[31] W. Kusakunniran, Q. Wu, H. Li, J. Zhang, Multiple views gait recognition using view transformation model based on optimized gait energy image, in: ICCV Workshops, 2010, pp. 1–6.
[32] S. Tong, Y. Fu, H. Ling, Multi-view gait identification based on stacked sparse auto-encoders, in: 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2018, pp. 57583–57596.
[33] S. Tong, Y. Fu, X. Yue, H. Ling, Multi-view gait recognition based on a spatial-temporal deep neural network, IEEE Access (2018) 1.
[34] Y. Liu, X. Jiang, T. Sun, K. Xu, 3D gait recognition based on a CNN-LSTM network with the fusion of SkeGEI and DA features, in: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8, http://dx.doi.org/10.1109/AVSS.2019.8909881.
[35] T. Huynh-The, C.-H. Hua, N.A. Tu, D.-S. Kim, Learning 3D spatiotemporal gait feature by convolutional network for person identification, Neurocomputing (2020).

[36] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, J. Feng, Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution, 2019, arXiv:1904.05049.
[37] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations (ICLR), 2017, p. 1.
[38] Z. Wu, Y. Huang, L. Wang, X. Wang, T. Tan, A comprehensive study on cross-view gait based human identification with deep CNNs, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2) (2017) 209–226.
[39] C.-B. database, CASIA-B database, 2020, http://www.cbsr.ia.ac.cn/china/Gait%20Databases%20CH.asp.
[40] SDU, SDU gait database, 2020, http://mla.sdu.edu.cn/info/1006/1195.htm.
[41] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, Y. Yagi, Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition, IPSJ Trans. Comput. Vis. Appl. 10 (4) (2018) 1–14.
[42] H. Iwama, M. Okumura, Y. Makihara, Y. Yagi, The OU-ISIR gait database comprising the large population dataset and performance evaluation of gait recognition, IEEE Trans. Inf. Forensics Secur. 7 (5) (2012) 1511–1521.
[43] google, Tensorflow library, 2020.
[44] Python, Python, 2020.
[45] G. Hinton, Visualizing high-dimensional data using t-SNE, Vigiliae Christ. 9 (2008) 2579–2605.

Aite Zhao received the Bachelor's degree in software engineering from Qingdao University of Technology in 2013, and received her Ph.D. degree in June 2020 in the College of Information Science and Engineering in Ocean University of China. She is a visiting Ph.D. researcher in the Department of Informatics, University of Leicester, Leicester, U.K. She is currently a Lecturer of College of Computer Science and Technology in Qingdao University. Her research interests include computer vision, pattern recognition, machine learning, data analysis and robotics.

Jianbo Li is currently a professor and the dean of College of Computer Science and Technology in Qingdao University. He received the bachelor and master degrees in computer science department from Qingdao University, China in 2002 and 2005, respectively, and the Ph.D. degree in computer science department from the University of Science and Technology of China in 2009. His research interests include urban computing, mobile social networks, and machine learning.

Manzoor Ahmed received the B.E. and M.S. degrees in Electrical engineering and Computer Science from Balochistan Engineering University, Khuzdar, Pakistan, in 1996 and 2010, respectively, and the Ph.D. degree in Communication and Information Systems from the Beijing University of Posts and Telecommunications, China, in 2015. From 1997 to 2000, he was a Lecturer with Balochistan Engineering University and a Telecom engineer in government-owned telecommunication service provider NTC, Pakistan, from 2000 to 2011. He was a postdoctoral researcher from the Electrical Engineering department, Tsinghua University, China, from 2015 to 2018. He is currently a faculty member with the department of Computer Science and Technology, Qingdao University. His research interests include resource allocation and offloading in vehicular communications and networking, fog and edge computing, socially aware D2D communication, physical layer security, and UAV. He has several research publications in IEEE top journals and conferences. He received several awards, including the distinction award from the President of Pakistan, the best employee award from NTC, and the best paper award from the 2014 GameNets conference.