J. Vis. Commun. Image R. 117 (2026) 104790

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Full Length Article

GGCN: Gait Recognition with Generate Network and Convolutional Neural Network☆

Hao Qin [a,1], Zhenxue Chen [a,*,1], Qingqiang Guo [a], Q.M. Jonathan Wu [b,2], Mengxu Lu [a]

[a] School of Control Science and Engineering, Shandong University, Jinan 250061, China
[b] Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada

ARTICLE INFO

Keywords:
Gait recognition
Generate network
Encoder network
Feature mapping network
Convolution neural network

ABSTRACT

Gait recognition is a biometric technology with wide application prospects, but it is easily affected by various covariates, which requires the gait recognition model is robust. In this paper, we design a robust gait recognition model named GGCN (Gait recognition with Generate network and Convolutional neural Network), which uses multi-type gait sequences as input and eliminates the effects of various covariates through a supervised mapping module. The GGCN processes the gait sequence in three steps. First, the generate network is used to extract low-level features and remove the features generated by interference. Then, the low-level features are input into the encoder network to obtain high-level features. Finally, the high-level features are input into the feature mapping network to acquire more recognizable features. The experimental results on the CASIA-B, OULP, and OUMVLP datasets demonstrate that our model outperforms current state-of-the-art methods.

1. Introduction

In recent years, more and more individual recognition methods based on biometric features have been widely applied, such as face recognition, fingerprint recognition, and gait recognition. Gait recognition identifies individuals through their walking styles [1]. Compared with faces and fingerprints, gaits can be acquired at a distance and do not require the cooperation of the object being recognized. Therefore, gait recognition is not easily detected by the subjects and has great development potential in the fields of monitoring, security, criminal investigation, and so on [2,3].

However, the main information source of gait is the image, which is easily affected by the carried objects, view, and occlusion. These factors make the intra-class distance greater than the inter-class distance in gait recognition, which limits the accuracy of gait recognition in practice. In practical applications, images are usually covered with varying degrees of occlusion, and it is very expensive to capture a walking sequence without occlusion. Therefore, the processing of images with occlusion is a challenge.

At present, there are two methods to process low-quality images in gait recognition: by extracting invariant features and by image generation. The method of extracting invariant features aims to extract features unrelated to occlusion and view through deep learning and then use these features for identification. Li et al. [4] obtained the most discriminative information to date by projecting Gait Energy Image (GEI) into different spaces. The GSP-CRC method they proposed has high computational efficiency and strong robustness against noise and data corruption. Xing et al. [5] extracted the common features of various gait sequences by a complete canonical correlation analysis (C3 A) method. Chao et al. [6] took the gait sequence as a unordered set and aggregated the feature maps of all frames in the sequence into one feature map through the SP module. The method of image generation is by using Generative Adversarial Networks (GAN) and other networks that can generate new images to map low-quality images. After mapping, the fake normal image is obtained and compared with the true normal image to complete the recognition. Xue et al. [7] proposed Frame-GAN to reduce the gap between adjacent frames, thus improving the frame rate of gait videos. Wang et al. [8] used a Two-Stream Generative Adversarial Network (TS-GAN) to map an image from any view to a standard view. Gupta [9] used Pose Energy Image (PEI) as the input to GAN to avoid the problem of input–output mismatch during training. Yu et al. [10] used two discriminators to supervise the training

☆ This paper has been recommended for acceptance by Junsong Yuan.

* Corresponding author.
E-mail addresses: 202014785@mail.sdu.edu.cn (H. Qin), chenzhenxue@sdu.edu.cn (Z. Chen), gqq@sdu.edu.cn (Q. Guo), jwu@uwindsor.ca (Q.M.J. Wu), 201500171046@mail.sdu.edu.cn (M. Lu).

[1] Both authors contributed equally.
[2] Senior Member, IEEE

https://doi.org/10.1016/j.jvcir.2026.104790
Received 4 July 2022; Received in revised form 4 January 2025; Accepted 24 March 2026
Available online 27 March 2026

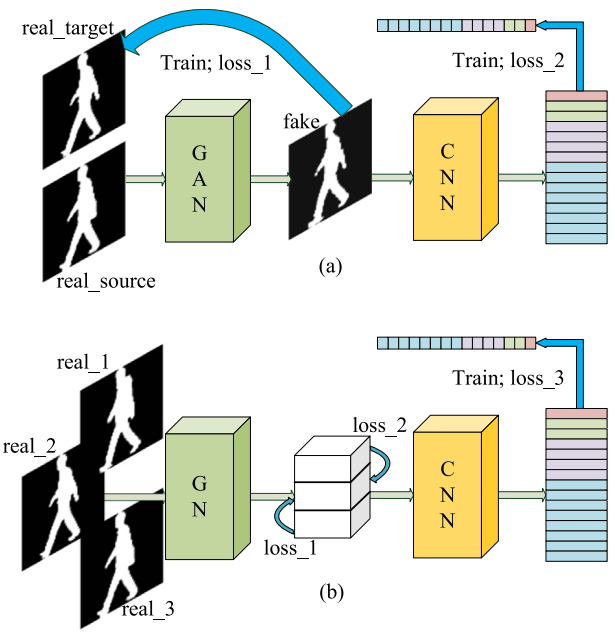

Fig. 1. (a): A schematic diagram of the application of GAN in gait recognition. **(b):** A schematic diagram of the training steps of GGCN.

of GAN, one to ensure the authenticity of the generated image and the other to ensure that the generated image contained valid information in relation to the source image. To avoid the problem of missing data, Chen et al. [11] proposed a Multi-view Gait Generative Adversarial Network (MvGGAN) that can be trained across datasets. MvGGAN extends the existing dataset and greatly improves the accuracy of gait recognition.

As a representative of image generation methods, GAN has facilitated the development of gait recognition. However, due to the swing of the legs and arms while walking, the image of each frame in a gait sequence varies greatly, so it is almost impossible to obtain two matching images as the input of GAN in practice. At the same time, the final training effect of generate network in gait recognition is judged by the accuracy of recognition, but the training of GAN and the training of recognition network are divided into two steps, which makes it difficult to adjust the model. Inspired by GAN and its problems, we propose a model named GGCN.

The generate network and recognition network are trained simultaneously in GGCN. That is to say, GGCN is an end-to-end network. As shown in Fig. 1, part (a) represents the traditional steps in the application of GAN for gait recognition [7–13]. First, the source image and target image are manually selected to train GAN. Then, low-quality images or multi-view images are input into the trained GAN to generate high-quality images or fixed-view images. Finally, the generated images are input into the recognition network as part of the dataset for identification. Part (b) represents the training steps of GGCN. First, the "source" image and "target" image are entered into generate network at the same time. And we use a supervised mapping module as the generator. Then, the feature map output by the mapping module is used as an input of the recognition network. Compared with previous work, GGCN is more robust, simpler, and easier to adjust. The framework of GGCN is shown in Fig. 2, and the details are described in Section 3. Previously, the method of extracting invariant features involved feeding all different quality gait sequences into the same feature extraction pathway, relying solely on the neural network's autonomous learning ability to extract invariant features. In contrast, by using multiple generative networks to extract features from gait sequences of varying qualities, our approach exhibits stronger rationality and interpretability.

In the practical application of gait recognition, it is difficult to obtain multiple ideal gait sequences, and it is more common to get several gait sequences with different image quality. Based on this, we set GGCN as a multi-input model. Even if there is only one actual acquired sequence, we can obtain new sequences to meet the input requirements of GGCN by simply adding interference manually. The commonly used gait datasets are CASIA-B, OULP and OUMVLP, of which only CASIA-B contains gait sequences in multiple states, and OULP and OUMVLP have relatively single states. We conduct experiments on the original dataset and the extended dataset with added interference, respectively. The experimental results and analysis are described in Section 4.

Over all, the contributions of our work are mainly as follows:

- We propose an end-to-end model called GGCN that can be used for gait recognition.
- We input multi-type gait sequences into the model to enhance model robustness and reduce the effect of covariates through a simple generate network.
- We reorganize and expand the OULP and OUMVLP datasets.
- Our model performs well on the **CASIA-B** [14], **OULP** [15], and **OUMVLP** [16] datasets. Experimental data shows that our model not only performs well on fixed datasets, but also has high accuracy on datasets with random interference.

2. Related works

In this section we will briefly introduce gait recognition, Generative Adversarial Network, and feature segmentation methods.

2.1. Gait recognition

Gait recognition methods can be divided into two types according to the types of gait features: model-based methods [17–22] and appearance-based methods [23–28].

The model-based methods usually build a human body model by input images and then take the parameters of the human body model as features. According to the dimensions of the extracted features, the model can be divided into a 2D and a 3D model. The 2D model only considers the representative features, which requires less computation but has poor fitting ability in complex cases. The 3D model can be used to model various parts of the human body and has strong robustness. Liao et al. [22] proposed PoseGait, which can extract temporal–spatial features from the 3D pose and improve the recognition rate. Combining the deterministic learning theory with the data stream of Kinect, Deng et al. [29] proposed a new model-based gait recognition method. Further, Li et al. [30] modeled the human body using the skinned multi-person linear (SMPL) model and estimated its parameters using a pre-trained human mesh recovery (HMR) network. They integrated 2D joints, 3D joints, and the contours of the human body in an end-to-end model.

Compared with the model-based method, the appearance-based method requires less calculation, but it is susceptible to interference from views and occlusion. There are two kinds of inputs for the appearance-based method, namely gait energy image (GEI) [31] and gait sequence. By aggregating a sequence into a single image, GEI can eliminate some interference but loses the timing information. Wang et al. [32] proposed frame-by-frame gait energy image (ff-GEI) to expand the amount of available GEI data and relax the limitations of current gait recognition methods on gait cycle segmentation. Gupta et al. [13] first determined the general posture in a gait cycle and then obtained Dynamic Gait Energy Image (DGEI) by calculating the gait features corresponding to this posture. They also used GAN to predict covariate-free DGEI and obtained superior identification results. Ben et al. [33] aligned gait energy images (GEIs) with the coupled bilinear discriminant projection (CBDP), reducing the impact of views on recognition rates. In recent years, with the development of deep

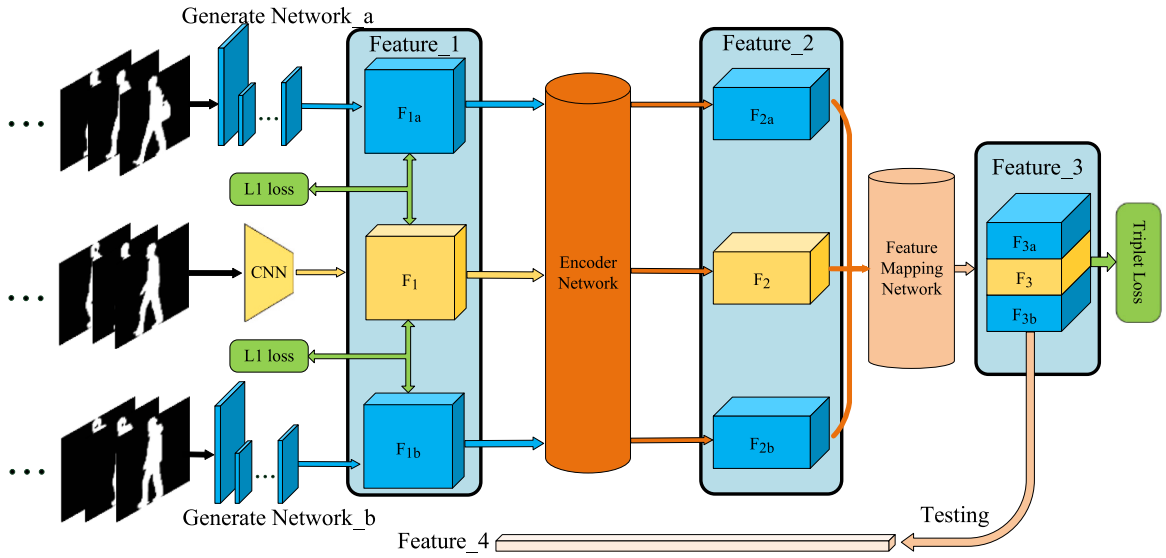

Fig. 2. The framework of GGCN. The **generate network** is a supervised mapping module whose inputs are normal gait sequences and occluded gait sequences. The occluded gait sequences can come from real walking videos or can be formed by adding interference to the normal sequences. The **encoder network** mainly consists of convolution operations and pooling operations. The **feature mapping network** mainly consists of feature cutting, splicing, and full concatenation operations.

learning, more and more models [6,27,34,35] have used gait sequence as input in order to extract more information. The accuracy of gait recognition has been improved by taking the gait sequence as a disordered set. However, too many covariables in the real scene still restrict the practical application of gait recognition. Compared to previous work, the main innovation of our method lies in using a multi-branch network architecture to handle gait sequences of different qualities. Additionally, we have introduced supervisory signals in the middle of the network to enable the model to eliminate covariates.

2.2. Generative adversarial network

GAN is a framework for estimating generative models through an adversarial process, including generator G and discriminator D submodules [36]. The training process of GAN is a game of generators and discriminators.

GAN is widely used in image generation tasks. Isola et al. [37] proposed a general solution to image-to-image translation problems based on conditional adversarial networks. Traditional GAN requires paired images as input in the training process, but it is very difficult to obtain paired images in nature. Therefore, it is necessary to solve the problem of training GAN when there are no paired images. Zhu et al. [38] proposed a translation method from the image of the source data domain X to the image of the target data domain Y. The dataset {X, Y} used in the training process is unpaired. To enhance the constraints on the translation process X→Y, the researchers added a reverse translation process Y→X and introduced Cyclic Consistency Loss to supervise the reverse translation process.

In gait recognition, GAN is mainly used to generate standard gait images [7–13,39]. GaitGANv2, proposed by Yu et al. [39] is an improvement of GaitGAN. It adopts a multi-loss strategy to optimize the network, which can increase the inter-class distance and decrease the intra-class distance. Zhang et al. [12] proposed a View Transformation Generative Adversarial Network (VT-GAN) to transform the gait image from any two views. They input both the source image and the target view into the model and used a view discriminator to ensure that the generated image was in the target view.

Currently, GAN can generate very realistic gait images with various covariates, but visually "realistic" images are not necessarily conducive to the training of Convolutional Neural Networks. Therefore, instead of using GAN to generate gait images, we use the generate network to generate feature maps. The details are described in Section 3.

2.3. Feature segmentation methods

Segmenting the features or images can yield more fine-grained features and improve feature recognizability. In the field of person re-identification (ReID), there are many models that improve recognition accuracy by feature segmentation [40–43]. For example, Fu et al. [43] proposed a Horizontal Pyramid Matching (HPM) method that reduces the negative effects of missing partial information in the image and provides more robust features for the ReID task.

Feature segmentation also performs well in the field of gait recognition. In order to extract more discriminative gait features, Zhang et al. [27] proposed a robust and effective loss function called angle center loss (ACL) and used a spatial transformer network to locate the suitable horizontal part of the body. Fan et al. [34] thought that each part of the human body should have its own separate feature map, so they divided the feature map into pieces and proposed the model GaitPart. Different parts of the body contribute differently to recognition under different conditions. Wu et al. [35] first calculated the weight of each part of the body through a condition-aware module, then adjusted the feature map based on the weight and finally recognized the identity according to the feature similarity. These methods show the importance of local body information in gait recognition, so our recognition network takes full account of local body information and integrates the relationship between adjacent body parts into features. The details are described in Section 3.

3. Method

In this section, we first introduce a general description of GGCN and the detailed operation of each submodule in GGCN. Then, we introduce our optimization objectives. Finally, we introduce the methods of training and testing the model. The overall pipeline is illustrated in Fig. 2.

3.1. Problem formulation

Converting gait sequences (as in Fig. 3) into recognizable features is an important part of gait recognition. If the gait sequence is G_i and the recognizable feature is $Feature_3$, then our work can be expressed as:

$$Feature_3 = GGCN(G_i) \tag{1}$$

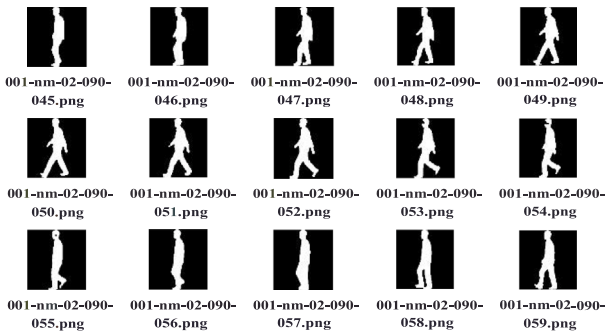

Fig. 3. A set of gait silhouettes from **CASIA-B**.

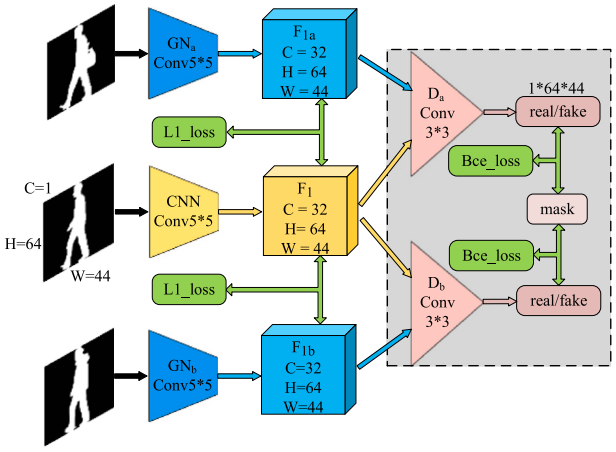

Fig. 4. Details of the generate network we use. The gray part represents the discriminator, which is not included in our final model. We only show the processing procedure of an image and the dimensions of its feature map. In the actual training process, the feature map will have one more "batch size" dimension.

where $G_i = \{g_i | i = 1, 2, \ldots, n\}$, g_i represents a frame in the gait sequence, n represents the number of input frames, and *GGCN* represents our proposed model. *GGCN* processes G_i in three steps.

(1) According to the input gait sequence, the corresponding low-level feature $Feature_1$ is generated through generate network:

$$Feature_1 = GN(G_i) \tag{2}$$

where *GN* represents the generate network and $Feature_1$ contains simple contour information and detail information.

(2) $Feature_1$ is input to the encoder network to obtain the high-level features $Feature_2$:

$$Feature_2 = EN(Feature_1) \tag{3}$$

where *EN* represents the encoder network and $Feature_2$ contains abstract body information.

(3) To make the features more discriminative, we input $Feature_2$ to the feature mapping network to obtain $Feature_3$:

$$Feature_3 = FMN(Feature_2) \tag{4}$$

where *FMN* represents the feature mapping network. After we obtain $Feature_3$, the similarity between the two gait sequences can be transformed into the Euclidean distance between the corresponding $Feature_3$.

3.2. Generate network

There are three types of gait images in the CASIA-B dataset: # NM, # BG, and # CL. These correspond to no occlusion, simple occlusion, and complex occlusion in real life. Since only normal images are available in the OULP and OUMVLP datasets, we produced simple occluded images and complex occluded images based on the original images. We designed a GAN-based generate network and simplified it to a simple mapping module (as in Fig. 4). Next, the generate network will be described in detail.

Definition: The generate network that we use is a supervised mapping module with an input of three gait sequences and an output of the corresponding three low-level features. Unlike traditional GANs that use images as the source and target domains, we use feature maps as the source and target domains.

Motivation: The gait sequences used for recognition may have various occlusions. In order to eliminate the influence of occlusions, we hope that the network can automatically ignore various occlusions when generating low-level features. That is, the low-level features of obscured sequences and the unobscured sequences of the same individual should be as similar as possible.

Operation: We input the three gait sequences into three convolutional layers:

$$F_{1a} = GN_a(G_i^a) \tag{5}$$

$$F_1 = CNN(G_i) \tag{6}$$

$$F_{1b} = GN_b(G_i^b) \tag{7}$$

where G_i^a, G, and G_i^b represent, respectively, the gait sequence with simple occlusion, the gait sequence without occlusion, and the gait sequence with complex occlusion. *CNN* is a convolutional layer, which is used to extract edge information from input images. GN_a and GN_b play the role of generator, which can not only extract low-level features of the input image, but also eliminate occlusion in the image. Our generator is also a convolutional layer, and its detailed architecture is shown in Table 1. We monitor this process through Mean Absolute Error (MAE, L1_loss):

$$L_{ga} = L(F_{1a}, F_1) = \frac{1}{N}\sum_{i=1}^{N} |F_{1a_i} - F_{1_i}| \tag{8}$$

$$L_{gb} = L(F_{1b}, F_1) = \frac{1}{N}\sum_{i=1}^{N} |F_{1b_i} - F_{1_i}| \tag{9}$$

where N represents the number of feature points in F_{1a} and F_{1a_i} represents the value of the ith feature point.

Initially, we designed a complete generative adversarial network as the generator part of GGCN, trained by a adversarial process of min–max gaming. As shown in Fig. 4, D_a is used to identify the validity of F_{1a} and D_b is used to identify the validity of F_{1b}. During the training, we considered F_1 as "real" and F_{1a} and F_{1b} as "fake". We monitored this process through Binary Cross Entropy Loss (BCE_loss):

$$L_{da} = L(real_a, mask_r) + L(fake_a, mask_f) \tag{10}$$

$$L(real_a, mask_r) = mean(\{l_1, l_2, \ldots, l_M\}^T) \tag{11}$$

$$l_i = -[y_i \cdot \log x_i + (1 - y_i) \cdot \log(1 - x_i)] \tag{12}$$

where $real_a$ represents the output of D_a when the input is F_1, and $mask_r$ represents the tensor of value 1 with the same size as $real_a$. $fake_a$ represents the output of D_a when the input is F_{1a}, and $mask_f$ represents the tensor of value 0 with the same size as $fake_a$. y_i represents the value of the ith point in $mask_r$, and x_i represents the value of the ith point in $real_a$. L_{db} is calculated in the same way as L_{da}.

The initial model is complex and difficult to train, so we pruned it to obtain the GGCN with simple structure and easy training. The final GGCN has no discriminator and no adversarial process, and is no longer related to GAN, but its accuracy is comparable to that of the original complex model. The specific experimental data is presented in Section 4.

Table 1
Detailed structure of our generator. "in_c", "out_c" and "act_func" stand for "in channel", "out channel" and "activation function", respectively.

type	kernel	stride	padding	bias	in_c	out_c	act_func
Conv	5	1	2	False	1	32	LeakyRELU

Table 2
Detailed structure of our encoder network. 'FConv' represents Focal Convolution.

Operation	Details	Output shape
Input	$T \times C \times H \times W$	30 × 32 × 64 × 44
layer 1	FConv[32, 3, 3, 64] -LeakyRELU	30 × 64 × 64 × 44
layer 2	FConv[64, 3, 3, 64] -LeakyRELU	30 × 64 × 64 × 44
Downsample	MaxPool2D[2, 2]	30 × 64 × 32 × 22
layer 3	FConv[64, 3, 3, 128] -LeakyRELU	30 × 128 × 32 × 22
layer 4	FConv[128, 3, 3, 128] -LeakyRELU	30 × 128 × 32 × 22
Downsample	MaxPool2D[2, 2]	30 × 128 × 16 × 11

3.3. Encoder network

The structure of the encoder network is shown in Table 2. F_{1a}, F_1, and F_{1b} share a common encoder network, but there is no interaction between their features. Next, the encoder network will be described in detail.

Definition: A coding network consisting of convolutional operations and pooling operations.

Motivation: $Feature_1$ only contains simple edge features, which cannot provide effective information for gait recognition. Therefore, we use a deeper convolutional network to extract more abstract high-level features.

Operation: 'FConv' in Table 2 represents Focal Convolution [34], which can be used to extract fine-grained features. The feature map is horizontally sliced before the convolution operation, and then the convolution operation is performed on each feature block separately. The pooling operation, by contrast, is performed on the whole feature map. The first two and the last two Focal Convolutions divide the feature map into 4 and 8 blocks, respectively. The size of each feature map and the parameters of the convolution and pooling operations are indicated in Table 2.

3.4. Feature mapping network

We use a Part Feature Relationship Extractor (**PFRE**) as the feature mapping network, the structure of which has been described in detail in [28]. Here, we will only briefly describe what each submodule does and how they are connected. The structure of the feature mapping network is shown in Fig. 5.

Definition: The feature mapping network is a network that can transform features from low identification space to high identification space. It mainly includes operations such as cutting, splicing, and fully connecting of features.

Motivation: In gait recognition, intra-class spacing is often greater than inter-class spacing. The features obtained only by convolution of gait images are susceptible to interference and cannot be used as an effective basis for recognition. Therefore, we want to obtain more feature information to assist gait recognition. There is a certain correlation between various parts of the human body during walking, and we use the feature mapping network to extract this correlation.

Operation: As shown in Fig. 5, the feature mapping network is composed of two parts: Adjacent Feature Relation Extractor (**AFRE**) and Total-Partial Feature Extractor (**TPFE**). We copy $Feature_2$ and enter it into two submodules:

$$F_{2A} = AFRE(Feature_2) \tag{13}$$

$$F_{2T} = TPFE(Feature_2) \tag{14}$$

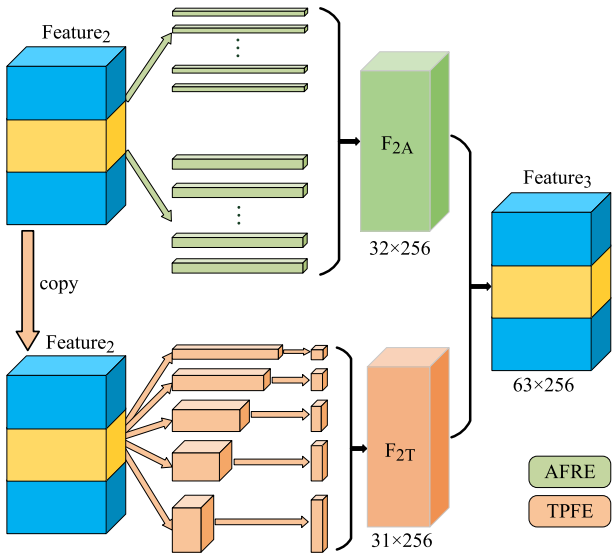

Fig. 5. The structure of the feature mapping network. The green part represents the **AFRE** module, and the yellow part represents the **TPFE** module. F_{2A} represents the output of AFRE, F_{2T} represents the output of TPFE, and the two are spliced together to obtain $Feature_3$. The numbers under the feature map indicate the dimension.

where F_{2A} contains the relationship between adjacent body parts and fine-grained features. F_{2T} contains the overall information and local information of different scales.

$Feature_3$ is obtained by splicing F_{2A} and F_{2T}:

$$Feature_3 = \{F_{2A}, F_{2T}\} \tag{15}$$

where "{}" means splicing operation. TPFE extracts multi-scale fine-grained features through the use of a feature pyramid, while AFRE simultaneously mines fine-grained features and captures the dependency relationships between adjacent feature blocks. The combination of both methods allows for the extraction of more comprehensive and robust gait features.

3.5. Optimization objective

We use the Batch All (BA+) triple loss function [44] to supervise the optimization process of the whole model. Let the input of the triple loss function be a triple as $r = (\alpha, \beta, \gamma)$, where α represents the anchor, β represents the positive sample with the same label as α, and γ represents the negative sample with a label different from α. Then, the BA+ triple loss is going to be:

$$L_r = max(D_{\alpha\beta} - D_{\alpha\gamma} + \xi, 0) \tag{16}$$

where $D_{\alpha\beta}$ represents the Euclidean distance between α and β, $D_{\alpha\gamma}$ represents the Euclidean distance between α and γ, and ξ is a hyperparameter.

By combining the L1_loss and BCE_loss mentioned above, we can get the full optimization objective:

$$L = \lambda_{ga}L_{ga} + \lambda_{gb}L_{gb} + \lambda_{da}L_{da} + \lambda_{db}L_{db} + L_r \tag{17}$$

where λ is a hyperparameter that controls the importance of different loss functions in the optimization process. For the initial model, we use L_{da}, L_{db} and L_r as loss functions. For the final model, we use L_{ga}, L_{gb} and L_r as the loss functions. We also try to supervise the model using the full loss function. The optimal value of each λ is discussed in Section 4.

Table 3
Average rank-1 accuracies on **CASIA-B** with cross conditions. The accuracies of all models except GGCN are extracted from their papers. GGCN(initial) represents a complex model containing GAN as the generator. GGCN(Ours) represents the final concise model. It can be seen that good results can be achieved by using only a simple generate network.

Gallery NM #1-4		0°-180°											mean
Probe		0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	
NM #5-6	CNN-Ensemble [45]	88.7	95.1	98.2	96.4	94.1	91.5	93.9	97.5	98.4	95.8	85.6	94.1
	GaitSet [6]	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 [34]	94.1	98.6	99.3	98.5	94.0	92.3	95.9	98.4	99.2	97.8	90.4	96.2
	Wu et al. [35]	90.6	97.9	98.5	97.6	93.9	89.7	94.4	98.1	97.8	96.2	87.9	94.8
	MvGGAN [11]	94.8	**99.0**	**99.7**	**99.2**	96.6	93.7	96.3	**98.6**	99.2	98.2	92.3	97.1
	GGCN(initial)	95.2	98.7	99.4	98.6	**96.9**	94.8	96.2	97.8	**99.3**	98.3	**92.4**	97.03
	GGCN(Ours)	**96.6**	**99.0**	99.6	98.6	96.6	**95.3**	**97.2**	98.4	**99.3**	**98.5**	92.0	**97.35**
BG #1-2	CNN-LB [45]	64.2	80.6	82.7	76.9	64.8	63.1	68.0	76.9	82.2	75.4	61.3	72.4
	GaitSet [6]	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 [34]	89.1	94.8	96.7	95.1	88.3	84.9	89.0	93.5	96.1	93.8	85.8	91.5
	Wu et al. [35]	84.0	91.8	94.8	92.7	85.7	82.1	86.7	91.7	94.4	91.9	81.3	88.8
	MvGGAN [11]	92.4	94.7	97.2	94.6	88.7	83.6	87.8	93.8	96.3	95.2	86.8	91.9
	GGCN(initial)	92.9	96.6	96.8	**95.9**	92.5	87.7	90.7	94.3	96.7	**95.6**	87.5	93.39
	GGCN(Ours)	**95.0**	**97.8**	**97.9**	95.7	**93.8**	**88.4**	**92.4**	**95.9**	**97.5**	**95.6**	**88.5**	**94.40**
CL #1-2	CNN-LB [45]	37.7	57.2	66.6	61.1	55.2	54.6	55.2	59.1	58.9	48.8	39.4	54.0
	GaitSet [6]	61.4	75.4	80.7	77.3	72.1	70.1	71.5	73.5	73.3	68.4	50.0	70.4
	GaitPart [34]	70.7	85.5	86.9	83.3	77.1	72.5	76.9	82.2	83.8	80.2	66.5	78.7
	Wu et al. [35]	75.9	87.5	**90.6**	**85.3**	81.5	76.5	**81.1**	**86.5**	**85.5**	82.1	66.8	81.8
	MvGGAN [11]	70.5	77.9	82.5	82.7	77.4	73.6	73.8	77.8	77.6	72.5	64.8	75.6
	GGCN(initial)	77.1	85.5	87.6	82.7	79.8	75.6	78.1	79.6	83.1	80.7	69.2	79.93
	GGCN(Ours)	**77.3**	**89.2**	89.8	84.5	**81.8**	**76.6**	80.0	84.1	84.6	**83.0**	**71.9**	**82.07**

Fig. 6. Schematic of the extended OULP dataset. The first to fourth rows represent 'Seq00', 'Seq01', 'Seq02', and 'Seq03', respectively, where 'Seq02' and 'Seq03' are generated from 'Seq01'.

3.6. Training and testing

Training: For the CASIA-B dataset, in each iteration, 6 sequences are randomly selected from # NM, 2 sequences are randomly selected from # BG, and 2 sequences are randomly selected from # CL of a subject. The correlation sequences of four subjects are selected at a time to calculate the loss value. Thus, the input of GGCN in the training phase can be expressed as 4×10.

For the OULP and OUMVLP datasets, we extend 'Seq02' and 'Seq03' (as shown in Fig. 6) from 'Seq01' by manually adding interference. We randomly select 4 sequences from 'Seq00', 'Seq01', 'Seq02', and 'Seq03', respectively. Four subjects are selected at a time. Thus, the input can be expressed as 4×16.

Testing: In the testing phase, each feature is expanded into a one-dimensional tensor. Then, the Euclidean distance between the probe feature and each gallery feature is calculated, and the feature in the gallery with the closest distance to the probe feature is considered to be from the same subject as the probe. Finally, the recognition labels are compared with the real labels to calculate the recognition accuracy.

4. Experiments

In this section, the proposed method is evaluated on several public gait datasets: CASIA-B, OULP, and OUMVLP. We will present the dataset details and parameter selection and compare our method with state-of-the-art works. Then, the effectiveness of each submodule is verified by ablation experiments. Finally, the impact of the weight of three loss functions on the model is discussed.

4.1. Datasets and training details

CASIA-B [14]: The CASIA-B dataset contains 124 subjects and is one of the most widely used gait datasets available. The walking sequences of each subject can be divided into 10 types according to what the subject is wearing at the time of filming, namely, NM1-NM6, BG1-BG2, and CL1-CL2. Among them, NM1-NM6 are filmed under normal conditions, BG1-BG2 are filmed under the condition of backpack, and CL1-CL2 are filmed under the condition of wearing a coat. Each type contains 11 sequences from different views (0°, 18°, …,162°, 180°). In the test phase, NM01-NM04 are used as the gallery set, while NM05-NM06, CL01-CL02, and BG01-BG02 are used as the probe set.

OULP [15]: The OULP dataset consisted of 4007 subjects, including 2135 men and 1872 women ranging in age from 1 to 94 years. Each subject's gait sequences are divided into two folders, 'Seq00' and 'Seq01', each containing four sequences from different views (55°, 65°, 75°, 85°). Seq00 is used as a gallery set, while Seq01 is used as a probe set. Compared with CASIA-B, OULP contains fewer covariates but includes a larger number of subjects. Therefore, the experiment on OULP dataset is evidence of the model's generalization ability.

OUMVLP [16]: The OUMVLP dataset consists of 10,307 subjects, including 5114 men and 5193 women ranging in age from 2 to 87 years. Each subject's gait sequences are divided into the two folders 'Seq00' and 'Seq01', each containing fourteen sequences from different views (0°, 15°, …, 90°, 180°, 195°, …, 270°). The use of OUMVLP is becoming more and more widespread due to the large number of views and subjects it contains.

Training Details: For CASIA-B, we take the first 74 subjects as the training set and the remaining 50 subjects as the test set. The size of the image in CASIA-B is reduced to 64×44 by the method in [6]. For OULP, we reduce the image size to 64×44 and expand the probe set

Table 4
Average rank-1 accuracies on **OULP** with cross conditions. 'Seq01', 'Seq02', and 'Seq03' represent the three probe sets, respectively.

Condition		Seq01					Seq02					Seq03				
Probe view	Method	Gallery view					Gallery view					Gallery view				
		55°	65°	75°	85°	mean	55°	65°	75°	85°	mean	55°	65°	75°	85°	mean
55°	GaitSet [6]	99.6	99.7	99.8	99.2	99.6	99.6	99.7	99.7	98.9	99.5	99.3	99.7	99.3	98.2	99.1
	GaitPart [34]	99.5	99.8	99.8	98.8	99.5	99.5	99.8	99.8	98.7	99.5	99.5	99.8	99.6	98.6	99.4
	Ours	99.7	99.9	99.9	99.6	**99.8**	99.6	99.9	99.9	99.6	**99.7**	99.7	99.9	99.9	99.5	**99.7**
65°	GaitSet [6]	99.4	99.5	99.6	99.3	99.4	99.3	99.4	99.7	99.3	99.4	99.1	99.2	99.4	98.9	99.2
	GaitPart [34]	99.3	99.6	99.7	99.1	99.4	99.3	99.6	99.6	99.0	99.3	99.1	99.6	99.6	98.6	99.2
	Ours	99.6	99.8	99.8	99.8	**99.7**	99.6	99.7	99.7	99.7	**99.7**	99.4	99.7	99.7	99.6	**99.6**
75°	GaitSet [6]	98.9	98.9	99.3	99.4	99.1	98.9	99.1	99.1	99.2	99.1	98.4	98.9	99.0	99.4	98.9
	GaitPart [34]	99.1	99.5	99.5	99.4	99.4	99.2	99.5	99.6	99.3	99.4	98.9	99.4	99.7	99.4	99.3
	Ours	99.4	99.5	99.8	99.4	**99.5**	99.5	99.5	99.8	99.5	**99.6**	99.3	99.4	99.6	99.5	**99.4**
85°	GaitSet [6]	98.4	99.1	99.8	99.8	99.3	98.5	99.1	99.6	99.7	99.2	98.2	99.1	99.6	99.8	99.1
	GaitPart [34]	98.5	99.5	99.7	99.9	99.4	98.5	99.3	99.7	99.8	99.3	98.2	99.2	99.7	99.9	99.2
	Ours	99.3	99.7	100	99.9	**99.7**	99.2	99.7	100	99.9	**99.7**	98.9	99.5	100	100	**99.6**
	Ours	99.5	99.7	99.9	99.7		99.5	99.7	99.9	99.7		99.3	99.6	99.8	99.7	

into three folders (see Fig. 6). Seq02 is formed by randomly adding 150 interference points with the size of 1 × 1 to Seq01, and Seq03 is formed by randomly adding 40 interference points with the size of 3 × 3 to Seq01. The coordinates of the upper left and lower right corners of the range where interference can be added are (15,15) and (48,55) respectively, and the interference values are randomly selected from 0, 86, 192, 255. Seq02 adds a lot of disturbance points, but the real gait profile is not affected too much. We use Seq02 as a simple occlusion probe set. In Seq03, the features of the arm and part of the knee are difficult to obtain, so we use Seq03 as a complex occlusion probe set. For OUMVLP, the method in [6] is used to reduce the image size to 64 × 44; other processing is the same as for OULP.

The optimizer chooses Adam [46], with a learning rate of 1e-4 and a momentum of 0.9. The margin in the BA+ triple loss is set to 0.2. λ_{g1} is set to 0.01, and λ_{g2} is set to 0.1. λ_{d1} and λ_{d2} are set to 0. Thirty randomly selected frames from each gait sequence are used as the input of GGCN during the training phase. Our experimental platform is Ubuntu 16.04, and the models are trained with 2 NVIDIA 2080TI GPUS. We trained our model with 160 K iterations on CASIA-B, 40 K iterations on OULP and 300 K iterations on OUMVLP.

4.2. Comparison with the state-of-the-art

CASIA-B: Table 3 shows the experimental results of other state-of-the-art models and GGCN. The accuracies of all models except GGCN are extracted from their papers. Data for all conditions (NM, BG, CL) are obtained by testing the same trained model. In the test phase, both the probe set and the gallery set contain eleven views, allowing us to obtain an accuracy matrix with the size of 11 × 11. After averaging the accuracy matrix across the gallery view dimension, we get the accuracy under each probe view shown in Table 3.

The data in Table 3 shows that our model achieves the highest mean accuracy in all conditions. The greatest improvement in accuracy is achieved in the BG condition compared to the previous best work, which indicates that our approach of using generate network to remove occlusions from images is effective. The accuracy in the NM condition is already high, and the dataset contains some unrecognizable images, so there is not much improvement. In the CL condition, although our model reaches the highest accuracy, it is still significantly lower than that in the NM condition. We believe that there are three reasons for this: **(1)** The coat on the subject obscures many leg features; **(2)** the dataset is produced with some invalid images mixed in, with the most invalid images in the CL condition; and **(3)** we use generative network at the low-level features of the model, which is good at eliminating simple occlusion but has limited ability to handle complex occlusion.

OULP: According to the official setting [47,48], we use CV01 as the training set and CV02 as the test set. The accuracy matrix obtained in the test phase is shown in Table 4. For the convenience of comparison, the three models in Table 4 use exactly the same hyperparameters in the training phase.

Our model achieves the highest accuracy in all three conditions. Viewing the data in Table 4 horizontally, it can be found that the accuracy is almost unaffected by random interference. That is, under the condition that the original information is valid, our model can well resist the influence of some random disturbances in nature.

OUMVLP: Due to the expansion of the dataset and the limitation of computing resources, we randomly select 1000 subjects as the training set and 1000 subjects as the test set. The experimental results are shown in Table 5. The last two columns of data are calculated on the original dataset, and the other data is calculated on the dataset with added interference. It can be seen that even the most advanced recognition networks are affected by interference, while our model is almost unaffected due to the presence of generate network.

4.3. The effectiveness of each submodule

In this part, we use experimental data and visual images to demonstrate the effectiveness of each submodule in GGCN.

Experimental data: To verify that each submodule plays its role and that the model has no redundancy, we designed a set of ablation experiments (Group A). The experimental results are shown in Table 6. It can be seen that the accuracy will decrease regardless of which submodule is missing. In the experiment A-b, a convolution layer and a pooling layer are added to the model after the encoder network is removed so as to ensure that the dimensions of the feature map match the feature mapping network. We think that these two layers support the role of the encoder network.

The results of experiment A-b are similar to those of experiment A-c. This is mainly because both generate network and the encoder network are essentially composed of convolutional layers and can accomplish the task of extracting image features. When the feature mapping network is removed (A-a), the accuracy of the model decreases greatly. This shows that it is difficult to complete the gait recognition task when only using convolution operation to find visually similar images.

Visual images: In practical applications, each subject's gait features occupy a certain position in Euclidean space. When the number of subjects is large, the positions occupied by each feature will overlap, which increases the recognition error rate. Therefore, we want to be able to accommodate as many subject features as possible within a fixed size of Euclidean space, with clear demarcation lines between each subject's features.

In order to investigate the clustering ability of our model for gait features, we use t-SNE to visualize the features generated at each stage

Table 5
Average rank-1 accuracies on **OUMVLP** with cross conditions. The last two columns of data are calculated on the original dataset, and the other data is calculated on the dataset with added interference.

Probe		0°	15°	30°	45°	60°	75°	90°	180°	195°	210°	225°	240°	255°	270°	mean
Seq01	GaitSet [6]	64.0	73.3	87.4	86.6	83.1	83.8	81.7	64.5	77.8	89.4	88.2	85.2	84.8	82.6	80.88
	GaitPart [34]	63.9	73.7	87.1	86.9	83.8	84.3	82.1	64.3	78.6	89.5	88.1	86.4	85.6	83.2	81.25
	Ours	**73.1**	**80.5**	**92.8**	**91.6**	**89.1**	**89.9**	**89.0**	**73.3**	**85.3**	**94.5**	**92.4**	**91.2**	**90.8**	**89.5**	**87.35**
Seq02	GaitSet [6]	63.1	72.3	86.5	86.1	82.6	83.0	80.9	64.0	76.4	88.6	87.5	85.0	84.5	82.1	80.17
	GaitPart [34]	63.1	73.0	86.8	86.5	83.1	83.5	81.2	63.7	78.0	88.8	87.6	85.7	84.9	82.7	80.61
	Ours	**72.7**	**80.2**	**92.3**	**91.3**	**88.5**	**89.7**	**88.8**	**73.2**	**85.0**	**94.2**	**92.2**	**90.8**	**90.5**	**89.3**	**87.05**
Seq03	GaitSet [6]	60.3	68.1	83.9	82.9	80.0	80.9	78.5	61.2	72.2	86.1	85.5	82.8	82.3	79.7	77.47
	GaitPart [34]	60.8	69.5	84.5	84.5	80.9	81.9	79.3	61.7	75.0	86.3	85.7	84.1	83.8	80.6	78.43
	Ours	**70.8**	**78.5**	**90.2**	**90.4**	**86.7**	**88.2**	**87.0**	**70.9**	**83.3**	**93.1**	**91.4**	**89.6**	**89.3**	**87.5**	**85.55**
GaitSet [6]		79.5	87.9	89.9	90.2	88.1	88.7	87.8	81.7	86.7	89.0	89.3	87.2	87.8	86.2	87.1
GaitPart [34]		82.6	88.9	90.8	91.0	89.7	89.9	89.5	85.2	88.1	90.0	90.1	89.0	89.1	88.2	88.7

Table 6
Ablation study, Group A. Experimental results when removing each submodule from GGCN.

Group A	GN	EN	FMP	NM	BG	CL
a	✓	✓		75.12	62.02	38.54
b	✓		✓	96.88	92.21	79.73
c		✓	✓	96.41	92.88	80.52
d	✓	✓	✓	**97.35**	**94.40**	**82.07**

Table 7
Ablation study, Group B. Experimental results when the loss functions have different weights.

Group B	λ_{ga}	λ_{da}	λ_{gb}	λ_{db}	NM	BG	CL
a	0.01	0.01	0.1	0.1	97.29	94.03	81.86
b	0	0.01	0	0.1	97.03	93.39	79.93
c	1	0	0.1	0	97.00	93.00	80.56
d	0.01	0	10	0	96.94	93.64	79.74
e	0.01	0	0.1	0	97.35	**94.40**	**82.07**
f	0	0	0	0	97.01	93.25	81.38
g	0	0	0.1	0	**97.48**	94.34	81.48
h	0.1	0	0	0	97.08	93.59	81.07
i	0	0	0	0.1	96.98	93.63	81.54
j	0	0.1	0	0	97.16	94.11	80.94

of the model (see Fig. 7). We randomly select 100 sequences from 10 subjects and input these sequences into the trained GGCN model. We visualize the features output from the generate network, the encoder network, and the feature mapping network as (a), (b), and (c) in Fig. 7, respectively. By observing the axes of (a) and (b), it can be found that the encoder network reduces the Euclidean space occupied by a subject's gait features. By observing the distribution of midpoints in (b) and (c), it can be found that the feature mapping network changes the state of features from indivisible to separable, which proves the effectiveness of the feature mapping network.

In order to show the change process of feature maps in GGCN more intuitively, we visualized the output feature maps of the generate network and the encoder network. The output of the feature mapping network has no height and width dimensions, and visualization of it is meaningless. As shown in Fig. 8, $Feature_1$ contains simple contour information and detail information, and $Feature_2$ contains abstract body information.

To further examine the generative network's capability in handling perturbed signals, we visualized the feature maps of both the original gait sequence and the gait sequence with added perturbations. The experimental results are shown in Fig. 9. The first column represents the undisturbed normal gait sequence, and the second column depicts its corresponding feature map. The third column displays the perturbed gait sequence obtained by introducing disturbances to the normal gait, while the fourth column shows its feature map. It can be observed that despite the random perturbations disrupting the gait appearance, our generative network still extracts high-quality feature maps that remain largely unaffected.

4.4. Discussion of loss function

Based on Eq. (17), there are five weight coefficients to be confirmed. We set the weight of triple loss to the fixed value 1 and obtained the experimental data in Table 7 by changing the other four coefficients (Group B).

B-b shows the experimental results of the initial model using GAN as a generator, where L_{da} and L_{db} are used to supervise the training of the GAN and L_r is used to supervise the whole model. The longitudinal comparison shows that the initial model is not very accurate, but requires a complex adversarial learning step. B-c, B-d and B-e show the experimental results of the final model using the supervised mapping module as a generator. Since the two generators process different types of images, their weights should also be different. It can be seen from experiments B-c, B-d, and B-e that the optimal ratio of λ_{ga} to λ_{gb} is about 1:10.

B-a shows the experimental results of supervising the training process of the model using all loss functions, and there is no adversarial process during the training process, and the parameters of all modules are updated simultaneously. Comparing B-a and B-e, we can see that without the adversarial training process, the "discriminator" has degenerated into a mapping module, and whether it is trained or not has almost no effect on the overall model.

B-f denotes using only the triplet loss function to supervise the model. By observing B-g, B-h, B-i, and B-j, it can be seen that the four loss functions have different impacts on the model under various conditions. Among them, when using only L_{gb}, the model achieves the optimal accuracy in the NM condition.

4.5. Cross dataset results

To further investigate the robustness of our approach, we conducted cross-dataset experiments: training on the OUMVLP dataset and testing on the CASIA-B dataset. The experimental results, as shown in Table 8, demonstrate that even in cross-domain scenarios, our model can achieve optimal performance. With the presence of a disturbance-eliminating generation module, our method exhibits significant advantages in the CL condition.

4.6. Ablation study of using multi-type gait sequences

GGCN is a model that accepts multi-type inputs, which enhances its robustness. In order to investigate whether the incorporation of multi-type gait sequences indeed has a positive impact on the model, we conducted ablation experiments. The experimental results are presented in

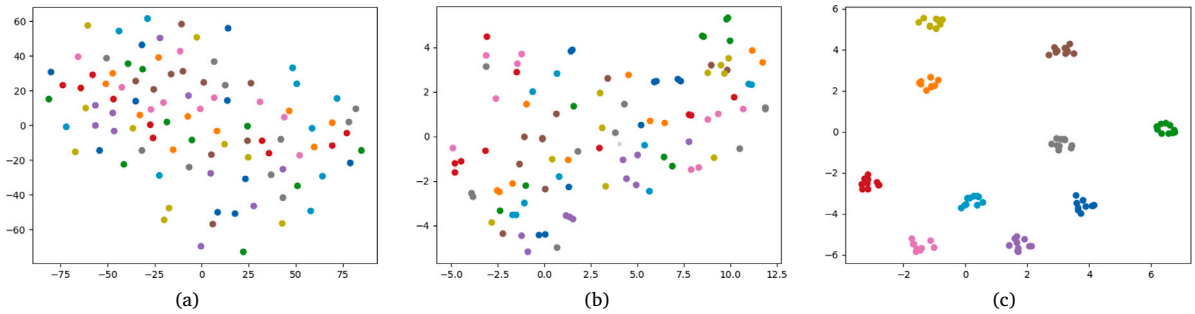

Fig. 7. The t-SNE visualization results of the feature maps at each stage in our model. Dots of the same color indicate features from the same subject. (a) represents the output of generate network, (b) represents the output of the encoder network, and (c) represents the output of the feature mapping network.

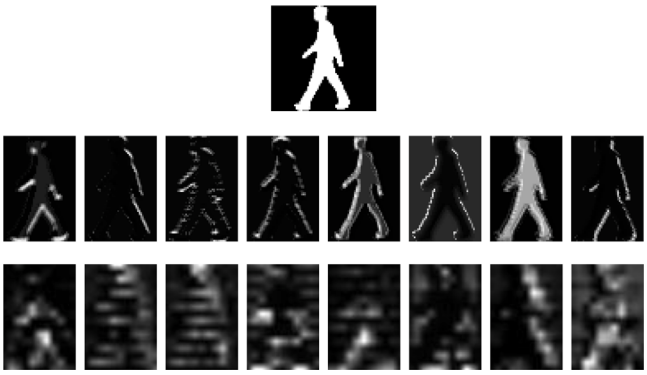

Fig. 8. Schematic diagram of the feature map visualization. The first row is a frame of the original video, the second row is the result of $Feature_1$ visualization, and the third row is the result of $Feature_2$ visualization.

Table 8
Ablation study, Group C. Cross-dataset experiments with different models: training on OUMVLP and testing on CASIA-B.

model	NM	BG	CL	mean
GaitSet	76.25	65.19	46.65	62.70
GaitPart	74.93	63.00	42.21	60.05
GGCN	**77.41**	**69.75**	**54.09**	**67.28**

Table 9
Ablation study, Group D. Ablation study of using multi-type gait sequence on CASIA-B. "Multi-type input" indicates that the gait sequences in NM, BG, and CL conditions are used as input, and "single-type input" indicates that the gait sequences in NM condition are used only.

input	NM	BG	CL	mean
multi-type input	97.35	**94.40**	**82.07**	**91.27**
single-type input	**98.66**	88.74	31.11	72.84

Table 9. It can be observed that single-type inputs slightly improved the recognition accuracy for that specific type, but significantly lowered the model's performance in other scenarios. This indicates that overly relying on a single input type will decrease the model's robustness.

4.7. Discussion about the limitations

Our method outperforms current state-of-the-art methods, but there is still significant room for exploration and some limitations, which we analyze here. Inputting various types of samples can enhance the model's robustness, and there are many possibilities for sample expansion methods. In this paper, to demonstrate the generality of the approach, we employed a method of adding random perturbations to

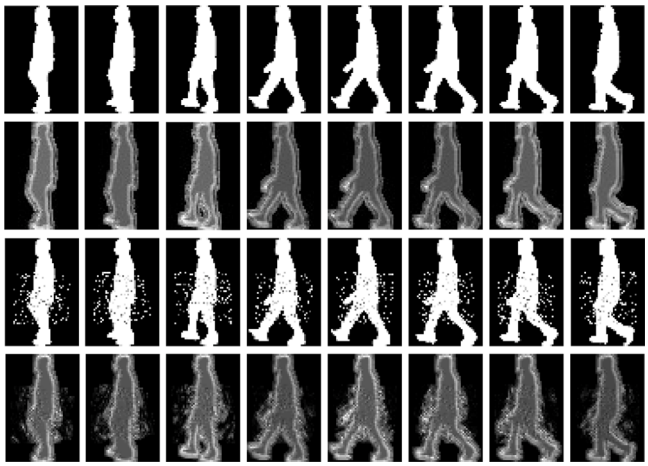

Fig. 9. Visualization of the feature map outputted by the generate network. From top to bottom: original gait sequence, feature map of the original gait sequence, gait sequence with added perturbation, and feature map of the perturbed gait sequence. It can be observed that our generative network effectively reduces the impact of perturbations.

expand the samples. A sample expansion method that better conforms to gait motion patterns may further enhance the model's performance. Additionally, the supervision signal for our generative model comes from shallow convolutional neural networks, which might lead to some limitations when dealing with extreme image cases. Using more complex supervision signals can improve the model's robustness in extreme scenarios. Finally, our model is a three-input structure, which could restrict the model's computational efficiency on some low-power devices.

5. Conclusion

In this paper, we propose a gait recognition method that combines a Generate Network with a Convolutional Neural Network and design a gait recognition model called GGCN. GGCN was designed inspired by GAN, which automatically completes the coupling between generate network and recognition network through the neural network and can adjust the model more intuitively compared with the existing methods using GAN. Unlike traditional methods, the output of the generate network in our model is a feature map rather than an image. Then, the feature map is processed by the encoder network and the feature mapping network into a tensor that can be easily identified. Through feature segmentation and feature fusion, our feature mapping network extracts local information from various body parts and captures the collaborative relationships between these parts, thereby enhancing the model's recognition accuracy. Experiments on CASIA-B, OULP, and

OUMVLP indicate that GGCN achieves the highest accuracy when compared with other state-of-the-art methods, and GGCN performs well even in the presence of interference. We also demonstrate the effectiveness of each submodule through ablation experiments. GGCN improves robustness by inputting multi-type sequences and reduces the effect of covariates by a simple supervised mapping module. We believe that, after refining this method, GGCN may be able to obtain good results in other fields, such as person re-identification.

CRediT authorship contribution statement

Hao Qin: Writing – original draft. **Zhenxue Chen:** Project administration. **Qingqiang Guo:** Project administration. **Q.M. Jonathan Wu:** Writing – review & editing. **Mengxu Lu:** Formal analysis.

Declaration of competing interest

We declare that we have no conflict of interest.

Acknowledgments

This work was supported in part by the Key R&D Project of Shandong Province, China (2022CXGC010503), in part by the National Key R&D Program of China (2019YFB1311000) and in part by the National Natural Science Foundation of China (61876099). Hao Qin and Zhenxue Chen contributed equally to this work and should be considered as the co-first authors.

Data availability

No data was used for the research described in the article.

References

[1] H.M. Thang, V.Q. Viet, N.D. Thuc, D. Choi, Gait identification using accelerometer on mobile phone, in: 2012 International Conference on Control, Automation and Information Sciences, 2012, pp. 344–348.
[2] P.K. Larsen, E.B. Simonsen, N. Lynnerup, Gait analysis in forensic medicine, J. Forensic Sci. 53 (5) (2008) 1149–1153.
[3] I. Bouchrika, M. Goffredo, J. Carter, M. Nixon, On using gait in forensic biometrics, J. Forensic Sci. 56 (4) (2011) 882–889.
[4] W. Li, C.C.J. Kuo, J. Peng, Gait recognition via GEI subspace projections and collaborative representation classification, Neurocomputing 275 (2018) 1932–1945.
[5] X. Xing, K. Wang, T. Yan, Z. Lv, A complete canonical correlation analysis with application to multi-view gait recognition, Pattern Recognit. 50 (2016) 107–117.
[6] H. Chao, K. Wang, Y. He, J. Zhang, J. Feng, GaitSet: Cross-view gait recognition through utilizing gait as a deep set, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 1.
[7] W. Xue, H. Ai, T. Sun, C. Song, Y. Huang, L. Wang, Frame-GAN: Increasing the frame rate of gait videos with generative adversarial networks, Neurocomputing 380 (2020) 95–104.
[8] Y. Wang, C. Song, Y. Huang, Z. Wang, L. Wang, Learning view invariant gait features with two-stream GAN, Neurocomputing 339 (2019) 245–254.
[9] S.K. Gupta, Reduction of covariate factors from silhouette image for robust gait recognition, Multimedia Tools Appl. (2021) 1–26.
[10] S. Yu, H. Chen, E.B. Garcia Reyes, N. Poh, Gaitgan: Invariant gait feature extraction using generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 30–37.
[11] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, Q. Tian, Multi-view gait image generation for cross-view gait recognition, IEEE Trans. Image Process. 30 (2021) 3041–3055.
[12] P. Zhang, Q. Wu, J. Xu, VT-GAN: View transformation GAN for gait recognition across views, in: 2019 International Joint Conference on Neural Networks, 2019, pp. 1–8.
[13] S.K. Gupta, P. Chattopadhyay, Gait recognition in the presence of co-variate conditions, Neurocomputing 454 (2021) 76–87.
[14] S. Yu, D. Tan, T. Tan, A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition, in: 18th International Conference on Pattern Recognition, vol. 4, 2006, pp. 441–444.
[15] 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.
[16] 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 (1) (2018) 1–14.
[17] J. Gu, X. Ding, S. Wang, Y. Wu, Action and gait recognition from recovered 3-D human joints, IEEE Trans. Syst. Man Cybern. B 40 (4) (2010) 1021–1033.
[18] 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.
[19] D. Kastaniotis, I. Theodorakopoulos, C. Theoharatos, G. Economou, S. Fotopoulos, A framework for gait-based recognition using kinect, Pattern Recognit. Lett. 68 (2015) 327–335.
[20] Y. Feng, Y. Li, J. Luo, Learning effective gait features using LSTM, in: 2016 23rd International Conference on Pattern Recognition, 2016, pp. 325–330.
[21] F. Jean, A.B. Albu, R. Bergevin, Towards view-invariant gait modeling: Computing view-normalized body part trajectories, Pattern Recognit. 42 (11) (2009) 2936–2949.
[22] R. Liao, S. Yu, W. An, Y. Huang, A model-based gait recognition method with body pose and human prior knowledge, Pattern Recognit. 98 (2020) 107069.
[23] L. Wang, T. Tan, H. Ning, W. Hu, Silhouette analysis-based gait recognition for human identification, IEEE Trans. Pattern Anal. Mach. Intell. 25 (12) (2003) 1505–1518.
[24] I. Rida, X. Jiang, G.L. Marcialis, Human body part selection by group lasso of motion for model-free gait recognition, IEEE Signal Process. Lett. 23 (1) (2016) 154–158.
[25] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, N. Wang, Gait recognition via disentangled representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4710–4719.
[26] X. Ben, P. Zhang, Z. Lai, R. Yan, X. Zhai, W. Meng, A general tensor representation framework for cross-view gait recognition, Pattern Recognit. 90 (2019) 87–98.
[27] Y. Zhang, Y. Huang, S. Yu, L. Wang, Cross-view gait recognition by discriminative feature learning, IEEE Trans. Image Process. 29 (2020) 1001–1015.
[28] H. Qin, Z. Chen, Q. Guo, Q.M.J. Wu, M. Lu, RPNet: Gait recognition with relationships between each body-parts, IEEE Trans. Circuits Syst. Video Technol. (2021) 1.
[29] M. Deng, C. Wang, Human gait recognition based on deterministic learning and data stream of microsoft kinect, IEEE Trans. Circuits Syst. Video Technol. 29 (12) (2019) 3636–3645.
[30] X. Li, Y. Makihara, C. Xu, Y. Yagi, S. Yu, M. Ren, End-to-end model-based gait recognition, in: Proceedings of the Asian Conference on Computer Vision, 2020.
[31] J. Han, B. Bhanu, Individual recognition using gait energy image, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2) (2006) 316–322.
[32] X. Wang, W.Q. Yan, Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory, Int. J. Neural Syst. 30 (01) (2020) 1950027.
[33] X. Ben, C. Gong, P. Zhang, R. Yan, Q. Wu, W. Meng, Coupled bilinear discriminant projection for cross-view gait recognition, IEEE Trans. Circuits Syst. Video Technol. 30 (3) (2020) 734–747.
[34] C. Fan, Y. Peng, C. Cao, X. Liu, S. Hou, J. Chi, Y. Huang, Q. Li, Z. He, Gaitpart: Temporal part-based model for gait recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14225–14233.
[35] H. Wu, J. Tian, Y. Fu, B. Li, X. Li, Condition-aware comparison scheme for gait recognition, IEEE Trans. Image Process. 30 (2021) 2734–2744.
[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[37] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[38] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.
[39] S. Yu, R. Liao, W. An, H. Chen, E.B. García, Y. Huang, N. Poh, GaitGANv2: Invariant gait feature extraction using generative adversarial networks, Pattern Recognit. 87 (2019) 179–189.
[40] S. Li, S. Bak, P. Carr, X. Wang, Diversity regularized spatiotemporal attention for video-based person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 369–378.
[41] W. Li, X. Zhu, S. Gong, Harmonious attention network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2285–2294.
[42] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, Q. Tian, Deep representation learning with part loss for person re-identification, IEEE Trans. Image Process. 28 (6) (2019) 2860–2871.
[43] Y. Fu, Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao, T. Huang, Horizontal pyramid matching for person re-identification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 8295–8302.

[44] A. Hermans, L. Beyer, B. Leibe, In defense of the triplet loss for person re-identification, 2017, ArXiv Preprint ArXiv:.07737.
[45] 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) (2016) 209–226.
[46] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6908.
[47] D. Muramatsu, Y. Makihara, Y. Yagi, View transformation model incorporating quality measures for cross-view gait recognition, IEEE Trans. Cybern. 46 (7) (2016) 1602–1615.
[48] D. Muramatsu, Y. Makihara, Y. Yagi, Cross-view gait recognition by fusion of multiple transformation consistency measures, IET Biom. 4 (2) (2015) 62–73.

Hao Qin was born in Shandong, China, in 1998. He received the B.S. degree in automation from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2020. He is currently working toward the M.S. degree in control science and engineering at the School of Control Science and Engineering, Shandong University, Jinan, China. His research interests include machine learning, deep learning, and gait recognition.

Zhenxue Chen was born in Shandong, China, in 1977. He received the B.S. degree in automatic from School of Electrical Engineering and Automation at Shandong Institute of Light Industry, Jinan, China, in 2000, the M.S. degree in computer science from School of Information Science and Engineering at Wuhan University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in pattern recognition and intelligent systems from Institute of Image Recognition and Artificial Intelligence at Huazhong University of Science and Technology, Wuhan, China, in 2007. From 2012 to 2013, he was a visiting scholar with the Michigan State University, East Lansing, Michigan, USA. He is currently a professor with the School of Control Science and Engineering, Shandong University. His main areas of interest include image processing, pattern recognition, and computer vision, with applications to face recognition. He has published over 100 papers in refereed international leading journals/conferences such as IEEE T-II, IEEE T-CSVT, IEEE T-IFS, IEEE T-VT, IEEE T-ITS, Information Sciences, Neurocomputing, Neural Computing and Applications, and SP-IC, etc.

Qingqiang Guo was born in Shandong, China, in 1971. He received the Ph.D. degree in Control Science and Engineering form School of Control Science and Engineering at Shandong University, Jinan, China, in 2010. Now, he is an associate professor with the School of Control Science and Engineering, Shandong University. His research interests include production scheduling, process control and optimization, system analyze and identification, and computer vision.

Q.M. Jonathan Wu (M'92-SM'09) received the Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990. He was with the National Research Council of Canada for ten years from 1995, where he became a Senior Research Officer and a Group Leader. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. He has published more than 300 peer-reviewed papers in computer vision, image processing, intelligent systems, robotics, and integrated microsystems. His current research interests include machine learning, 3-D computer vision, video content analysis, interactive multimedia, sensor analysis and fusion, and visual sensor networks. Dr. Wu holds the Tier 1 Canada Research Chair in Automotive Sensors and Information Systems. He was Associate Editor for IEEE Transactions on Systems, Man, and Cybernetics Part A, and the International Journal of Robotics and Automation. Currently, he is an Associate Editor for the IEEE Transaction on Neural Networks and Learning Systems and the journal of Cognitive Computation. He has served on technical program committees and international advisory committees for many prestigious conferences.

Mengxu Lu was born in Jiangsu, China, in 1997. She received the B.S. degree in automation from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2019. She is currently working toward the M.S. degree in control science and engineering at the School of Control Science and Engineering, Shandong University, Jinan, China. Her research interests include machine learning, deep learning, and semantic segmentation.