PAPER Special Section on Computational Intelligence and Big Data for Scientific and Technological Resources and Services

CJAM: Convolutional Neural Network Joint Attention Mechanism in Gait Recognition

Pengtao JIA[†a)], ***Nonmember***, **Qi ZHAO**[†b)], ***Member***, **Boze LI**[†], ***and*** **Jing ZHANG**[†], ***Nonmembers***

SUMMARY Gait recognition distinguishes one individual from others according to the natural patterns of human gaits. Gait recognition is a challenging signal processing technology for biometric identification due to the ambiguity of contours and the complex feature extraction procedure. In this work, we proposed a new model - the convolutional neural network (CNN) joint attention mechanism (CJAM) - to classify the gait sequences and conduct person identification using the CASIA-A and CASIA-B gait datasets. The CNN model has the ability to extract gait features, and the attention mechanism continuously focuses on the most discriminative area to achieve person identification. We present a comprehensive transformation from gait image preprocessing to final identification. The results from 12 experiments show that the new attention model leads to a lower error rate than others. The CJAM model improved the 3D-CNN, CNN-LSTM (long short-term memory), and the simple CNN by 8.44%, 2.94% and 1.45%, respectively.

key words: *image classification, gait recognition, deep learning, convolutional neural networks, attention mechanism*

1. Introduction

Biometric recognition, which identifies individuals using biometric features, is an essential field of computer vision. Biometric recognition includes iris recognition [1], fingerprint recognition [2], face recognition [3] and gait recognition [4]. In law enforcement scenarios in which suspects are being arrested, it is difficult for the police and related institutions to obtain the fingerprints or facial features of strangers. However, with the help of security cameras, the gait video of suspects cab be obtained effortlessly. Hence, gait recognition, as a new biometric method, requires more research. Gait recognition also has potential applications in fields such as person re-identification, visual surveillance and object detection. Nevertheless, since the contours of humans may be blurred and images may appear stumpy and poorly defined, recognizing an individual by gait presents more difficulties than other recognition methods.

Gait recognition is a major public problem, and the crucial points focus on utilizing the contour information and temporal information among the former and latter frames in a sequence. There are two sorts of approaches for gait recognition: traditional methods and neural network methods. Traditional methods often use manually notated features to complete the identification. Therefore, the traditional process is a painstaking and time-consuming task that requires manual preprocessing to extract gait features and cannot effectively use gait information. Neural network methods could extract gait features automatically, although the accuracy of recognition depends on the structure of the neural network. Although there are abundant images in a person's gait sequence, neural networks can solve the challenging task by distinguishing a person automatically and effortlessly. A key frame in a gait sequence has two kinds of pixels: important pixels and unimportant pixels. The unfit and useless images, which are named "unimportant" pixels, may be very similar to other people's contours. The unimportant pixels cannot show a person's feature and, as a result, they may have serious impacts on the predictive power of a neural network model. The vital points of data are named "important" pixels since they can show the gait features effectively. Common convolutional neural network models cannot eliminate the influence of unimportant information on the images in a sequence, which may lead to a bad result. Therefore, we proposed a model, named the convolutional neural network joint attention mechanism (CJAM), that joins an attention mechanism and a convolutional neural network (CNN) together, and it is depicted in Fig. 1. This model can be used to discern between "important" and "unimportant" pixels.

A CNN is adept at extracting useful features from the initial input, and an attention mechanism is essentially a

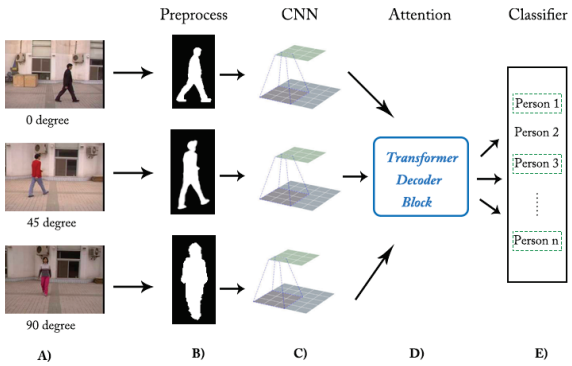

Fig. 1 CJAM for gait recognition. A) The input is the key frames of a sequence of RGB-D videos. B) The raw pixels are preprocessed, including graying, binarization and compression. C) Binary images are passed through the CNN to obtain the gait signatures. D) An attention model gives various values to different parts. E) A final decision is formed by a softmax classifier and an identification result is output.

Manuscript received November 11, 2020.
Manuscript revised March 14, 2021.
Manuscript publicized April 28, 2021.
[†]The authors are with the Xi'an University of Science and Technology, China.
a) E-mail: jiapengtao@xust.edu.cn
b) E-mail: zhaoqiovo@gmail.com
DOI: 10.1587/transinf.2020BDP0010

weight assignment that allows a computer to focus the picture in a manner similar to that of human eyes. Hence, the model has the ability to ignore those "unimportant" pixels in a sequence and set high weights for those "important" pixels. The CJAM extracted features of the initial gait images' contours with the CNN and then used the attention mechanism to recognize those features and finalize the classification.

The rest of this paper structured as follows. In Sect. 2, the related works and background knowledge section, we introduce the formal research studies on gait recognition, including traditional methods and neural network methods, and provide basic background knowledge of the attention model. In Sect. 3, the CJAM model, including a CNN encoder for feature extraction and an attention model for classification, that is used for image sequence classification is explained. The experimental results and analysis are presented in the experiment section, which is Sect. 4 of this paper. In Sect. 5, we conclude our studies and present an expectation about the field of gait recognition and potentially related areas.

2. Related Works and Background Knowledge

2.1 Related Works

Studies associated with gait recognition began later but have produced promising results for challenging classification problems. In terms of the feature extraction approaches, gait recognition studies can be divided into two stages: the traditional stage and the neural network stage. The traditional stage was based on methods that extract gait features manually while neural network methods have become more popular, especially when deep learning was introduced. The neural network methods auto extract gait features using neural networks and then implement classification. In this part, we will introduce traditional methods and then concentrate on studies based on neural networks for gait recognition.

Traditionally, there are three main types of established traditional methods used in image classification tasks. It is well known that the performance of any learning algorithm is heavily dependent on the choice of the data representation. The first approach to recognize gait sequences depends on some traditional models. For example, Kale et al. applied a hidden Markov model to human body contours to classify humans [5]. Derltka M and Bogdan M ensembled the kNN classifiers for human gait recognition and reached the 97.3% accuracy [42]. Sharma et al. applied the artificial neural networks for gait recognition and compared the performance with BPNN (Back Propagation Neural Network), that ANN performance of the recognition method depends significantly on the quality of the extracted binary silhouettes [43]. The second way approach classifies the gait sequence using manually extracted features. Han and Bhanu first took the mean contour of the whole gait cycle, called the gait energy image (GEI) [6], as an effective feature. The GEI is basically equal to the average silhouette over one gait cycle. The GEI, or the average silhouette representation, has been widely adopted due to its simplicity and effectiveness. The GEI saves both computation time and storage space. Wang et al. represented feature information with human body contour column mass vectors [7] and used a support vector machine to conduct recognition and improve the accuracy. These authors used principal component analysis to reduce the dimensionality of the input feature space and to extract the unique gait features. More recently, Chen et al. proposed a gait recognition method based on tracking the center of gravity [8]. The third is using fused features, which are built to make up for the deficiency that a single feature might lose gait information, for gait recognition. For example, Li et al. applied canonical correlation analysis after the gait features were fused for classification and recognition [9]. Chai el al. introduced the dynamic region variance feature (DRV) [10] to describe the motion of body parts, and then feature-level and decision-level strategies were respectively used to fuse three types of features.

Although traditional methods have made significant achievements for the task of gait recognition, these methods cannot extract feature automatically, while neural network could deal those problems easily. The neural methods are introduced below.

Lecun was the first to use the backpropagation algorithm on the convolution neural network and improved the performance of the CNN so that it could be applied in practical work [11]. However, due to the limitations of hardware computing abilities and other constraints, neural networks were not given more attention. In 2006, Hinton and others proposed a method that could train deep belief networks fast [12], and they published an article in "Science" [13], which opened up the field of deep learning. In 2012, Krizhevsky et al. won the ImageNet championship. These authors obtained a particularly good result with deep convolutional neural networks [14], piquing the interests of industry and academics in deep learning.

Convolutional neural networks have become a better way to extract the spatial information of images since these researcher studies were conducted. Furthermore, a CNN was adopted to carry out image recognition and other related aspects of this work.

Many more studies on gait recognition using deep convolutional neural networks have been carried out [15], [16]. Wu and his research group trained a general deep convolutional neural network to recognize the most discriminative changes in a human's identity, and the method achieved a state-of-the-art 94% recognition rate only under the condition that the cross-view angle was no less than 36 degrees [17]. An empirical comparison of the 2D-CNN, 3D-CNN and ResNet using the CASIA-B dataset was conducted by Castro et al. [18], and the results show that multimodel feature fusion could achieve the best image classification. The accuracy and performance of the algorithm mostly depends on the neural network architecture, but it could achieve much better results than using human eyes to examine the various model structures to identify gaits.

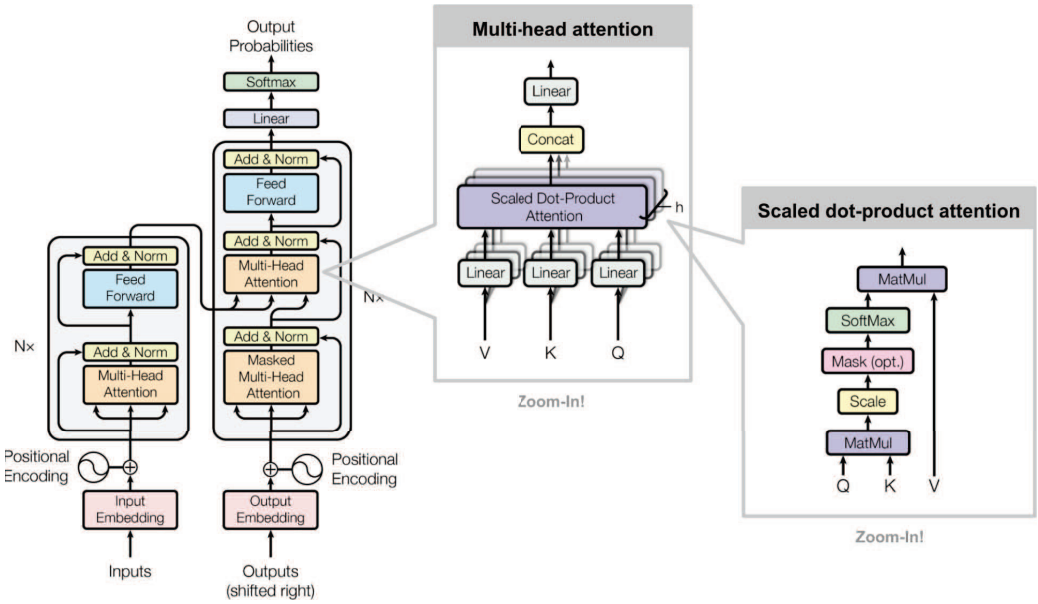

Fig. 2 The transformer structure proposed at [29].

Most convolutional neural networks are 2D networks specializing in two-dimensional problems with spatial relations such as image classification. However, the common deep learning methods used to recognize image sequences are the 3D-CNN or CNN-LSTM. The 3D convolution has the ability to combine temporal and spatial information to handle the relevant multiframe pictures, so it could be used to address image sequence recognition tasks, such as motion and gait recognition. In 2012, Ji et al. first proposed the 3D convolutional neural network to resolve the problem of human motion recognition [19]. In addition, in 2014, Karpathy et al. applied the 3D convolution to changed frames of videos [20]. In 2016, the Google DeepMind team applied a 3D convolution neural network called LipNet to lip reading at the sentence level [21]. In addition, Thomas Wolf et al. applied the 3D convolution neural network to the field of multiview gait recognition and achieved remarkable results [22].

Unlike convolutional neural networks, recurrent neural networks (RNNs) are more suitable for dealing with time series problems, such as generating text or speech recognition, but RNNs also perform well in computer vision tasks. While a general RNN unit cannot eliminate the problem of long-distance dependence, Jürgen Schmidhuber et al. proposed LSTM [23], which can prevent the vanishing and exploding gradients caused by RNN training over a long distance. Once LSTM was proposed, many scholars applied it to diverse fields. For example, Fei Li et al. used the bidirectional LSTM model to classify relationships [24]. Yi et al. used the bidirectional GRU (a variant of LSTM) to study Chinese classical poetry and achieved excellent results [25]. To achieve better performance and accuracy, researchers also joined an ordinary 2D CNN and RNN together to address video sequence-related tasks. Donahue et al. first put a CNN and LSTM together to present a new model for visual recognition and description [26]. Medel et al. used a CNN-LSTM network architecture for the first time to achieve the automatic prediction of video sequences [27]. Recently, Zhang Z et al. proved that the CNN-LSTM could achieve better performance than a simple CNN model [28]. The attention model, which was proposed by Google in 2017, has been proved to achieve significant results in time series problems. In this paper, a combined CNN and attention model was proposed for gait recognition, and satisfactory results were obtained by our experiments.

2.2 Attention Model of Transformer

The attention mechanism has been widely used in natural language processing. Vaswani et al. 2017 introduced a model called a transformer [29] that used self-attention. Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence.

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

The inputs of the attention encoder consist of a set of key-value pairs (K, V), including the keys of dimension d_k and values of dimension d_v. In the decoder, the transformer computes the dot products of the queries with all keys, divides each by $\sqrt{d_k}$, and applies a softmax function to obtain the weights on the values and pack a set of queries together into a matrix Q. The keys and values are also packed together into matrices K and V. The transformer adopts the scaled dot-product attention. The outputs can be computed using Eq. (1).

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Rather than only computing the attention once, the multihead mechanism computes the scaled dot-product attention multiple times in parallel, as shown in Fig. 2. The queries, keys and values are linearly projected h times with different, learned linear projections to d_k, d_k, and d_v dimensions, respectively. For each of these projection versions of queries, keys and values, the multihead attention performs the scaled dot-product attention function in parallel. These independent attention outputs are concatenated and transformed into expected dimensions, resulting in the final values that are calculated using Eq. (2).

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \ldots, head_h) \\ where\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^f) \end{aligned} \quad (2)$$

In the transformer, the encoder and the decoder are both stacked to N=6 sublayers, and the multihead mechanism runs h=8 parallel scaled dot-product attention layers. For each of these layers, the dimensions of the keys, values and output $d_k = d_v = \frac{d_{model}}{h} = 64$. Each encoder attention sublayer is a self-attention layer with a fully connected feed-forward network. All of the keys, values and queries come from the same input.

This paper proposed a new CJAM structure, based on the experience of predecessors and the combination of a CNN and an attention model, to conduct gait recognition. By conducting comparison experiments between the CJAM structure, the 3D-CNN and the CNN-LSTM, we analyzed the performance of various network structures on the gait recognition task.

3. CNN Joint Attention Mechanism

3.1 The Description of CNN Joint Attention Mechanism

The attention mechanism has been largely adopted in the field of language modeling, but too little work has been devoted to the application of the attention model in image sequence tasks. Since the transformer is designed for machine translation, which is a sequence-to-sequence classification problem, it cannot work for simple image sequence classification, such as gait recognition. Therefore, by modifying the transformer from machine translation, we proposed a new attention model for image sequence classification and applied to gait recognition. We altered the transformer by eliminating its decoder and adding a randomly initialized decoder query vector. The CJAM uses the CNN model to transform the initial input image sequences into feature vectors since the input of the attention model must be vectors. The attention model in the CJAM also has an encoder for encoding sequence information and a decoder for decoding the information. The CNN model and the attention model are illustrated in detail in the following sections.

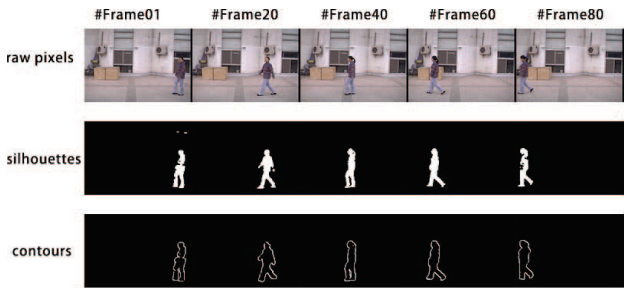

Fig. 3 Samples that some key frames of continuous gait sequence in CASIA-A.

3.2 CNN Model for Feature Extraction

The CNN structure is used to extract image features so that each image will be transformed into a vector after the CNN model.

A human's gait sequence, as shown in Fig. 3, contains many continuous image frames. In our CASIA-A experimental dataset, the number of key frames of each sequence ranges from 27 to 127. Because the temporal complexity to process an original sequence is a bit longer, we divided it into several segments instead of using the whole original sequence as the input, and each segment becomes a shorter subsequence of the original sequence. The length of the subsequence is a crucial factor affecting the accuracy and efficiency of the model. If the length is too short, the model may not learn sufficient information; but if the length is too long, the training procedure will be very slow. However, the most relevant features for gait sequence frames are only relevant with a short temporal range and are not actually associated with other temporally distant frames. In our approach, the frame sequences overlap within the test or training set as in Wolf et al. 2016 [22]. Considering both the performance and the accuracy, we set the length of each subsequence at a moderate length of 5 frames. Therefore, the CNN accepted 5 preprocessed gait images at a time and encoded them via several layers. Assuming that there are 75 frames in our initial gait sequence, the initial gait sequence would be divided into clips that have five frames each, such as frame (1-5), (6-10), (11-15), ...(71-75). Using the five frames of every subsequence in the same initial long sequence, we calculate the mean value of the class probabilities as the final prediction of the test set and use the max probability class as the exact class recognized by the CJAM.

The encoded vectors of images would be transmitted to the attention model in temporal order. Using encoded vectors as the attention mechanism's input, we can finish the classification task using the CJAM.

The architecture of the CNN in the CJAM is shown in Fig. 4, and the description of the CJAM is given in the following paragraph.

The 1st layer: The input layer of the CJAM accepts data with the format of [batch size, 5, 120, 176]. The first parameter, the batch size, is the size of batch in the training step. The "5" indicates that the number of neighboring im-

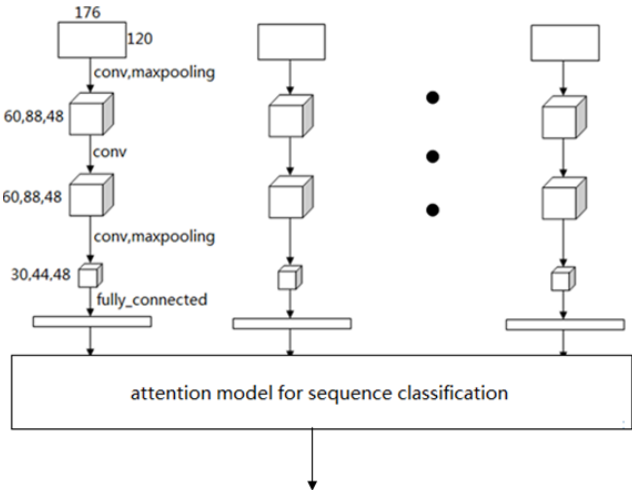

Fig. 4 Convolutional neural network for gait feature extraction. The input image is our preprocessed silhouettes gait photos, which compressed to 176×120 pixels. Linear CNN with three 2D convolutions, two max pooling, and 1 fully connected layers and output the gait feature to attention model for sequence classification.

ages in temporal order is 5. The "120" and the "176" represent the size of the compressed images (see the details in the experimental section), which has transformed into 120×176 matrices. Then, the input data would be transformed into the format of [batch size×5, 120, 176, 1], which was convenient to for the CNN to calculate as its input.

The 2nd layer: The second layer is a convolutional layer with a filter size of 3×3 and a stride size of 1, and it outputs 48 feature maps.

The 3rd layer: The third layer conducts the max pooling operation, where the pooling window size and the stride size are both 2.

The 4th and the 5th layers: The fourth and fifth layers are two additional convolutional layers, and the parameters of both are consistent with those of the first layer.

The 6th layer: The sixth layer is the second pooling layer, and its operation is equal to that of the third layer.

The 7th layer: The seventh layer is a fully connected layer that is accountable for transmuting the previous 6th layer's output into vectors with a length of 512. After this transformation, the CNN model modifies the input with the format of [batch size, 5, 120, 176] into the output with a format of [batch size×5, 512]. It can be seen that the input images have been encoded into vectors with a length of 512 using the CNN.

All of the layers make up the entire CNN model architecture in the CJAM. The attention structure of the CJAM was designed similar to that of the CNN architecture, which will be described in Sect. 3.3.

3.3 Attention Model for Sequence Classification

In this work, a special attention model combined with the CNN was designed to classify sequences, and the model is depicted in Fig. 5.

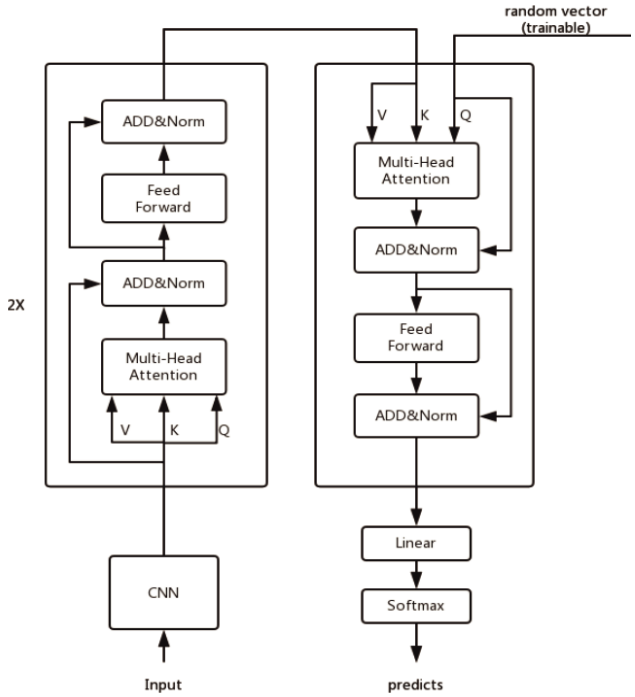

Fig. 5 Attention model for sequence classification. We take advantage of N=2 identical layer as the encoder and our decoder contains only one layer.

The difference between attention model in the CJAM and the transformer in Google is that the CJAM's encoder inputs are the CNN's outputs instead of word embeddings being used. Simply, the CJAM take advantage of N=2 identical layers as the encoder. The two encoder attention sublayers both could be considered as a self-attention layer with a fully connected feedforward network. The decoder of the CJAM contains only one layer. In the transformer, all of the keys, values and queries come from the same place. However, in the CJAM, the keys and values of the decoder come from the output of the encoder, and the queries come from a randomly initialized vector, which can be modified during training. Furthermore, the attention model in the CJAM adopted a residual connection around the encoder and the decoder, followed by layer normalization.

Similarly and conveniently, the CJAM adopted the multihead attention, which is described in Sect. 2.2. In this work, the CJAM has 8 parallel attention layers, and $d_k = d_v = \frac{d_{model}}{h} = 64$ in each of these layers.

After the decoder, the output would make a transformation with a linear operation and then acquire the predictions using a softmax function since gait recognition must handle hundreds and thousands of classes.

4. Experiment and Discussion

4.1 Dataset

We had tested the CJAM model using CASIA dataset A [31] and CASIA dataset B [32], which were collected by the In-

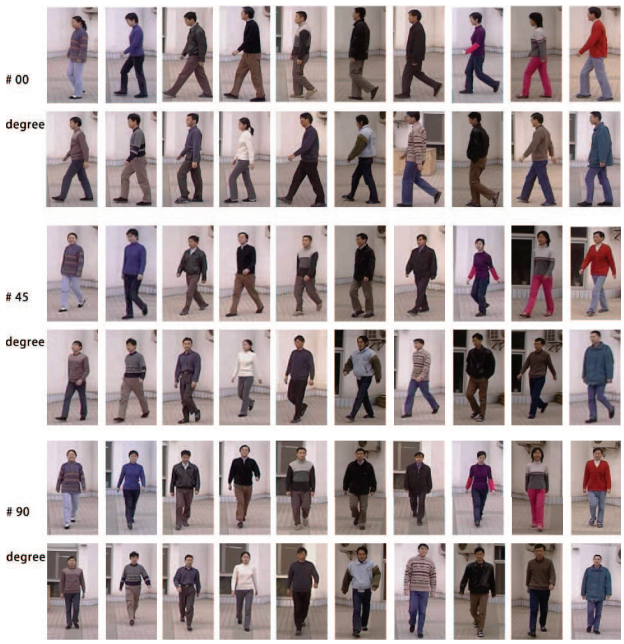

Fig. 6 Samples of CASIA-A. The dataset contains 20 people's gaits, each person has 12 image sequences in three different moving directions (00, 45, 90 degree).

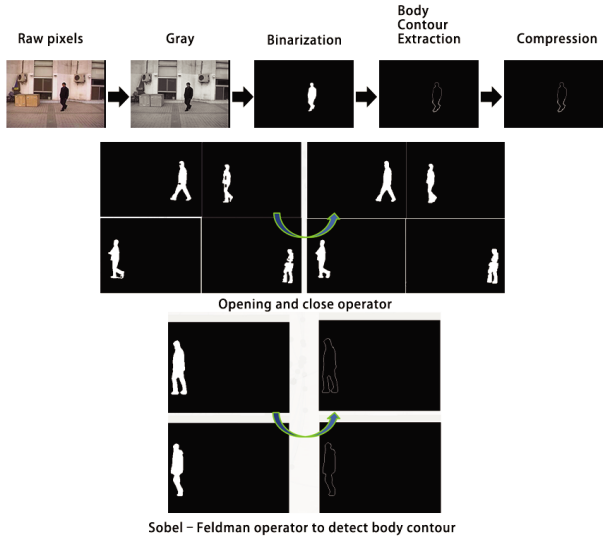

Fig. 7 The procedure of data preprocessing, which include 5 functional model, the transformation from RGB to gray image, the binarization, image morphology processing, the extraction of body contours and the compression.

telligent Recognition & Digital Security Group. CASIA dataset A was constructed and released in 2001. As shown in Fig. 6, this dataset contains 20 people's gait data, and each person has 12 image sequences in three different moving directions. The sequences are 0, 45, and 90 degrees to the image plane, respectively, with 4 sequences for each direction. The length of the sequences depends on the different moving speeds. For each sequence, we divided the original sequence into several subsequences that have 5 different frames. Namely, the CASIA-A have 240 sequences that divided to 3828 sub-sequences. To avoid the influence of clothes and color, the contour data are only used in our experiments, and the contour images are shown in Fig. 3. The images only consist of values of 0 and 1, where 0 represent black pixels and 1 represents white pixels.

In our experiments, the dataset, CASIA-A, was divided into two mutually exclusive parts, a training set and a test set, at a ratio of 9:1 ratio via cross-validation. For each 10 sub-sequences, we picked 9 sub-sequences for training and 1 for testing. To obtain more precise data of every angle, these gait sequences were separated into different angles to train or test specific tasks. For example, the model trained using the 0 degree images could test various gait sequences without considering the angle or the index of the sequence. All the gait sequences, which encompass 3828 sub-sequences and 19139 total frames, were divided into the training set and the test set, which contain 3441 sub-sequences that 17205 frames and 386 sub-sequences that 1930 frames, respectively.

CASIA-B [32] dataset is a popular and comprehensive gait dataset. It contains 124 subjects (labeled in 001-124), 3 walking conditions and 11 views $(0, 18, \ldots, 180)$. The walking condition contains normal (NM) (6 sequences per subject), walking with bag (BG) (2 sequences per subject) and wearing coat or jacket (CL) (2 sequences per subject). Namely, each subject has $11 \times (6 + 2 + 2) = 110$ sequences. As there is no official partition of training and test sets of this dataset, we conduct experiments on the settings which are popular in current literatures. The first 74 subjects were used for training and the rest 50 subjects were leaved for test. Given a probe sequence, the goal is to retrieve all the sequences with the same identity in gallery set. In the test sets of all three settings, the first 4 sequences of the NM condition (NM #1-4) are kept in gallery, and the rest 6 sequences are divided into 3 probe subsets, i.e. NM subsets containing NM #5-6, BG subsets containing BG #1-2 and CL subsets containing CL #1-2.

4.2 Data Preprocessing

The initial CASIA contour data include 240×352 images. To effectively use the data to train and calculate the model, the initial images have to be preprocessed. We transform the initial 240×350 images into new 120×176 contour images, which means that the height and width of the new images are both halved compared with the initial images.

In the preprocessing, as shown in Fig. 7, the main coding aimed to achieve 5 functional steps, which include the transformation from RGB to grayscale images, the binarization, the morphological processing, the extraction of body contours and the compression.

The common format of color image is RGB, with a 24-bit image bit depth. Due to the massive of information contained in the picture, the calculations require substantial temporal and spatial resources. The purpose of grayscale and binarization is to convert the RGB image into a bi-

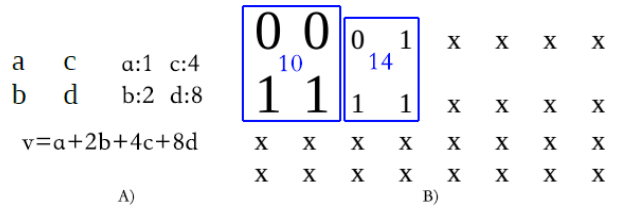

Fig. 8 Compression algorithm. A) Give the different weights to specific pixel position. B) Examples of our compression algorithm, the new value calculated by the equation v=a+2b+4c+8d.

nary image. Although the operations will lose the color and grayscale values of the original image, for portrait images, it still retains the outline texture information of the portrait. Furthermore, the operations are improved due to the extreme reduction in the size of the image data, and the efficiency is improved for subsequent processing.

Morphology was originally denoted as the study of the morphology and structure of living things. In graphics, it is mainly used to represent the content of digital morphology. Mathematical morphology is used as a tool to extract the useful components of an image's expression and the shape of the depicted area, such as the boundaries, skeletons, and convex hulls. As shown in Fig. 7, after obtaining the grayscale image, there are defects in the human body and the connected parts that should not appear in the image. For this case, the opening and closing operations were performed on the defective images.

To further increase the speed of the operations, the binarization reduced amount of calculation of the pixels inside the contour on the premise of preserving the shape and size of human figures. The edge part in the grayscale image is caused by the discontinuous or sudden change of the grayscale value of the adjacent area. Generally, the edge was detected by the first and second derivatives. At the edge position, the amplitude value of the first derivative will appear at the local extreme value, and the amplitude value of the second derivative will appear at the zero crossing point. Hence, the edge position can be determined by calculating the grayscale derivative and detecting the local extreme point or zero crossing point. There are three general types of edge detection operators: the Sobel, Laplace and Canny operators. This contour extraction uses the edge detection of the Sobel operator [33] (structure shown in Fig. 7) to perform edge detection on the person and then obtain the contour information.

The compression algorithm combines the adjacent four pixels into a new pixel. Because the value of the pixel is 0 or 1 in 4 positions, there are 16 combinations for four pixels. Our data compression method is showed in Fig. 8. Every digit of the four adjacent pixel values has a different weight. If we stipulate that the position identification and weight of four pixels is as that shown in Fig. 8 A), the novel pixel value would be v=a+2b+4c+8d. Hence, the novel value would be one digit from 0 to 15.

For the example in Fig. 8 B), there is a block of pixel values in the initial image. The four values in the upper left corner make up a novel value v=. . . = 10. Similarly, the new value composed of the four right values would be 14. According to the equation described in Fig. 7 A), all of the gait contour pixels could be condensed after converting the 240×350 images into 120×176 images. Compared with the original images, the compressed images have fewer parameters, thus greatly reducing the time and spatial complexity. Moreover, the compression algorithm can guarantee the integrity of the information while the neural network is calculating. After the compression, each novel pixel value was divided by 15 and normalized into values from 0 to 1. Then, the normalized value of each pixel was subtracted from that pixel's value as the final data processing procedure.

4.3 Experimental Procedures

The CASIA-A dataset described in Sect. 4.1 contains 20 people's gait contours and 12 gait sequences with 3 angles: 0°, 45° and 90°. The experiment runs on two GPUs, GeForce GTX TITAN X, in the Linux server. The procedures on CASIA-A was described as follows.

Step 1: Using the cross-validation method, each gait sequence in the dataset was divided into a training set and a test set at a ratio of 9:1. Then, the method was trained using the first three sequences and tested with the last sequence and the other sequences of each angle in the following steps.

Step 2: The gait images have to be preprocessed and condensed, as described in Sect. 4.2, to optimize the performance of the training and testing processes.

Step 3: Three difference approaches were applied to train the gait recognition method using images from various angles. The first experiment identified gait sequences using the 3D-CNN, which could make complete use of the temporal and spatial information. Nevertheless, the number of parameters causes it to train slowly and require substantial memory space. The convolution window size is 5 in the temporal dimension, representing a set of five frames in each group. The second experiment used the CNN-LSTM. The CNN model was used to extract the features of a minor sequence and the LSTM model was used to classify the gait sequence to conduct person identification. In the last experiment, the CJAM approach we mentioned before was trained to conduct gait recognition. In order to observe how the Attention model affects the efficiency of the gait recognition, we removed the attention layers for experimentation.

Step 4: Test the performance of every model using different cross-views.

Step 5: The performance and the accuracy of each model was compared.

After completing the experiments on the CASIA-A dataset, we used the same experimental method as GaitSet [39] and GaitNet [40] to conduct experiments on the CASIA-B dataset.

4.4 Experimental Results and Discussion

In the experiments, the CJAM approach was used for the

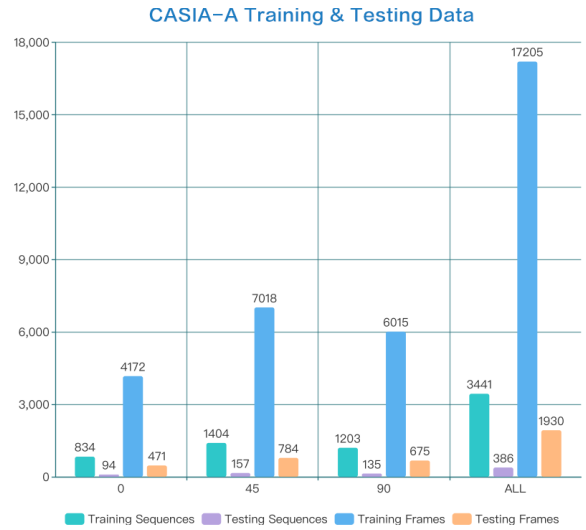

Fig. 9 The quantity of training and testing data on each cross view in CASIA-A.

Table 1 Comparison on CASIA-A with cross view and conditions. Three models are trained and tested for different cross view.

Training angle	Testing angle	Testing accuracy on different models			
		CJAM (ours)	3D-CNN [22]	CNN-LSTM [30]	CNN [41]
0°	0°	**100**	95	99.2	96.1
	45°	72.5	57.5	**76.25**	28.1
	90°	**30**	11.25	16.25	25.8
45°	0°	75	72.5	**82.5**	40.0
	45°	**100**	90	**100**	97.6
	90°	**70**	51.25	53.75	55.9
90°	0°	12.5	**21.25**	13.75	15.3
	45°	10	12.5	12.5	**14.7**
	90°	**99.8**	80	90	96.5
All	All	**97.8**	92	95.3	96.35
Average accuracy		**66.76**	58.325	63.825	56.635
Average accuracy with similar training and testing angle		**99.93**	88.3	96.4	96.73

identification of the gait dataset. We considered the accuracy of the model to evaluate the performance. Through a comparative experimental analysis, a conclusion could be drawn that the effect of each model depends on the cross-view of data. Furthermore, as the amount of data to be processed increases, the effect of the CJAM model becomes superior to those of the other models.

First, the whole dataset is separated by 9-fold cross validation (K-CV) [34] into mutually exclusive parts: the training set and the testing set. The cross-validation effectively makes use of the limited data. Furthermore, the evaluation results can be as close as possible to the performance of the model on the test set, and an indicator can be taken advantage of for model optimization. The specific sizes of the separated data set can be shown in Fig. 9 as follows. The size of each training set and testing set is subject to the ratio we mentioned in Sect. 4.1, and the all item includes various 00, 45 and 90 degree images.

Perhaps the most important part of this section is that comparisons of the various approaches were made over the course of the 12 experiments and the results are shown in Table 1. In order to reflect the superiority of the CJAM, we compared with the more popular models, 3D-CNN [22] and CNN-LSTM [30]. Moreover, in order to highlight the improvement of the model's performance by the attention mechanism, we conducted a comparative experiment between simple CNN [41] and CJAM. Obviously, the CJAM can perform better than others regarding accuracy. However, the 3D-CNN has a partial advantage for the 90 degrees training and 0 degrees testing groups and the CJAM and the CNN-LSTM have significant advantages for the other groups; furthermore, the CJAM reached the highest average accuracy overall.

When the training and testing sets are from the same direction, the CJAM has the best accuracy with 100% on both the 0 and 45 degree subsets and 99.8% on the 90 degree subset. Compared with other methods, the CJAM could learn the features more effectively. When training all angle sequences were used at once, the CJAM can achieve a recognition rate of up to 97.8%.

Regardless of the training set, the empirical evidence shows that the CJAM is always the best for the 90 degree images, which means the attention model in the CJAM has the ability to address the irrelevant details to some extent. There are significant variances between the 0 degree and 90 degree images in that 0 degree images vary due to the distinctive locations and gestures, but the 90 degree images are often similar to each other. It is difficult to recognize 90 degree images through the model trained on 0 degree images. The 3D-CNN and CNN-LSTM only achieved accuracies of 11.25% and 16.25%, respectively, in this scenario. However, the CJAM achieved better accuracy at 30%. The reason that our model has the best predictions is that attention model could ignore those irrelevant images in a sequence and set high weights on those "important" images. The 3D-CNN consumes substantial computing time and memory resources, which significantly increases the spatial complexity and time complexity. The CNN-LSTM model cannot eliminate the influence of those unimportant images which may lead to a slightly worse result. In the 45 degree training set and the 90 degree testing set, the CJAM achieved 70% accuracy but the 3D-CNN and CNN-LSTM only achieved 51.25% and 53.75%, respectively. When all of the gait data in CASIA-A were used for training and testing, the CJAM still had the best accuracy of 97.9% compared with 92% and 95.3%, respectively.

Obviously, the evidence on CASIA-A have shown that the architecture of CJAM is superior to the 3D-CNN and CNN-LSTM. Table 2 depicts the accuracy of the normal walking condition on CASIA-B with various training and testing angle. When the training angle and the testing angle are similar, the CJAM always achieve the perfect recognition accuracy of 100%. Moreover, the CJAM could got the significant accuracy when the cross-view of camera below 54°. For example, the model trained in 54°-126° has 100% rank-1 accuracy to recognize the 90° gait sequences.

Table 2 The accuracy on the normal walking condition of CASIA-B in various angle.

Training angle(°) \ Testing angle(°)	0	18	36	54	72	90	108	126	144	162	180	MEAN (Include Training Angle)	MEAN (Exclude Training angle)
0	**100.0**	**100.0**	92.0	74.0	56.0	52.0	58.0	68.0	78.0	90.0	96.0	78.55	76.4
18	98.0	**100.0**	**100.0**	92.0	74.0	68.0	66.0	80.0	82.0	92.0	88.0	85.45	84.0
36	84.0	**100.0**	**100.0**	**100.0**	98.0	76.0	76.0	90.0	88.0	86.0	76.0	88.55	87.4
54	64.0	88.0	**100.0**	**100.0**	**100.0**	94.0	92.0	94.0	86.0	68.0	56.0	85.64	84.2
72	54.0	80.0	96.0	98.0	**100.0**	**100.0**	98.0	96.0	88.0	60.0	42.0	82.9	81.2
90	56.0	68.0	90.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	96.0	58.0	48.0	83.28	81.6
108	52.0	68.0	84.0	90.0	**100.0**	**100.0**	**100.0**	98.0	96.0	68.0	58.0	83.1	81.4
126	56.0	78.0	86.0	92.0	96.0	98.0	**100.0**	**100.0**	**100.0**	90.0	60.0	86.9	85.6
144	74.0	80.0	92.0	94.0	90.0	90.0	94.0	**100.0**	**100.0**	98.0	78.0	90.0	89.0
162	86.0	86.0	88.0	72.0	62.0	62.0	72.0	92.0	96.0	**100.0**	92.0	82.55	80.8
180	92.0	84.0	70.0	58.0	46.0	40.0	44.0	68.0	68.0	92.0	**100.0**	69.3	66.2

Table 3 Average accuracy(%) of cross-view gait recognition on CASIA-B. Excluding identical view cases.

Gallery NM#1-4		0°-180°					mean
Probe View(°)		0	54	90	144	180	
NM#5-6	LSTM [37]	63.6	83.8	60.0	-	-	69.1
	3D-CNN [38]	87.1	94.6	88.3	96.5	85.7	92.1
	GaitSet [39]	90.9	**96.9**	91.7	**98.9**	85.8	**95.0**
	GaitNet [40]	**91.2**	95.6	92.6	92.9	**89.0**	91.6
	CJAM(ours)	88.55	95.64	**93.28**	97.64	87.3	**92.48**
BG#1-2	3D-CNN [38]	64.2	76.9	63.1	82.2	61.3	72.4
	GaitSet [39]	83.8	**88.8**	81.0	**92.2**	79.0	**87.2**
	GaitNet [40]	83.0	86.6	74.8	85.8	-	82.6
	CJAM(ours)	**86.5**	83.52	**83.9**	91.54	**85.51**	**86.2**
CL#1-2	3D-CNN [38]	37.3	61.1	54.6	58.9	39.4	54.0
	GaitSet [39]	61.4	**77.3**	70.1	73.5	50.0	66.46
	GaitNet [40]	42.1	70.7	70.6	69.4	-	63.2
	CJAM(ours)	**65.4**	73.4	**72.6**	**74.0**	**58.3**	**68.74**

A comprehensive comparisons between the CJAM and the state-of-art gait literatures on CASIA-B are shown in Table 3. Except of ours, other results are directly taken from their original papers. Most of the methods compared are both neural networks instead of traditional machine learning methods. All of the results are averaged on the 11 trained views and the identical views are excluded, for that identical view always made 100% recognition accuracy. The CJAM model is superior to most of the current mainstream models in deep learning for gait recognition, and the attention model has more potential to better performance.

On the CASIA-A dataset and on average, the CJAM still achieved the best performance among the three neural network methods. Although the CJAM has many advantages for a great majority of the angles, more studies are still required to improve the performance for the 0 degree training set, the 90 degree training set and the 0 and 45 degree test sets, which may be due to overfitting.

5. Conclusion

To reduce the impact of the distorted pixels on gait recognition, it is important to develop a novel approach to assign higher weights to vital pixels. In this paper, we propose a novel CJAM approach for gait recognition, where the features of images are extracted and classified by a CNN and an attention mechanism, respectively. An extensive method to preprocess and condense the initial image was applied so that the neural network model can be more efficiently trained. Extensive experiments on the CASIA-A and CASIA-B gait datasets in tasks shows the great advantages of our proposed CJAM model compared with two other main deep learning methods, namely, the 3D-CNN and the CNN-LSTM.

In the future, the approaches mentioned above could be applied to other databases or sequencing problems. The attention mechanism ignoring irrelevant details is also practicable in many other fields. Furthermore, we intend to extend the domain of the proposed models to specific scenarios, such as generalized recognition from any angle.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61902311). We gratefully acknowledge the foundation and the invaluable cooperation we received in preparing this manuscript. We also would like to acknowledge the dataset collected by Professor Liming Shi's team at the Chinese Criminal Investigation College.

References

[1] P. Kumar, M. Ahirwar, and A. Deen, "A Survey on Iris Recognition System," International Journal of Computer Sciences and Engineering, vol.7, no.7, pp.302–307, 2019.
[2] H. Jan, A. Ali, S. Mahmood, et al., "Statistical descriptors-based automatic fingerprint identification: Machine learning approaches," 2019.
[3] L. Li, Y. Peng, G. Qiu, Z. Sun, and S. Liu, "A survey of virtual sample generation technology for face recognition," Artificial Intelligence Review, vol.50, no.1, pp.1–20, 2018.
[4] P. Connor and A. Ross, "Biometric recognition by gait: A survey of modalities and features," Computer Vision and Image Understanding, vol.167, pp.1–27, Feb. 2018.
[5] A. Kale, A. Sundaresan, A.N. Rajagopalan, N.P. Cuntoor, A.K. Roy-Chowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait," IEEE Transactions on image processing, vol.13, no.9, pp.1163–1173, 2004.
[6] J. Han and B. Bhanu, "Individual recognition using gait energy image," IEEE transactions on pattern analysis and machine intel-

ligence, vol.28, no.2, pp.316–322, 2006.
[7] K.J. Wang and T.Q. Yang, "Gait recognition method based on column mass vector and support vector machine," Computer Engineering and Applications, vol.51, no.7, pp.169–173, 2015.
[8] X. Chen and T. Yang, "Gait recognition method without influence of dress and carrying," Computer Engineering and Applications, vol.52, no.5, pp.141–146, 2016.
[9] L. Li, G. Gu, and C. Wang, "Features fusion gait recognition by combining energy image and canonical correlation analysis," Journal of Chinese Computer Systems, vol.35, no.11, pp.2558–2561, 2014.
[10] Y. Chai, T. Xia, and W. Han, "Gait recognition algorithm based on multi-featured fusion," Journal of Chinese computer Systems, vol.35, no.3, pp.636–641, 2014.
[11] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural computation, vol.1, no.4, pp.541–551, 1989.
[12] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol.18, no.7, pp.1527–1554, 2006.
[13] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science, vol.313, no.5786, pp.504–507, 2006.
[14] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," International Conference on Neural Information Processing Systems, pp.1097–1105, Curran Associates Inc., 2012.
[15] K. Leyden, M. Koller, M. Niemier, et al., "Kinect image processing by CNN algorithm for gait recognition," Cnna; International Workshop on Cellular Nanoscale Networks & Their Applications, VDE, 2017.
[16] M. Rauf, C. Song, Y. Huang, L. Wang, and N. Jia, "Knowledge transfer between networks and its application on gait recognition," IEEE International Conference on Digital Signal Processing, IEEE, 2016.
[17] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A Comprehensive Study on Cross-View Gait Based Human Identification with Deep CNNs," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.39, no.2, pp.209–226, 2016.
[18] F.M. Castro, M.J. Marín-Jiménez, N. Guil, and N.P.D. Blanca, "Multimodal feature fusion for CNN-based gait recognition: an empirical comparison," Neural Computing and Applications, vol.32, no.17, pp.14173–14193, 2020.
[19] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," IEEE transactions on pattern analysis and machine intelligence, vol.35, no.1, pp.221–231, 2012.
[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp.1725–1732, 2014.
[21] Y.M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "Lipnet: Sentence-level lipreading," 2016. arXiv preprint arXiv:1611.01599
[22] T. Wolf, M. Babaee, and G. Rigoll, "Multi-view gait recognition using 3D convolutional neural networks," IEEE International Conference on Image Processing, pp.4165–4169, IEEE, 2016.
[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol.9, no.8, pp.1735–1780, 1997.
[24] F. Li, M. Zhang, G. Fu, T. Qian, and D. Ji, "A Bi-LSTM-RNN model for relation classification using low-cost sequence features," arXiv preprint arXiv:1608.07720, 2016.
[25] X. Yi, R. Li, and M. Sun, "Generating chinese classical poems with rnn encoder-decoder," Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, vol.10565, pp.211–223, Springer, Cham, 2017.
[26] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2625–2634, 2015.
[27] J.R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," arXiv preprint arXiv:1612.0390, 2016.
[28] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, "Gait Recognition via Disentangled Representation Learning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.4705–4714, 2019.
[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, and I. Polosukhin, "Attention Is All You Need," arXiv preprint arXiv:1706.03762, 2017.
[30] J. Gao, P. Gu, Q. Ren, J. Zhang, and X. Song, "Abnormal Gait Recognition Algorithm Based on LSTM-CNN Fusion Network," IEEE Access, vol.7, pp.163180–163190, 2019, doi: 10.1109/ACCESS.2019.2950254
[31] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhoutte analysis based gait recognition for human identification," IEEE trans Pattern Analysis and Machine Intelligence(PAMI), vol.25, no.12, pp.1505–1518, 2003.
[32] S. Yu, D. Tan, and T. Tan, "A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition," Proc. 18'th International Conference on Pattern Recognition (ICPR06), Hong Kong, China, Aug. 2006.
[33] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, Sobel Edge Detector, http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm, 2003.
[34] J.D. Rodriguez, A. Perez, and J.A. Lozano, "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.32, no.3, pp.569–575, March 2010. doi: 10.1109/TPAMI.2009.187
[35] V.A. Chenarlogh and F. Razzazi, "Multi-stream 3D CNN structure for human action recognition trained by limited data," IET Computer Vision, vol.13, no.3, pp.338–344, 2019. doi: 10.1049/iet-cvi.2018.5088
[36] R. Yang, S.K. Singh, M. Tavakkoli, N. Amiri, Y. Yang, M.A. Karami, and R. Rai, "CNN-LSTM deep learning architecture for computer vision-based modal frequency detection," Mechanical Systems & Signal Processing, vol.144, N.PAG, 2020. doi: 10.1016/j.ymssp.2020.106885
[37] Y. Feng, Y. Li, and J. Luo, "Learning effective Gait features using LSTM," 2016 23rd International Conference on Pattern Recognition (ICPR), pp.325–330, 2016.
[38] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep CNNs," IEEE TPAMI, vol.39, no.2, pp.209–226, 2017.
[39] H. Chao, Y. He, J. Zhang, and J. Feng, "Gaitset: Regarding gait as a set for cross-view gait recognition," Proceedings of the AAAI Conference on Artificial Intelligence, vol.33, 2019.
[40] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, "Gait Recognition via Disentangled Representation Learning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.4705–4714, 2019.
[41] X. Wang, J. Zhang, and W.Q. Yan, "Gait recognition using multichannel convolution neural networks," Neural Computing and Applications, vol.32, no.2, 2020.
[42] M. Derlatka and M. Bogdan, "Ensemble kNN classifiers for human gait recognition based on ground reaction forces," International Conference on Human System Interactions, IEEE, 2015.
[43] T. Sharma, S.S. Dub, and B. Gupta, "Performance Analysis of ANN based Gait Recognition," International Journal of Scientific Research in Science and Technology, pp.112–125, 2018.

Pengtao Jia was born in Pucheng County, Shaanxi Province, China, in 1977. She received her B.S. and M.S. degrees in computer application technology from Xi'an University of Science and Technology, China, in June 2002 and her Ph.D. degree in computer science and technology from Northwestern Polytechnical University, China, in June 2008. Since 2015, she has been a Professor with the School of Computer Science and Technology, Xi'an University of Science and Technology. Professor Jia's research interests include data mining, applications of artificial intelligence theory and visualizations of coal mine safety data. She has published more than 20 scientific research papers and one monograph, and she holds more than 10 software copyrights and utility model patents. Professor Jia has received the Shaanxi Science and Technology Award for Excellence.

Qi Zhao was born in Weinan County, Shaanxi Province, China, in 1998. She received her B.S. degree in network engineering from North University of China in 2019. She is currently pursuing her M.S. degree in computer science and technology at Xi'an University of Science and Technology. Her research interests include deep learning, convolutional neural networks, defects detection, and the manufacture of intelligent robots. Miss. Zhao has received awards and honors from the Computer Application Competition in Five North China Provinces.

Boze Li was born in Shaanxi Province, China, in 1992. He received his B.S. degree in computer science and technology at Xi'an University of Science and Technology in 2014, and he received his M.S. degree in 2018 at the same college. After graduation, he worked at Deeply curious Technology Co., Ltd. Util 2020. His research interests includes natural language processing, neural models, reinforcement learning, and symbolic computing.

Jing Zhang was born in Xi'an, Shaanxi, China in 1988. She received the B.S. degree in Mathematics department from Northwest University, Shaanxi, China, in 2010, the M.S. degree in 2013 and the Ph.D. degree in 2018 in computer science and technology from Northwest University, Shaanxi, China. From 2018 to 2019, she was a lecturer with the Xi'an University of Science and Technology, Shaanxi, China. Her research interest includes the image processing and 3D reconstruction techniques, fundamental study of signal processing and signal imaging. Dr. Author has been conducting 1 NSFC project and 1 project from Shaanxi Science and Technology Department. She has been published 10 papers in SCI and EI journal.