Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

GaitGCN++: Improving GCN-based gait recognition with part-wise attention and DropGraph

Md. Bakhtiar Hasan *, Tasnim Ahmed, Sabbir Ahmed, Md. Hasanul Kabir

Department of Computer Science and Engineering, Islamic University of Technology, Board Bazar, Gazipur, 1704, Dhaka, Bangladesh

ARTICLE INFO

Article history:
Received 27 December 2022
Revised 2 June 2023
Accepted 25 June 2023
Available online 4 July 2023

Keywords:
Biometric authentication
Hop extraction
ResGCN
Joint position
Graph Convolutional Network

ABSTRACT

Gait recognition is becoming one of the promising methods for biometric authentication owing to its self-effacing nature. Contemporary approaches of joint position-based gait recognition generally model gait features using spatio-temporal graphs which are often prone to overfitting. To incorporate long-range relationships among joints, these methods utilize multi-scale operators. However, they fail to provide equal importance to all joint combinations resulting in an incomplete realization of long-range relationships between joints and important body parts. Furthermore, only considering joint coordinates may fail to capture discriminatory information provided by the bone structures and motion. In this work, a novel multi-scale graph convolution approach, namely 'GaitGCN++', is proposed, which utilizes joint and bone information from individual frames and joint-motion data from consecutive frames providing a comprehensive understanding of gait. An efficient hop-extraction technique is utilized to understand the relationship between closer and further joints while avoiding redundant dependencies. Additionally, traditional graph convolution is enhanced by leveraging the 'DropGraph' regularization technique to avoid overfitting and the 'Part-wise Attention' to identify the most important body parts over the gait sequence. On the benchmark gait recognition dataset CASIA-B and GREW, we outperform the state-of-the-art in diversified and challenging scenarios.

1. Introduction

Biometric authentication pertains to identifying and re-identifying human individuals by analyzing their physical and behavioral characteristics (Rani and Kumar, 2023). In this regard, the use of gait as a method of non-intrusive biometric authentication is getting more popular day by day (Sepas-Moghaddam and Etemad, 2022; Parashar et al., 2022; Minaee et al., 2023). Gait denotes the pattern generated by the movement of body parts of subjects during locomotion over a surface. The human gait cycle can be considered as a sequence of repetitive steps involving the muscles and skeleton in coordination with the nervous, cardiac, and respiratory systems (Physiopedia, 2022). It becomes an intrinsic part of our life as people tend to start walking from an early age. Consequently, this can be utilized to identify humans, which is known as gait recognition. It has applications in biometric authentication in security systems to prevent unauthorized access (Nambiar et al., 2019; Filipi Gonçalves dos Santos et al., 2022; Singh et al., 2022), clinical applications and healthcare to assist elderly patients and detect abnormalities (Muro-De-La-Herran et al., 2014; Archila et al., 2022; Cicirelli et al., 2022; Russo et al., 2023; Meng et al., 2023), sports science for performance analysis and injury prevention (Echterhoff et al., 2018; Mason et al., 2023), style and affect analysis used in psychology, marketing, and human–computer interaction for better understanding of human behavior and designing more effective products and services (Etemad and Arya, 2016; Azhar et al., 2023), etc.

Gait recognition systems can be divided into two major categories: wearable and non-wearable (Marsico and Mecca, 2019). Wearable gait recognition systems require the use of external devices, such as accelerometers, gyroscopes, smartphones, and so on (Shen et al., 2023). However, the use of such devices can be costly for large-scale applications and may cause inconvenience for the end users. On the other hand, non-wearable gait recognition

[*] Corresponding author.
E-mail addresses: bakhtiarhasan@iut-dhaka.edu (M.B. Hasan), tasnimahmed@iut-dhaka.edu (T. Ahmed), sabbirahmed@iut-dhaka.edu (S. Ahmed), hasanul@iut-dhaka.edu (M.H. Kabir).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101641
1319-1578/© 2023 The Author(s). Published by Elsevier B.V. on behalf of King Saud University.

systems are mostly vision-based (Sepas-Moghaddam and Etemad, 2022; Mogan et al., 2022). These systems utilize imaging sensors to capture the gait of the subject. As a result, they do not require cooperation from the subject and can identify them from a distance. However, the performance of these systems may suffer due to i) variations in the appearance of the subject; ii) variations in the viewpoint or camera angle; iii) occlusion resulting from appearance or viewpoint, and iv) variations in the environment (Sepas-Moghaddam and Etemad, 2022). To alleviate these issues, recent joint position-based approaches to gait recognition depend on extracting the physical structure of a subject's body (Liao et al., 2017; Liao et al., 2020). In the recent past, these approaches were mostly avoided due to their high computational requirement (Wang et al., 2010). Nowadays, the advances in pose estimation techniques have made them feasible again (Sepas-Moghaddam and Etemad, 2022). Moreover, the data extracted by pose estimators contain only joint positions, which can provide critical information regarding the gait devoid of environmental noises. This allows the gait recognition systems to focus entirely on extracting spatial features. Additionally, the temporal changes extracted from gait video sequences can show how the position of the subject of interest differs in consecutive frames, which can provide valuable insights. To extract the temporal information, 3D-CNNs (Lin et al., 2020; Huang et al., 2021) or LSTMs (Liao et al., 2020; Yue et al., 2022) have been used in the literature. These approaches can comprehend spatio-temporal information but require high computational resources.

To utilize spatio-temporal features with relatively low computational resources, Graph Convolutional Networks (GCNs) have recently gained much traction owing to their capability to harness the power of arbitrarily structured graphs using convolution filters (Monti et al., 2017). The adjacency matrix of the underlying graph, along with a feature vector associated with each node can be used to model both the spatial information and temporal relationships available in the gait sequence in order to learn discriminative and robust features. Instead of working with the entire video sequence, the network considers the set of nodes and edges to perform different operations, reducing the overall computation. Recent approaches (Li et al., 2020a; Teepe et al., 2021) extract gait features by forming a spatio-temporal graph from the available video sequences. These approaches mostly perform local convolutions. However, since walking is accomplished using various body parts that are distant from each other, local movements conducted by a few adjacent joints could be ambiguous when differentiating between different gaits.

To aggregate the effects of distant body parts, higher-order polynomials of the adjacency matrix have been proposed in the existing literature (Kipf and Welling, 2017). Unfortunately, this formulation is affected by the *biased weighting problem* due to the cyclic walks present in the graph representing human gait. This results in a higher priority towards closer joints than the further ones (Hasan et al., 2022). Another desirable trait of gait recognition systems is the propensity to leverage the discriminatory information provided by the joint coordinates, their bone structures, and motion (Wang et al., 2022a). This necessitates the use of a deep neural architecture that is able to combine the latent information from the said features. Additionally, it should be able to identify and prioritize body parts that are useful in gait recognition, which requires the use of an attention mechanism (Vaswani et al., 2017). Moreover, depending on the class distribution, data distribution, and depth, deep learning architectures often tend to overfit (Ioffe and Szegedy, 2015). To overcome this problem, various regularization techniques are used to increase the generalization capability of the networks. However, these techniques may be ineffective in GCNs due to the strong correlation between features in single and consecutive frames (Cheng et al., 2020). Hence, model-specific regularization approaches need to be explored.

Based on the discussion above, the motivation of this work is to develop a multi-stream aggregation technique that can effectively model the relationship between closer and distant limbs while also prioritizing important body parts for gait recognition. Our specific contributions can be summarized as follows:

1. 'GaitGCN++', a gait recognition system is proposed that utilizes a multi-stream Graph Convolution Network is constructed that combines the joint, bone, and motion features.
2. Utilizing a hop extraction-based adjacency technique, the system is able to exploit the relationship between nearby and distant joints while avoiding redundant dependencies.
3. DropGraph technique is applied to ensure that the system learns to avoid overfitting in order to learn generalized features that are helpful in recognizing unseen gait samples.
4. Part-wise attention module is introduced to identify and prioritize only specific body parts that are helpful in recognizing gait.
5. By combining all these techniques mentioned above, the system has outperformed the state-of-the-art gait recognition methods in indoor and outdoor environments in terms of average accuracy on CASIA-B and GREW datasets.

2. Literature review

The use of gait to identify individuals started in the late 1960s (Murray et al., 1964; Murray, 1967), which utilized intra-class similarities and inter-class differences among human gait. Later on, studies on the ability of humans in recognizing other people based on their gait solidified the base for gait recognition in biometric and forensic applications (Johansson, 1973; Cutting and Kozlowski, 1977; Cutting et al., 1978). Since then, prominent use of gait can be seen utilizing video data (Niyogi and Adelson, 1994), sensors (Addlesee et al., 1997), and mobile devices (Mantyjarvi et al., 2005). In this work, we focus on vision-based gait recognition (henceforth called gait recognition) that utilizes video data to develop machine vision.

The recent influx in the utilization of deep neural networks in almost every task has seen the application of a hierarchy of frameworks to extract high-level features using nonlinear functions and use them to classify human gait (Sepas-Moghaddam and Etemad, 2022; Mogan et al., 2022; Minaee et al., 2023; Filipi Gonçalves dos Santos et al., 2022). These networks include 2D Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN), Capsule Networks (CapsNet), Recurrent Neural Networks (RNN), 3D Convolutional Neural Networks (3D CNN), Graph Convolutional Networks (GCN), etc.

One of the most common architectures for gait recognition, Convolutional Neural Networks (CNN) employs a set of convolution and pooling layers and activation functions to generate activation maps that encode the skeleton joints and silhouettes in gait video sequences. The extracted activation maps are then passed through a set of flatten and fully-connected layers, which are then fed through a softmax function to classify based on the probability distribution of the individual classes. Common CNN-based architectures that are utilized in gait recognition are GEINet (Shiraga et al., 2016), Ensemble CNNs (Wu et al., 2016), EV-Gait (Wang et al., 2019a), GaitNet (Song et al., 2019), GaitSet (Chao et al., 2019), Joint-CNN (Zhang et al., 2019) GaitRNNPart (Sepas-Moghaddam and Etemad, 2020), GaitPart (Fan et al., 2020), SMPL (Li et al., 2020b), CapsGait (Sepas-Moghaddam et al., 2021), Distilled Light GaitSet (Song et al., 2022), etc. All CNN architectures mentioned here, except for GaitNet, require less than ten layers to extract relevant features. These models consist of 2–6 convolution layers, 0–2 pooling layers, and 1–3 fully-connected layers. This relatively less number of layers can be attributed to the fact that CNNs are capable of extracting informative texture information using various layers, which is mostly absent in gait modalities, such as skeleton joints or silhouettes. Additionally, the extraction of structural information alone poses one of the critical limitations of using 2D CNNs in gait recognition as they fail to extract temporal information, thus having a lesser understanding of how the subject moves from one frame to another.

Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) combine generator and discriminator(s) to address the issue of viewing angle, clothing, and carrying condition invariance in gait recognition. GANs can be used to change viewing angles and clothing types or remove carried objects. To preserve the identifying information while also modifying the appearance, two discriminator networks are often employed by GAN-based architectures: one for distinguishing real and fake samples and one for preserving the identifying information. Different GAN-based architectures such as MGAN (He et al., 2018), DiGAN (Hu et al., 2018), and TS-GAN (Wang et al., 2019b) are commonly used in gait recognition. These approaches require substantial computational resources since multiple architectures need to be trained to generate, discriminate, and identify gait. Additionally, their sheer sizes make them infeasible to be implemented in low-end devices.

Capsule Networks (CapsNet) (Sabour et al., 2017) have been used in gait recognition to model the structural relationships between different body parts by preserving different positional information. The information is encoded using a set of capsule blocks. The network is utilized in gait recognition because of its capability to understand intrinsic view-invariant features that help recognize gait from different camera angles. It has been used separately (Xu et al., 2019) and in combination with other networks (Zhao et al., 2020; Sepas-Moghaddam et al., 2021). While CapsNets are capable of extracting view-invariant features, they often struggle with processing complex data making them infeasible for handling entire gait sequences.

To exploit the temporal relationship among the consecutive frames of a gait sequence, Recurrent Neural Networks (RNN), such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014), have been applied. These architectures feed either skeleton joints extracted via external devices (Liu et al., 2016) or spatial features extracted by CNNs (Feng et al., 2016; Battistone and Petrosino, 2019; Sepas-Moghaddam and Etemad, 2020; Sepas-Moghaddam et al., 2021; Qi et al., 2022) to the RNN, which then generates the class label for the provided sequence. Since RNNs introduce more parameters to encode the temporal information, they take additional computational resources to train. They are also prone to overfitting the training data as it is difficult to implement regularization techniques (Pascanu et al., 2013). On the other hand, the use of 2D CNNs in combination with RNNs also introduces the shortcomings of 2D CNNs in extracting structural information from silhouette or joint data.

To combine the spatial and temporal information of gait sequence, 3D Convolutional Neural Networks (3D CNN) have been used to extract view- and appearance-invariant features. However, due to the variability in the number of frames in the gait sequence, directly applying 3D CNNs is not possible. As a result, multiple 3D CNNs of varying scales and filter sizes are used in the existing gait recognition literature (Wolf et al., 2016; Lin et al., 2020; Lin et al., 2021b; Lin et al., 2021a; Lin et al., 2021c). The use of varying scales and sizes to process gait sequences is also considered as one of the shortcomings. The use of several convolution blocks repeated multiple times can result in a substantially large network, increasing the overall computational cost for training and inference.

Graph Convolutional Networks (GCN) (Monti et al., 2017) were developed as an extension of CNNs that utilize higher dimensional graph structures and adjacency matrix-based convolution filters. GCNs exploit the inherent graph-like nature of human posture. The advantage of this approach is that it can combine both structural information from a single frame and temporal relationships among consecutive frames. At the same time, the extracted features can be view and appearance-invariant. Li et al. (2020a) was the pioneer in using GCN in combination with Joint Relationship Pyramid Mapping considering the joint positions as vertices and the bones connecting them as edges. Recently, Gao et al. (2022) used a combination of ST-GCN (Chen et al., 2021) and Canonical Polyadic Decomposition (Sorber et al., 2013) to improve the performance. In another work, Zheng et al. (2022) combined ST-GCN with an Angle Estimator module that helps predict the viewing angles making the network robust to variation in viewing angle. Wang et al. (2022) proposed frame-level refinement-based graph convolutional networks along with self-attention to differentiate relations between joints in consecutive frames and focus on important body joints. Liao et al. (2022) utilized pose estimation maps combined with GCNs for gait recognition. Liu et al. (2022b) utilized Symmetry-Driven Hyper Feature GCN for learning dynamic patterns and semantic features. Shopon et al. (2021) introduced the concept of residual connection in gait recognition models. However, these approaches only considered the joint-stream data for gait recognition. Powered by the multi-stream feature extraction of ResGCN (Song et al., 2020), Teepe et al. (2021); Teepe et al. (2022) further enhanced this idea by combining bone and motion data with joints. A similar approach was followed by Wang et al. (2022a). Recently, Hasan et al. (2022) utilized the hop-extraction technique to provide equal importance for long-range and short-range relationships among joints to improve the recognition performance compared to the earlier works. However, the authors did not consider prioritizing crucial joint information. One of the critical limitations of GCNs is that their performance often varies based on the extracted poses. Nowadays, the advancement in pose estimation networks has made them a viable alternative for extracting joint positions for GCN-based gait recognition (Toshev and Szegedy, 2014; Pishchulin et al., 2016; Cao et al., 2021; Fang et al., 2017; Zhang et al., 2020a). Despite that, any wrongly estimated pose can affect the performance of GCNs in gait recognition. Attention mechanisms can be used in this regard to avoid focusing on poorly estimated poses, while also prioritizing important body parts that are helpful in gait recognition. Since human gait involves the coordination between various body joints, in conclusion, GCNs can be considered as the most suitable approach to effectively capture this relationship compared to other architectures.

All the issues discussed above necessitate a gait recognition pipeline that can effectively capture the spatial and temporal relationship of gait that does not overfit and can prioritize important body parts using low computational resources.

3. Proposed method

The GaitGCN++ pipeline for the skeleton joint position-based gait recognition system comprises of multiple stages. Initially, the joint positions depicting the human pose are determined by feeding each frame of the gait video sequence into a pose estimation network. These joint positions are then utilized to generate the graph representation of the gait sequences by considering the joint positions as vertices and the connection between joints (bones) as edges. Next, the sequence is preprocessed to remove low-confidence predictions. After that, the preprocessed joint sequence is passed through a graph convolutional network where the bone structure and joint-motion features are generated along with the joint positions. Each data stream is then fed to a set of basic convolution and residual bottleneck blocks consisting of graph convolution and temporal 2D convolution layers. The graph convolution layers are enhanced using the hop extraction technique for multi-scale feature aggregation that can extract a comprehensive understanding of gait considering the relationship between closer and further joints. The residual bottleneck blocks in the deeper layers are combined with a 'Part-wise Attention module' to identify and prioritize specific body parts that are useful in gait recognition. The 'DropGraph' regularization technique is employed in the training phase of the network to avoid overfitting while also learning generalized features that are useful in identifying unseen gait samples. Finally, based on the activation maps generated from gait sequences, class labels are generated identifying the subjects to whom the sequence pertains. A pictorial view of the proposed pipeline can be seen in Fig. 1.

3.1. Estimation of joint pose

A gait video sequence consists of N frames of RGB images $f_1, f_2, \ldots, f_N$. Each image is fed to a pose estimation network to extract K key points representing K joints. In this work, we employ Higher Resolution Net (HRNet)[1] (Sun et al., 2019) to extract the key points from each frame. The network follows the typical pose estimation pipeline composed of a stem, a main body, and a regressor. The stem contains two strided convolution layers that decrease the resolution of the input. The main body outputs a feature map having the same size as the input. The regressor estimates the heatmaps generating the key points.

Let us consider that each RGB image has a size of $W \times H \times C$, where W, H, and C denote the width, height, and the number of channels in the image, respectively. To generate K key points from each RGB image $f_i (1 \leqslant i \leqslant N)$, HRNet employs a sequential high-resolution subnetwork to maintain the high resolution of the image throughout the pose estimation process. It connects high-to-low and low-to–high subnetworks in parallel to exchange information via a set of downsampling and upsampling process. This generates K heatmaps with size $W\prime \times H\prime$ and a set $\{M_1, M_2, \ldots, M_K\}$. Here, $M_i (1 \leqslant i \leqslant K)$ denotes the confidence of the network in predicting the ith keypoint. K key points are extracted from the heatmaps that denote the joint positions representing the vertices of the graph. For HRNet, $K = 17$. The joint connections representing the edge information are created using the configuration provided by Teepe et al. (2021) (Fig. 2).

3.2. Preprocessing

Along with the 2D coordinates of the joints, HRNet also generates a confidence score for each of its predictions. The higher the confidence score, the better the prediction. In occluded conditions, such as walking while carrying a bag and walking while heavily clothed, the confidence for the generated 2D coordinates can be significantly low. At the same time, the relative orientation and the distance of the subject can affect the estimation accuracy of the 2D coordinates due to scale and noise (Liu et al., 2022a). These coordinates, being a feature and the source for generating other data streams, play an important role in gait recognition. However, as illustrated in Fig. 3, the predictions can have low confidence resulting in a poor estimation of the joint position.

We hypothesize that any joint coordinate with low confidence can capture random noise hampering both the training process and consequently the recognition performance of the graph convolutional network. The average prediction confidence of HRNet for a particular frame is calculated as follows:

$$\text{Confidence} = \frac{\sum_{i=1}^{K} c_i}{K} \times 100\% \tag{1}$$

Here,

c_i = Confidence score for the ith joint,
K = The number of key points.

After that, if the average confidence of the frame is less than a threshold, T_C, the frame is removed to alleviate the effect of poor predictions.

In addition, since the GCN learns to recognize gait based on the relationship between different joints in both spatial and temporal dimensions, missing information due to an error in the calculation of the coordinates can have a detrimental effect on the training and recognition process (Teepe et al., 2021). To address the issue, if any frame did not contain the 2D coordinates and the corresponding confidence score of all 17 joints, that frame was also removed.

3.3. Graph convolutional network

Graph Convolutional Networks (GCN) are designed to work with graph-structured data (Bhatti et al., 2023; Ren et al., 2022). Graphs are mathematical representations of collections of objects and the connections among them, comprised of separate nodes and edges. In a GCN, the nodes of the graph are associated with feature vectors, and the objective is to learn a function that maps these feature vectors to some downstream tasks e.g., node classification, link prediction, graph classification, etc. (Cao et al., 2022). Convolutional operations are carried out on the graph using filters that account for its local structure. The key insight is to use a variant of the standard convolutional operation from image processing, but instead of convolving over a regular grid of pixels, the convolution is performed over the graph structure. The graph convolutional operation can be expressed mathematically as follows:

$$h_i^{l+1} = \sigma\left(\sum_{j\epsilon\mathcal{N}(i)} \frac{1}{C_{ij}} W^{(l)} h_j^{(l)}\right) \tag{2}$$

where $h_i^{(l)}$ represents the feature vector associated with node i in the l-th layer of the network, $W^{(l)}$ is a weight matrix associated with the l-th layer, $\sigma(\cdot)$ is an activation function, $\mathcal{N}(i)$ is the set of neighbors of node i, and c_{ij} is a normalization constant that depends on the degree of nodes i and j. The aforementioned equation can be interpreted as a weighted sum of the feature vectors of the neighbors of node i, where the weights are given by the entries of the weight matrix $W^{(l)}$, scaled by the inverse of the normalization constant c_{ij}. The activation function $\sigma(\cdot)$ introduces non-linearity into the model.

In this regard, Residual Graph Convolutional Network (ResGCN) (Song et al., 2020) incorporates the bottleneck structure and residual links to improve the efficiency and effectiveness of spatio-temporal graph convolutional networks. The bottleneck structure involves inserting two 1×1 convolutional layers before and after the existing convolutional layer, reducing the number of feature channels with a reduction rate, r. By replacing the spatial and temporal basic blocks with the bottleneck structure, the model achieves a faster implementation of model training and inference. The ResGCN module consists of a sequential execution of one spatial block and one temporal block, and residual links are added over the blocks. The residual links can be of three types: block, module,

[1] https://github.com/HRNet/HRNet-Human-Pose-Estimation.

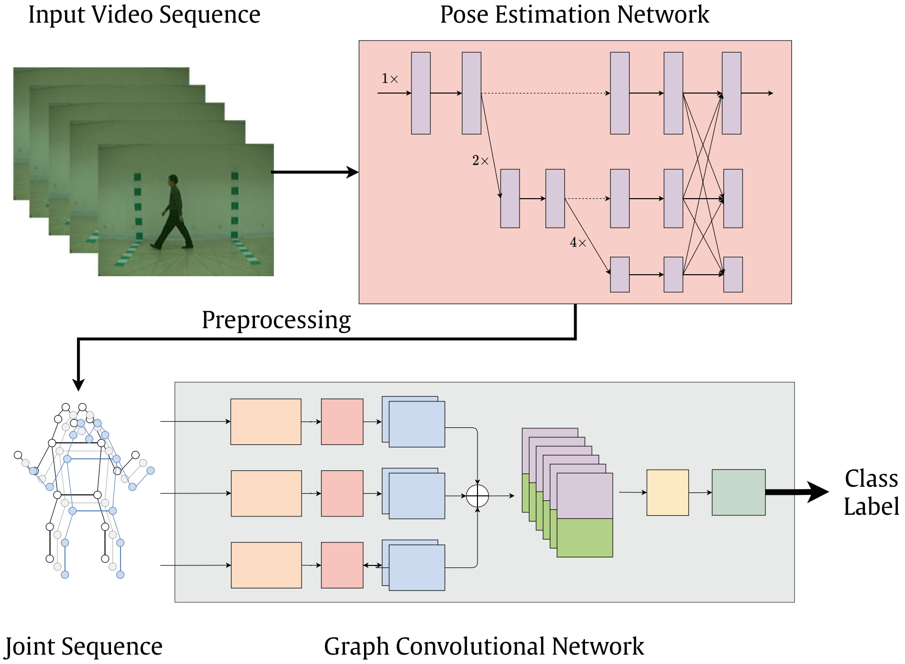


Fig. 1. Overview of the proposed pipeline for gait recognition. Joint poses are extracted from input video sequences using the pose estimation architecture, HRNet. The extracted joints are preprocessed and fed to the Graph Convolution Network, ResGCN which generates a feature vector used to determine the class label.

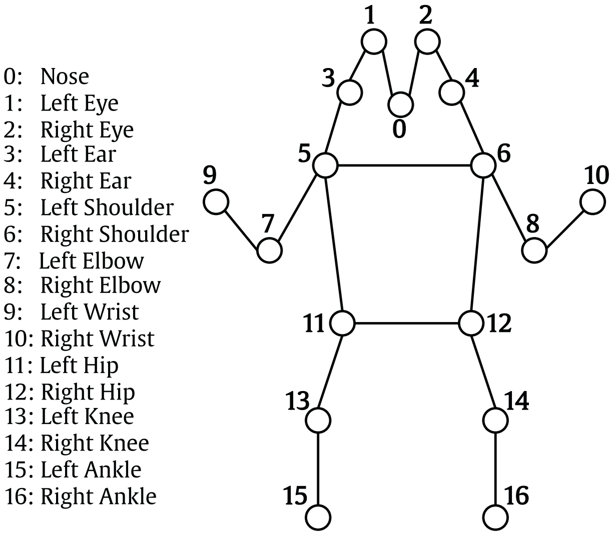


Fig. 2. Joint data extracted from HRNet. The joints are labeled using numbers and their corresponding names are shown on the left. The bone configuration is taken from Teepe et al. (2021).

and dense, with the appropriate type selected to balance the trade-off between the compactness of the model and the memory costs.

ResGCN architecture, pretrained on two benchmark action recognition datasets NTU RGB + D (Shahroudy et al., 2016) and NTU RGB + D 120 (Liu et al., 2019), has been adapted to our task. The code of the original model, along with the pretrained weights, are available online [2].

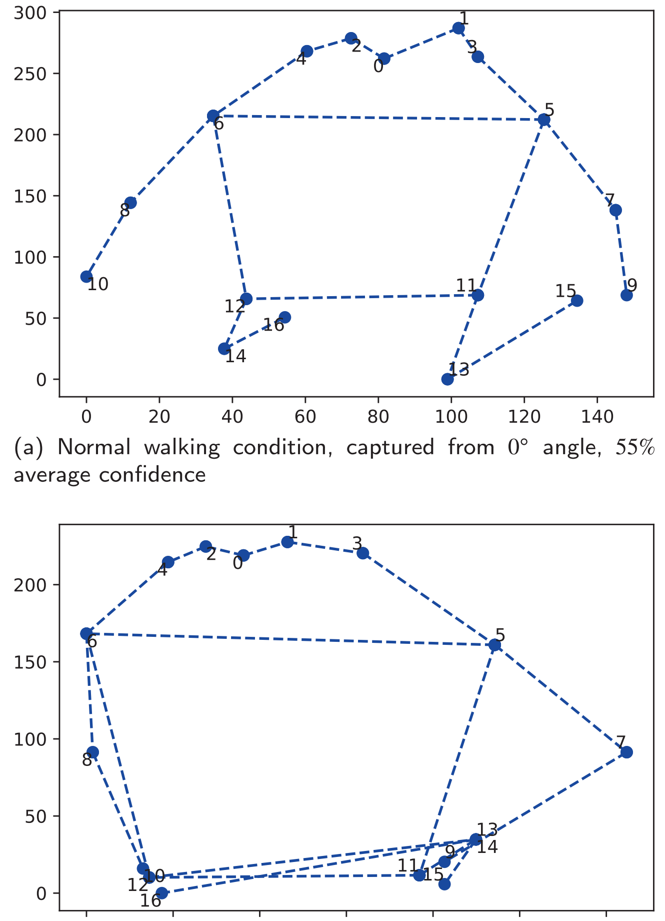
As shown in Fig. 4, the joint, bone, and joint motion data, combined in a batch normalized multi-stream input, undergoes processing through a sequence of ”Basic” blocks. These blocks, comprising convolution and batch normalization layers, have the option of incorporating a residual connection in some cases. The output of each Basic block is generated by applying a ReLU activation function to it. The input is then reconstructed using “Bottleneck” blocks, which consist of up-convolution and down-convolution layers with batch normalization in between. If the Bottleneck block is residual, the input is transformed with convolution and batch normalization before being added to the output. The final output is produced by passing the output of the Bottleneck blocks through a ReLU activation function, followed by average pooling and mapping to the output units through a fully-connected layer.

3.3.1. Multi-stream input

In addition to joint information generated using HRNet, bone and joint-motion information is fed to the ResGCN architecture to provide supplementary insights to the model. Each stream can capture different aspects of the data. For instance, the joint stream provides information about the relative positions of the body parts, while the bone stream encodes information about the skeletal structure. The joint-motion stream, on the other hand, captures the temporal dynamics of the movements. By using all three streams, the model can capture a more comprehensive representation of the input data. At the same time, each stream can specialize in feature extraction relevant to its type of data, allowing the model to extract richer and more diverse features from the input data. The extracted features can then be combined to create a more robust representation of gait. In addition to that, the use of multiple streams can help resolve ambiguity in the input data. For instance, in the case of occlusion, different joints can be hard to detect, and combining information from multiple streams can help to compensate for missing/erroneous data. Last but not the least, the evaluation of such multi-stream features early in the network can help reduce the overall complexity of the model by reducing the number of layers that would have been required otherwise to infer such features deep in the network (Xu et al., 2021; Zhang et al., 2023).

Bone Stream A bone is considered as a link between two joints. In our work, we define the bone as a vector pointing from one joint

[2] https://gitee.com/yfsong0709/ResGCNv1.



(a) Normal walking condition, captured from 0° angle, 55% average confidence

(b) Heavily clothed condition, captured from 0° angle, 58% average confidence

Fig. 3. Illustration showing poorly predicted joint positions by the pose estimation network, HRNet.

to another. We consider the coordinates of the joints in the 2D plane as vectors with respect to the origin, $(0, 0)$. Let us assume that a joint in frame $p(0 \leqslant p < N)$ is denoted as $\overrightarrow{v}_{i,p} = (x_{i,p}, y_{i,p})$ where $x_{i,p}$ and $y_{i,p}$ indicates the x and y-coordinate of the joint i, respectively. Let another joint be denoted as $\overrightarrow{v}_{j,p} = (x_{j,p}, y_{j,p})$ where $x_{j,p}$ and $y_{j,p}$ indicates the x and y-coordinate of the joint j, respectively. Then we calculate the difference in the x and y values between the two vectors to define the bone vector, $\overrightarrow{b}_{i,p}$ as:

$$\begin{aligned} \overrightarrow{b}_{i,p} &= \overrightarrow{v}_{j,p} - \overrightarrow{v}_{i,p} \\ &= (x_{j,p} - x_{i,p}, y_{j,p} - y_{i,p}) \end{aligned} \tag{3}$$

The resulting vector, $\overrightarrow{b}_{i,p}$ points from $(x_{i,p}, y_{i,p})$ to $(x_{j,p}, y_{j,p})$. This vector represents the displacement between the two points in the same frame encoding the bone structure. It is calculated for each possible pair of joints that are connected by an edge in each frame as defined in the earlier section. The introduction of these features, as shown in Fig. 5(a), can encode rich structural information that may be helpful for the network to understand how the joints are connected and provide further insight into their interaction.

Motion Stream The motion data indicates the change of coordinates for the same joint in subsequent frames. We define the joint-motion as a vector pointing from one joint in one frame to the same joint in the subsequent frame (Fig. 5(b)). Let us assume that a joint in frame $p(0 \leqslant p < N - 1)$ is denoted as $\overrightarrow{v}_{i,p} = (x_{i,p}, y_{i,p})$. Then, the same joint in the subsequent frame $(p + 1)$ is denoted as $\overrightarrow{v}_{i,p+1} = (x_{i,p+1}, y_{i,p+1})$. The joint-motion vector, $\overrightarrow{jm}_{i,p}$ can be calculated by considering the difference in the x and y values between the two joints as:

$$\begin{aligned} \overrightarrow{jm}_{i,p} &= \overrightarrow{v}_{i,p+1} - \overrightarrow{v}_{i,p} \\ &= (x_{i,p+1} - x_{i,p}, y_{i,p+1} - y_{i,p}) \end{aligned} \tag{4}$$

The resulting vector, $\overrightarrow{jm}_{i,p}$ points from $(x_{i,p}, y_{i,p})$ to $(x_{i,p+1}, y_{i,p+1})$. This vector represents the displacement between the two points in consecutive frames encoding the motion information. It is calculated for each pair of consecutive frames in the gait video sequence. The introduction of motion features in this manner can encode the temporal information present in the graph structure of the gait sequence.

3.3.2. Bottleneck and residual connection

ResGCN utilizes a bottleneck structure with residual connections, pioneered by ResNet (He et al., 2016), for faster convergence and to reduce the performance tuning cost. The bottleneck architecture, as shown in Fig. 6, is composed of a sequence of spatial graph convolution module and temporal 2D convolution module. The graph convolution module and the temporal convolution module can help aggregate information from a single frame of a data stream and multiple frames of a data stream, respectively.

The bottleneck structure enables the reduction of feature channels by using two 1×1 convolution layers right before and after regular convolution layers. Before each spatial and temporal block, the bottleneck structure is used to reduce the number of parameters in the overall architecture. This results in faster optimization of the model.

With the increase in the number of layers in GCN, one unstable behavior can be noticed. Since gradients in the deeper layers are

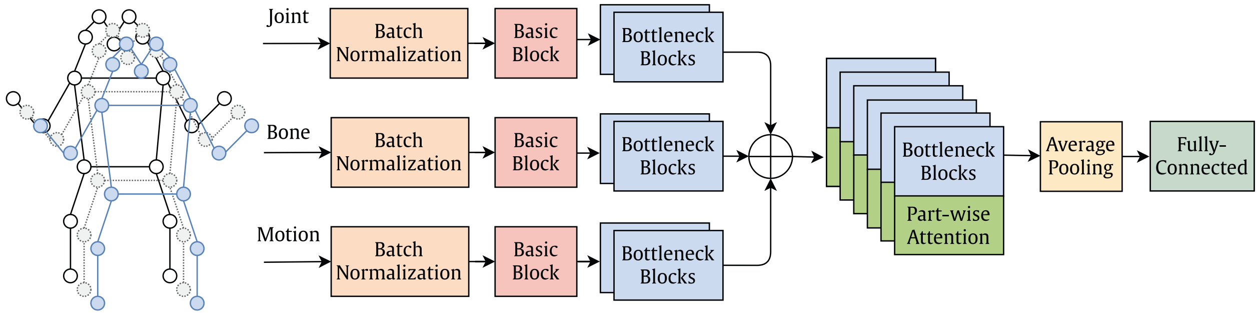
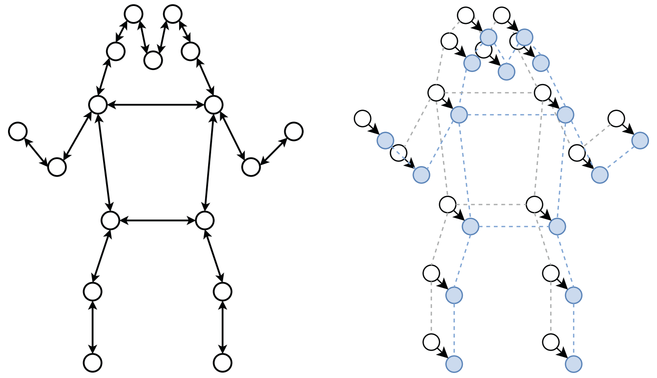


Fig. 4. Overview of the ResGCN architecture. Adapted and modified from Song et al. (2020).



(a) Generated bone data stream. Here, the directions indicate that the bone data was generated considering both directions.

(b) Generated motion data stream. Here, the black nodes are in frame f_p and the blue nodes are in frame f_{p+1}.

Fig. 5. Data streams generated from joint positions.

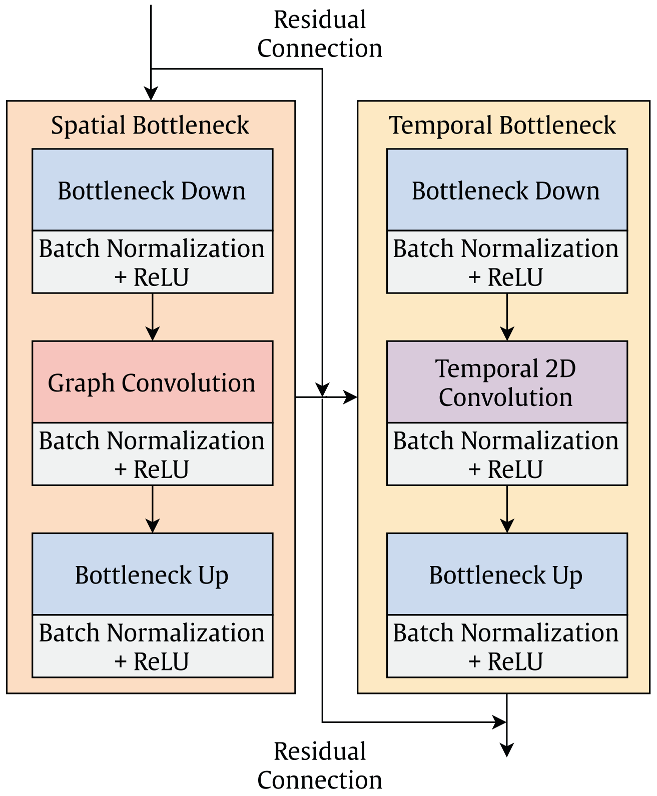


Fig. 6. Structure of ResGCN Bottleneck with Block Residual Links. It includes a module for spatial graph convolution and another module for temporal 2D convolution. These modules collaborate to collect information from single frames and multiple frames, respectively.

calculated as the product of several gradient values, smaller gradients tend to have minimal effect on the weight update, resulting in longer convergence time of the model. This is known as the vanishing gradient problem (Song et al., 2020). The residual connection is used as a skip connection between the spatial and temporal bottleneck blocks. It can help propagate larger gradients to earlier layers resulting in faster convergence of the model (He et al., 2016). Here, block residual connections are used to facilitate better feature learning as suggested by Song et al. (2020).

3.3.3. Hop extraction technique

Let, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the human skeleton graph. Here, $\mathcal{V} = \{v_1, v_2, \ldots, v_P\}$ denotes the set of P vertices corresponding to each keypoint in a data stream. For the joint and bone data stream, $P = M$ and for joint-motion data stream, $P = M - 1$. Set $\mathcal{E}$ denotes the edges corresponding to the connections between each keypoint. Each vertex consists of a pair of values and the edge information is represented using an adjacency matrix $\mathbf{A} \in \mathbb{R}^{P\times P}$ where

$$A_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Note that, $\mathbf{A}$ encodes the structural information of the skeleton and is symmetric due to the undirected nature of $\mathcal{G}$.

For each data stream, the human gait can be represented using a set of vertex features $\mathbf{X} = \{x_{i,j} \in \mathbb{R}^C \,|\, i,j \in \mathbb{Z}; 1 \leqslant i \leqslant N; 1 \leqslant j \leqslant P\}$ where $x_{i,j} \in \mathbb{R}^C$ for vertex v_j at frame i. That means, $\mathbf{X} \in \mathbb{R}^{N\times P\times C}$. And since HRNet extracts 2D coordinates for each joint, $C = 2$. Again, $\mathbf{X}$ encodes the feature information for the skeleton.

The layer-wise update function that is applied on frame f is:

$$X_f^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}_f^{(l)}\Theta^{(l)}\right) \tag{6}$$

Here,

$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, that conserves the identified features of the skeleton by introducing self-loops,
$\tilde{\mathbf{D}}$ = A diagonal matrix comprising the degrees of $\tilde{\mathbf{A}}$ along its main diagonal,
$\Theta^{(l)}$ = The weight matrix for layer l that can be updated during the learning process, and.
$\sigma(\cdot)$ = The sigmoid activation function.

The purpose of the diagonal degree matrix, $\tilde{\mathbf{D}}$ is to normalize the features to prevent vanishing/exploding gradients (Kipf and Welling, 2017).

The spatial aggregation framework utilizes higher-order polynomials of the adjacency matrix to aggregate multi-scale structural information.

$$X_f^{(l+1)} = \sigma\left(\sum_{k=0}^{K}\hat{\mathbf{A}}^k\mathbf{X}_f^{(l)}\Theta_{(k)}^{(l)}\right) \tag{7}$$

Here,

K = The scale of aggregation,
$\hat{A} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$.

Note that, $A_{i,j}^k = A_{j,i}^k$ is the total number of length k walks between v_i and v_j. That means, $\hat{\mathbf{A}}^k\mathbf{X}_f^{(l)}$ can be used to perform feature aggregation weighted by the number of such walks.

As illustrated in Fig. 7, due to the nature of walks involving hops between vertex i and vertex j, with the possibility of i being the same as j, cyclic walks can occur, primarily centered around the originating vertex. Moreover, the inclusion of self-loops, which aim to preserve identity features, can contribute to the proliferation of such walks. Consequently, the adjacency matrix exhibits higher values for vertices in close proximity to the starting vertex and lower values for those located farther away. This creates a bias in feature aggregation, rendering the process less effective in capturing the long-range relationship between joints (Liu et al., 2020; Hasan et al., 2022). This issue is addressed by defining a k-adjacency matrix as:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

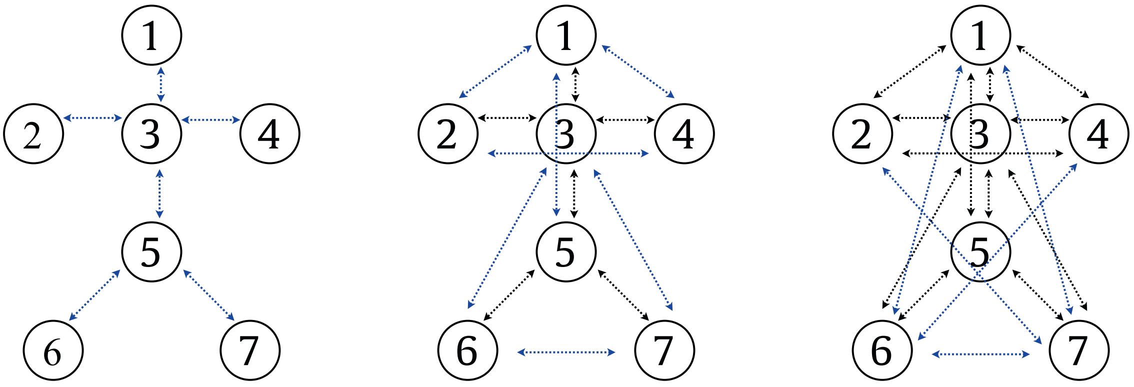
(a) Adjacency matrix for A^1

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

(b) Adjacency matrix for A^2

$$\begin{bmatrix} 0 & 0 & 4 & 0 & 0 & 1 & 1 \\ 0 & 0 & 4 & 0 & 0 & 1 & 1 \\ 4 & 4 & 0 & 4 & 6 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 1 & 1 \\ 0 & 0 & 6 & 0 & 0 & 3 & 3 \\ 1 & 1 & 0 & 1 & 3 & 0 & 0 \\ 1 & 1 & 0 & 1 & 3 & 0 & 0 \end{bmatrix}$$

(c) Adjacency matrix for A^3



(d) Corresponding joint connections for A^1

(e) Corresponding joint connections for A^2

(f) Corresponding joint connections for A^3

Fig. 7. Illustration of the Biased Weighting problem considering 7 joints with length 3 walks. Here, self-loops are avoided for simplicity. The blue connections are the only joints that need to be considered. The black connections are redundant but still considered due to the property of walks in a graph.

$$\left[\tilde{\mathbf{A}}_{(k)}\right]_{i,j} = \begin{cases} 1, & \text{if } d(v_i, v_j) = k \\ 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Here,

$d(v_i, v_j)$ is the shortest distance between v_i and v_j considering the number of hops.

Given that $\mathcal{G}$ is an undirected graph, the values are determined by employing the Breadth-First Search (BFS) algorithm (Zuse, 1972; Moore, 1959; Lee, 1961) to identify the k-hop neighbors. The algorithm starts from a certain joint and explores all joints in the current depth before moving on to the next one.

Now, consider that, $\tilde{\mathbf{A}}_{(1)} = \tilde{\mathbf{A}}$ and $\tilde{\mathbf{A}}_{(1)} = \mathbf{I}$. Thereafter, incorporating Eq. 8 with Eq. 7, we get:

$$X_f^{(l+1)} = \sigma\left(\sum_{k=0}^{K} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \tilde{\mathbf{A}}_{(k)} \tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \mathbf{X}_f^{(l)} \Theta_{(k)}^{(l)}\right) \tag{9}$$

Unlike Eq. 7, which relies on the number of length $k-1$ walks to determine the total number of length k walks, Eq. 9 addresses the issue of biased weighting by assigning equal significance to vertices in both closer and further neighborhoods. As a result, this approach effectively takes into account long-range relationships between joints. A sample illustration considering 7 joints and hop distance of 3 can be seen in Fig. 8.

3.3.4. DropGraph regularization

Since ResGCN has to handle multiple data streams in a large network, it is very likely that overfitting issues can occur in the model (Cheng et al., 2020). As a result, the model might try to learn the structural noise in the training data instead of learning the underlying patterns and characteristics of the gait sequence, which in turn can have negative consequences on the performance of the model in unseen test data.

Our initial experiments with different baseline architectures of the existing state-of-the-art model showed that even after careful consideration via augmentation of training samples and various measures taken to control the convergence, the difference between the training and test accuracies was substantial. For example, one of the baseline architectures had a remarkable 88.60% training accuracy. However, the test accuracy was only 69.90%. These results further enforced the notion of overfitting in ResGCN while used in our task. The introduction of a regularization technique can help mitigate the risk of overfitting and ensure that the model learns meaningful representations from the available training data.

One simple regularization technique to avoid overfitting could be the use of dropout layers (Srivastava et al., 2014). During training, the dropout layer is used to ignore a set of randomly selected key points. In a graph convolutional network, this would translate to randomly selecting a set of vertices and setting their activation to 0 (Fig. 9(a)). This minimizes the complex co-adaptation of the network layers forcing the network to learn a sparse representation that is more helpful in identifying unseen samples.

However, as illustrated in Fig. 9(b), due to the closely related nature of the key points, the features of a key point can be estimated from the features of the neighboring key points. This is due to the fact that graph convolution is a unique variant of Laplacian smoothing that combines the characteristics of a node with those of its neighboring nodes (Li et al., 2018). As a result, if a particular node is dropped, the information for that node can be extracted from the neighboring nodes as activation units are correlated between neighbor nodes, leading to overfitting (Cheng et al., 2020). To solve this issue, DropGraph is utilized by dropping the entire node set in a neighborhood.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

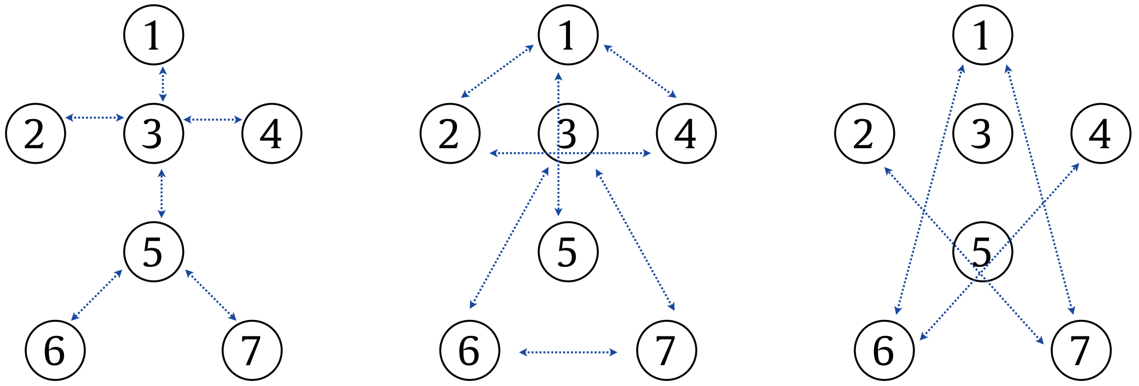
(a) Adjacency matrix for A_1

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(b) Adjacency matrix for A_2

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) Adjacency matrix for A_3

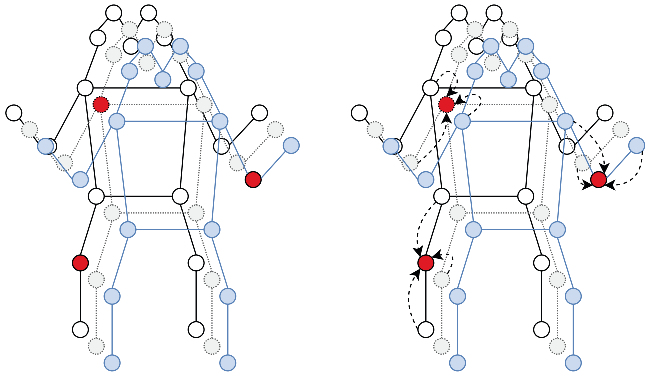


(d) Corresponding joint connections for A_1

(e) Corresponding joint connections for A_2

(f) Corresponding joint connections for A_3

Fig. 8. Solution to the Biased Weighting problem using hop extraction considering 7 joints and up to hop distance 3. Here, self-loops are avoided for simplicity.



(a) Dropout applied to graph convolutional network. Vertices are randomly selected (red colored) and their activation is set to 0.

(b) Estimation of a keypoint based on neighboring key points resulting in the nullification of the effect of dropout. (A few of the possible connections are shown for simplicity)

Fig. 9. Dropout regularization technique and its problem in graph convolutional network. Using dropout layers is a common regularization technique to prevent overfitting. However, in graph convolutional networks, dropping vertices may not effectively remove their contribution due to the shared information among neighboring vertices. This can lead to overfitting as the dropped vertices' information can still be estimated from neighboring vertices.

The DropGraph regularizer is used to generate distortions from data (Xiang et al., 2021). It helps deep neural networks to prevent feature co-adaptations. The network samples stochastic spatial feature vectors and then incorporates graph reasoning methods to generate feature map distortions. The technique is particularly effective as a regularizer for graph convolutional networks as it promotes better adaptation to unseen or perturbed inputs, improving the model's ability to handle variations and noise in the data (Cheng et al., 2020).

The key idea behind DropGraph regularization involves randomly dropping a fraction of the graph connections or edges during training, effectively inducing sparsity in the graph structure. It selects a vertex v_{base} from the graph with a small probability λ. Then, the nodes that are at most T_D steps away from v_{base} consist in a region called the Drop Area. The area includes v_{base} as well. All the nodes in the Drop Area are dropped by setting their activation to 0. The activation map is then normalized with respect to the number of nodes dropped. Since v_{base} is selected based on a Bernoulli distribution with probability λ, the expected size of the Drop Area, $E[drop]$ can be estimated via:

$$E[drop] = 1 + \sum_{i=1}^{T_D} \left(d_M \times (d_M - 1)^{i-1} \right) \tag{10}$$

Here,

d_M = The average degree of each node, calculated using $\frac{2l}{k}$.
l = Number of edges.
k = Number of nodes.

The probability that a certain unit of the activation map will be kept, $P[keep]$ is calculated using:

$$P[keep] = 1 - \lambda \times E[drop] \tag{11}$$

It should be noted that there could be instances where the Drop Areas overlap, thus it is important to recognize that Eq. 11 and Eq. 10 serve as approximate representations. In our experiments, we set the value of $P[keep]$ to 0.90, as per the recommendation of Cheng et al. (2020), and then calculate the value of λ accordingly.

Note that, in gait recognition, the input to the DropGraph is a spatio-temporal activation map. Hence, our implementation consists of two components: Spatial DropGraph and Temporal DropGraph.

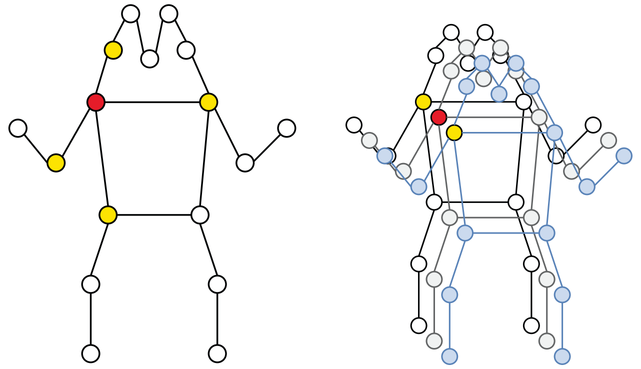

(a) DropGraph applied in spatial dimension (b) DropGraph applied in temporal dimension

Fig. 10. Demonstration of DropGraph applied in both spatial and temporal dimensions. Here, the red node denotes the V_{base} node and the yellow nodes are its neighbors, where $K = 1$.

The spatial component of the graph corresponds to the physical structure of the human body, which includes a specific set of joints. To obtain spatial information for each frame, we compress the activation map's absolute values by applying average pooling across both the channel and temporal dimensions. Once we sample v_{base}, we extend the Drop Area to encompass its neighboring spatial nodes. Subsequently, we distribute the Drop Area information to all temporal frames. This approach is referred to as Spatial DropGraph.

On the other hand, the temporal component of the graph corresponds to the linking of consecutive frames along the temporal dimension. To obtain temporal information for each frame, we compress the activation map's absolute values by applying average pooling across both the channel and spatial dimensions. Once we sample v_{base}, we extend the Drop Area to encompass its neighboring temporal nodes. Subsequently, we distribute the Drop Area information to all body joints. This approach is referred to as Temporal DropGraph.

By incorporating both Spatial and Temporal DropGraph, we aim to introduce regularization effects into the network by randomly dropping nodes while ensuring that the dropped values cannot be reconstructed from neighboring nodes. This approach encourages sparsity in both the spatial and temporal dimensions, facilitating improved performance in gait recognition tasks. The processes of applying the DropGraph approach in spatial and temporal dimensions are illustrated in Fig. 10.

3.3.5. Part-wise attention

Not all body parts contribute equally to gait. For instance, while walking, hand and leg movements play a crucial role. Therefore, it might be beneficial to focus on the changes in joint positions that are related to these body parts to recognize gait better. By prioritizing the relevant joints, a gait recognition system can effectively capture the distinctive features and patterns related to hand and leg movements, leading to improved accuracy in recognizing gait. Another benefit of prioritizing important body parts is that it can also learn to ignore noisy joints. Since we use an off-the-shelf pose estimation network trained on a different dataset, it may not be able to predict the joint positions accurately, which can negatively impact the performance of the gait recognition system. One solution could be to make the system learn to provide lower importance to these noisy joints during the training phase. This ability to disregard unreliable joints across the training samples can result in better performance when dealing with unseen test samples, where similar inaccuracies may be present. To this end, attention mechanisms provide a means to selectively emphasize or suppress specific parts of the input data (Vaswani et al., 2017). In the context of our task, attention mechanisms can allow the model to assign higher importance to the joints associated with hand and leg movements (for example), enabling it to capture the most discriminative features for gait recognition. By focusing on these crucial body parts, the model becomes more sensitive to the relevant patterns and variations, while also ignoring noisy joints, leading to improved recognition accuracy.

Motivated by the Split Attention of the ResNeSt model (Zhang et al., 2020b), a Part-wise Attention block is used to understand the importance of different body parts and obtain more explainable representations for different gait sequences. It is a neural network that can be used to focus on specific parts of an input. In general, it works by dividing the input into different parts and then calculating the attention for each part. The attention values are then used to weight the features of each part before they are combined and passed through the rest of the network. This allows the network to focus on the most important parts of the input and can improve its performance on tasks such as classification or recognition.

As depicted in Fig. 11, the attention weights are learned by a Multilayer Perceptron (MLP) consisting of two layers with sigmoid activation functions. The output of the MLP is then normalized across all parts to ensure that the sum of all attention weights is equal to one. The final feature representation is obtained by pooling the weighted feature maps of all parts. To achieve this, the joints are divided into five body parts to apply Part-wise Attention. The joints for each part are selected manually from the input feature following the suggestions of Song et al. (2020). The features of each part are then concatenated and passed through a sequence of average pooling in the temporal dimension, fully-connected, batch normalization, and ReLU layers. Average pooling in the temporal dimension helps extract global contextual feature maps as opposed to spatial attention that focuses on each frame individually. The batch normalization layer stabilizes the learning process by distributing the weights of each layer in a distributed manner (Lyu et al., 2022). Applying the ReLU activation function ensures that only positive values are considered, effectively emphasizing the importance of certain parts in the input features. This non-linearity allows the attention mechanism to assign higher weights to more relevant or informative features, helping the model to focus on the most salient parts of the input and improve its discriminative power (Agarap, 2018). After that, five fully-connected layers and a part-level softmax are used to calculate the attention of each part and multiplied by the feature values of the respective parts. That means the feature values of each part, p are calculated using:

$$f_p = f_{in}(p) \otimes \delta(\theta(pool(f_{in})W)W_p) \tag{12}$$

Here,

f_{in} = Input feature map.
$\otimes$ = Point-wise multiplication.
$pool(\cdot)$ = Temporal average pooling.
$\delta(\cdot)$ = Part-level softmax.
$\theta(\cdot)$ = ReLU Activation.

W and W_p are learnable parameters, where W is shared among all body parts that help in dimension reduction and W_p is calculated for each body part, denoting the attention weights. Finally, the feature vectors are concatenated to rebuild the entire skeleton representation, f_{out}.

$$f_{out} = Concat(\{f_p | p = 1, 2, \ldots, 5\}) \tag{13}$$

The module is designed to enhance the explainability and stability of the model as it can now provide insights into which features or areas contribute the most to the final prediction. At the

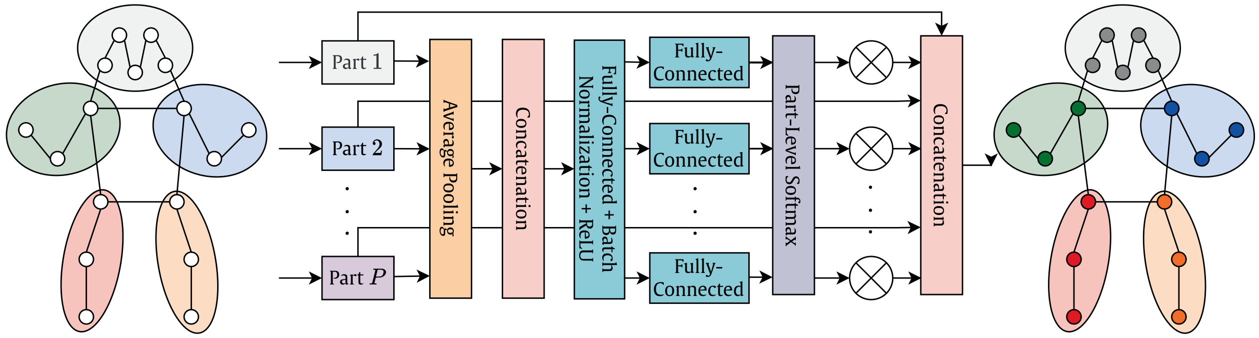


Fig. 11. Part-wise Attention Module used to compute attention weights for different body parts to further improve the discriminative capability of the features. The joints are divided into five body parts and the features of each part are concatenated and passed through a sequence of average pooling in the temporal dimension, fully-connected, batch normalization, and ReLU layers to extract global contextual feature maps. Five fully-connected layers and a part-level softmax are then used to calculate the attention of each part and multiplied by the feature values.

same time, the module can reduce computational costs in model training by selectively attending to informative parts, enabling the model to effectively allocate its resources and computation to the most relevant parts. All these are benefits can be achieved while also gaining a commendable performance due to the model being more robust to variations, distractions, or irrelevant information as it can ignore noisy parts of the input and at the same time, the model can capture context-dependent relationships of the input activation maps to make more informed decisions.

3.4. Class label generation

In accordance with the recommendation provided by Chao et al. (2019), we adopt a subject-independent protocol for generating class labels. This protocol ensures that the training set and test set are mutually exclusive, meaning that the subjects included in the training set do not appear in the test set. Our pipeline, GaitGCN++, is trained using the gait sequences from the training set.

To assess the performance of our pipeline, we further divide the gait sequences in the test set into two subsets: the gallery set and the probe set. We then extract activation maps from both sets using GaitGCN++. In order to assign a class label to a probe sequence, we employ a classifier that compares the activation map of the probe sequence with the maps of all the sequences in the gallery set. The objective is to identify the most similar gait patterns by computing the closest normalized distance. Based on this comparison, we can label the probe sequence as belonging to the same subject as the gallery sequence with the closest resemblance.

This subject-independent protocol is widely used in gait recognition studies as it allows us to evaluate the generalization capability of the GaitGCN++ pipeline across different individuals. By ensuring that the training and test sets do not overlap in terms of subjects, we can assess the effectiveness of our approach in recognizing gait patterns from unseen individuals. This evaluation helps us understand the effectiveness of our approach in recognizing gait patterns from individuals who were not seen during the training phase, enhancing the overall reliability and applicability of our pipeline.

4. Results and discussion

4.1. Dataset

To evaluate the performance of the proposed pipeline, we utilize two of the largest and most popular gait recognition datasets, CASIA-B (Yu et al., 2006) and GREW (Zhu et al., 2021).

4.1.1. CASIA-B

The CASIA-B dataset is one of the most widely used datasets for evaluating the performance of gait recognition architectures. It has been used in our work to perform hyperparameter tuning, ablation study, and comparison with state-of-the-art models. The dataset provides RGB videos of 124 subjects (93 male, 31 female). For each subject, ten videos are provided considering three walking conditions: 6 of them captured the subject walking normally (NM), 2 of them captured the subject walking while carrying a bag (BG), and 2 of them captured the subject walking while heavily clothed (CL). Each video sequence is captured simultaneously from 11 angles, namely: $0^\circ, 18^\circ, 36^\circ, 54^\circ, 72^\circ, 90^\circ, 108^\circ, 126^\circ, 144^\circ, 162^\circ$, and 180°.

4.1.2. GREW

To better understand the effectiveness of our pipeline in real-life scenarios, we have experimented with GREW dataset (Zhu et al., 2021), as it has been constructed in the wild. The dataset contains 128,671 sequences from 26,345 subjects, captured using 882 cameras. It provides silhouettes, flow information, and 2D/3D poses estimated for each subject. The variety in human attributes (age group, gender), carrying conditions, and clothing of the captured subjects in unconstrained environments poses a significant challenge for gait recognition systems.

4.2. Experimental setup

4.2.1. Environment

The proposed system was implemented in Google Colaboratory[3] using PyTorch framework[4]. The environment provides an NVIDIA Tesla T4 GPU with a VRAM of 11 GB and CUDA version 11.2. An Intel Xeon CPU with two cores, each having a base clock speed of 2.2 GHz and 55 megabytes of cache was used. The total usable memory of the system was 13 GB.

4.2.2. Dataset split

Due to the lack of any official split for CASIA-B, the dataset split recommended by Sepas-Moghaddam and Etemad (2022) was used. The first 74 subjects were kept in the training set and the rest in the test set. Further, the test set was divided into a gallery set and a probe set. The gallery set contains the first four clips in normal walking conditions (NM01 - NM04). The remaining clips (NM05 - NM06, CL01 - CL02, BG01 - BG02) were kept in the probe set. The GCN architecture extracted features from both gallery sets and probe sets. Then the features of the probe set were matched with the features from the gallery set to determine the most

[3] https://colab.research.google.com/.

[4] https://pytorch.org/.

similar sample as the class label. The test results were reported considering the accuracy averaging over 11 gallery views excluding the identical views. The normal walking condition of the gallery set can be beneficial in the sense that to get the best performance, the network has to learn to extract features that are invariant to the walking conditions. Finally, a commendable performance of the gait recognition pipeline in this testing scenario can mean that the pipeline can be implemented with only a small amount of reference samples in new scenarios.

A similar method was followed for GREW dataset. As suggested by Zhu et al. (2021), we divided the dataset into a training set and a test set containing 20,000 and 6,000 samples, respectively. For each subject in the test set, there were two sequences for the gallery set and two sequences for the probe set.

4.2.3. Augmentation

Data augmentation techniques were used to increase the number of samples and make the model robust against various noises. The video frames were inverted to synthesize the effect of walking in reverse. Mirroring with respect to the vertical axis mimicked the situation of the subject walking in the opposite direction. To make the model resistant to pose estimation inaccuracies, Gaussian Noise with a standard deviation of 0.10 was added to each joint. These augmentations were performed only in the training set during runtime. They help avoid data leakage issues where augmentation before splitting the dataset can "leak" different variations of the same image into different splits (Kaufman et al., 2012). As a result, the network might get one variation of the image in the training set and another variant in the test set, resulting in a misleading increase in accuracy (Ahmed et al., 2022).

4.2.4. Batch size

Too small batch size can decrease the rate of convergence, whereas too large size can make the training procedure behave erratically. The batch size was chosen to be 128 following the mini-batch gradient descent technique (Khirirat et al., 2017), which is small compared to the total number of frames in the training set. This often works as a regularizer helping to reduce the generalization error due to the noise introduced by a small number of samples in each batch (Rahman et al., 2022). At the same time, the small size of the skeleton joint data allowed us to avoid even smaller batch sizes as we were able to easily fit the training samples into memory.

4.2.5. Epoch and learning rate

The system was trained for 400 epochs before convergence. The learning rate (LR) was set to 0.01 initially to ensure rapid learning and divergence from local minima. However, it is often recommended to reduce the LR over time to ensure that the learning does not stagnate and regularize the learning process (Goodfellow et al., 2016). For this reason, the learning rate is reduced by 10% after every 100 epochs. This helps the model to converge to an optimal solution more smoothly by reducing the stochastic noise and oscillation near the optimal point while training a graph convolutional network (Wan et al., 2022; Nakamura et al., 2021).

4.2.6. Optimizer and loss function

Adam optimizer is used to reduce the effect of noise generated due to the estimation of joints (Kingma and Ba, 2015). It can help models converge faster as it is computationally faster, enables the use of an adaptive learning rate, and has small memory requirements since it only stores the first and second moments of the gradient (Nanni et al., 2021). It is also well-suited for our case since it can handle a large amount of data with high-dimensional parameter spaces.

Supervised Contrastive Loss (Khosla et al., 2020), which is an extension of self-supervised batch contrastive loss, was used to calculate the loss. It can introduce a normalization effect by reducing the intra-class distance and increasing the inter-class distance in the embedding space (Bae et al., 2023). At the same time, the loss function provides stable performance in the presence of data augmentation allowing optimal hyperparameter tuning (Yang et al., 2022; Yu et al., 2023).

4.3. Hyperparameter tuning

In this section, we discuss the experiments that were performed to determine specific hyperparameter values for our pipeline.

4.3.1. Confidence threshold, T_C

To determine the threshold T_C for removing a low confidence frame, we evaluated the average accuracy of the baseline architecture based on different thresholds (Table 1).

Here, $T_c = 0$ indicates the vanilla dataset where no frames were removed from the samples. According to Table 1, having a 60% confidence threshold increased the accuracy by the highest amount. Similar accuracy can be obtained by keeping 50% threshold, however, it increased the number of frames that are used for making inference. On the other hand, increasing the threshold to more than 70% removed around 40,000 frames which led to the sequence not having enough information to represent the gait. This negatively affected the overall performance of the architecture.

In conclusion, to achieve the highest accuracy while also keeping the number of frames minimum, we removed any frame that had an average confidence below 60%. This enabled our architecture to better prevent errors caused by unreliable data that are false positives. In this regard, it is worth mentioning that preprocessing in this manner removed only 0.9% frames from the dataset.

4.3.2. DropGraph neighborhood threshold, T_D

To determine the number of neighbors to be dropped along with the selected base node, we evaluated the average accuracy of the baseline architecture based on different values of T_D in both spatial and temporal dimensions. It is worth mentioning that a combination of both spatial and temporal DropGraph was used as a regularizer during the training phase of ResGCN.

Spatial DropGraph For each frame, HRNet can extract 17 joint positions. Hence, we considered the values of T_D to be 0, 1, 2, 3, and 4 in the spatial dimension. As seen in Table 2, for $K = 0$ in the spatial dimension, the DropGraph technique is equivalent to Dropout as only a single node is dropped, which was not much effective. Dropping 1 neighbor resulted in effective regularization, whereas setting $T_D > 1$ may have caused too strong regularization. For this reason, the value of K was set to 1 for the spatial dimension. That means, in each frame, with a small probability, a root vertex and its 1-hop neighbors were dropped.

Temporal DropGraph The gait sequences in the training set of the CASIA-B dataset consist of 93 frames on average. However, the smallest video sequence consists of 30 frames. That means, on the temporal dimension, we have at least 30 nodes. Considering the small size in the temporal dimension, we considered the values of T_D to be 0, 1, 2, 3, and 4. As shown in Table 3, for $T_D = 0$, the temporal DropGraph is equivalent to Dropout as only a single node is dropped, which was not very effective. The effectiveness of DropGraph increased up to $T_D = 3$, which then decreased. For this reason, we set the value of T_D to 3 for the temporal dimension. That means, for each gait sequence, with a small probability, a root vertex and its 3-hop neighbors in earlier and later frames were dropped.

Table 1
Performance of the baseline architecture for different confidence values, T_C for frame removal. The values under the Δ column denote the increase/decrease in accuracy compared to the baseline.

Threshold	Frame	Accuracy (%)	Δ
0 (Baseline)	508211	69.89	0.00
50	507942	71.57	1.68
60	505410	**71.59**	**1.70**
70	500476	71.54	1.65
80	463773	66.72	−3.17

Table 2
Performance of the architecture for different confidence values, T_D in the spatial dimension. Here, the values under the Δ column denote the increase/decrease in accuracy compared to the network with no dropout.

T_D	Accuracy (%)	Δ
No dropout	79.67	0.00
0	79.86	0.19
1	**80.79**	**1.12**
2	80.31	0.64
3	80.04	0.37
4	79.89	0.22

Table 3
Performance of the architecture for different values of T_D in the temporal dimension. Here, the values under the Δ column denote the increase/decrease in accuracy compared to the network with no dropout.

T_D	Accuracy (%)	Δ
No dropout	79.67	0.00
0	79.86	0.19
1	80.09	0.48
2	80.35	0.68
3	**80.98**	**1.31**
4	80.55	0.88

4.4. Ablation study

We evaluate the performance of each module of our pipeline to understand their effect on the overall recognition performance. Vanilla ResGCN was implemented for gait recognition as the baseline model. The results are shown in Table 4.

4.4.1. Effect of preprocessing

To evaluate the performance of our overall preprocessing technique, we compared the performance of the baseline architecture with the vanilla dataset and our preprocessed dataset. In comparison to the baseline presented in Table 4, our preprocessing methods demonstrated enhancements across all aspects of network performance. Particularly, the improvements were more significant for BG and CL walking conditions, which are more susceptible to pose estimation inaccuracies, as opposed to NM. The reduction of noisy and inaccurate frames led to an average accuracy increase of 2.27%. The highest increase in accuracy is seen in CL conditions denoting that HRNet struggled to estimate poses correctly when the subject is heavily clothed. As a result, once we removed the frames with low confidence, the performance of the architecture improved significantly.

4.4.2. Effect of multi-stream input

Instead of using any excessively sophisticated and over-parameterized architecture for learning the discriminative features over all skeleton joints, we employ an early fusion of multi-stream input that enables us to capture rich spatial configurations and temporal dynamics from the skeleton data. One key benefit of using these three branches that consider the joint position, bone feature, and motion data is that it reduces the overall complexity of the model, without sacrificing the accuracy. According to Table 4, this multi-stream input injects a better insight of the spatial configuration and temporal dynamics of the joint configuration into the ResGCN model, improving the overall accuracy by 2.20% on average compared to only preprocessing. This increase in accuracy highlights the benefit of using multi-stream input. The model achieves this accuracy while also having at most 34 times less number of parameters compared to other popular graph convolutional networks used in gait recognition, such as ST-GCN, DGNN, etc. (Song et al., 2020). At the same time, despite processing multi-stream input, due to this small size, it can process around 1.57 times more frames per second compared to the mentioned architectures (Song et al., 2020).

4.4.3. Effect of hop extraction

Replacing the higher-order polynomials with the hop extraction technique to capture the long-distance relationships among the joints resulted in an overall increase in accuracy. As seen in Table 4, the significant increase in performance in all walking conditions proves the superiority of this technique. In fact, among all the techniques applied, Hop Extraction demonstrated the maximum increase in terms of accuracy. The removal of redundant dependencies between joint positions from different neighborhoods enabled the vanilla ResGCN to effectively capture video-wide joint relationships of human skeletons. Hence, the augmented ResGCN with hop extraction in the spatial dimension allowed the model to perform multi-scale reasoning without being affected by the biased weighting problem despite the introduction of additional edges. This improves the existing ResGCN architecture, making it an important building block to the overall pipeline.

4.4.4. Effect of DropGraph

To understand the overfitting issue, we observed the difference between the training and testing accuracies of the model as mentioned in Table 5.

Even after careful consideration in the training process via dynamic learning rate adjustment through each cycle, a 6% difference between the training accuracy and test accuracy was noticed. The addition of DropGraph to the model decreased the training accuracy by 0.3%, but it increased the test accuracy by 1.9%. That means, due to the introduction of the DropGraph, the model was

Table 4
Effect of different components on the performance of the overall architecture. Here, the values under the Δ column denote the increase/decrease in accuracy compared to the baseline model with no added components.

Configuration of Components					Accuracy (%)					
Preprocessing	Multi- Stream	Hop Extraction	DropGraph	Part-wise Attention	NM	Δ	BG	Δ	CL	Δ
×	×	×	×	×	84.1	0.0	73.2	0.0	65.5	0.0
✓	×	×	×	×	85.2	1.1	75.5	2.3	68.9	3.4
✓	✓	×	×	×	88.9	4.8	77.6	4.4	69.7	4.2
✓	✓	✓	×	×	93.3	9.2	87.5	14.3	82.3	16.8
✓	✓	✓	✓	×	94.1	10.0	89.9	16.7	85.2	19.7
✓	✓	✓	✓	✓	96.5	12.4	93.0	19.8	90.1	24.6

Table 5
Comparison of training and test accuracy after integrating DropGraph into our model on CASIA-B dataset.

Strategy	Accuracy (%)			
	Train	Δ	Test	Δ
Hop Extraction	85.7	0.0	79.7	0.0
DropGraph	85.4	−0.3	81.6	1.9

Table 6
Comparison of gait recognition performance with existing state-of-the-art models that utilize joint-positions.

Gallery NM#1–4		Viewing angles											Mean
Probe	Ref.	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	
NM#5–6	CrossGait (Qi et al., 2022)	57.7	68.5	72.5	75.5	70.9	69.4	71.4	73.8	76.3	68.3	55.8	68.9
	PoseGait (Liao et al., 2020)	55.3	69.6	73.9	75.0	68.0	68.2	71.1	72.9	76.1	70.4	55.4	68.7
	PoseMapGait (Liao et al., 2022)	59.9	76.2	81.7	83.1	76.8	76.1	76.3	81.1	79.6	75.4	66.1	75.7
	SDHF-GCN (Liu et al., 2022b)	77.3	82.8	85.1	86.0	85.5	85.4	83.7	81.5	80.5	83.9	77.6	82.7
	JointsGait (Li et al., 2020a)	68.1	73.6	77.9	76.4	77.5	79.1	78.4	76.0	69.5	71.9	70.1	74.4
	GaitGraph2 (Teepe et al., 2022)	78.5	82.9	85.8	85.6	83.1	81.5	84.3	83.2	84.2	81.6	71.8	82.0
	Gaitgraph (Teepe et al., 2021)	85.3	88.5	91.0	92.5	87.0	86.5	88.4	89.2	87.9	85.9	81.9	87.6
	MAST-GCN (Zheng et al., 2022)	87.7	89.2	89.6	89.9	90.3	90.0	88.5	88.1	88.5	86.1	81.8	88.2
	Gait-D(Gao et al., 2022)	87.7	92.5	93.6	95.7	93.3	92.4	92.8	93.4	90.6	88.6	87.3	91.6
	FR-GCN(Wang et al., 2022)	90.6	91.6	93.5	92.3	91.9	93.0	92.1	91.6	93.3	89.3	86.0	91.4
	MS-Gait (Wang et al., 2022a)	89.4	91.7	91.6	90.2	90.6	90.6	90.4	90.9	90.4	88.5	85.6	90.0
	HeatGait (Hasan et al., 2022)	91.7	93.8	93.8	94.7	92.6	94.6	94.3	94.4	93.2	91.5	91.5	93.3
	RGCNN (Shopon et al., 2021)	94.8	**98.5**	**96.9**	**98.3**	**96.8**	**98.9**	96.9	**98.8**	**97.9**	93.9	**95.9**	**97.0**
	GaitGCN++ (ours)	**95.2**	95.7	96.6	96.8	96.5	97.6	**97.2**	97.2	96.0	**97.1**	95.3	96.5
BG#1–2	CrossGait (Qi et al., 2022)	37.4	49.3	50.1	51.5	46.7	43.0	47.5	50.9	49.3	45.4	33.8	45.9
	PoseGait (Liao et al., 2020)	35.3	47.2	52.4	46.9	45.5	43.9	46.1	48.1	49.4	43.6	31.1	44.5
	PoseMapGait (Liao et al., 2022)	47.7	56.1	63.9	63.3	64.2	59.5	58.1	61.5	61.9	58.2	44.3	58.1
	SDHF-GCN (Liu et al., 2022b)	67.5	73.9	73.2	74.3	68.5	68.5	70.5	69.0	62.2	68.7	60.1	68.8
	JointsGait (Li et al., 2020a)	54.3	59.1	60.6	59.7	63.0	65.7	62.4	59.0	58.1	58.6	50.1	59.1
	GaitGraph2 (Teepe et al., 2022)	69.9	75.9	78.1	79.3	71.4	71.7	74.3	76.2	73.2	73.4	61.7	73.2
	Gaitgraph (Teepe et al., 2021)	75.8	76.7	75.9	76.1	71.4	73.9	78.0	74.7	75.4	75.4	69.2	74.8
	MAST-GCN (Zheng et al., 2022)	74.8	76.6	78	78.2	75.2	73.1	72.0	74.3	77.4	75.8	72	75.2
	Gait-D (Gao et al., 2022)	78.2	80.1	79.3	80.2	78.4	77.6	80.4	78.6	79.1	80.2	76.5	79.0
	FR-GCN(Wang et al., 2022)	77.9	85.2	84.0	81.2	82.5	78.9	81.3	79.5	80.2	77.8	71.0	80.0
	MS-Gait (Wang et al., 2022a)	75.7	84.8	83.7	83.2	80.6	80.1	82.2	79.8	79.1	75.9	71.1	79.7
	HeatGait (Hasan et al., 2022)	86.9	87.2	89.6	89.5	86.5	88.0	89.4	86.9	87.1	85.6	85.5	87.5
	RGCNN (Shopon et al., 2021)	88.8	**93.8**	91.5	88.5	91.6	90.0	91.9	**91.4**	92.4	89.7	89.0	90.8
	GaitGCN++ (ours)	**92.4**	92.4	**95.1**	**93.7**	**92.5**	**94.1**	**93.9**	90.3	**93.1**	**93.1**	**92.0**	**93.0**
CL#1–2	CrossGait (Qi et al., 2022)	26.8	32.1	43.6	39.3	41.2	39.6	39.9	43.7	42.5	35.7	24.2	37.2
	PoseGait (Liao et al., 2020)	24.3	29.7	41.3	38.8	38.2	38.5	41.6	44.9	42.2	33.4	22.5	35.9
	PoseMapGait (Liao et al., 2022)	30.4	41.9	45.2	48.9	47.3	48.1	46.5	44.9	36.0	34.5	29.6	41.2
	SDHF-GCN (Liu et al., 2022b)	63.4	65.4	66.7	64.8	63.0	66.2	69.1	63.3	61.1	65.9	60.7	64.5
	JointsGait (Li et al., 2020a)	48.1	46.9	49.6	50.5	51.0	52.3	49.0	46.0	48.7	53.6	52.0	49.8
	GaitGraph2 (Teepe et al., 2022)	57.1	61.1	68.9	66.0	67.8	65.4	68.1	67.2	63.7	63.6	50.4	63.6
	Gaitgraph (Teepe et al., 2021)	69.6	66.1	68.8	67.2	64.5	62.0	69.5	65.6	65.7	66.1	64.3	66.3
	MAST-GCN (Zheng et al., 2022)	68.8	71.6	76.8	75.4	76.6	74.8	71.5	66.7	74.4	75.0	71.0	73.0
	Gait-D (Gao et al., 2022)	73.2	71.7	75.4	73.2	74.6	72.3	74.1	70.5	69.4	71.2	66.7	72.0
	FR-GCN(Wang et al., 2022)	74.0	74.3	76.0	78.8	80.4	79.3	78.0	79.5	74.8	70.5	67.0	75.7
	MS-Gait (Wang et al., 2022a)	75.1	79.7	80.5	84.7	84.0	82.4	79.8	80.4	78.3	78.0	70.9	79.4
	HeatGait (Hasan et al., 2022)	82.8	81.3	87.4	83.9	82.5	84.1	83.9	81.1	80.7	78.5	78.8	82.3
	RGCNN (Shopon et al., 2021)	87.9	**91.1**	90.9	89.8	88.8	89.8	89.2	**90.0**	**91.0**	**91.0**	**89.4**	89.9
	GaitGCN++ (ours)	**88.9**	88.3	**93.2**	**90.9**	**91.6**	**92.0**	**90.2**	89.8	89.2	89.4	87.1	**90.1**

able to learn more generalized features to perform better on an unseen dataset.

DropGraph works as a regularizer by reducing overfitting and increasing the generalization capability of the model. The test accuracy is further expressed in detail in Table 4. When DropGraph was applied on top of the preprocessed and hop-extracted state, it resulted in a promising improvement for all walking conditions. It is evident from the table that DropGraph increased the accuracy of the model by allowing it to learn general features to classify unseen samples better.

4.4.5. Effect of part-wise attention

Part-wise attention block was added to identify specific body parts that are helpful in identifying people. It does so by learning to generate larger weights for important parts of the body. These weights enable the underlying GCN to discover the most informative body parts for each gait sequence and then aggregate their features to obtain a more explainable and accurate representation of different gait sequences, i. As seen in Table 4, the introduction of the part-wise attention block resulted in higher accuracy. Compared to the NM walking condition, the accuracy improvement was higher in BG and CL conditions. This can be attributed to the fact that the model already has a high accuracy in NM, considering the joint pose estimated by the pose estimation model. However, since there are certain noises introduced due to the occlusion and appearance in BG and CL conditions, the attention block helps the network to determine a specific portion of the body that can still be used to identify gait in a better way.

4.5. Comparison with state-of-the-art models

4.5.1. CASIA-B dataset

Joint position-based Approaches Table 6 compares the performance of the proposed system with existing state-of-the-art

Table 7
Comparison of gait recognition performance with existing state-of-the-art models that utilize appearance.

Ref.	Accuracy (%)			
	NM	BG	CL	Mean
GaitNet (Song et al., 2019)	91.6	85.7	58.9	78.7
GaitSet (Chao et al., 2019)	95.0	87.2	70.4	84.2
DL-GaitSet (Song et al., 2022)	94.0	88.6	71.3	84.6
GaitRNNPart (Sepas-Moghaddam and Etemad, 2020)	95.2	89.7	74.7	86.5
GaitSet + Loss (Han et al., 2022)	96.0	91.6	74.8	87.5
P-GOFI (Xu et al., 2023)	95.4	79.0	91.6	88.7
GaitPart (Fan et al., 2020)	96.2	91.5	78.7	88.8
GLN (Hou et al., 2020)	96.9	94.0	77.5	89.5
SRN + CBlock (Hou et al., 2021)	97.5	**94.3**	77.7	89.8
HMRNet (Li et al., 2021)	**97.9**	93.1	77.6	89.5
3DCNN (Lin et al., 2020)	96.7	93.0	81.5	90.4
GaitGL (Lin et al., 2021b)	96.4	92.7	83.0	90.7
SRN (Hou et al., 2021)	97.1	94.0	81.8	90.9
Multi3D (Lin et al., 2021a)	97.6	94.1	81.2	90.9
GLFE (Lin et al., 2021c)	97.4	94.5	83.6	91.8
Vi-GaitGL (Chai et al., 2021)	96.2	92.9	87.2	92.1
GaitGCN++ (ours)	96.5	93.0	**90.1**	**93.2**

Table 8
Comparison of gait recognition performance in the wild with existing state-of-the-art models.

Ref.	Accuracy (%)			
	Rank-1	Rank-5	Rank-10	Rank-20
GEINet (Shiraga et al., 2016)	6.82	13.42	16.97	21.01
GaitSet (Chao et al., 2019)	46.28	63.58	70.26	76.82
GaitPart (Fan et al., 2020)	44.01	60.68	67.25	73.47
GaitGL (Lin et al., 2021b)	47.29	63.56	69.32	74.18
RealGait (Zhang et al., 2022)	54.12	71.47	77.57	81.71
TransGait (Li et al., 2022)	56.27	72.72	78.12	82.51
GaitGCN++ (ours)	**58.65**	**73.82**	**80.13**	**84.41**

models in gait recognition. Our system improved the accuracy by a commendable margin in varying viewing angles and walking conditions.

Appearance-based Approaches Even though appearance-based methods tend to achieve a better result in gait recognition compared to model-based approaches, our system took a huge step to close this gap. Additionally, we achieved a higher average accuracy in the CASIA-B dataset compared to the appearance-based methods. A comparison can be seen in Table 7.

One key advantage of joint position-based methods is that they are invariant to walking conditions. As evident from the table, our method performed significantly better than appearance-based approaches when the subject walks while wearing heavy clothes. In other walking conditions, our model provided comparable performance to the state-of-the-art appearance-based methods.

4.5.2. GREW dataset

To determine how our pipeline will perform in real-life gait recognition scenarios, we evaluated the performance of the model using GREW dataset. A comparison of performance with the state-of-the-art models is shown in Table 8.

According to the table, the performance of our proposed method exceeds the current state-of-the-art in terms of accuracy. However, the performance still needs to be improved for Rank-1 accuracy. This can be attributed to the difficulty of recognizing gait in real-life environments. As a result, there still exists some room for improvement.

5. Conclusion

In conclusion, our gait recognition pipeline, GaitGCN++, consisting of an effective preprocessing and a robust and generalized feature extractor based on a graph convolutional network achieved state-of-the-art performance in challenging conditions. The joint-position based features were robust to variations, and the multi-stream features captured relationships among joints and while the Part-wise Attention technique prioritized important body parts. The preprocessing, DropGraph technique, and Part-wise Attention module prevented overfitting and ensured high test accuracy without sacrificing training accuracy. Future research endeavors may focus on exploring the effect of varying the number of joints considered on recognition performance, evaluating better pose estimation techniques, finding the optimal configuration for the part-wise attention module, and investigating the combination of joint position-based and appearance-based techniques to handle both poor pose estimation and appearance variance.

Declaration of Competing Interest

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

Acknowledgements

The authors are thankful to all the faculty members of Department of Computer Science and Engineering, Islamic University of Technology for their continuous support and suggestions throughout the entire study. Md. Bakhtiar Hasan would like to give special thanks to his wife for providing encouragement and motivation while undertaking the research and writing the manuscript. He is deeply grateful for her patience, love, and unwavering belief in him.

References

Addlesee, M., Jones, A., Livesey, F., Samaria, F., 1997. The ORL active floor [sensor system]. IEEE Pers. Commun. 4, 35–41. https://doi.org/10.1109/98.626980. URL: https://ieeexplore.ieee.org/document/626980.

Agarap, A.F., 2018. Deep Learning using Rectified Linear Units (ReLU). CoRR, abs/1803.08375. URL: http://arxiv.org/abs/1803.08375. arXiv:1803.08375.

Ahmed, S., Hasan, M.B., Ahmed, T., Sony, M.R.K., Kabir, M.H., 2022. Less is more: lighter and faster deep neural architecture for tomato leaf disease classification. IEEE Access 10, 68868–68884. https://doi.org/10.1109/ACCESS.2022.3187203. URL: https://ieeexplore.ieee.org/document/9810234.

Archila, J., Manzanera, A., Martínez, F., 2022. A multimodal Parkinson quantification by fusing eye and gait motion patterns, using covariance descriptors, from non-invasive computer vision. Comput. Methods Programs Biomed. 215, 106607. https://doi.org/10.1016/j.cmpb.2021.106607. URL: https://www.sciencedirect.com/science/article/pii/S0169260721006817.

Azhar, M., Ullah, S., Raees, M., Rahman, K.U., Rehman, I.U., 2023. A real-time multi view gait-based automatic gender classification system using kinect sensor. Multimedia Tools Appl. 82, 11993–12016. https://doi.org/10.1007/s11042-022-13704-3. URL: https://link.springer.com/article/10.1007/s11042-022-13704-3.

Bae, S., Kim, S., Ko, J., Lee, G., Noh, S., Yun, S., 2023. Self-Contrastive Learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 197-205. URL: https://ojs.aaai.org/index.php/AAAI/article/view/25091/24863. https://doi.org/10.1609/aaai.v37i1.25091.

Battistone, F., Petrosino, A., 2019. TGLSTM: A time based graph deep learning approach to gait recognition. Pattern Recogn. Lett. 126, 132–138. https://doi.org/10.1016/j.patrec.2018.05.004. URL: https://www.sciencedirect.com/science/article/abs/pii/S0167865518301703.

Bhatti, U.A., Tang, H., Wu, G., Marjan, S., Hussain, A., 2023. Deep learning with graph convolutional networks: an overview and latest applications in computational intelligence. Int. J. Intell. Syst. 2023, 8342104. https://doi.org/10.1155/2023/8342104. URL: https://www.hindawi.com/journals/ijis/2023/8342104/.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., Sheikh, Y., 2021. OpenPose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. 43, 172–186. https://doi.org/10.1109/TPAMI.2019.2929257. URL: https://ieeexplore.ieee.org/abstract/document/8765346.

Cao, P., Zhu, Z., Wang, Z., Zhu, Y., Niu, Q., 2022. Applications of graph convolutional networks in computer vision. Neural Comput. Appl. 34, 13387–13405. https://doi.org/10.1007/s00521-022-07368-1. URL: https://link.springer.com/article/10.1007/s00521-022-07368-1.

Chai, T., Mei, X., Li, A., Wang, Y., 2021. Silhouette-Based View-Embeddings for Gait Recognition Under Multiple Views. In: 2021 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 2319–2323, URL: https://ieeexplore.ieee.org/abstract/document/9506238. https://doi.org/10.1109/ICIP42928.2021.9506238.

Chao, H., He, Y., Zhang, J., Feng, J., 2019. GaitSet: regarding gait as a set for cross-view gait recognition. In; Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8126–8133. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4821. https://doi.org/10.1609/aaai.v33i01.33018126.

Chen, Z., Li, S., Yang, B., Li, Q., Liu, H., 2021. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 1113–1122. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16197. https://doi.org/10.1609/aaai.v35i2.16197.

Cheng, K., Zhang, Y., Cao, C., Shi, L., Cheng, J., Lu, H., 2020. Decoupling GCN with DropGraph module for skeleton-based action recognition. In: European Conference on Computer Vision (ECCV). Springer, Cham, pp. 536–553. https://doi.org/10.1007/978-3-030-58586-0_32. URL: https://link.springer.com/chapter/10.1007/978-3-030-58586-0_32.

Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: encoder-decoder approaches. In: Dekai Wu, Marine Carpuat, Xavier Carreras, Eva Maria Vecchi (Eds.), Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014, Association for Computational Linguistics, pp. 103–111, URL: https://aclanthology.org/W14-4012/. https://doi.org/10.3115/v1/W14-4012.

Cicirelli, G., Impedovo, D., Dentamaro, V., Marani, R., Pirlo, G., D'Orazio, T., 2022. Human gait analysis in neurodegenerative diseases: a review. IEEE J. Biomed. Health Informat. 26, 229–242. https://doi.org/10.1109/jbhi.2021.3092875. URL: https://ieeexplore.ieee.org/abstract/document/9466394.

Cutting, J.E., Kozlowski, L.T., 1977. Recognizing friends by their walk: Gait perception without familiarity cues. Bull. Psychon. Soc. 9, 353–356. https://doi.org/10.3758/bf03337021. URL: https://link.springer.com/article/10.3758/BF03337021.

Cutting, J.E., Proffitt, D.R., Kozlowski, L.T., 1978. A biomechanical invariant for gait perception. J. Exp. Psychol. Hum. Percept. Perform. 4, 357. https://doi.org/10.1037/0096-1523.4.3.357. URL: https://psycnet.apa.org/record/1980-00173-001.

Echterhoff, J.M., Haladjian, J., Brügge, B., 2018. Gait and jump classification in modern equestrian sports. In: Proceedings of the 2018 ACM International Symposium on Wearable Computers, Association for Computing Machinery (ACM), pp. 88–91, URL: https://dl.acm.org/doi/abs/10.1145/3267242.3267267. https://doi.org/10.1145/3267242.3267267.

Etemad, S.A., Arya, A., 2016. Expert-driven perceptual features for modeling style and affect in human motion. IEEE Trans. Human-Mach. Syst. 46, 534–545. https://doi.org/10.1109/thms.2016.2537760. URL: https://ieeexplore.ieee.org/abstract/document/7446325.

Fan, C., Peng, Y., Cao, C., Liu, X., Hou, S., Chi, J., Huang, Y., Li, Q., He, Z., 2020. GaitPart: temporal part-based model for gait recognition. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14225–14233. URL: https://ieeexplore.ieee.org/abstract/document/9156784. https://doi.org/10.1109/cvpr42600.2020.01423.

Fang, H.-S., Xie, S., Tai, Y.-W., Lu, C., 2017. RMPE: regional multi-person pose estimation. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, pp. 2353–2362, URL: https://ieeexplore.ieee.org/document/8237518. https://doi.org/10.1109/ICCV.2017.256.

Feng, Y., Li, Y., Luo, J., 2016. Learning effective Gait features using LSTM. In: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE, pp. 325–330, URL: https://ieeexplore.ieee.org/abstract/document/7899654. https://doi.org/10.1109/icpr.2016.7899654.

Filipi Gonçalves dos Santos, C., Oliveira, D. d. S., A. Passos, L., Gonçalves Pires, R., Felipe Silva Santos, D., Pascotti Valem, L., P. Moreira, T., Cleison S. Santana, M., Roder, M., Paulo Papa, J., Colombo, D., 2022. Gait recognition based on deep learning: a survey. ACM Comput. Surv. 55. URL: https://dl.acm.org/doi/10.1145/3490235. https://doi.org/10.1145/3490235.

Gao, S., Yun, J., Zhao, Y., Liu, L., 2022. Gait-D: Skeleton-based gait feature decomposition for gait recognition. IET Comput. Vision 16, 111–125. https://doi.org/10.1049/cvi2.12070. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi2.12070.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Goodfellow, I.J., Bengio, Y., Courville, A.C., 2016. Deep learning. Adaptive Computation and Machine Learning. MIT Press. URL: http://www.deeplearningbook.org/.

Han, F., Li, X., Zhao, J., Shen, F., 2022. A unified perspective of classification-based loss and distance-based loss for cross-view gait recognition. Pattern Recogn. 125, 108519. https://doi.org/10.1016/j.patcog.2021.108519. URL: https://www.sciencedirect.com/science/article/pii/S0031320321006956.

Hasan, M.B., Ahmed, T., Kabir, M.H., 2022. HEATGait: hop-extracted adjacency technique in graph convolution based gait recognition. In: 2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), IEEE, pp. 1–6, URL: https://ieeexplore.ieee.org/document/9849799. https://doi.org/10.1109/CTISC54888.2022.9849799.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. URL: https://ieeexplore.ieee.org/abstract/document/7780459. https://doi.org/10.1109/cvpr.2016.90.

He, Y., Zhang, J., Shan, H., Wang, L., 2018. Multi-task GANs for view-specific feature learning in gait recognition. IEEE Trans. Inf. Forensics Secur. 14, 102–113. https://doi.org/10.1109/tifs.2018.2844819. URL: https://ieeexplore.ieee.org/abstract/document/8374898.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735. URL: https://ieeexplore.ieee.org/abstract/document/6795963.

Hou, S., Cao, C., Liu, X., Huang, Y., 2020. Gait lateral network: learning discriminative and compact representations for gait recognition. In: European Conference on Computer Vision. Springer, Cham, pp. 382–398. https://doi.org/10.1007/978-3-030-58545-7_22. URL: https://link.springer.com/chapter/10.1007/978-3-030-58545-7_22.

Hou, S., Liu, X., Cao, C., Huang, Y., 2021. Set residual network for silhouette-based gait recognition. IEEE Trans. Biomet. Behav. Ident. Sci. 3, 384–393. https://doi.org/10.1109/TBIOM.2021.3074963. URL: https://ieeexplore.ieee.org/abstract/document/9410587.

Huang, Z., Xue, D., Shen, X., Tian, X., Li, H., Huang, J., Hua, X.-S., 2021. 3D local convolutional neural networks for gait recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) pp. 14920–14929. URL: https://ieeexplore.ieee.org/document/9709936. doi:10.1109/ICCV48922.2021.01465.

Hu, B., Gao, Y., Guan, Y., Long, Y., Lane, N.D., Ploetz, T., 2018. Robust cross-view gait identification with evidence: a discriminant Gait GAN (DiGGAN) approach on 10000 people. CoRR, abs/1811.10493. URL: http://arxiv.org/abs/1811.10493. arXiv:1811.10493.

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach, Francis, Blei, David (Eds.), Proceedings of the 32nd International Conference on Machine Learning, Lille, France: PMLR volume 37 of Proceedings of Machine Learning Research, pp. 448–456, URL: https://proceedings.mlr.press/v37/ioffe15.html.

Johansson, G., 1973. Visual perception of biological motion and a model for its analysis. Percept. Psychophys. 14, 201–211. https://doi.org/10.3758/bf03212378. URL: https://link.springer.com/article/10.3758/BF03212378.

Kaufman, S., Rosset, S., Perlich, C., Stitelman, O., 2012. Leakage in data mining: formulation, detection, and avoidance. ACM Trans. Knowledge Discov. Data 6, 1–21. https://doi.org/10.1145/2382577.2382579. URL: https://dl.acm.org/doi/abs/10.1145/2382577.2382579.

Khirirat, S., Feyzmahdavian, H.R., Johansson, M., 2017. Mini-batch gradient descent: Faster convergence under data sparsity. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 2880–2887. URL: https://ieeexplore.ieee.org/abstract/document/8264077. https://doi.org/10.1109/cdc.2017.8264077.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised Contrastive Learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc. pp. 18661–18673. URL: https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Yoshua Bengio, Yann LeCun (Eds.), 3rd International Conference on Learning Representations (ICLR). URL: http://arxiv.org/abs/1412.6980. https://doi.org/10.48550/arXiv.1412.6980.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net. URL: https://openreview.net/forum?id=SJU4ayYgl.

Lee, C.Y., 1961. An algorithm for path connections and its applications. IRE Trans. Electronic Comput. EC-10, 346–365. https://doi.org/10.1109/TEC.1961.5219222. URL:https://ieeexplore.ieee.org/abstract/document/5219222.

Liao, R., Cao, C., Garcia, E.B., Yu, S., Huang, Y., 2017. Pose-Based Temporal-Spatial Network (PTSN) for gait recognition with carrying and clothing variations. In: Zhou, Jie, Wang, Yunhong, Sun, Zhenan, Xu, Yong, Shen, Linlin, Feng, Jianjiang, Shan, Shiguang, Qiao, Yu., Guo, Zhenhua, Yu, Shiqi (Eds.), Biometric Recognition. Springer, Cham, pp. 474–483. https://doi.org/10.1007/978-3-319-69923-3_51. URL: https://link.springer.com/chapter/10.1007/978-3-319-69923-3_51.

Liao, R., Yu, S., An, W., Huang, Y., 2020. A model-based gait recognition method with body pose and human prior knowledge. Pattern Recogn. 98, 107069. https://doi.org/10.1016/j.patcog.2019.107069. URL: https://www.sciencedirect.com/science/article/pii/S003132031930370X.

Liao, R., Li, Z., Bhattacharyya, S.S., York, G., 2022. PoseMapGait: A model-based gait recognition method with pose estimation maps and graph convolutional networks. Neurocomputing 501, 514–528. https://doi.org/10.1016/j.neucom.2022.06.048. URL: https://www.sciencedirect.com/science/article/pii/S0925231222007652.

Li, Q., Han, Z., Wu, X.-M., 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 32. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11604. https://doi.org/10.1609/aaai.v32i1.11604.

Li, N., Zhao, X., Ma, C., 2020a. A model-based Gait Recognition Method based on Gait Graph Convolutional Networks and Joints Relationship Pyramid Mapping. CoRR, abs/2005.08625. URL: https://arxiv.org/abs/2005.08625.

Li, X., Makihara, Y., Xu, C., Yagi, Y., Yu, S., Ren, M., 2020b. End-to-end model-based gait recognition. In: Proceedings of the Asian Conference on Computer Vision. Springer, Cham. URL: https://link.springer.com/chapter/10.1007/978-3-030-69535-4_1. https://doi.org/10.1007/978-3-030-69535-4_1.

Li, X., Makihara, Y., Xu, C., Yagi, Y., Yu, S., Ren, M., 2021. End-to-end model-based gait recognition. In: Ishikawa Hiroshi, Liu Cheng-Lin, Pajdla Tomas, Shi Jianbo (Eds.), Computer Vision – ACCV 2020 (pp. 3–20). Cham: Springer, Cham. URL: https://link.springer.com/chapter/10.1007/978-3-030-69535-4_1. https://doi.org/10.1007/978-3-030-69535-4_1.

Li, G., Guo, L., Zhang, R., Qian, J., Gao, S., 2022. TransGait: Multimodal-based gait recognition with set transformer. Appl. Intell. 1–13. URL: https://link.springer.com/article/10.1007/s10489-022-03543-y. https://doi.org/10.1007/s10489-022-03543-y.

Lin, B., Zhang, S., Bao, F., 2020. Gait recognition with multiple-temporal-scale 3D convolutional neural network. In: Proceedings of the 28th ACM International Conference on Multimedia, Association for Computing Machinery (ACM), pp. 3054–3062. URL: https://dl.acm.org/doi/abs/10.1145/3394171.3413861. https://doi.org/10.1145/3394171.3413861.

Lin, B., Zhang, S., Liu, Y., Qin, S., 2021a. Multi-scale temporal information extractor for gait recognition. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 2998–3002. IEEE. URL: https://ieeexplore.ieee.org/abstract/document/9506488. https://doi.org/10.1109/ICIP42928.2021.9506488.

Lin, B., Zhang, S., Yu, X., 2021b. Gait recognition via effective global-local feature representation and local temporal aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14648–14656, URL: https://ieeexplore.ieee.org/document/9710710. https://doi.org/10.1109/ICCV48922.2021.01438.

Lin, B., Zhang, S., Yu, X., 2021c. Gait recognition via effective global-local feature representation and local temporal aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, pp. 14648–14656, URL: https://ieeexplore.ieee.org/abstract/document/9710710/. https://doi.org/10.1109/ICCV48922.2021.01438.

Liu, D., Ye, M., Li, X., Zhang, F., Lin, L., 2016. Memory-based gait recognition. In: Wilson, Richard C., Hancock, Edwin R., Smith, William A.P. (Ed.), Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19–22, 2016. BMVA Press. URL: http://www.bmva.org/bmvc/2016/papers/paper082/index.html. https://doi.org/10.5244/C.30.82.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., Kot, A.C., 2019. NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding. IEEE Trans. Pattern Anal. Mach. Intell. 42, 2684–2701. https://doi.org/10.1109/tpami.2019.2916873. URL: https://ieeexplore.ieee.org/document/8713892.

Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W., 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 143–152, URL: https://ieeexplore.ieee.org/abstract/document/9156556. https://doi.org/10.1109/CVPR42600.2020.00022.

Liu, W., Bao, Q., Sun, Y., Mei, T., 2022a. Recent advances of monocular 2D and 3D human pose estimation: a deep learning perspective. ACM Comput. Surv. 55. https://doi.org/10.1145/3524497. URL: https://dl.acm.org/doi/abs/10.1145/3524497.

Liu, X., You, Z., He, Y., Bi, S., Wang, J., 2022b. Symmetry-driven hyper feature GCN for skeleton-based gait recognition. Pattern Recogn. 125, 108520. https://doi.org/10.1016/j.patcog.2022.108520. URL: https://www.sciencedirect.com/science/article/pii/S0031320322000012.

Lyu, K., Li, Z., Arora, S., 2022. Understanding the generalization benefit of normalization layers: Sharpness reduction. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc., pp. 34689–34708, URL: https://papers.nips.cc/paper_files/paper/2022/hash/dffd1c523512e557f4e75e8309049213-Abstract-Conference.html.

Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S.-M., Ailisto, H., 2005. Identifying users of portable devices from gait pattern with accelerometers. In: Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, vol. 2. pp. ii/973–ii/976, URL: https://ieeexplore.ieee.org/abstract/document/1415569. https://doi.org/10.1109/icassp.2005.1415569.

Marsico, M.D., Mecca, A., 2019. A survey on gait recognition via wearable sensors. ACM Comput. Surv. 52. https://doi.org/10.1145/3340293. URL: https://dl.acm.org/doi/abs/10.1145/3340293.

Mason, R., Pearson, L.T., Barry, G., Young, F., Lennon, O., Godfrey, A., Stuart, S., 2023. Wearables for running gait analysis: a systematic review. Sports Med. 53, 241–268. https://doi.org/10.1007/s40279-022-01760-6. URL: https://link.springer.com/article/10.1007/s40279-022-01760-6.

Meng, L., Pang, J., Yang, Y., Chen, L., Xu, R., Ming, D., 2023. Inertial-based gait metrics during turning improve the detection of early-stage parkinson's disease patients. IEEE Trans. Neural Syst. Rehabil. Eng. 31, 1472–1482. https://doi.org/10.1109/TNSRE.2023.3237903. URL: https://ieeexplore.ieee.org/abstract/document/10025615.

Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M., Zhang, D., 2023. Biometrics recognition using deep learning: a survey. Artif. Intell. Rev. https://doi.org/10.1007/s10462-022-10237-x. URL: https://link.springer.com/article/10.1007/s10462-022-10237-x.

Mogan, J.N., Lee, C.P., Lim, K.M., 2022. Advances in vision-based gait recognition: from handcrafted to deep learning. Sensors, 22. URL: https://www.mdpi.com/1424-8220/22/15/5682. https://doi.org/10.3390/s22155682.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5115–5124. URL: https://ieeexplore.ieee.org/document/8100059. https://doi.org/10.1109/cvpr.2017.576.

Moore, E.F., 1959. The shortest path through a maze. In: Proceedings of the International Symposium on the Theory of Switching, Harvard University Press, pp. 285–292.

Muro-De-La-Herran, A., Garcia-Zapirain, B., Mendez-Zorrilla, A., 2014. Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications. Sensors 14, 3362–3394. https://doi.org/10.3390/s140203362. URL: https://www.mdpi.com/1424-8220/14/2/3362.

Murray, M.P., 1967. Gait as a total pattern of movement: including a bibliography on gait. Am. J. Phys. Med. Rehabil. 46, 290–333. URL https://journals.lww.com/ajpmr/Citation/1967/02000/Gait_As_A_Total_Pattern_of_Movement__Including_A.26.aspx.

Murray, M.P., Drought, A.B., Kory, R.C., 1964. Walking patterns of normal men. J. Bone Joint Surg. 46, 335–360. URL: https://journals.lww.com/jbjsjournal/Abstract/1964/46020/Walking_Patterns_of_Normal_Men.9.aspx.

Nakamura, K., Derbel, B., Won, K.-J., Hong, B.-W., 2021. Learning-rate annealing methods for deep neural networks. Electronics 10. https://doi.org/10.3390/electronics10162029. URL: https://www.mdpi.com/2079-9292/10/16/2029.

Nambiar, A., Bernardino, A., Nascimento, J.C., 2019. Gait-based person re-identification: a survey. ACM Comput. Surv. 52, 1–34. https://doi.org/10.1145/3243043. URL: https://dl.acm.org/doi/abs/10.1145/3243043.

Nanni, L., Maguolo, G., Lumini, A., 2021. Exploiting Adam-like Optimization Algorithms to Improve the Performance of Convolutional Neural Networks. CoRR, abs/2103.14689. URL: https://arxiv.org/abs/2103.14689.

Niyogi, S.A., Adelson, E.H., 1994. Analyzing and recognizing walking figures in XYT. In: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 469–474, URL: https://ieeexplore.ieee.org/abstract/document/323868. https://doi.org/10.1109/cvpr.1994.323868.

Parashar, A., Shekhawat, R.S., Ding, W., Rida, I., 2022. Intra-class variations with deep learning-based gait analysis: A comprehensive survey of covariates and methods. Neurocomputing 505, 315–338. https://doi.org/10.1016/j.neucom.2022.07.002. URL: https://www.sciencedirect.com/science/article/pii/S0925231222008578.

Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: Dasgupta Sanjoy, McAllester David (Eds.), Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA:

PMLR volume 28 of Proceedings of Machine Learning Research, pp. 1310–1318, URL: https://proceedings.mlr.press/v28/pascanu13.html.

Physiopedia, 2022. Gait — Physiopedia. URL: https://www.physio-pedia.com/index.php?title=Gait&oldid=295011 [Online; accessed 25-April-2022].

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., Schiele, B., 2016. DeepCut: joint subset partition and labeling for multi person pose estimation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4929–4937. URL: https://ieeexplore.ieee.org/document/7780902. https://doi.org/10.1109/CVPR.2016.533.

Qi, Y.J., Kong, Y.P., Zhang, Q., 2022. A cross-view gait recognition method using two-way similarity learning. Mathe. Probl. Eng. 2022, 2674425. URL: https://www.hindawi.com/journals/mpe/2022/2674425/. https://doi.org/10.1155/2022/2674425.

Rahman, A.B.M.A., Hasan, M.B., Ahmed, S., Ahmed, T., Ashmafee, M.H., Kabir, M.R., Kabir, M.H., 2022. Two decades of bengali handwritten digit recognition: a survey. IEEE Access 10, 92597–92632. https://doi.org/10.1109/ACCESS.2022.3202893. URL: https://ieeexplore.ieee.org/abstract/document/9869842.

Rani, V., Kumar, M., 2023. Human gait recognition: A systematic review. Multimedia Tools Appl. https://doi.org/10.1007/s11042-023-15079-5. URL: https://link.springer.com/article/10.1007/s11042-023-15079-5.

Ren, H., Lu, W., Xiao, Y., Chang, X., Wang, X., Dong, Z., Fang, D., 2022. Graph convolutional networks in language and vision: A survey. Knowl.-Based Syst. 251, 109250. https://doi.org/10.1016/j.knosys.2022.109250. URL: https://www.sciencedirect.com/science/article/pii/S0950705122006220.

Russo, M., Amboni, M., Barone, P., Pellecchia, M.T., Romano, M., Ricciardi, C., Amato, F., 2023. Identification of a gait pattern for detecting mild cognitive impairment in parkinson's disease. Sensors 23. https://doi.org/10.3390/s23041985. URL: https://www.mdpi.com/1424-8220/23/4/1985.

Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic Routing Between Capsules. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Ed.), Advances in Neural Information Processing Systems. vol. 30, Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2017/hash/2cad8fa47bbef282badbb8de5374b894-Abstract.html.

Sepas-Moghaddam, A., Etemad, A., 2020. View-invariant gait recognition with attentive recurrent learning of partial representations. IEEE Trans. Biomet., Behavior Ident. Sci. 3, 124–137. https://doi.org/10.1109/tbiom.2020.3031470. URL: https://ieeexplore.ieee.org/abstract/document/9229117.

Sepas-Moghaddam, A., Etemad, A., 2022. Deep gait recognition: a survey. IEEE Trans. Pattern Anal. Mach. Intell. https://doi.org/10.1109/tpami.2022.3151865. URL: https://ieeexplore.ieee.org/abstract/document/9714177.

Sepas-Moghaddam, A., Ghorbani, S., Troje, N.F., Etemad, A., 2021. Gait Recognition using multi-scale partial representation transformation with capsules. In: 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 8045–8052). IEEE. URL: https://ieeexplore.ieee.org/abstract/document/9412517. https://doi.org/10.1109/icpr48806.2021.9412517.

Shahroudy, A., Liu, J., Ng, T.-T., & Wang, G. (2016). NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1010–1019). IEEE. URL: https://ieeexplore.ieee.org/abstract/document/7780484. https://doi.org/10.1109/cvpr.2016.115.

Shen, S., Sun, S.-S., Li, W.-J., Wang, R.-C., Sun, P., Wang, S., Geng, X.-Y., 2023. A classifier based on multiple feature extraction blocks for gait authentication using smartphone sensors. Comput. Electr. Eng. 108, 108663. https://doi.org/10.1016/j.compeleceng.2023.108663. URL: https://www.sciencedirect.com/science/article/pii/S0045790623000836.

Shiraga, K., Makihara, Y., Muramatsu, D., Echigo, T., Yagi, Y., 2016. GEINet: View-invariant gait recognition using a convolutional neural network. In: 2016 International Conference on Biometrics (ICB), pp. 1–8. IEEE. URL: https://ieeexplore.ieee.org/abstract/document/7550060. https://doi.org/10.1109/icb.2016.7550060.

Shopon, M., Bari, A., Gavrilova, M.L., 2021. Residual connection-based graph convolutional neural networks for gait recognition. Visual Comput. 37, 2713–2724. https://doi.org/10.1007/s00371-021-02245-9. URL: https://link.springer.com/article/10.1007/s00371-021-02245-9.

Singh, N.K., Khare, M., Jethva, H.B., 2022. A comprehensive survey on person re-identification approaches: various aspects. Multimedia Tools Appl. 81, 15747–15791. https://doi.org/10.1007/s11042-022-12585-w. URL: https://link.springer.com/article/10.1007/s11042-022-12585-w.

Song, C., Huang, Y., Huang, Y., Jia, N., Wang, L., 2019. GaitNet: An end-to-end network for gait based human identification. Pattern Recogn. 96, 106988. https://doi.org/10.1016/j.patcog.2019.106988. URL: https://www.sciencedirect.com/science/article/abs/pii/S0031320319302912.

Song, Y.-F., Zhang, Z., Shan, C., Wang, L., 2020. Stronger, faster and more explainable: a graph convolutional baseline for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, Association for Computing Machinery, New York, NY, USA, pp. 1625–1633, URL: https://dl.acm.org/doi/abs/10.1145/3394171.3413802. https://doi.org/10.1145/3394171.3413802.

Song, X., Huang, Y., Huang, Y., Shan, C., Wang, J., Chen, Y., 2022. Distilled light GaitSet: Towards scalable gait recognition. Pattern Recogn. Lett. 157, 27–34. https://doi.org/10.1016/j.patrec.2022.03.019. URL: https://www.sciencedirect.com/science/article/pii/S0167865522000848.

Sorber, L., Van Barel, M., De Lathauwer, L., 2013. Optimization-based algorithms for tensor decompositions: canonical polyadic decomposition, decomposition in Rank-$(L\_r, L\_r, 1)$ terms, and a new generalization. SIAM J. Optim. 23, 695–720. https://doi.org/10.1137/120868323. URL: https://epubs.siam.org/doi/abs/10.1137/120868323.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Machine Learn. Res. 15, 1929–1958. https://doi.org/10.5555/2627435.2670313. URL: https://dl.acm.org/doi/10.5555/2627435.2670313.

Sun, K., Xiao, B., Liu, D., Wang, J., 2019. Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, pp. 5693–5703. URL: https://ieeexplore.ieee.org/abstract/document/8953615. https://doi.org/10.1109/cvpr.2019.00584.

Teepe, T., Khan, A., Gilg, J., Herzog, F., Hörmann, S., Rigoll, G., 2021. Gaitgraph: graph convolutional network for skeleton-based gait recognition. In: 2021 IEEE International Conference on Image Processing (ICIP). IEEE, Anchorage, Alaska, USA, pp. 2314–2318. https://doi.org/10.1109/icip42928.2021.9506717.

Teepe, T., Gilg, J., Herzog, F., Hörmann, S., Rigoll, G., 2022. Towards a deeper understanding of skeleton-based gait recognition. In: 17th IEEE Computer Society Workshop on Biometrics 2022. IEEE/CVF. URL: https://arxiv.org/abs/2204.07855. https://doi.org/10.48550/arxiv.2204.07855.

Toshev, A., Szegedy, C., 2014. Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 1653–1660, URL: https://ieeexplore.ieee.org/document/6909610/. https://doi.org/10.1109/CVPR.2014.214.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Informat. Process. Syst. 30. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wan, Y., Yuan, C., Zhan, M., Chen, L., 2022. Robust graph learning with graph convolutional network. Informat. Process. Manage. 59, 102916. https://doi.org/10.1016/j.ipm.2022.102916. URL: https://www.sciencedirect.com/science/article/pii/S0306457322000413.

Wang, J., She, M., Nahavandi, S., Kouzani, A., 2010. A review of vision-based gait recognition methods for human identification. In 2010 International Conference on Digital Image Computing: Techniques and Applications, IEEE, pp. 320–327, URL: https://ieeexplore.ieee.org/abstract/document/5692583. https://doi.org/10.1109/DICTA.2010.62.

Wang, Y., Du, B., Shen, Y., Wu, K., Zhao, G., Sun, J., Wen, H., 2019a. EV-gait: event-based robust gait recognition using dynamic vision sensors. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 6351–6360, URL: https://ieeexplore.ieee.org/abstract/document/8953966. https://doi.org/10.1109/cvpr.2019.00652.

Wang, Y., Song, C., Huang, Y., Wang, Z., Wang, L., 2019b. Learning view invariant gait features with Two-Stream GAN. Neurocomputing 339, 245–254. https://doi.org/10.1016/j.neucom.2019.02.025. URL: https://www.sciencedirect.com/science/article/abs/pii/S0925231219302395.

Wang, L., Chen, J., Liu, Y., 2022. Frame-level refinement networks for skeleton-based gait recognition. Comput. Vis. Image Underst. 222, 103500. https://doi.org/10.1016/j.cviu.2022.103500. URL: https://www.sciencedirect.com/science/article/pii/S1077314222000972.

Wang, L., Chen, J., Chen, Z., Liu, Y., Yang, H., 2022a. Multi-stream part-fused graph convolutional networks for skeleton-based gait recognition. Connect. Sci. 34, 652–669. https://doi.org/10.1080/09540091.2022.2026294. URL: https://www.tandfonline.com/doi/full/10.1080/09540091.2022.2026294.

Wolf, T., Babaee, M., Rigoll, G., 2016. Multi-view gait recognition using 3D convolutional neural networks. In: 2016 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 4165–4169, URL: https://ieeexplore.ieee.org/abstract/document/7533144. https://doi.org/10.1109/icip.2016.7533144.

Wu, Z., Huang, Y., Wang, L., Wang, X., Tan, T., 2016. A comprehensive study on cross-view gait based human identification with deep CNNs. IEEE Trans. Pattern Anal. Mach. Intell. 39, 209–226. https://doi.org/10.1109/tpami.2016.2545669. URL: https://ieeexplore.ieee.org/abstract/document/7439821.

Xiang, T., Zhang, C., Song, Y., Liu, S., Yuan, H., Cai, T.W., 2021. Partial graph reasoning for neural network regularization. CoRR, abs/2106.01805. URL: https://arxiv.org/abs/2106.01805. arXiv:2106.01805.

Xu, Z., Lu, W., Zhang, Q., Yeung, Y., Chen, X., 2019. Gait recognition based on capsule network. J. Vis. Commun. Image Represent. 59, 159–167. https://doi.org/10.1016/j.jvcir.2019.01.023. URL: https://www.sciencedirect.com/science/article/abs/pii/S1047320319300318.

Xu, Q., Zheng, W., Song, Y., Zhang, C., Yuan, X., Li, Y., 2021. Scene image and human skeleton-based dual-stream human action recognition. Pattern Recogn. Lett. 148, 136–145. https://doi.org/10.1016/j.patrec.2021.06.003. URL: https://www.sciencedirect.com/science/article/pii/S0167865521001902.

Xu, J., Li, H., Hou, S., 2023. Attention-based gait recognition network with novel partial representation PGOFI based on prior motion information. Digital Signal Process. 133, 103845. https://doi.org/10.1016/j.dsp.2022.103845. URL: https://www.sciencedirect.com/science/article/pii/S1051200422004626.

Yang, Y., Miao, R., Wang, Y., Wang, X., 2022. Contrastive Graph Convolutional Networks with adaptive augmentation for text classification. Informat. Process. Manage. 59, 102946. https://doi.org/10.1016/j.ipm.2022.102946. URL: https://www.sciencedirect.com/science/article/pii/S0306457322000681.

Yue, J., Mei, Z., Ivanov, K., Li, Y., He, T., Zeng, H., 2022. Gait recognition by sensing insole using a hybrid CNN-attention-LSTM network. In: Deng Weihong, Feng Jianjiang, Huang Di, Kan Meina, Sun Zhenan, Zheng Fang, Wang Wenfeng, He Zhaofeng (Eds.), Biometric Recognition, Cham: Springer Nature Switzerland, pp. 484–492.

Yu, S., Tan, D., Tan, T., 2006. A Framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In: 18th International Conference on Pattern Recognition (ICPR'06), vol. 4, IEEE, pp. 441–444, URL: https://ieeexplore.ieee.org/abstract/document/1699873. https://doi.org/10.1109/icpr.2006.67.

Yu, W., Wan, S., Li, G., Yang, J., Gong, C., 2023. Hyperspectral image classification with contrastive graph convolutional network. IEEE Trans. Geosci. Remote Sens. 61, 1–15. https://doi.org/10.1109/TGRS.2023.3240721. URL: https://ieeexplore.ieee.org/document/10032180.

Zhang, Y., Huang, Y., Wang, L., Yu, S., 2019. A comprehensive study on gait biometrics using a joint CNN-based method. Pattern Recogn. 93, 228–236. https://doi.org/10.1016/j.patcog.2019.04.023. URL: https://www.sciencedirect.com/science/article/pii/S0031320319301694.

Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C., 2020a. Distribution-aware coordinate representation for human pose estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7091–7100. URL: https://ieeexplore.ieee.org/abstract/document/9157744. https://doi.org/10.1109/CVPR42600.2020.00712.

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., Smola, A., 2022. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2735–2745, URL: https://ieeexplore.ieee.org/document/9857221/. https://doi.org/10.1109/CVPRW56347.2022.00309.

Zhang, S., Wang, Y., Chai, T., Li, A., Jain, A.K., 2022. RealGait: Gait Recognition for Person Re-Identification. CoRR, abs/2201.04806. URL: https://arxiv.org/abs/2201.04806. https://doi.org/10.48550/arXiv.2201.04806. arXiv:2201.04806.

Zhang, H., Liu, X., Yu, D., Guan, L., Wang, D., Ma, C., Hu, Z., 2023. Skeleton-based action recognition with multi-stream, multi-scale dilated spatial-temporal graph convolution network. Appl. Intell., URL: https://link.springer.com/article/10.1007/s10489-022-04365-8. https://doi.org/10.1007/s10489-022-04365-8.

Zhao, A., Li, J., Ahmed, M., 2020. SpiderNet: A spiderweb graph neural network for multi-view gait recognition. Knowl.-Based Syst. 206, 106273. https://doi.org/10.1016/j.knosys.2020.106273. URL: https://www.sciencedirect.com/science/article/abs/pii/S0950705120304597.

Zheng, L., Zha, Y., Kong, D., Yang, H., Zhang, Y., 2022. Multi-branch angle aware spatial temporal graph convolutional neural network for model-based gait recognition. IET Cyber-Syst. Robot. 4, 97–106. https://doi.org/10.1049/csy2.12052. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/csy2.12052.

Zhu, Z., Guo, X., Yang, T., Huang, J., Deng, J., Huang, G., Du, D., Lu, J., Zhou, J., 2021. Gait Recognition in the Wild: A Benchmark. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14769–14779. URL: https://ieeexplore.ieee.org/document/9710045. https://doi.org/10.1109/ICCV48922.2021.01452.

Zuse, K., 1972. Der plankalkül. URL: http://zuse.zib.de/item/gHI1cNsUuQweHB6.