IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 March 2023, accepted 3 April 2023, date of publication 11 April 2023, date of current version 19 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266252

RESEARCH ARTICLE

Gait Recognition for 2-Second Walks Using Viewpoint Normalization and Sliding Window Process

PIYA LIMCHAROEN, NIRATTAYA KHAMSEMANAN, AND CHOLWICH NATTEE
Sirindhorn International Institute of Technology, Thammasat University, Khlong Luang, Pathum Thani 12120, Thailand

Corresponding authors: Nirattaya Khamsemanan (nirattaya@siit.tu.ac.th) and Cholwich Nattee (cholwich@siit.tu.ac.th)

This work was supported by the Thammasat University Research Unit in Gait Analysis and Intelligent Technology (GaitTech).

ABSTRACT Many events, such as robberies, missing people, and other suspicious activities, are often captured by cameras. However, these videos are, more often than not, short and not from an optimal angle. Biometric recognition techniques such as facial or iris recognition are inadequate for situations like this. Gait recognition techniques are more suitable than other types of biometrics in such situations. In this work, we propose a new gait recognition technique using viewpoint normalization and a sliding window process. The proposed technique is designed to handle short walking videos captured from any angle. The proposed technique consists of 3 steps. First, a 2-second walk is preprocessed using the sliding window process. This step allows us to generate more gait data in a form of a set of sliding windows from only a 2-second walk. Then sliding windows are transformed into the optimal viewpoint using *ViewNet*, a proposed neural network designed for finding and transforming sliding windows into the optimal viewpoint angle. Finally, local joint movement information is extracted from sliding windows and used to identify a person using *IdenNet*, a proposed neural network designed for identifying a person from local joint movements. Four evaluation methods, the top k accuracy test, the precision-recall curves, the cumulative matching characteristic curves, and the gallery-size test, are used to assess the proposed technique. The experimental results show that the proposed technique outperforms existing techniques on all four tests. In particular, the proposed technique can provide a small group, not more than 5, of suspects with a chance above 90% that the real person of interest is in the group. Moreover, the proposed technique still maintains high accuracy even when used with a larger pool of people.

INDEX TERMS Gait recognition, human identification, microsoft kinect, viewpoint, camera, biometric recognition.

I. INTRODUCTION

With cameras almost everywhere, security cameras, web cameras, car cameras, or mobile phone cameras, important events, such as robberies, abductions, or even terrorist activities, are recorded by these cameras. However, video clips that contain a person of interest are often concise with poor quality. Conventional biometric recognition techniques such as facial or iris recognition are unsuitable in these situations since they require close and steady information. Gait recognition, which can be used to identify a person from afar and does require high-resolution images or videos, is more suitable to use in these situations.

Gait is the locomotion of an animal. The basic locomotion of a human being is a bipedal walking with one foot in front of another, one step at a time. To put it in an easier term, gait is how a person moves his or her body from head to toe while walking. Gait is a person's biometric characteristic that includes physical and behavioral data. Physical or static biometrics in gait are limb lengths and a body's structure. Behavioral or dynamic biometrics are movement patterns of limbs or an entire body. The gait of a person is unique

The associate editor coordinating the review of this manuscript and approving it for publication was Vincenzo Conti.

and hard to modify. Gait data is not limited to providing biometrics information. Comparing the gait analysis of an individual to others can lead to assessment and feedback tools for areas such as athlete examination, persons with movement disorders, and predicting fall detection.

Gait recognition is a field of study that uses gait, both physical and behavioral characteristics, to identify a person. Unlike other biometric recognition techniques, a gait recognition technique can be done from afar without a subject's awareness [1], [2], [3], [4]. Gait recognition techniques can be used as tracking or surveillance and authentication systems.

Gait recognition techniques consist of two main steps. First, gait features are extracted from walking videos as input (gait feature extraction step). Then, these gait features are compared to the gait features of a known subject using various techniques. An output of the process is a label of gait features closest to the input (classification step). There are two main categories of gait recognition techniques; model-free (or appearance-based) and model-based.

II. RELATED WORK

A. MODEL-FREE GAIT RECOGNITION

Model-free gait recognition techniques typically use gait features extracted by isolating a person's silhouette from the background. This means that a person's silhouette is crucial information in the model-free gait recognition field. However, silhouettes and appearances change drastically with different clothes, carried objects, lighting environment, and different observation angles. As a result, the accuracies of model-free gait recognition techniques are based on silhouette qualities as discussed in [1], [4], [5], and [6]. Many model-free approaches such as [5], [6], [7], [8], [9], and [10] offer different ways to handle the viewpoint issue.

A gait energy image (GEI) is an image that is created by combining all spatial-temporal silhouettes of a subject while walking within a limited duration into one 2-dimensional image [11]. GEIs are standard gait features widely used in many model-free approaches [5], [6], [8], [10], [11]. GEI-based gait recognition approaches commonly combine silhouettes extracted from a sequence of frames in a walking video into one GEI, then find the similarities between GEIs. Most model-free gait features have higher feature dimensions which could lead to a heavy computation burden. Some GEI-based gait recognition techniques try to lighten the computation burden by reducing the dimensions, such as work by Zhang et al. [12] that uses a simple approach of GEI-based methods with the principal component analysis or PCA to reduce gait feature dimensions. Many model-free GEI images are obtained by averaging silhouettes from frames during a certain period of time. As a result, the temporal information is removed. In order to keep the temporal information, some model-free gait recognition works [13], [14] used the Long Short-Term Memory or LSTM on sequential silhouette images.

B. MODEL-BASED GAIT RECOGNITION

Model-based gait recognition techniques use structural information such as skeleton data and joint coordinates as gait features. This field of study has received more attention after low-cost sensors such as Microsoft Kinect became more available. Microsoft Kinect cameras were initially designed as input interface devices for gaming consoles such as XBOX-360. Microsoft Kinect cameras as well as their SDKs, generate a 3-dimensional model-based skeleton stream directly from a video stream. Many model-based recognition approaches such as [15], [16], [17], [18], [19], [20], [21], and [22] use gait features obtained from Microsoft Kinect devices. An advantage of the model-based gait recognition technique is that its data are kept in a more structural form with fewer dimensions, leading to a lighter load of computations. The model-based approaches also rely mostly on coordinates of body joints. These models minimize problems caused by changes in silhouettes and appearances (viewpoint, clothing, and carried objects).

Early model-based gait recognition works, [16], [17], [23] were designed to handle gait data from fixed-direction walks (view-dependent). Gaits features used in these works are average values of specific dynamic characteristics, such as stride lengths or angles of some three connected joints, over a period of time or a walk cycle, along with static characteristics, such as limb lengths or height. A walk is preprocessed and represented in the form of a vector. Among these works, a technique by Yang et al. [23] outperforms the rest in a fixed-direction walk dataset. Yang et al. [23] propose a technique using both standard deviations and mean values of static features (the limb lengths and the subject's height) and dynamic features (distances between non-connected joints). In the classification step, the technique from [23] uses k-NN with the Manhattan distance, similar to [17]. The results confirm that static features alone are not enough to achieve high accuracies. Dynamic features are needed to accomplish that goal. The best parameters for k-NN are the Manhattan distance function with the parameter of $k = 1$. However, Yang et al. [23] and other early works do not perform well with view-independent walks where gait features are collected from different observation angles. This problem is sometimes referred to as a viewpoint issue.

Khamsemanan et al. [20] propose a coordinate system called the Center-of-Body (CoB) relative coordinate system. The COB system uses the four center joints of the body (hip-center, hip-left, hip-right, and spine) as the fixed reference. Gait features are based on the CoB coordinates. Since all coordinates of all joints are relative to the center of the body, the technique proposed in [20] is equipped to handle the viewpoint issue. Intuitively, the entire body in each frame is rotated so that the body is always facing forward. The technique significantly outperforms earlier works, such as [17], [23], on a view-independent dataset collected from different observation angles.

Limcharoen et al. [21] also propose a model-based gait recognition technique that can handle the viewpoint issue by using a technique called Joint Replacement Coordinates (JRC). JRC is a new coordinate system that represents the relations of three connected joints. Therefore, this coordinate system focuses more on local movement, unlike [20], which focuses on the entire body movement. The technique from [21] outperforms [20], which suggests that the local movement of body parts is crucial in identifying a person from gaits.

Ahmed et al. [24] propose a new concept called Joint Relative Angle (JRA). A JRA is an angle between a vector from the hip-center joint to a joint, and a vector from the hip-center joint to another joint. Later, Ahmed et al. [19] introduce a technique called Joint Relative Distance (JRD). A JRD is a distance between a pair of skeletal joints (including non-connected joints). Both [19] and [24] use the Dynamic Time Warping (DTW) to measure the similarity between walks. The hip-center joint is the main contributor to both JRA and JRD approaches. Both techniques cannot be used when the hip-center joint is not detected. Moreover, both JRA and JRD approaches require data over a complete walking cycle, which means that both techniques may have serious issues when used with incomplete walking cycle data.

Following [24] and [19], Bari et al. introduce two view-independent gait features based on the joint relative cosine dissimilarity (JRCD) and the joint relative triangle area (JRTA) [25], [26]. JRCD is the cosine distance of any two joints. JRTA is an area of a triangle formed by the spine joint and the other two joints. A deep neural network consisting of four blocks of multi-layer perceptrons (MLP) is used in the classification process. Their works show that the best performance technique comes from an ordinary cross-entropy loss with hyperbolic tangent activation function (tanh) for activation layers. Even though JRCD and JRTA outperform JRA and JRD approaches and the works from Preis et al. [16], and Ball et al. [15], the same issues as in JRA and JRD approaches remains since the technique by Bari et al. still relies on a fixed joint, namely the spine joint, and it requires a complete walking cycle.

Lately, some model-based gait recognition techniques rely on skeleton data from Pose Estimation techniques. The OpenPose, [27], [28], is a pose estimation program that extracts 2-dimensional and 3-dimensional coordinates of body parts such as shoulders, elbows, hands, hips, knees, ankles, and feet from an image or a video based on a neural network model.

Liao et al. [29], [30], [31] produce a series of gait recognition works based on the pose estimation. Their work proposed gait recognition on top of joint coordinates from the OpenPose network.

In [29], Liao et al. propose a gait recognition technique called the pose-based temporal-spatial network (PTSN). PTSN is designed to obtain temporal-spatial features from gait pose sequences. A convolutional neural network (CNN) is used to extract spatial features, and LSTM is used to extract temporal features from the gait pose of a frame.

In [31], Liao et al. introduce a network called PoseGait. PoseGait extracts four types of features from 3-dimensional coordinates, which are *fpose* (coordinate of joints), *fangle* (two angles between two adjacent joints), *flimb* (Euclidean distances between two adjacent joints), and *fmotion* (coordinate difference between the same joint on two adjacent frames). Input for a classification network are matrices that combine four types of features. Their classification network consists of multiple CNN layers and two loss functions (Softmax and Center losses). Their experiments show that CNN extracts the important features efficiently and achieves better results than LSTM or recurrent neural network (RNN).

Other types of sensors are also used to obtain gait data. A work by An et al. [32] uses gait features obtained from Inertial Motion Unit (IMU) sensors. These sensors provide a 3-axis accelerometer and 3-axis gyroscope information, and a bend sensor. They use this information to estimate step length, stride length, and other joint angles. Their work also utilizes a deep neural network technique using Generative Adversarial Network (GAN) to create a synthetic dataset and combine it with a real dataset to achieve better regression results.

In this work, we propose a new model-based gait recognition technique designed to deal with the viewpoint issue and the short walking video issue. The proposed technique generates more information from a 2-second walk using the sliding window process, finds a viewpoint that provides optimal results in an unsupervised manner using a proposed neural network called *ViewNet*, extracts local joint movements, and uses another proposed network called *IdenNet* to identify a person. Unlike previous works, such as [19], [20], [21], [24], [25], [26], [29], and [31], our proposed technique does not rely on any particular angles of observation. Moreover, our proposed technique is designed to be used with gaits from at most 2-second walks and does not require complete walking cycle data. The experimental results show that our proposed technique outperforms existing techniques significantly on a view-independent 2-second walk dataset. Moreover, our proposed technique performs well under the gallery size test, Precision-Recall curve as well as the Cumulative Matching Characteristic test. This suggests that the proposed technique is suitable for real-world use.

III. METHOD

The proposed gait recognition technique consists of two main processes; viewpoint normalization and identification. The overview of the proposed technique is shown in Fig. 1.

Input of our technique is a skeleton stream of a 2-second walk captured by Microsoft Kinect cameras (with various observation angles, i.e., different viewpoints). One input consists of 40 consecutive frames of skeleton data where a frame consists of 3-dimensional coordinates of 20 joints. An input

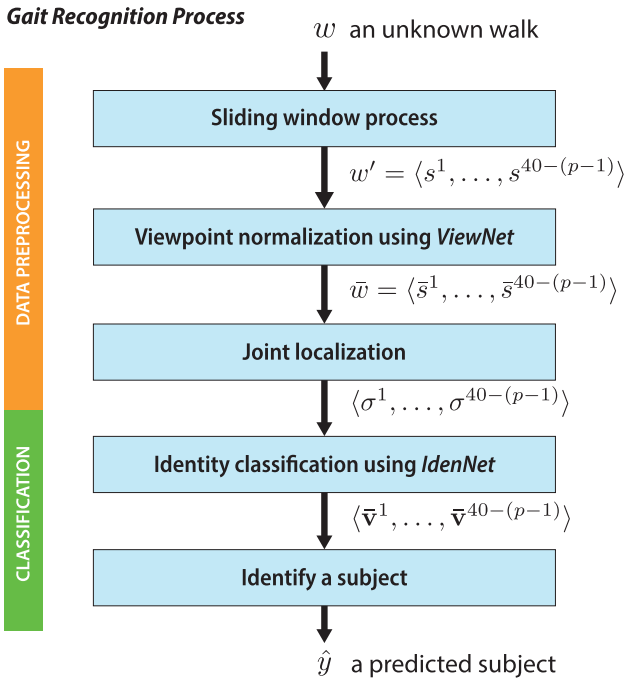


FIGURE 1. Overview the proposed gait recognition process.

walk w is defined as

$$w = \langle F^1, \ldots, F^{40} \rangle, \tag{1}$$

where F^k is frame k, for $k = 1, \ldots, 40$ of walk w. Frame F^k is defined as

$$F^k = \begin{bmatrix} \mathbf{f}_1^k \\ \vdots \\ \mathbf{f}_{20}^k \end{bmatrix}, \tag{2}$$

where $\mathbf{f}_l^k$ is a 3-dimensional coordinate (x, y, z) of a joint l, for $l = 1, \ldots, 20$, in a frame k. An input walk w has dimensions of 40 (frames) $\times$ 20 (joints) $\times$ 3 (x, y, z coordinate).

A. PRE-PROCESSING AND SLIDING WINDOWS

The proposed technique is designed to identify a person from just a 2-second walk. An input of a 2-second walk by itself may not contain enough information. We apply a sliding window technique to obtain and generate more information from a 2-second walk. A sliding window, in this case, is a series of p consecutive frames. Since a 2-second walk contains 40 frames, a parameter p can be between 1 and 40. We employ the sliding window technique to turn a 2-second walk with only 40 frames into a sequence of $40 - (p - 1)$ sliding windows. In particular, we create a sliding window walk w' from a 2-second w by letting

$$w' = \langle s^1, \ldots, s^{40-(p-1)} \rangle, \tag{3}$$

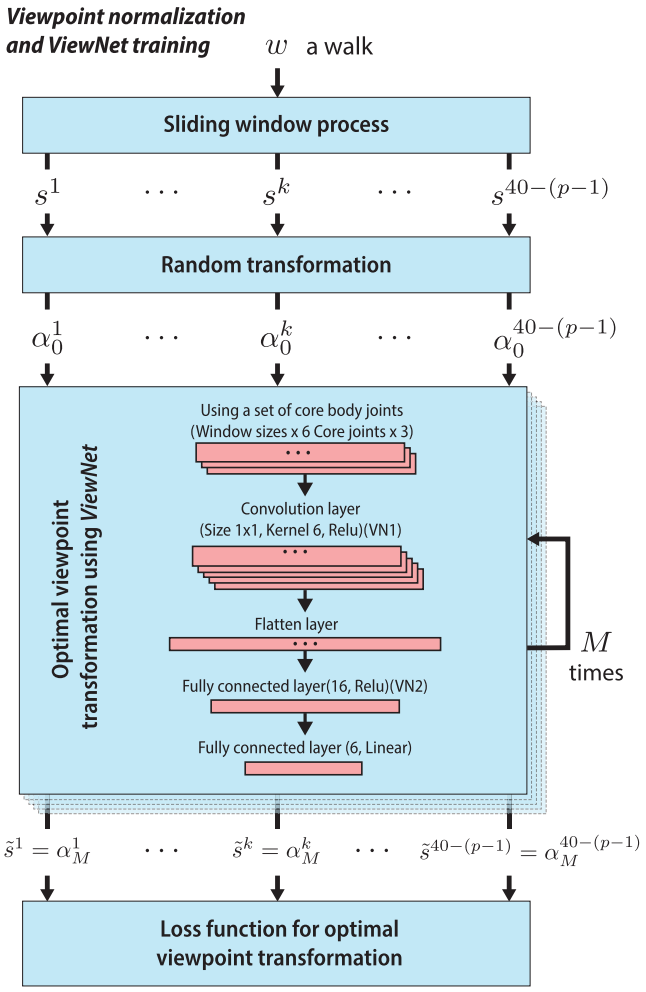


FIGURE 2. Proposed network for the viewpoint normalization model.

where s^k is a sliding window from frame k to frame $k+p-1$, for $k = 1, \ldots, 40 - (p - 1)$, defined as follows

$$s^k = \langle F^k, \ldots, F^{k+(p-1)} \rangle = \begin{bmatrix} \mathbf{f}_1^k & \cdots & \mathbf{f}_1^{k+(p-1)} \\ \vdots & \ddots & \vdots \\ \mathbf{f}_{20}^k & \cdots & \mathbf{f}_{20}^{k+(p-1)} \end{bmatrix}, \tag{4}$$

A sliding window walk w' has dimensions of $40 - (p - 1)$ (sliding windows) $\times$ p (sliding window size) $\times$ 20 (joints) $\times$ 3 (x, y, z coordinate).

The proposed sliding window technique generates $40 - (p - 1)$ walks with p frames from a single 40-frame walk. According to [20], a quick movement provides a high human identification accuracy. The sliding window technique produces many different sequences of quick movements, which, in turn, provide more ways to identify a person from a 2-second walk. Each of these new shorter walking sequences can be considered as another new and different quick movement.

The sliding window also reduces noise. Some joints detected from Microsoft Kinect are labeled as "inferred" or "not tracked" statuses. Joints with these statuses are consid-

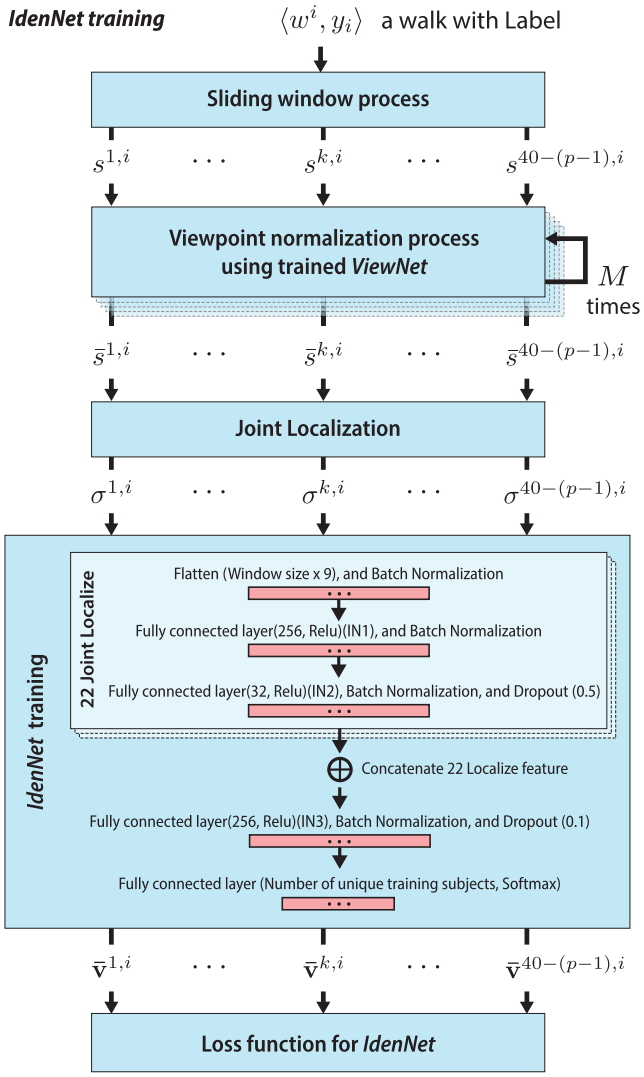


FIGURE 3. Proposed network for the identity classification model.

ered as noise because joints are extrapolated in the cases of ''inferred'' and not detected in the cases of ''not tracked''. A full 2-second walk contains more of these joints. By applying the sliding window technique to an original 2-second walk, we can obtain smaller walking sequences, where many of which contain much less or none of these noise.

We conduct an experiment to investigate the necessity of the sliding window process. The result of the investigation is in Table 6.

B. VIEWPOINT NORMALIZATION AND ViewNet TRAINING

The viewpoint normalization process aims to find a viewpoint that the center of the body stays at the same place as much as possible. Consequently, an optimal viewpoint for this work is defined as an observation point such that the torso or the core of bodies stay more or less in the same location regardless of an original observation angle. Mathematically, the optimal viewpoint is an angle such that a sum of the Manhattan distances of core body joints of any pair of frames is minimized. The purposes of this process are (1) to find such optimal viewpoint and (2) to find a way to transform a viewpoint of each frame into the optimal viewpoint. Note that different frames of walks may be required different transformations.

In a way, we *normalize* a viewpoint of a walk. We create a neural network called *ViewNet* to identify such optimal viewpoint and find a way to rotate and translate (shift) an original viewpoint of a walk into the optimal viewpoint. Since each walk may be obtained from a different angle of observation, different walks may have to be rotated and shifted differently. The *ViewNet* is designed to provide a way to normalize all ways to achieve the optimal result.

The viewpoint normalization and *ViewNet* training process consist of 3 steps, (1) Random geometric transformation (III-B1) where each sliding window walk is randomly rotated and translated, (2) Optimal viewpoint transformation (III-B2) where *ViewNet* rotates and translates each sliding window repeatedly M times into a new viewpoint, and (3) Loss calculation (III-B3) where displacements of the core body joints are minimized. An overview of the viewpoint normalization model is shown in Fig. 2.

1) RANDOM GEOMETRIC TRANSFORMATION

In this step, a sliding window from a walk is rotated and translated randomly. Note that different sliding windows in the same walk may be translated and rotated differently. The process of this step is described below.

For $\Omega = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6) \in \mathbb{R}^6$ and $\mathbf{v} \in \mathbb{R}^3$, let $t : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^3$ defined as

$$t(\Omega, \mathbf{v}) = R(\Omega)\mathbf{v} + \begin{bmatrix} \omega_4 \\ \omega_5 \\ \omega_6 \end{bmatrix} \tag{5}$$

where

$$R(\Omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_1 & -\sin\omega_1 \\ 0 & \sin\omega_1 & \cos\omega_1 \end{bmatrix} \begin{bmatrix} \cos\omega_2 & 0 & \sin\omega_2 \\ 0 & 1 & 0 \\ -\sin\omega_2 & 0 & \cos\omega_2 \end{bmatrix} \times \begin{bmatrix} \cos\omega_3 & -\sin\omega_3 & 0 \\ \sin\omega_3 & \cos\omega_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A transformation t rotates an input vector $\mathbf{v}$ counterclockwise ω_1 degrees along the x-axis, ω_2 degrees along y-axis and ω_3 degrees along z-axis, then shifts ω_4 units along the x-axis, ω_5 units along y-axis and ω_5 unit along z-axis.

Let $T : \mathbb{R}^6 \times \mathbb{R}^{p\times 20\times 3} \to \mathbb{R}^{p\times 20\times 3}$ defined as

$$T(\Omega, s^k) = \begin{bmatrix} t(\Omega, \mathbf{f}_1^k) & \cdots & t(\Omega, \mathbf{f}_1^{k+(p-1)}) \\ \vdots & \ddots & \vdots \\ t(\Omega, \mathbf{f}_{20}^k) & \cdots & t(\Omega, \mathbf{f}_{20}^{k+(p-1)}) \end{bmatrix}, \tag{6}$$

where t is as defined in (5).

For a sliding window s^k of a walk w, a new sliding window α_0^k is obtained from randomly rotating and translating s^k. This means $\alpha_0^k = T(\Omega_k, s^k)$, where Ω_k is selected randomly.

TABLE 1. Architecture of the proposed *ViewNet*.

Type	Size	Kernel size, Number of kernel	Activation
Input; a sliding window	$p \times 20 \times 3$		
Slice for 6 core joints	$p \times 6 \times 3$		
Convolution2D	$p \times 6 \times 6$	$1 \times 1 \times 3, 6$	ReLU
Flatten	$36p$		
Fully-connected	16		ReLU
Fully-connected	6		Linear
Output; a 6-dimensional vector	6		

Intuitively, this step is applied to the input gait data to handle the viewpoint issue further. Since the gait data are randomly rotated and transformed, gait features from different observation angles are added to the training process.

2) OPTIMAL VIEWPOINT TRANSFORMATION

We create a new network called *ViewNet*. The proposed *ViewNet* takes a sliding window α_0^k from a walk k as an input. An output of *ViewNet* is a 6-dimensional vector. The architecture of *ViewNet* is shown in Table 1. The proposed *ViewNet* is designed to take a sliding window and return a way to rotate and translate that particular sliding window.

This means that $T(ViewNet(\alpha_0^k), \alpha_0^k)$ is a new sliding window obtained from rotating and translating α_0^k according to $ViewNet(\alpha_0^k)$.

We repeat this process M times. Each time a sliding window is rotated and translated slightly to a new viewpoint according to *ViewNet*. This process is described as follows. Let

$$\begin{aligned}\alpha_1^k &= T(ViewNet(\alpha_0^k), \alpha_0^k)\\ \alpha_m^k &= T(ViewNet(\alpha_{m-1}^k), \alpha_{m-1}^k), m = 2, \ldots, M\\ \tilde{s}^k &= \alpha_M^k.\end{aligned}$$

An output of this step is a sliding window $\tilde{s}^k$ that has been translated and rotated M times.

Note that from the preliminary result, applying *ViewNet* once in a training process is not as robust. Applying *ViewNet* multiple times in the training process provides a much better result. This may be due to applying *ViewNet* once in the training process may have rotated and translated joints in a step that is too small to obtain an optimal result.

Figure 4 shows examples of inputs and outputs of *ViewNet*. The input frames on the left show different original observation viewpoints of walks. The output frames on the right show that all joints are transformed into the optimal viewpoint where the core joints stay, more or less, in the same location of each frame.

3) LOSS CALCULATION FOR VIEWPOINT NORMALIZATION

The objective of this proposed *ViewNet* is to create a tool that is able to find a way to rotate and translate a walk from any angle of observations into the viewpoint that the cores

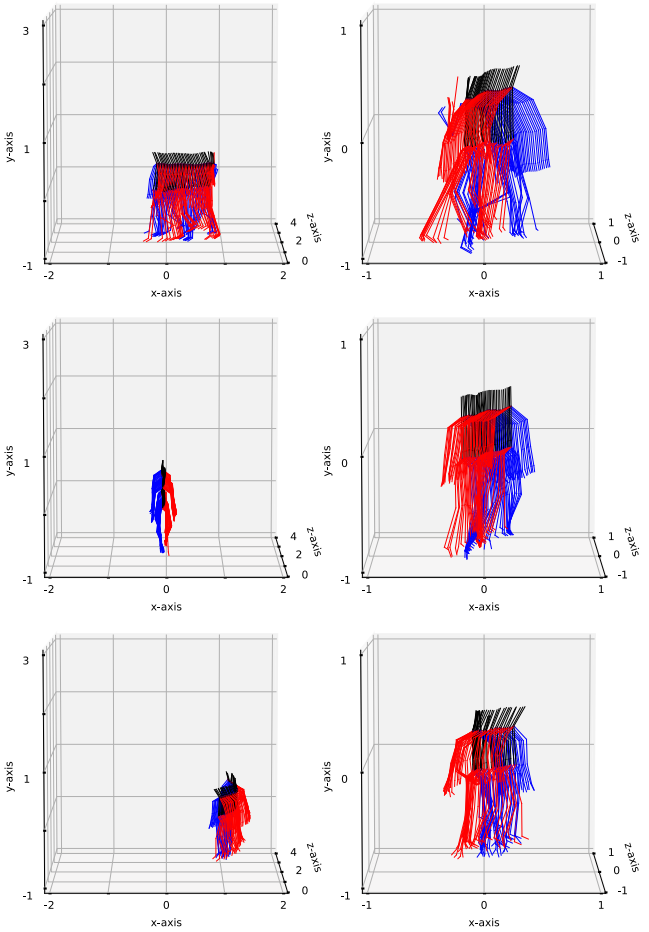


FIGURE 4. The *ViewNet* input (left) and output (right) of a walk without sliding window process.

of the bodies in all sliding windows stay at the same place. To do so, we need to focus on minimizing displacements of 6 core joints of the bodies, i.e., Hip-Center, Hip-Left, Hip-Right, Shoulder-Center, Shoulder-Left, and Shoulder-Right. To achieve this goal, we define a loss function for updating *ViewNet*'s weights as follows.

If $\tilde{s}^{k,i}$ is an output from the previous step of a sliding window k in a walk i and $\tilde{s}^{l,j}$ is an output from the previous step of a sliding window l in a walk j, then

$$\tilde{s}^{k,i} = \begin{bmatrix} \tilde{\mathbf{f}}_1^{k,i} & \cdots & \tilde{\mathbf{f}}_1^{k+(p-1),i} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{f}}_{20}^{k,i} & \cdots & \tilde{\mathbf{f}}_{20}^{k+(p-1),i} \end{bmatrix},$$

where $\tilde{\mathbf{f}}_\beta^{k,i}$ is a joint β of a sliding window s^k of a walk i that has been translated and rotated M times ($\beta = 1, \ldots, 20$).

Similarly, if $\tilde{s}^{l,j}$ is an output from the previous step of a sliding window l in a walk j, then

$$\tilde{s}^{l,j} = \begin{bmatrix} \tilde{\mathbf{f}}_1^{l,j} & \cdots & \tilde{\mathbf{f}}_1^{l+(p-1),j} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{f}}_{20}^{l,j} & \cdots & \tilde{\mathbf{f}}_{20}^{l+(p-1),j} \end{bmatrix}.$$

Let $C = \{c_1, c_2, \ldots, c_6\}$ be a set of indices of 6 core joints, and the loss function L_v is defined as

$$L_v = \sum_{i \neq j} \sum_{k=1,l=1}^{40-(p-1)} \sum_{\beta \in C} d(\tilde{\mathbf{f}}_{\beta}^{k,i}, \tilde{\mathbf{f}}_{\beta}^{l,j}), \tag{7}$$

where $d(\cdot, \cdot)$ is a Manhattan distance function.

The loss function L_v is minimized and is used to update weights in *ViewNet*.

The trained *ViewNet* is then used in the Identity Classification process.

C. IDENTITY CLASSIFICATION AND IdenNet TRAINING

Identity classification is a process that identifies a subject from a walk. In this work, we create a new network called *IdenNet* to find the probability that a sliding window belongs to a particular subject.

Let $(w')^i = \langle s^1, \ldots, s^{40-(p-1)} \rangle$ be a sliding window walk i in a dataset as defined in Eq. 3. We define

$$\bar{w}^i = \langle \bar{s}^1, \ldots, \bar{s}^{40-(p-1)} \rangle, \tag{8}$$

where $\bar{s}^{k,i} = T(ViewNet(s^{k,i}), s^{k,i})$, $T(\cdot, \cdot)$ is as defined in (6) and *ViewNet* is a trained network as described in section III-B.

For identity classification, a training data is $\{\langle \bar{w}^i, y_i \rangle\}_{i \in q}$, where q is the total number of walks in training set, $\bar{w}^i$ is as defined in (8) and y_i is a label of $\bar{w}^i$.

The identity classification and *IdenNet* training consists of 3 steps; (1) Joint Localization, (2) *IdenNet* Training and (3) Loss Calculation of Identity classification

1) JOINT LOCALIZATION

Since previous works such as [21] show that gait features from local movements of connected joints perform well with data from multi-view points, we create a new gait feature vector focusing on groups of connected joints instead of focusing on joint coordinates separately.

For a $\bar{s}^{k,i}$ is a sliding window k, rotated and translated according to the trained *ViewNet*, of a walk i, we construct a new gait feature of $\bar{s}^{k,i}$ as

$$\sigma^{k,i} = \begin{bmatrix} \mu_1^{k,i} & \cdots & \mu_1^{k+(p-1),i} \\ \vdots & \ddots & \vdots \\ \mu_{22}^{k,i} & \cdots & \mu_{22}^{k+(p-1),i} \end{bmatrix},$$

where

$$\mu_r^{k,i} = \begin{bmatrix} \text{LeftJoint}_r^{k,i} - \text{MiddleJoint}_r^{k,i} \\ \text{MiddleJoint}_r^{k,i} \\ \text{RightJoint}_r^{k,i} - \text{MiddleJoint}_r^{k,i} \end{bmatrix},$$

and $\text{LeftJoint}_r^{k,i}$, $\text{MiddleJoint}_r^{k,i}$, and $\text{RightJoint}_r^{k,i}$ are joins shown in Table 2.

2) IDENTITY CLASSIFICATION

We create a neural network called *IdenNet* to identify a person from gait features. The architecture of *IdenNet* is shown in Fig. 3 and Table 3.

TABLE 2. Index of joint localization.

Index r	LeftJoint_r	MiddleJoint_r	RightJoint_r
1	Knee Right	Ankle Right	Foot Right
2	Hip Right	Knee Right	Ankle Right
3	Hip Center	Hip Right	Knee Right
4	Spine	Hip Center	Hip Right
5	Knee Left	Ankle Left	Foot Left
6	Hip Left	Knee Left	Ankle Left
7	Hip Center	Hip Left	Knee Left
8	Spine	Hip Center	Hip Left
9	Head	Shoulder Center	Shoulder Right
10	Spine	Shoulder Center	Shoulder Right
11	Shoulder Center	Shoulder Right	Elbow Right
12	Shoulder Right	Elbow Right	Wrist Right
13	Elbow Right	Wrist Right	Hand Right
14	Head	Shoulder Center	Shoulder Left
15	Spine	Shoulder Center	Shoulder Left
16	Shoulder Center	Shoulder Left	Elbow Left
17	Shoulder Left	Elbow Left	Wrist Left
18	Elbow Left	Wrist Left	Hand Left
19	Hip Left	Hip Center	Hip Right
20	Shoulder Left	Shoulder Center	Shoulder Right
21	Shoulder Center	Spine	Hip Center
22	Head	Shoulder Center	Spine

An input of *IdenNet* is a gait feature $\sigma^{k,i}$ of a sliding window k of a walk i. An output of *IdenNet* is a probability vector of dimension N, the number of subjects in the dataset. This means that, for each $\sigma^{k,i}$ we have

$$\bar{\mathbf{v}}^{k,i} = IdenNet(\sigma^{k,i}) = \begin{bmatrix} v_1^{k,i} \\ \vdots \\ v_N^{k,i} \end{bmatrix},$$

where $v_j^{k,i}$ is a probability that $\sigma^{k,i}$ belong to a subject u in the dataset, $u = 1, \cdots, N$.

We use Softmax Cross-Entropy Loss to update weights in *IdenNet* and use a training batch size of 128 and "Adam" as a network optimizer.

D. GAIT RECOGNITION PROCESS

An overview of the proposed gait recognition process is shown in Fig. 1 and Table 4.

Let w be a walk of an unknown subject. The proposed gait recognition process is as follows:

1) Apply the sliding window process (as in III-A) to w, we get
$$w' = \langle s^1, \ldots, s^{40-(p-1)} \rangle.$$
2) Apply the viewpoint normalization (as in III-B) to each s^k, $k = 1, \ldots, 40 - (p-1)$ we get
$$\bar{s}^k = T(ViewNet(s^k), s^k),$$
where $T(\cdot, \cdot)$ is as defined in (6) and *ViewNet* is a trained network as described in section III-B.

TABLE 3. Architecture of the proposed *IdenNet* for predicting an identity of input window.

Type	Size	Activation
Input window (window sizes, joint localize, coordinates)	$p \times 22 \times 9$	
Slice on joint localize (22 localize of $(p \times 9)$)	$22 \times (p \times 9)$	
- Flatten - Batch normalization	$9p$	
- Fully-connected - Batch normalization	256	ReLU
- Fully-connected - Batch normalization - Dropout (0.5)	32	ReLU
Concatenate (22 localize $\times 32 = 704$)	704	
Fully-connected Batch normalization Dropout (0.1)	256	ReLU
Fully-connected	Number of subjects (N)	Softmax
Output	Number of subjects (N)	

3) Apply the joint localization process (as in III-C1) to each $\bar{s}^k$, we get a proposed gait features σ^k extracted from $\bar{s}^k$.
4) Apply the identity classification process (as in III-C2) to each σ^k, we get a probability vector

$$\bar{\mathbf{v}}^k = IdenNet(\sigma^k) = \begin{bmatrix} v_1^k \\ \vdots \\ v_N^k \end{bmatrix}.$$

5) Find an average vector of all probability vectors $\bar{\mathbf{v}}^k, k = 1, \ldots, 40 - (p-1)$

$$\bar{\mathbf{v}} = \frac{1}{40-(p-1)} \sum_{k=1}^{40-(p-1)} \bar{\mathbf{v}}^k.$$

Identify a subject for a walk w from the highest average probability,

$$\hat{y} = \underset{u=1,\ldots,N}{\arg\max}\, \bar{\mathbf{v}}_u. \tag{9}$$

An output of the proposed gait recognition is $\hat{y}$.

IV. EXPERIMENTS

A. DATASETS

The data used in this work comes from the dataset SIIT-CN-C [21]. SIIT-CN-C is a dataset collected by Microsoft Kinect cameras, where 130 unique subjects were asked to walk freely in any direction in an area of 335 cm by 250 cm. Each subject where asked to walk six rounds (30-60 seconds each round) where Microsoft Kinect cameras were placed at different heights and angles (235, 185, 150, and 100 cm from the ground and tilted 27, 20, 10, and 0 degrees down from the horizontal line, respectively).

We randomly extract 100 2-second walks (40 consecutive frames) from one subject in SIIT-CN-C so that there is no overlap between any 2-second walks. This means that the data in the experiments consists of 13,000 2-second walks (multi viewpoints) from 130 unique subjects (different heights, ages, weights, and genders).

TABLE 4. Output dimension of the proposed gait recognition process.

Type	Output dimension
Input walk (number of frames, joints, coordinates x, y, z)	$40 \times 20 \times 3$
Sliding window process	$40 - (p-1) \times p \times 20 \times 3$
Viewpoint normalization process, using $T(ViewNet(\cdot), \cdot)$ with $M = 10$	$40 - (p-1) \times p \times 20 \times 3$
Joint localization process	$40 - (p-1) \times p \times 22 \times 9$
Identity classification process, using $IdenNet(\cdot)$	$40 - (p-1) \times$ Number of subjects (N)
Identify subject for a walk	1
Output	1

B. EXPERIMENTAL SET-UP

We perform a 10-fold cross-validation technique on all experiments. The dataset is divided into ten sets. Each set contains ten different 2-second walks from a subject. This means that one set contains 1,300 2-second walks. In each round of the experiments, nine sets are used as training data, and one set is used as a test set.

In this paper, we implement our proposed techniques, including deep neural network models, using Python with TensorFlow library on a machine with an Intel i7-8700 CPU, 32 GB RAM, NVIDIA GeForce RTX 2060 GPU, and 6 GB GPU RAM.

C. PERFORMANCE EVALUATIONS

To assess the performances of the proposed techniques, we employ three popular evaluation methods for gait recognition techniques; (1) Top-k accuracy, (2) Precision-Recall Curves, (3) Cumulative Matching Characteristic (CMC) Curves, and (4) Gallery Size Test.

1) TOP-k ACCURACY TEST

Top-k accuracy test is one of the most crucial evaluation tools in biometric recognition techniques. This test assesses whether a particular biometric recognition technique can accurately identify a subject within the first k ranks. In other words, this test provides accuracies (percentages) when a subject is one of the top k predicted subjects from the particular technique. Top-k accuracy test is useful in real-world usages because, most of the time, authority figures are interested in a small group of people that contains the person of interest. The top k accuracy is calculated by

$$\text{Top-}k \text{ accuracy} = \frac{1}{n} \sum_{i=1}^{n} \Phi(i), \tag{10}$$

where n is the number of walk samples in the gallery

$$\Phi(i) = \begin{cases} 1, & \text{if the top } k \text{ ranked predict subjects contain} \\ & \text{the same subject (class) as } i; \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

2) PRECISION-RECALL (PR) CURVES

A Precision-Recall (PR) curve is a matric to evaluate the relevancy of a technique. A PR curve shows a relation between the precision of a technique (y-axis) against the recall of the technique (y-axis). Precision and Recall of a technique are defined as

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}, \tag{12}$$

and

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{13}$$

A technique with a larger area under the PR curve has higher relevancy.

3) CUMULATIVE MATCHING CHARACTERISTIC (CMC) CURVES

A Cumulative Matching Characteristic (CMC) Curve shows the performances of a particular biometric recognition technique based on the accuracy of each rank. A CMC curve shows a rank k on the x-axis against a top-k accuracy as defined in (10) on the y-axis.

4) GALLERY-SIZE TEST

A gallery-size test assesses how well a biometric recognition performs when the gallery size (dataset size) is increased. This test is used to confirm that a biometric recognition technique performs well not just when the number of unique subjects is small but also when the number of unique subjects in the dataset increases.

We evaluate a top-1 accuracy under increasing gallery size, starting from 10 subjects, increasing by 10 more each step until the gallery reaches 130 subjects. A gallery-size test is shown in the form of a curve where top-1 accuracies are shown on the y-axis, and gallery sizes are shown along the x-axis.

5) COMPARISON TO EXISTING TECHNIQUES

To check the robustness of the proposed technique, we also implement the JRC-CNN and JRC-KNN techniques [21], COB-CNN and COB-KNN techniques [20], the JRTA, JRCD with DNN, and the JRA, JRD with DNN techniques [25], techniques (PoseGait with 7 convolution layers and PoseGait with 20 convolution layers) by Liao et al. [31], and a technique by Yang et al. [23]. We also conduct all four evaluation methods, top-k accuracy test, PR curves, CMC curves, and the gallery size test on the proposed techniques and existing techniques mentioned above using the same dataset.

V. RESULTS AND DISCUSSIONS

A. TOP k ACCURACY TEST

Results of top 5 accuracy tests of the proposed technique (sliding window sizes $p = 1, 5, 10, 15, 20$) and existing techniques [20], [21], [25], [31] (PoseGait with 7 convolution layers and PoseGait with 20 convolution layers), and [23] are reported in the Table 5.

The experimental results show that the proposed technique with the sliding window size 10 yields the highest accuracies in all ranks $k = 1, \ldots, 5$. In theory, longer walking sequences contain more information and should have provided higher accuracies. But the experimental results show that the accuracies are dropped when the sliding window is more than 10. This may due to more noise in longer walking sequences. The experimental results support that the sliding window technique reduces noise in the data.

Furthermore, these accuracies from the proposed technique with the sliding window size 10 are significantly higher than existing techniques. This suggests that our proposed technique is better equipped to handle situations with short gait videos and the viewpoint issue than existing techniques.

The proposed technique produces high accuracies even with low ranks; 82.50 % accuracy in rank 1 to 95.76% in rank 5. This implies that, when used in real-world situations, the proposed technique would provide a small group of people (5 people) that 95% of the time would have a person of interest in this group.

The accuracies obtained from the JRC-CNN technique [21] of rank 1 to 5 are in the 70% to high 80%. The JRC-CNN technique also focuses on the movements of local joints. This indicates that gait recognition techniques focusing on local joint movements, such as our proposed technique and [21], provide better accuracies in a small rank than techniques using other types of gait features. However, [21] returns lower accuracies than the proposed technique. This demonstrates that only focusing on local joint movements alone is not enough to achieve high accuracies in the dataset of short walking videos. The proposed technique uses a sliding window process to generate more gait features from a short 2-second video where [21] does not. This key difference between the proposed and the JRC-KNN techniques [21] confirms that the sliding window process is crucial. Interestingly, the JRC-KNN technique [21] yields lower accuracies than the JRC-CNN techniques by almost 20% in all ranks from 1 to 5. This suggests that simple machine learning techniques such as k-NN might not be suitable for gait recognition, especially when walking videos are short and observation angles are not fixed.

The work from [20] focuses on quick movements of the entire body. This may be a reason why both COB-MLP and COB-KNN techniques [20] provide lower accuracies than the proposed technique and [21] in all ranks. This further indicates that local joint movements are better gait features. Accuracies from the COB-KNN technique are also lower than those from the COB-MLP technique. This, again, confirms

TABLE 5. Performance of the proposed techniques compared to the existing techniques.

Techniques	Top-k Accuracy (%) 1	2	3	4	5
Proposed gait viewpoint normalization					
Viewpoint normalization, 1 Frame	73.84$^{\dagger}\pm$ 1.39	83.49$^{\dagger}\pm$ 1.14	87.62$^{\dagger}\pm$ 1.06	90.07$^{\dagger}\pm$ 0.76	91.81$^{\dagger}\pm$ 0.68
Viewpoint normalization, 5 Frames	82.12 $\pm$ 1.39	89.55$^{\dagger}\pm$ 1.02	92.72$^{\dagger}\pm$ 0.72	94.47 $\pm$ 0.65	95.48 $\pm$ 0.64
Viewpoint normalization, 10 Frames	**82.50 $\pm$ 1.32**	**90.34 $\pm$ 0.83**	**93.21 $\pm$ 0.87**	**94.85 $\pm$ 0.64**	**95.76 $\pm$ 0.40**
Viewpoint normalization, 15 Frames	80.82$^{\dagger}\pm$ 1.63	88.27$^{\dagger}\pm$ 1.31	91.88$^{\dagger}\pm$ 0.89	93.78$^{\dagger}\pm$ 0.65	95.02$^{\dagger}\pm$ 0.66
Viewpoint normalization, 20 Frames	78.08$^{\dagger}\pm$ 2.01	86.85$^{\dagger}\pm$ 1.57	90.56$^{\dagger}\pm$ 1.14	92.57$^{\dagger}\pm$ 0.94	94.03$^{\dagger}\pm$ 0.87
Limcharoen et al. (JRC-CNN) [21]	70.68$^{\dagger}\pm$ 1.43	79.48$^{\dagger}\pm$ 1.18	83.88$^{\dagger}\pm$ 1.17	86.29$^{\dagger}\pm$ 1.05	88.05$^{\dagger}\pm$ 1.09
Limcharoen et al. (JRC-KNN) [21]	50.83$^{\dagger}\pm$ 1.89	60.36$^{\dagger}\pm$ 1.66	65.51$^{\dagger}\pm$ 1.50	69.15$^{\dagger}\pm$ 1.65	71.73$^{\dagger}\pm$ 1.81
Khamsemanan et al. (COB-MLP) [20]	51.14$^{\dagger}\pm$ 1.53	64.12$^{\dagger}\pm$ 1.57	71.21$^{\dagger}\pm$ 1.63	76.03$^{\dagger}\pm$ 1.39	79.60$^{\dagger}\pm$ 1.26
Khamsemanan et al. (COB-KNN) [20]	39.50$^{\dagger}\pm$ 1.45	49.11$^{\dagger}\pm$ 1.64	55.22$^{\dagger}\pm$ 1.74	59.32$^{\dagger}\pm$ 1.81	62.38$^{\dagger}\pm$ 1.55
Bari et al. (JRTA, JRCD and DNN) [25]	28.33$^{\dagger}\pm$ 1.22	40.45$^{\dagger}\pm$ 1.57	48.55$^{\dagger}\pm$ 1.88	54.47$^{\dagger}\pm$ 2.26	59.15$^{\dagger}\pm$ 2.48
Bari et al. (JRA, JRD and DNN) [19], [25]	28.73$^{\dagger}\pm$ 1.27	41.07$^{\dagger}\pm$ 1.74	49.09$^{\dagger}\pm$ 1.61	54.88$^{\dagger}\pm$ 1.78	59.40$^{\dagger}\pm$ 1.81
Liao et al. (PoseGait 7 convolution layers) [31]	29.35$^{\dagger}\pm$ 6.21	41.50$^{\dagger}\pm$ 7.86	48.98$^{\dagger}\pm$ 8.17	54.75$^{\dagger}\pm$ 8.25	58.96$^{\dagger}\pm$ 8.21
Liao et al. (PoseGait 20 convolution layers) [31]	23.09$^{\dagger}\pm$ 2.23	34.17$^{\dagger}\pm$ 2.72	41.81$^{\dagger}\pm$ 3.32	47.53$^{\dagger}\pm$ 3.39	52.39$^{\dagger}\pm$ 3.54
Yang et al. [23]	21.21$^{\dagger}\pm$ 0.61	29.33$^{\dagger}\pm$ 1.05	34.85$^{\dagger}\pm$ 1.05	39.31$^{\dagger}\pm$ 1.11	42.82$^{\dagger}\pm$ 1.14

Note: techniques in boldface font show our proposed techniques; accuracy in boldface font shows the highest performance between techniques and ranks.
† indicates significantly different from the highest accuracy with the 95% confidence level.

that simple machine learning techniques are not suitable for gait recognition.

Accuracies obtained from [25] and [31] are comparable and quite low in all ranks. In particular, the accuracies from [25] and [31] are around 20% – 30%. These accuracies increase when the rank is higher but not above 60% in rank 5. Gait recognition techniques by [25] and [31] perform poorly even though they are designed to handle the viewpoint issue. One reason for this might be because [25], [31] use gait data from fixed points of reference, e.g. JRA uses a hip-center joint. In other words, techniques in [25] and [31] transform gait data from multi-viewpoints to a single fixed viewpoint. Moreover, both [20] and [21] also use this same principle. However, this fixed viewpoint may not be an optimal viewpoint. The proposed technique is designed with this problem in mind. The *ViewNet* network is created to resolve this problem. Another reason that may lead [25] and [31] to have low performances is because [25] and [31] require a longer walking sequence. These techniques perform best with complete walking cycles. Consequently, [25] and [31] are unsuitable for short walking sequences. This illustrates that the sliding window process used in the proposed technique is important for datasets with short walking sequences.

Yang et al. [23] return the lowest accuracies in all 5 ranks. This technique uses static gait features such as limb lengths as well as dynamic gait features such as distance between non-connected joints. A machine learning technique used in [23] is k-NN with the Manhattan distance function. The poor performance of [23] implies that simple direct gait features and less complicated machine learning techniques are not adequate for the viewpoint issues with short walking videos.

B. PRECISION-RECALL (PR) CURVES

Precision-Recall curves of the proposed technique (sliding window sizes $p = 10$) and existing techniques [20], [21], [25], [31] (PoseGait with 7 convolution layers and PoseGait with 20 convolution layers), and [23] are shown in the Fig. 5.

The results of PR curves show that the proposed technique with 10 frames yields the highest area under the curve with 89.07%. The area under the curve of the proposed technique is 13.16% higher than the area under the curve of JRC-CNN [21] (75.91%), and much higher than other existing techniques. Only the proposed technique and the JRC-CNN [21] obtain the area under the PR curves above 75%, while the area under the PR curves of JRC-KNN [20], [21] (COB-MLP and COB-KNN) AUC are between 51.02% to 39.73%, and those of [23], [25], and [31] are below (between 25.05% to 21.13%.)

Compared to the proposed technique, the precision rates of other existing techniques decrease with high rates when the recall rate increases. At the recall rate of 50%, the proposed technique can maintain precision above 98%, while the JRC-CNN [21] precision rate is dropped to 92%, and the JRC-KNN [21] precision rate decreases quickly to 51%.

The PR curves suggest that the proposed technique yields much better discrimination performance between classes than other existing techniques. With the average accuracy of the retrieved set of 98%, the proposed technique can retrieve a larger set of results compared to those from the other existing techniques. The results show that when used in real-world situations with a certain threshold, the proposed technique can retrieve a larger set of results than existing techniques.

C. CUMULATIVE MATCHING CHARACTERISTIC (CMC) CURVES

Results of CMC curves, with the first 80 ranks of 130 ranks, of the proposed technique (sliding window sizes $p = 1, 5, 10, 15, 20$) and existing techniques [20], [21], [25], [31] (PoseGait with 7 convolution layers and PoseGait with 20 convolution layers), and [23] are shown in the Fig. 6.

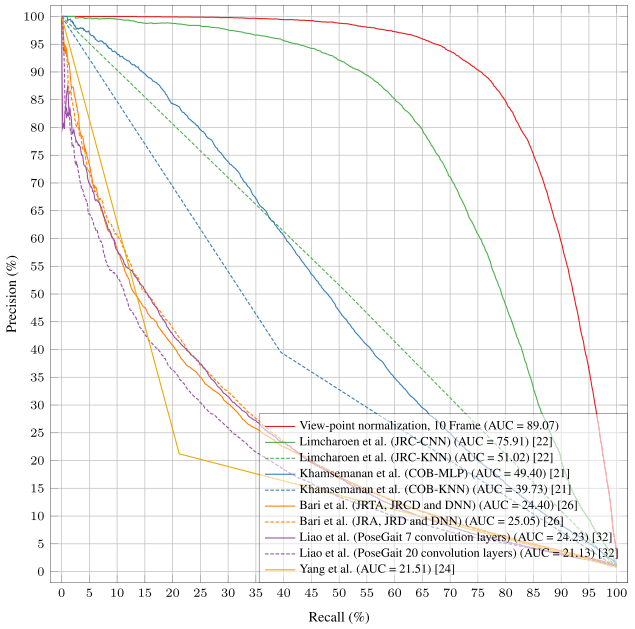


FIGURE 5. Micro-average precision-recall curves of the proposed and existing techniques.

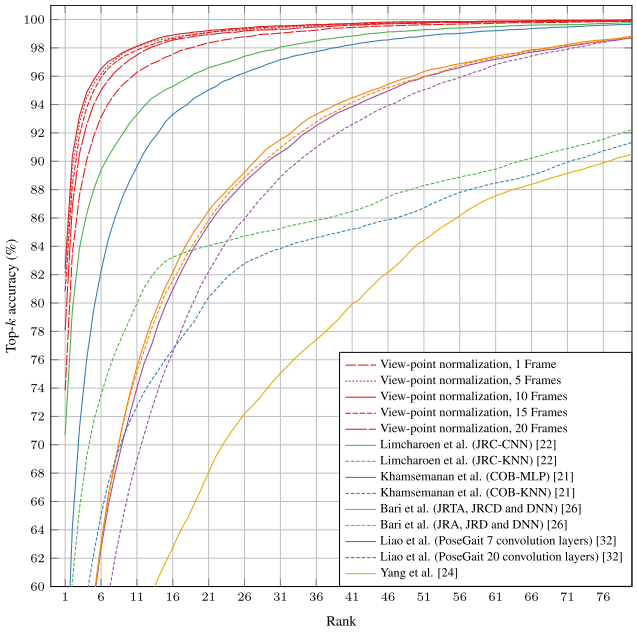


FIGURE 6. CMC curves of the proposed and existing techniques reported with 80 of 130 ranks.

From the experimental results, the proposed technique with 10 frames achieves 98% accuracy at rank 11 and above, whereas it takes until rank 31, 41, 71, and 72 for JRC-CNN [21], COB-MLP [20], and [31] to achieve 98% accuracy, respectively. The JRC-KNN, COB-KNN and Yang et al. do not obtain accuracies above 90% until rank 66, 71, and 77, respectively.

The results from the CMC curves suggest that, when used in real-world situations where walking videos are short and come from many observation angles, the proposed technique can provide authorities with a much smaller group of people with a percentage that the real person of interest is in that group. In contrast, other techniques would require much bigger groups of people with lower accuracies.

D. GALLERY-SIZE TEST

Results of the gallery-size test of the proposed technique (sliding window sizes $p = 1, 5, 10, 15, 20$) and existing techniques [20], [21], [25], [31] (PoseGait with 7 convolution layers and PoseGait with 20 convolution layers), and [23] are shown in Fig. 7.

In the gallery size test, the proposed technique with 10 frames obtains above 97% accuracy when the number of unique subjects in the dataset is 10 and maintains accuracies above 82% when the number of subjects is increased to 130. The results of the gallery size test show that the proposed technique performs very well even when the gallery size increases. Accuracies of the proposed technique do not drop quite quickly even when the gallery size increases from 10 to 130.

The JRC-CNN [21] starts with an accuracy of around 96% when the number of subjects is 10, and the accuracies

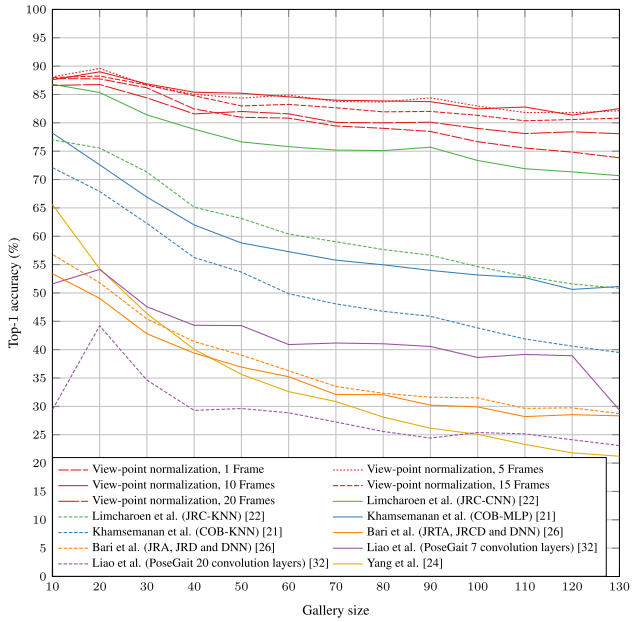


FIGURE 7. Prediction accuracies under different gallery sizes.

decrease down to just above 70% at 130 subjects. Accuracies of the JRC-KNN, COB-MLP, and COB-KNN start between 70% to 80% when there are 10 subjects and decrease more than 20% when there are 130 subjects in the gallery. Accuracies of [25], and [31] start below 60% at gallery size 1 and drop quickly down to below 30% when the gallery size is 130. The results illustrate that gait recognition techniques from [25], and [31] perform much worse, and the accuracies drop down much more quickly than the proposed technique when the number of people in the gallery size is increased.

TABLE 6. The proposed techniques with ablation studies.

Technique components			Top-1 accuracy (%)
Sliding window	ViewNet	IdenNet	
Included[a]	**Included**	**Included**	**82.50 ± 1.32**
Included[a]	Included	Not included[b]	35.42[†] ± 1.42
Included[a]	Not included	Included	80.47[†] ± 2.01
Included[a]	Not included	Not included[b]	14.75[†] ± 0.94
Not included	Included	Included	59.33[†] ± 1.79
Not included	Included	Not included[b]	15.22[†] ± 1.28
Not included	Not included	Included	60.31[†] ± 1.65
Not included	Not included	Not included[b]	8.52[†] ± 0.79

Note: the technique with a boldface font is the proposed viewpoint normalization with sliding window size 10 technique from Table 5; an accuracy in a boldface font shows the highest performance.
[†] indicates significantly different from the highest performance at the 95% confidence level.
[a] indicates the sliding window process with sliding window size 10.
[b] indicates the 1-NN with Euclidean distance function.

TABLE 7. The proposed techniques with adjusted network structure.

ViewNet					
No. of kernels (VN1)	6	3	12	6	6
No. of hidden units (VN2)	16	8	32	16	16
Total no. of parameters	5.9k	1.5k	23.3k	5.9k	5.9k
IdenNet					
No. of hidden units (IN1)	256	256	256	128	512
No. of hidden units (IN2)	32	32	32	16	64
No. of hidden units (IN3)	256	256	256	128	512
Total no. of parameters	941.6k	941.6k	941.6k	384.7k	2,596k
Top-1 accuracy (%)	82.50 ± 1.32	81.88 ± 1.83	**82.77 ± 1.91**	80.55[†] ± 1.77	81.43[†] ± 1.73

Note: an accuracy in boldface font shows the highest performance between techniques. The technique in blue (column 2) is the proposed viewpoint normalization with sliding window size 10 technique from Table 5.
[†] indicates significantly different from the highest performance with the 95% confidence level.

Interestingly the accuracy from Yang et al. is 65% at the gallery size 10, which is higher than those of [25], and [31] but drops even more quickly when the gallery size is increased. This shows that when there are not many subjects in the gallery, simple techniques such as Yang et al. may perform better than the rest, but their accuracy drops much more quickly when there are more people.

E. NETWORK COMPONENT ANALYSIS

To validate and study all three major components (sliding window, *ViewNet*, and *IdenNet*), we construct an experiment to investigate each component and all possible combinations of the three components. The results of the network components study of the proposed technique are reported in Table 6.

The result shows that the highest accuracy is achieved when all three components are included and it is significantly higher than other combinations of components with the 95% confidence level. This validates that all three components are needed in order to obtain the highest performance. When the *IdenNet* is not included, the accuracy is decreased quickly to 35.42% and when the *IdenNet* is employed alone without the other two components, the accuracy is 60.31%. This shows that the *IdenNet* plays a bigger role in the overall performance than the other two components. Similarly, the result also shows that the sliding window technique contributes to the overall performance more than the *ViewNet*. However, the *ViewNet* is not without its merit. Without the *ViewNet* component, the accuracy cannot reach it full potential.

The proposed technique with all three components achieves the highest accuracy 82.50% with statistical significance compared to other settings. This show that all three components are necessary and complement each other.

F. NETWORK PARAMETER ANALYSIS

We conduct an experiment to study the effects of the parameters of our proposed techniques on the accuracies. We vary parameters by doubling and decreasing parameters by half parameters in both *ViewNet* and *IdenNet*. The result of the parameter study is shown in Table 7.

Table 7 shows that the highest accuracy is obtained when the number of parameters of *ViewNet* is doubled but the number of parameters of *IdenNet* remains the same as the originally proposed technique. However, the highest accuracy is not significantly different from the accuracy obtained by the originally proposed technique. This implies that it is not necessary to increase parameters (bigger computational burden) in the proposed technique since the accuracies do not change much.

However, the accuracy is decreased significantly when parameters are reduced. This suggests that the parameter size of the proposed technique should not be reduced, otherwise, the optimal accuracy may not be achieved.

G. LIMITATIONS AND FUTURE DIRECTION

Even though the proposed technique outperforms other existing techniques significantly, it is not without its limitations.

Since the proposed technique is designed to be used on skeleton data, in particular joint coordinates, special devices such as Kinect cameras or other devices or programs that can provide joint data are needed. The proposed technique cannot be used directly on clips from typical security cameras. Extra steps are needed.

Secondly, two out of three major components of the proposed technique (*ViewNet* and *IdenNet*) are deep neural networks. They require computational power and time to process. Consequently, at the current state, the proposed technique may not be suitable for real-time human identification.

To address these two limitations, we are planning on improving our work in two main directions. First, we will improve our technique such that it can be used with images and video from typical security cameras directly without special sensors or special cameras such as Kinects. Second, we are also planning on modifying our technique to use less computational time and, hence, can be used to identify a person in real time.

VI. CONCLUSION

This paper proposes a new model-based gait recognition technique suitable for short walking videos (2 seconds) obtained from different observation angles (the viewpoint issue). The proposed technique consists of three major components: the sliding window, the *ViewNet* and the *IdenNet*. The proposed method works as follows.

First, a sliding window technique is applied to a 2-second walking video to generate shorter walking sequences. Next, the *ViewNet* finds an optimal viewpoint, where the sum of the displacements of core body joints over all walks and all frames are minimized. The *ViewNet* finds a different way to transform a shorter walking sequence into the optimal viewpoint. These transformations are different for different walking sequences. Finally, the *IdenNet* identifies a person's identity based on an output of the *ViewNet*.

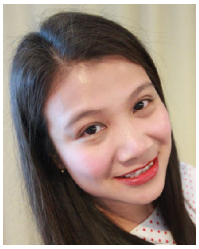
We conduct experiments using the 10-fold cross-validation techniques. We also employ popular evaluation tools for biometric recognition techniques, namely the top k accuracy test, the Precision-Recall (PR) curves, the cumulative matching characteristic (CMC) curves, and the gallery-size test, to assess the proposed technique against existing techniques by [20], [21], [23], [25], and [31]. The experimental results show that the proposed technique with 10 frames achieves the highest accuracies in all top 5 ranks (82.50% in rank 1, 90.34% in rank 2, 93.21% in rank 3, 94.85% in rank 4 and 95.76% in rank 5) and these accuracies are significantly higher than existing techniques. The top k accuracy test indicates that the proposed technique is more suitable for short walking videos from multiple observation angles than existing techniques. Moreover, the proposed technique performs well with the CMC curves, the PR curves and the gallery-size test. These show that the proposed technique can provide a small group of people with high accuracy that a person of interest is in the group (CMC curves), and the proposed technique can provide high accuracies even when used with a larger population (the gallery size test). In layman's terms, the proposed technique can provide a group of 5 candidates with above 95% chance that one of five is the person of interest from a short 2-second walking video obtained from any observation angle.

REFERENCES

[1] J. P. Singh, S. Jain, S. Arora, and U. P. Singh, "Vision-based gait recognition: A survey," *IEEE Access*, vol. 6, pp. 70497–70527, 2018.
[2] D. Matovski, M. S. Nixon, and J. N. Carter, *Gait Recognition*. Boston, MA, USA: Springer, 2014, pp. 309–318.
[3] J. E. Boyd and J. J. Little, *Biometric Gait Recognition*. Berlin, Germany: Springer, 2005, pp. 19–42.
[4] I. Rida, N. Almaadeed, and S. Almaadeed, "Robust gait recognition: A comprehensive survey," *IET Biometrics*, vol. 8, no. 1, pp. 14–28, Jan. 2019.
[5] Y. Makihara, T. Tanoue, D. Muramatsu, Y. Yagi, S. Mori, Y. Utsumi, M. Iwamura, and K. Kise, "Individuality-preserving silhouette extraction for gait recognition," *IPSJ Trans. Comput. Vis. Appl.*, vol. 7, pp. 74–78, Jul. 2015.
[6] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep CNNs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 209–226, Feb. 2017.
[7] Y. Makihara, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi, "Gait recognition using a view transformation model in the frequency domain," in *Computer Vision—ECCV*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 151–163.
[8] J. Lu and Y.-P. Tan, "Uncorrelated discriminant simplex analysis for view-invariant gait signal computing," *Pattern Recognit. Lett.*, vol. 31, no. 5, pp. 382–393, 2010.
[9] H. Chao, Y. He, J. Zhang, and J. Feng, "GaitSet: Regarding gait as a set for cross-view gait recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019 pp. 8126–8133.
[10] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "On input/output architectures for convolutional neural network-based cross-view gait recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2708–2719, Sep. 2019.
[11] J. Man and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.
[12] Y. Zhang, Y. Huang, L. Wang, and S. Yu, "A comprehensive study on gait biometrics using a joint CNN-based method," *Pattern Recognit.*, vol. 93, pp. 228–236, Sep. 2019.
[13] F. Battistone and A. Petrosino, "TGLSTM: A time based graph deep learning approach to gait recognition," *Pattern Recognit. Lett.*, vol. 126, pp. 132–138, Sep. 2019.
[14] Y. Feng, Y. Li, and J. Luo, "Learning effective gait features using LSTM," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 325–330.
[15] A. Ball, D. Rye, F. Ramos, and M. Velonaki, "Unsupervised clustering of people from 'skeleton' data," in *Proc. 7th ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, Mar. 2012, pp. 225–226.
[16] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien, "Gait recognition with Kinect," in *Proc. 1st Int. Workshop Kinect Pervasive Comput.*, Jun. 2012, pp. 1–4.
[17] O. V. Andersson and R. M. D. Araujo, "Person identification using anthropometric and gait data from Kinect sensor," in *Proc. AAAI*, 2015, pp. 1–7.
[18] E. Gianaria, M. Grangetto, M. Lucenteforte, and N. Balossino, "Human classification using gait features," in *Biometric Authentication* (Lecture Notes in Computer Science). V. Cantoni, D. Dimov, and M. Tistarelli, Eds. Cham, Switzerland: Springer, 2014, pp. 16–27.
[19] F. Ahmed, P. P. Paul, and M. L. Gavrilova, "DTW-based kernel and rank-level fusion for 3D gait recognition using Kinect," *Vis. Comput.*, vol. 31, nos. 6–8, pp. 915–924, Jun. 2015.
[20] N. Khamsemanan, C. Nattee, and N. Jianwattanapaisarn, "Human identification from freestyle walks using posture-based gait feature," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 119–128, Jan. 2018.
[21] P. Limcharoen, N. Khamsemanan, and C. Nattee, "View-independent gait recognition using joint replacement coordinates (JRCs) and convolutional neural network," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3430–3442, 2020.
[22] P. Limcharoen, N. Khamsemanan, and C. Nattee, "Gait recognition and re-identification based on regional LSTM for 2-second walks," *IEEE Access*, vol. 9, pp. 112057–112068, 2021.
[23] K. Yang, Y. Dou, S. Lv, F. Zhang, and Q. Lv, "Relative distance features for gait recognition with Kinect," *J. Vis. Commun. Image Represent.*, vol. 39, pp. 209–217, Aug. 2016.
[24] F. Ahmed, P. P. Paul, and L. M. Gavrilova, "Kinect-based gait recognition using sequences of the most relevant joint relative angles," *J. WSCG*, vol. 23, pp. 147–156, Jan. 2015.
[25] A. H. Bari and M. L. Gavrilova, "Artificial neural network based gait recognition using Kinect sensor," *IEEE Access*, vol. 7, pp. 162708–162722, 2019.
[26] A. H. Bari and L. M. Gavrilova, "Multi-layer perceptron architecture for Kinect-based gait recognition," in *Advances in Computer Graphics* (Lecture Notes in Computer Science), M. Gavrilova, J. Chang, N. M. Thalmann, E. Hitzer, and H. Ishikawa, Eds. Cham, Switzerland: Springer, 2019, pp. 356–363.
[27] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
[28] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4645–4653.

[29] R. Liao, C. Cao, B. E. Garcia, S. Yu, and Y. Huang, "Pose-based temporal-spatial network (PTSN) for gait recognition with carrying and clothing variations," in *Biometric Recognition* (Lecture Notes in Computer Science), J. Zhou, Y. Wang, Z. Sun, Y. Xu, L. Shen, J. Feng, S. Shan, Y. Qiao, Z. Guo, and S. Yu, Eds. Cham, Switzerland: Springer, 2017, pp. 474–483.
[30] W. An, R. Liao, S. Yu, Y. Huang, and C. P. Yuen, "Improving gait recognition with 3D pose estimation," in *Biometric Recognition* (Lecture Notes in Computer Science), J. Zhou, Y. Wang, Z. Sun, Z. Jia, J. Feng, S. Shan, K. Ubul, and Z. Guo, Eds. Cham, Switzerland: Springer, 2018, pp. 137–147.
[31] R. Liao, S. Yu, W. An, and Y. Huang, "A model-based gait recognition method with body pose and human prior knowledge," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107069.
[32] S. An, Y. Tuncel, T. Basaklar, G. K. Krishnakumar, G. Bhat, and U. Y. Ogras, "MGait: Model-based gait analysis using wearable bend and inertial sensors," *ACM Trans. Internet Things*, vol. 3, no. 1, pp. 1–24, Oct. 2021.

PIYA LIMCHAROEN received the bachelor's, master's, and Ph.D. degrees in computer sciences from the Sirindhorn International Institute of Technology, Thammasat University, Thailand. He is currently a Research Assistant with the Thammasat University Research Unit in Gait Analysis and Intelligent Technology (GaitTech), Sirindhorn International Institute of Technology, Thammasat University.



NIRATTAYA KHAMSEMANAN received the bachelor's degree (cum laude) in mathematics from Cornell University, USA, and the master's and Ph.D. degrees in mathematics from the University of California at Los Angeles (UCLA), USA. She is currently an Associate Professor with the Sirindhorn International Institute of Technology, Thammasat University, Thailand.

CHOLWICH NATTEE received the bachelor's degree in computer engineering from Chulalongkorn University, Thailand, and the master's and D.Eng. degrees in computer science from the Tokyo Institute of Technology, Japan. He is currently an Associate Professor with the Sirindhorn International Institute of Technology, Thammasat University, Thailand.