IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 17, 2019, accepted October 30, 2019, date of publication November 7, 2019, date of current version November 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952065

Artificial Neural Network Based Gait Recognition Using Kinect Sensor

A. S. M. HOSSAIN BARI AND MARINA L. GAVRILOVA
Department of Computer Science, University of Calgary, Calgary, AB T2N 1N4, Canada

Corresponding author: A. S. M. Hossain Bari (asmhossain.bari@ucalgary.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada for the Discovery Grant (DG) of Machine Intelligence for Biometric Security (Principal Investigator: M. Gavrilova) under Grant 10007544.

ABSTRACT Accurate gait recognition is of high significance for numerous industrial and consumer applications, including video surveillance, virtual reality, on-line games, medical rehabilitation, collaborative space exploration, and others. This paper proposes a new architecture designed using deep learning neural network for a highly accurate and robust Kinect-based gait recognition. Two new geometric features: joint relative cosine dissimilarity and joint relative triangle area are introduced. Both of the proposed features are view and pose invariant, thus enhancing recognition performance. The proposed neural network model is trained using the feature vector of dynamic joint relative cosine dissimilarity and joint relative triangle area. Subsequent application of Adam optimization method minimizes the loss of the objective function iteratively. The performance of the proposed deep learning neural network architecture is evaluated on two publicly available 3D skeleton-based gait datasets recorded with the Microsoft Kinect sensor. It is experimentally proven that the accuracy, precision, recall, and F-score of the proposed neural network architecture, trained using introduced dynamic geometric features, is superior to other state-of-the-art methods for Kinect skeleton-based gait recognition.

INDEX TERMS Kinect-based gait recognition, human motion, Microsoft Kinect, joint relative cosine dissimilarity, joint relative triangle area, deep learning neural network.

I. INTRODUCTION

Gait recognition is one of the tasks commonly used in a multitude of industrial and consumer applications, such as video surveillance, virtual reality, on-line games, medical rehabilitation, collaborative space exploration, and others. Biometric gait is a type of behavioral biometric, which provides one of the most popular unobtrusive means of remote authentication. Since biometric gait is acquired from a distance, the identity of an individual can be recognized remotely [1]. Therefore, gait recognition system can be used for person authentication [2], human action recognition [3], gender recognition [4], surveillance [5], analysis of abnormal gait in physiological treatment [6], and psychological state assessment [7]. Gait analysis is highly dependent on the motion of different body joints. Although physical injuries and tiredness may affect the walking pattern of an individual, it is still quite difficult to imitate another person's gait [8], [9]. For this reason, biometric gait is well-suited for robust unobtrusive authentication system where a subject can be authenticated without direct interaction with the system.

The deep learning methodology is a popular machine learning (ML) technique, which opens new doors for advanced analysis of human motion. Unlike deep learning, traditional machine learning methodology has been previously applied for gait recognition [2], [9]–[13]. While those methods achieved reasonable results on Kinect-based gait sequences, they suffer from severe limitations. First, although prior researches extracted classifying features from the skeleton model of Kinect, feature extraction methodologies were limited to selective body joints of the skeleton model of Kinect. Relative motions and directions considering all combinations of body joints were not considered. Therefore, more distinctive robust feature extraction methodology is needed to consider relative motions of all body joints. Second, recognition methodologies were limited to traditional machine learning methods. Powerful universal approximation algorithm, such as neural network, provides new opportunity to improve recognition performance of Kinect-based gait recognition. Third, while all combinations of body joints are considered for feature extraction, co-dependencies of body joints must

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

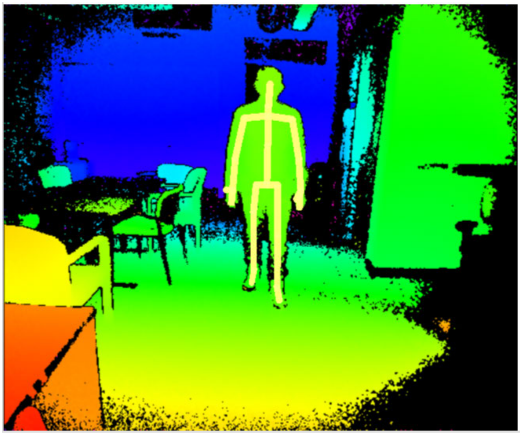


FIGURE 1. Human skeleton identification from the depth image acquired from the Microsoft Kinect.

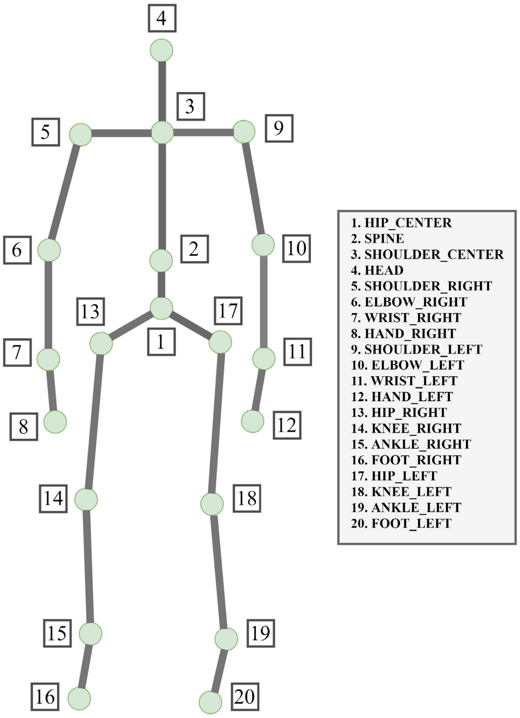


FIGURE 2. Representation of human body joints by Microsoft Kinect.

be prevented. Therefore, incorporation of the regularization method is necessary to achieve robust recognition performance.

In this paper, we introduce a deep learning neural network architecture for Kinect-based gait recognition that solves the above-mentioned deficiencies. Microsoft Kinect produces color-based depth video frame with the human skeleton from the 2D color image shown in Figure 1. The proposed system utilizes the 3D coordinates of the body joints of the 3D skeleton as an input, acquired by Microsoft Kinect (see Figure 2). The coordinates of each of the joints of the human skeleton are used to extract the distinctive features for training a neural network architecture. The major contributions of this paper can be outlined as follows. First, multiple gait cycles are extracted from a gait sequence. Intuitively, the extraction of multiple gait cycles acts as the data augmentation in the neural network architecture. Second, two unique view and pose invariant geometric features: joint relative triangle area (JRTA) and joint relative cosine dissimilarity (JRCD) are introduced. We extract dynamic JRTA and JRCD features from each of the frames of the gait cycle to create highly distinctive gait signatures. Then, JRTA and JRCD features are fused in such a way that resulting feature vector is compatible as input to the proposed deep learning neural network. Third, a new architecture of deep learning neural network is designed for Kinect-based gait recognition. Fourth, the proposed deep leanring neural network is optimized by the Adam optimization [14] method to determine the optimal weights and biases. Furthermore, the performance of the Adam optimization method is compared with the Root Mean Square Propagation (RMSProp) [15] and Stochastic Gradient Descent (SGD) [16] optimization methods to find out which optimization method is better suited for the proposed deep learning neural network. Publicly available UPCV gait dataset [17] and Kinect Gait Biometry dataset [2] are used to evaluate the performance of the proposed deep learning neural network based gait recognition method. Results of the cross-validation experimentation demonstrate that the proposed system outperforms all other recently proposed state-of-the-art Kinect-based gait recognition methods [10], [11], [12], [13], [9] in terms of recognition accuracy, precision, recall, and F-score on both datasets. The preliminary results on the use of deep learning for biometric gait recognition were reported in [18].

The rest of the paper is organized as follows. Background research work related to Kinect-based gait recognition is presented in Section II. Proposed Kinect-based gait recognition methodology is described in Section III including multiple gait cycle detection, unique feature extraction, and the proposed deep learning neural network architecture. The details of the two benchmark datasets are described in Section IV. Experimental results of the proposed methodology on two datasets are reported in Section V. Performance of the proposed neural network is compared with the traditional machine learning classifier in Section VI. Performances of the recent Kinect-based gait recognition methods are compared in Section VII. Conclusion and future direction of the research work are presented in Section VIII.

II. RELATED WORK

The research on person identification using biometric gait analysis has been extensively conducted for many years. Gait analysis relies on video sequence recorded by the motion capture system. Motion-based gait recognition can be either model-free or model-based [19]. In the model-free approach, binary segmented version of the silhouette of the gait sequence is generated to analyze the changes of the motion of the human body. Han and Bhanu identified silhouette sequences and extracted gait energy images (GEI) by averaging the silhouette sequences over the gait cycle [20].

Li and Chen [21] developed a better version of GEI image by determining both foot energy image and head energy image from the silhouette sequences and fusing both energy images. Active energy image [22], gait flow image [23], and structural gait energy image [21] are other examples of model-free based gait recognition. Although the model-free approaches are not computationally expensive, the recognition accuracy suffers from the changes of view angle and scale variation of the subject. Moreover, these approaches do not exploit 3D data as input. As a result, model-free approaches are not effective in real-life scenarios.

In the model-based approach, the model is constructed and updated over time by estimating changes of parameters considering different body parts from the video input of the gait sequence. Model generation is computationally expensive and therefore, it is not a robust approach for many practical applications [24]. BenAbdelkader *et al.* [3] extracted features by computing the stride length and the speed of walking. Yam *et al.* [25] automatically recognized a person by identifying the changes of motion between walking and running. This method is view and scale-invariant but requires extensive computation for model generation. These works were conducted before the invention of Microsoft Kinect.

Research on model-based gait recognition have emerged after the release of Microsoft Kinect as a low cost commercial sensor. Initially Kinect was envisioned as a consumer console focusing specifically on motion capturing tasks in the indoor environment. Due to its low cost, portability, and convenience of data access, it rapidly evolved from a human motion research in a controlled environments to more practical and versatile applications, such as fall detection in hospitals [26], [27], design of companion robots for elderly [28], or interpretation of hand signals for emergency response [29]. Currently, there are applications of Kinect sensor in outdoor environments. For instance, in 2017, autonomous systems equipped with the Kinect sensor were introduced for the rescue operations [30], [31], [32]. A study on the use of multiple Kinect sensors to increase the outdoor tracking coverage was carried out by Banerjee *et al.* [33].

Studies comparing Kinect based technology with traditional ways to capture motion data have been abundant. Kinect convenience stems from the fact that it can generate human skeleton model in 3D space in real-time [34] without using any marker-based equipment attached to the human body. The Kinect sensor can provide depth information with acceptable accuracy without causing any noticeable systematic errors [35]. Moreover, Schmitz *et al.* [36] concluded that the accuracy and precision of joint angle measurement in the 3D skeleton model obtained from Kinect is comparable with the marker-based system. Therefore, 3D data of the skeleton model of Kinect can be utilized for model-based gait recognition.

There was a surge in new methodologies developed in the research community for Kinect-based gait recognition. Preis *et al.* [11] extracted thirteen anthropometric features—eleven static and two dynamic features—from the Kinect skeleton model and used traditional classifier, such as Naïve Bayes for person identification. Ball *et al.* [10] applied unsupervised K-means clustering algorithm on the motion data extracted from the lower body part. Kastaniotitis *et al.* [17] proposed to calculate Euler angle to represent the direction of the human body limb while walking. They investigated K-nearest neighbors (KNN) and support vector machine (SVM) classifiers for the recognition purpose. Moreover, Ahmed *et al.* [12] proposed joint relative angle (JRA) and joint relative distance (JRD) features and utilized dynamic time warping to measure the dissimilarity score of the training and testing sets. On the other hand, Yang *et al.* [13] combined twenty relative distance features and twenty anthropometric features from the skeleton model of the Kinect. Random subspace method was applied to reduce the size of the feature vector incorporation with KNN classifier. Furthermore, Sun *et al.* [9] also trained KNN classifier using eight static features and four groups of angle feature considering upper and lower limbs. Recognition methodologies of the aforementioned related works are limited to traditional machine learning techniques. On the other hand, multi-layer perceptron based model using a single hidden layer was designed by Andersson and Araujo [2]. But the multi-layer perceptron model designed by them did not show better recognition performance than KNN classifier.

In this paper, we investigate whether improved machine learning model can be designed specifically for Kinect-based gait recognition and whether new features can be proposed to train this machine learning model for further performance improvement. In the subsequent section, the details of the proposed gait recognition method are described.

III. PROPOSED METHOD

A. OVERALL METHODOLOGY

Our proposed framework presents several key contributions. We introduce two unique geometric features: joint relative cosine dissimilarity (JRCD) and joint relative triangle area (JRTA). The geometric features have the properties of exhibiting dynamic spatio-temporal characteristics of human while walking. The spatio-temporal distinctive features are extracted over the frame of a gait cycle or multiple gait cycles. JRCD and JRTA features of a gait cycle are merged to represent the feature vector of the walking pattern of an individual. The feature vector is utilized to train a neural network. We propose a deep learning neural network, specialized for Kinect-based gait recognition. Since the gait cycle is different among people, the feature vector of joint relative cosine dissimilarity and joint relative triangle area are resampled to a vector of a fixed size. The feature vector of fixed size becomes the input to train the proposed deep learning neural network. The deep learning neural network architecture is designed by cascading fully connected dense layer, batch normalization layer, activation layer, and dropout layer. The hyper-parameters of each layer of the neural network are adjusted to secure the best performance for gait recognition.

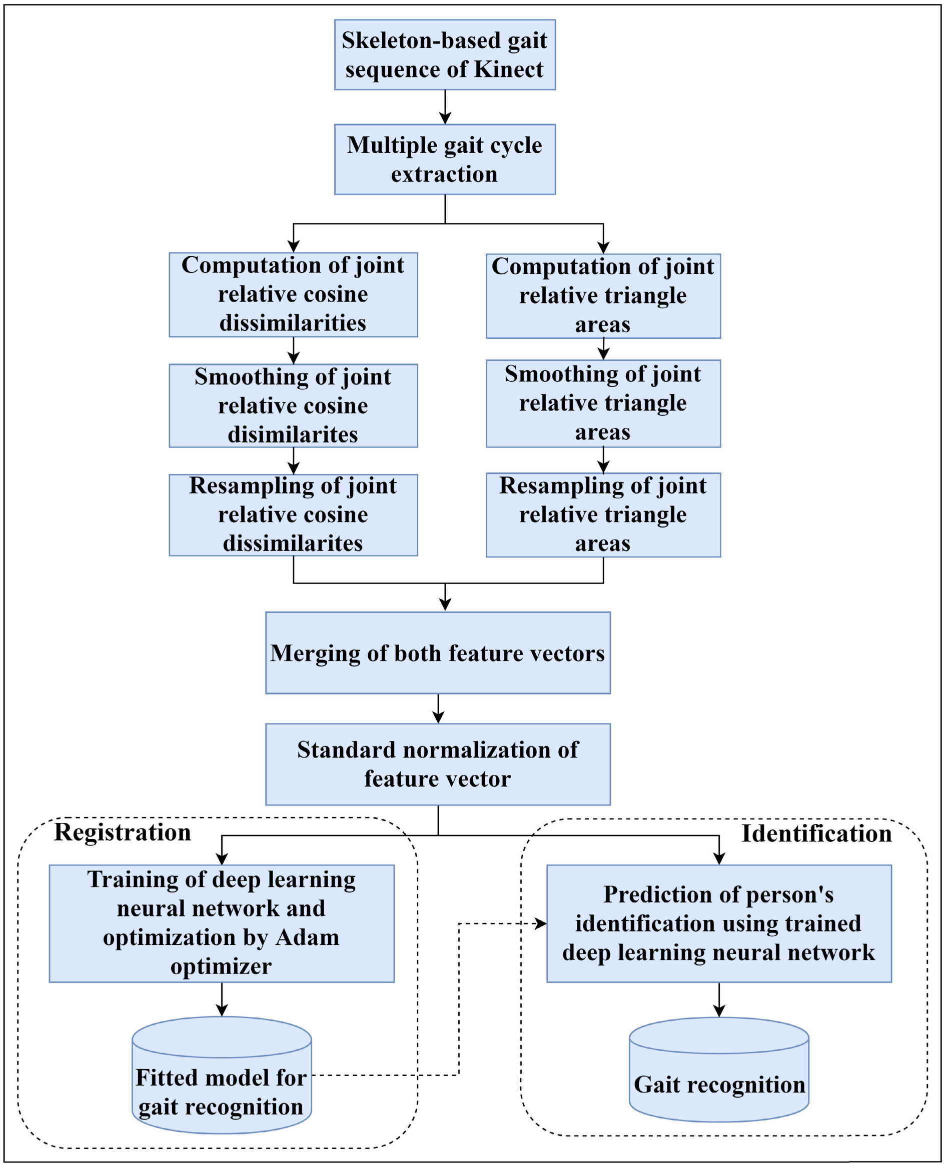

Skeleton-based gait sequence of Kinect
Multiple gait cycle extraction
Computation of joint relative cosine dissimilarities
Computation of joint relative triangle areas
Smoothing of joint relative cosine disimilarites
Smoothing of joint relative triangle areas
Resampling of joint relative cosine dissimilarites
Resampling of joint relative triangle areas
Merging of both feature vectors
Standard normalization of feature vector
Registration
Identification
Training of deep learning neural network and optimization by Adam optimizer
Prediction of person's identification using trained deep learning neural network
Fitted model for gait recognition
Gait recognition

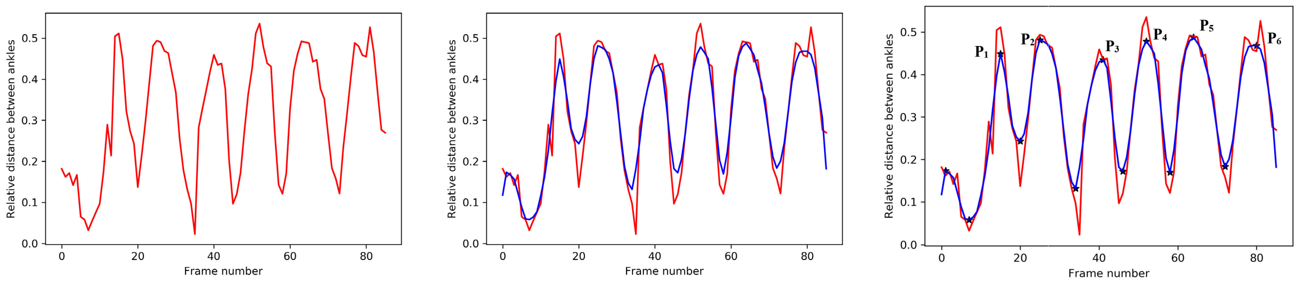
FIGURE 3. Overall system flowchart of the proposed framework, where new user registration process is shown on the left and subsequent identification on the right.

We design our proposed model in such a way that does not suffer from overfitting. After the design of the model, we propose to use Adam optimizer to minimize the loss of the cost function for the deep learning neural network. Figure 3 shows the flowchart of the overall gait recognition methodology. The proposed method has two phases: the registration phase and the identification phase. In the registration phase, proposed JRCD and JRTA features are extracted from the skeleton-based gait sequence of Kinect of the training set. The feature vectors of the training set are fed into the proposed deep learning neural network for training. After the training, the fitted model is stored to be applied for the prediction in the identification phase. The skeleton-based gait sequences of Kinect of the testing tests are used in the identification phase. The prediction of the person's identification is performed using the trained deep learning neural network.

B. MULTIPLE GAIT CYCLE EXTRACTION

During walking, the motion of the different joints of the human body has a resemblance to a cyclic pattern. As a result, each frame of a gait cycle exhibits recurrent classifying features. Features extracted from each frame of a gait cycle comprise a unique gait signature for the corresponding subject. This unique gait signature is the input of the machine learning model for training.

At the time of walking, heel strike is the first occurrence, left and right ankles are in their farthest positions. During a standing position, left and right ankles are in the nearest positions. Distance between two ankles become maximum again at the time of heel off position. This cyclic pattern is repeated over the gait sequence. To determine a gait cycle, Euclidean norm is calculated between left and right ankles. A complete gait cycle consists of three successive maxima of

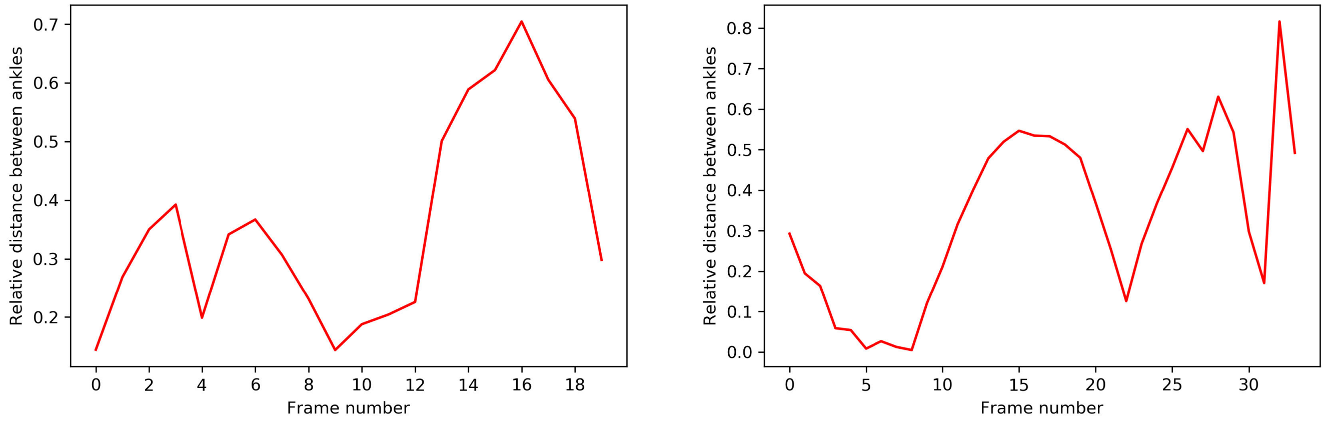


(a) Relative distances between left and right

(b) After smoothing on distance vector.

(c) Local maxima and minima.

FIGURE 4. Steps of gait cycle detection algorithm tracking the Euclidean distances between two ankles over the gait sequence.



(a) A inconsistent gait sequence of person 5 in UPCV gait dataset.

(b) A inconsistent gait sequence of person 22 in UPCV gait dataset.

FIGURE 5. Gait sequences do not exhibit consecutive three local maxima consistently because the number of frames from first maxima to second maxima does not match with the number of frames from second maxima to third maxima.

the Euclidean distances over the gait sequence. The first maxima is the beginning of a gait cycle and the gait cycle is ended by the third maxima. Euclidean distances between two ankles of a gait sequence are shown in Figure 4a. To determine the maxima, moving average filter and subsequent median filter are applied to suppress the noise in the distance vector, as shown in Figure 4b. Local maxima and minima are shown in Figure 4c. We extract more than one gait cycle if a gait sequence has multiple gait cycles. Otherwise, a single gait cycle is extracted. In Figure 4c, P_1, P_2, P_3, P_4, P_5, and P_6 represent the local maxima of the distance vector. Therefore, four gait cycles are detected from this distance vector, such as $P1$ to P_3, P_2 to P_4, P_3 to P_5, and P_4 to P_6. Features are extracted separately from each of the gait cycles. Inclusion of multiple gait cycle and feature extraction from multiple gait cycles act as a data augmentation for model training.

In practical situation, inconsistent gait cycles can be exhibited by an individual. It is worth mentioning that inconsistent gait cycles are observed in the dataset for our study. Figures 5a and 5b are the examples of inconsistent gait sequences found in the UPCV gait dataset. Three local maxima are determined to extract a gait cycle. If three local maxima are not found in a gait sequence, a gait cycle is assumed starting from the first frame and ending with the last frame of the corresponding gait sequence. This situation occurs when a gait sequence does not have a sufficient number of frames to accommodate a complete gait cycle because of errors in data collection.

C. GAIT FEATURE EXTRACTION

Distinctive feature extraction is crucial for gait recognition. In prior researches, various degrees of success were achieved in locating motion of body joints. Very effective features based on the relationship of distance and angle of body joints were introduced by Ahmed *et al.* [12]. We make one step further through extracting more robust and discriminating features to find the distinctive motion of human body joints. These features are Joint Relative Triangle Area (JRTA) and Joint Relative Cosine Dissimilarity (JRCD)—extracted from different body joints. The relative motions of different body joints while walking are identified using these dynamic features. Both of these dynamic features are extracted from each of the frames of a gait cycle. Since JRTA identifies the geometry of the motion of joint and JRCD captures the directional motions of body joints, their combination enhances the recognition accuracy. Consider N_g is the number of the gait cycle in a gait sequence, N_f is the number of frames in a complete gait cycle, N_C is the length of feature vector of the joint relative cosine dissimilarity, and N_T is the length of the feature vector of the joint relative triangle area. We extract $N_f * N_C$ numbers of joint relative cosine dissimilarities and $N_f * N_T$ numbers of joint relative triangle areas. Since the number of the gait cycle in a gait sequence is N_g, we extract N_g rows of $N_f * N_C$ numbers of joint relative cosine dissimilarities. Similarly, we extract N_g rows of $N_f * N_T$ numbers of joint relative triangle areas.

1) JOINT RELATIVE TRIANGLE AREA (JRTA)

The area of triangle in 3D space is determined considering three 3D coordinates. We represent three body joints as $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, and $C(x_3, y_3, z_3)$ Cartesian 3D points. The area of triangle considering three 3D points is calculated using the Equation (4) where vector $\overrightarrow{AB}$ and $\overrightarrow{BC}$ are defined in Equation (1) and (2) respectively. The area of parallelogram is required to determine the area of triangle. The area of parallelogram is the cross-product of $\overrightarrow{AB}$ and $\overrightarrow{BC}$ defined in Equation (3) and half of the area of parallelogram is the area of triangle. In Equation (3), $\|*\|$ represents the norm of the vector and $(\times)$ denotes the cross-product of two vectors. In Equation (4), $\triangle ABC$ represents the area of triangle enclosing A, B, and C points where B is the reference point.

$$\overrightarrow{AB} = B - A \tag{1}$$

$$\overrightarrow{BC} = C - B \tag{2}$$

$$Area_{parallelogram} = \|\overrightarrow{AB} \times \overrightarrow{BC}\| \tag{3}$$

$$\triangle ABC = \frac{\|\overrightarrow{AB} \times \overrightarrow{BC}\|}{2} \tag{4}$$

The physical interpretation of joint relative triangle area feature is that it represents the geometry of the motions of human joints calculated with respect to the most stable body joint (SPINE body joint) in the human skeleton. The joint relative triangle area features are calculated from each of the frames of a gait cycle. The fluctuation of the SPINE body joint (see Figure 6) is the smallest among all of the other body joints, according to [37]. Therefore, the SPINE body joint is chosen as the reference point for determining the joint relative triangle area feature. If there are N_b body joints in the skeleton, length of a joint relative triangle area feature vector (N_T) is $((N_b - 1) * (N_b - 2))$. This means that there are N_T triangles formed with respect to the SPINE body joint in the skeleton model. In Figure 6, P is the reference point (SPINE body joint), $\triangle POQ$ and $\triangle POR$ are the examples of joint relative triangles. Each of the joint relative triangles undergoes unique transformation while the person is walking. Therefore, $\triangle POQ$ and $\triangle QOP$ triangles are treated as two different triangles in our study. As the SPINE body joint is the most stable body joint, it serves as an excellent reference point to compute and compare the geometry of the motion of the joint. In addition, the area of a triangle is shown to be less susceptible to noise and random distortions than other geometric features, such as direct distances between pairs of absolute coordinates [38].

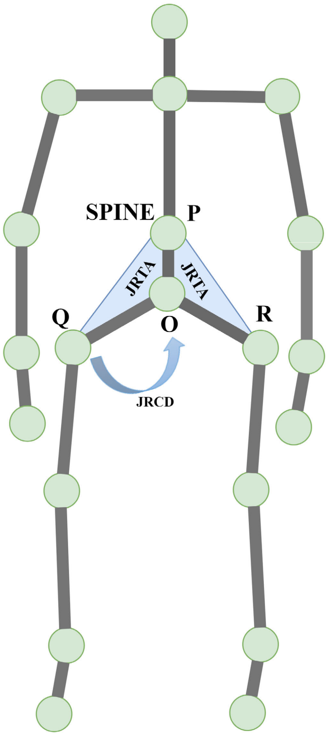


FIGURE 6. **Representation of JRCD and JRTA features. *P*, *O*, *Q*, and *R* are the body joints. JRTA feature is represented by the shaded triangle and the cosine dissimilarity between *Q* and *O* body joints is represented by a curved arrow.**

2) JOINT RELATIVE COSINE DISSIMILARITY (JRCD)

Joint relative cosine dissimilarity is the cosine distance between two points in the vector space. Consider $A(x_1, y_1, z_1)$ and $B(x_2, y_2, z_2)$ are the 3D coordinates of two skeleton joints. Cosine similarity between $\overrightarrow{A}$ and $\overrightarrow{B}$ is defined by the Equation (5) and (6). In Equation (5), $(\cdot)$ represents the dot product of two vectors.

$$\overrightarrow{A} \cdot \overrightarrow{B} = \|\overrightarrow{A}\|\|\overrightarrow{B}\| \cos\theta \tag{5}$$

$$\cos\theta = \frac{\overrightarrow{A} \cdot \overrightarrow{B}}{\|\overrightarrow{A}\|\|\overrightarrow{B}\|} \tag{6}$$

Cosine dissimilarity is defined by the Equation (7).

$$\delta_{cosine}(A, B) = 1 - \frac{\overrightarrow{A} \cdot \overrightarrow{B}}{\|\overrightarrow{A}\|\|\overrightarrow{B}\|} \tag{7}$$

The physical interpretation of the new joint relative cosine dissimilarity feature is that joint relative cosine dissimilarity uniquely identifies the recurrent directional motions of all body joints of a person who is walking. This is highly valuable in distinguishing one person's gait against others, through recurrent yet unique motions of body joints. The cosine dissimilarities of each of the body joints are calculated with the rest of the body joints; these body joints can be connected or not. The recurrent directional motion of each of the body joints relative to the rest of the body joints is identified by the cosine dissimilarity feature. The cosine dissimilarities of each of the body joints are calculated considering all combination of the rest of body joints. For example,

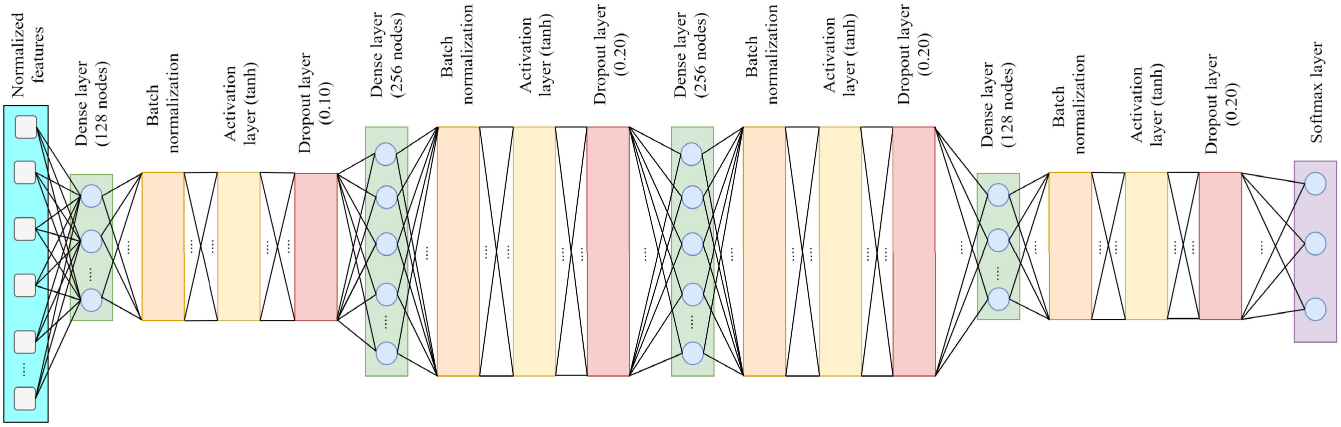


FIGURE 7. Proposed deep learning neural network architecture.

in Figure 6, we extract cosine dissimilarity between O and Q body joints. Similarly, cosine dissimilarities are calculated using $\{O, R\}$, $\{P, O\}$, and $\{P, R\}$ body joints. If there are total N_b body joints in the skeleton model of Kinect, length of a joint relative cosine dissimilarity feature feature vector (N_C) is $(N_b * (N_b - 1))/2$. This means that there are N_C directional motions identified using the joint relative cosine dissimilarity feature vector. Identifying recurrent directional motions of a person during the process of walking allows to achieve high discriminability during gait recognition.

D. PROPOSED DEEP LEARNING NEURAL NETWORK

This paper presents a new deep learning neural network for Kinect-based gait recognition. The proposed deep learning neural network architecture is shown in Figure 7. The proposed architecture takes the input of normalized features of joint relative triangle area and joint relative cosine dissimilarity. In a pre-processing step, the feature vectors are normalized by the standard normalization method to eliminate the effect of outliers while training of the neural network model. The feature vectors are transformed to make the mean equal to 0 and variance equal to 1 of the normalized feature vectors [39]. The identification labels of every person are transformed into one-hot encoded format. Then, normalized feature vectors and one-hot encoded identification label are fed into hidden blocks for training the deep learning neural network.

There are four hidden blocks in the proposed architecture of deep learning neural network. The hidden units of four hidden blocks are stacked in such a way that the output of one layer becomes the input to the next layer. There are 128, 256, 256, and 128 fully connected dense nodes in the first, second, third, and fourth hidden block respectively. The weights of each of the dense layers of all hidden blocks are initialized using *He Normal* initialization method [40]. *He Normal* initializer selects the random value for weights from the truncated normal distribution which is centered on 0 and the standard deviation of the truncated normal distribution is of value $\sqrt{2/N_h}$, where N_h is the number of hidden units in any particular hidden block. The batch normalization layer, activation layer, and dropout layer are successively added after the dense layer. The order of the dense layer, batch normalization layer, activation layer, and dropout layer is similar to all hidden blocks. Mini-batch wise training and mini-batch gradients' calculation are performed in the proposed deep learning neural network. The purpose of the addition of batch normalization layer is for performance improvement and stability to each of the mini-batch of each hidden layer. Batch normalization layer decreases the internal covariate-shift [41] by reducing the oscillation of gradient descent. The output of mini-batch of each layer is normalized and become the input to the next layer using batch normalization method. Furthermore, the accuracy of the model can be improved using batch normalization method [41].

In our study, hyperbolic tangent (tanh) activation function is used to add non-linearity to the deep learning neural network. The hyperbolic tangent activation function is used according to Equation (8) instead of logistic activation function defined in Equation (9). Although the derivatives of both hyperbolic tangent and logistic activation functions are not monotonic, hyperbolic tangent activation function is used in our research for three reasons. First, the length of the feature vector is large. Second, the derivatives of hyperbolic tangent activation are bigger than derivatives of logistic activation. Third, we can minimize objective function faster than the logistic activation function in the neural network.

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

$$logistic(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

Furthermore, we investigate the performance of the rectified linear unit (ReLU) activation function. ReLU is one of the most used activation functions in deep learning architecture. ReLU activation function is formally defined in Equation (10).

$$ReLU(x) = max(0, x) \tag{10}$$

TABLE 1. Type and count of parameters of the proposed deep learning neural network architecture without considering the number of nodes in the decision layer.

Type of parameters	Count of parameters
Trainable parameters	4493056
Non-trainable parameters	1536
Total parameters	4494592

The propoerty of the result and derivative of the ReLU function is monotonic. Since ReLU function clips the negative values to 0, it lowers the capability of training the model properly because of the dead neuron problem. The performance of ReLU activation is compared with tanh activation function to find out which activation function works better in the proposed deep learning neural network architecture for gait recognition. Besides, we introduce dropout regularizer [42] by setting weights to zero to random hidden units to subdue the high-variance problem in the deep learning neural network. Thus, the co-dependency of each of the hidden units is prevented. The dropout rate of 10%, 20%, 20%, and 20% are set in the dropout layer of first, second, third, and fourth hidden block respectively.

The softmax activation function is applied in the decision layer to classify persons' identities in a multi-class gait recognition system. Softmax activation function—defined in Equation (11)—results probabilistic outputs within the range of [0, 1].

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{11}$$

If there are gait sequences of N_y persons in the dataset, Softmax activation function in the decision layer generates a vector (S) of probabilities of length N_y. In Equation (11), y_i is ith value of the vector y for the Softmax activation function and $S(y_i)$ is probability of ith value of the vector y. Thus, the model gives the prediction probabilities of each of the classes for a sample. The highest probability denotes the rank-1 prediction from the features extracted from a gait cycle.

The type and count of parameters of the proposed deep learning neural network architecture is presented in Table 1. Based on the number of class label, the number of nodes in decision layer is set. Since the number of class label is not same in both datasets considered in this study, the counts of trainable and non-trainable parameters are reported in Table 1 without considering the number of nodes in the decision layer.

E. OPTIMIZATION METHOD

The objective function of the proposed deep learning neural network is minimized iteratively by applying the optimization method. The optimized weights of the proposed deep learning neural network are determined using Adam optimization method [14]. The optimization result of Adam optimizer is compared with Root Mean Square Propagation (RMSProp) and Stochastic Gradient Descent (SGD) optimizer. There are several benefits of using Adam stochastic optimizer over other optimizers. First, it is computationally efficient and simple to implement. Second, it is an empirically established optimization method for optimizing a deep neural network. Third, the benefits of both AdaGrad [43] and RMSProp [15] optimizers are exploited in Adam optimization method. Furthermore, performance result mentioned in the section V substantiate the choice of Adam optimizer for the proposed deep learning neural network.

Hyper-parameters of each of the optimization methods are adjusted empirically using the grid-search method. Hyper-parameters of Adam optimizer, such as learning rate (η_a), exponential decay rate of first moment (β_1), exponential decay rate of second moment (β_2), decay of learning rate ($decay_a$) over each of the updates while back-propagation, and numerical constant (ϵ_a), are tuned for the high performance in terms of recognition accuracy. We register the value of 0.001 to η, 0.90 to β_1, 0.999 to β_2, 0.0 to $decay$, and 10^{-8} to ϵ_a. Similarly, hyper-parameters of RMSProp optimizer—learning rate (η_r), exponential weighted average of all the gradient (ρ), numerical constant (ϵ_r), and decay of learning rate ($decay_r$) over each of the updates while back-propagation— are tuned to reduce the loss of cost function at maximum. We set the value of 0.0001 to η_r, 0.9 to ρ, 0.0 to $decay_r$, and 10^{-8} to ϵ_r. Furthermore, the learning rate (η_s), momentum and decay of learning rate ($decay_s$) over each of the updates while back-propagation are initialized to 0.001, 0.90, and 10^{-6} respectively for the hyper-parameters of SGD optimizer. In our study, Nesterov momentum is included in SGD optimizer because the performance of Nesterov momentum is found to be better than standard momentum in the SGD optimizer. Adam, RMSProp, and SGD optimizer minimize the categorical cross-entropy loss function while training the deep learning neural network. Binary cross-entropy loss is extended to categorical cross-entropy loss using the Equation (12) because multi-class gait recognition is studied in this paper. In Equation (12), C denotes the total number of participants available in a particular dataset, y_{true} is one-hot encoded vector of ground-truth and $y_{predict}$ is the vector of probabilities of the prediction for each of the classes by the Softmax activation function, and L is the loss.

$$L(y_{true}, y_{predict}) = -\sum_{c}^{C} y_{true}^{(c)} log(y_{predict}^{(c)}) \tag{12}$$

Since Softmax activation function is used in the decision layer, the categorical cross-entropy loss function is a good match with the probabilistic result of the Softmax activation function.

To train the deep learning neural network batch-wise, we set the batch size to 32. The categorical cross-entropy loss function is applied to minimize the training loss iteratively. We add callback function for reducing the learning rate by a factor if learning does not change within a predetermined number of epochs. Thus, the learning rate is reduced adaptively.

IV. DATASETS

In this section, two benchmark datasets: UPCV gait dataset [17] and Kinect gait biometry dataset [2] are described. Both datasets are publicly available. The descriptions of the both datasets are as follows.

A. UPCV GAIT DATASET

Kastaniotis *et al.* [17] released UPCV gait dataset having the gait sequences of 15 males and 15 females. Participants were directed to walk normally in a straight direction. The Kinect sensor was located at a position which was 1.70 m above the ground level. Videos were recorded at the speed of 30 fps from the left side of the person's walking direction. The recorded video had the gait sequences maintaining a 30-degree angle with the Kinect sensor. There were five walking sequences for each of the participants. Each gait sequence consisted of approximately 55–120 frames. In our paper, UPCV gait dataset is represented as UPCV dataset.

B. KINECT GAIT BIOMETRY DATASET

Andersson and Araujo collected Kinect Gait Biometry Dataset of skeleton-based gait sequences of 164 individuals [2]. The participants were directed to walk in a semi-circular path. The subjects walked from left to right following clockwise direction and then right to left to the starting point. The gait sequence was captured using the X-Box 360 Kinect sensor attached to a spinning dish. The purpose of utilizing the spinning dish was to track the participant while walking. Thus, walking sequences were recorded without distortion keeping the participant in the center of the field of view of Kinect sensor. There were variable walking sequences for the participants. Each walking sequence had a total of approximately 500–600 frames and 6–12 gait cycles. In our paper, Kinect gait biometry dataset is represented as GaitBiometry dataset.

V. EXPERIMENTAL RESULTS

The proposed gait recognition methodology is validated using two Kinect-based skeleton datasets of gait discussed in the section IV. The gait cycle detection algorithm is applied to detect multiple gait cycles from the gait sequences of both datasets. Joint relative triangle area and joint relative cosine dissimilarity features are extracted from the beginning to the ending frame of each of the gait cycles. Proposed features are normalized after merging. The normalized features are employed to train the proposed Deep Learning Neural Network (DLNN) and the fitted model is stored in the registration phase. In the identification phase, the trained model is used for the prediction of the person's identification.

The experiments are conducted based on the K-fold cross-validation with K equal to 5. In the 5-fold cross-validation experiment, all data samples are randomly split into equal size of five sets. The performance of the proposed gait recognition method is evaluated in terms of recognition accuracy, precision, recall, and F-score. The recognition accuracy is calculated for each of the testing sets and the average recognition accuracy is reported. In our study, the precision, recall, and F-score metrics are reported by determining the macro-average for each class. To calculate the macro-average, the precision, recall, and F-score metrics are calculated for each class and the corresponding unweighted average is measured.

We arrange multiple experiments to compare our proposed neural network architecture with other activation functions and optimization methods. First, ReLU and tanh activation functions are applied in the proposed deep learning neural network. Second, the performance of the proposed deep learning neural network architecture is observed replacing the dense layer with the Maxout network [44]. Maxout network is chosen for comparison because the maxout activation is easily applicable for deep learning neural network and robust model can be designed using maxout network with dropout regularizer. Third, the objective function is minimized applying the SGD, RMSProp, and Adam optimization methods.

The performance results of Maxout network, deep learning neural network with ReLU activation and SGD optimizer, deep learning neural network with ReLU activation and RMSProp optimizer, deep learning neural network with ReLU activation and Adam optimizer, deep learning neural network with tanh activation and SGD optimizer, deep learning neural network with tanh activation and RMSProp optimizer and deep learning neural network with tanh activation and Adam optimizer are calculated in our study. The average recognition performance of the proposed deep learning neural network with different activation functions and optimization methods on UPCV dataset is reported in Table 2. The recognition accuracy, precision, recall, and F-score are determined to evaluate the performance of the proposed deep learning neural network. Maxout network shows the accuracy of 81.96%, the precision of 80.25%, the recall of 81.30%, and the F-score of 80.78%. This is the lowest recognition performance among all configurations of the neural network considered in this study. The hyperbolic tangent activation provides better non-linearity than ReLU activation on UPCV dataset. The recognition accuracy, precision, recall, and F-score of proposed deep learning neural network with ReLU activation function and SGD optimizer are 88.60%, 85.47%, 85.70%, and 83.56% respectively whereas the recognition accuracy, precision, recall, and F-score of proposed deep learning neural network with tanh activation function and SGD optimizer are 92.94%, 91.13%, 91.16%, 90.19% respectively. A similar comparison can be performed between the RMSProp and Adam optimizer. Deep learning neural network optimized by Adam optimizer secures the best result. Precisely, the highest recognition accuracy of 95.30%, precision of 94.40%, recall of 94.02%, and F-score of 93.27% are achieved while proposed features are trained with the proposed deep learning neural network with tanh activation and Adam optimization method. It is worth pointing out that precision, recall, and F-score of deep learning neural network with tanh activation and RMSProp optimizer are better than deep learning neural network with ReLU activation

TABLE 2. Average recognition performance of proposed deep learning neural network with different activation functions and optimization methods on UPCV dataset.

Proposed features + neural network	Accuracy	Precision	Recall	F-score
Maxout network	81.96	80.25	81.30	80.78
Proposed DLNN + ReLU + SGD	88.60	85.47	85.70	83.56
Proposed DLNN + ReLU + RMSProp	89.81	87.50	87.30	86.42
Proposed DLNN + tanh + SGD	92.94	91.13	91.16	90.19
Proposed DLNN + tanh + RMSProp	93.73	92.45	92.42.	92.03
Proposed DLNN + ReLU + Adam	94.51	90.70	91.27	90.10
Proposed DLNN + tanh + Adam	**95.30**	**94.40**	**94.02**	**93.27**

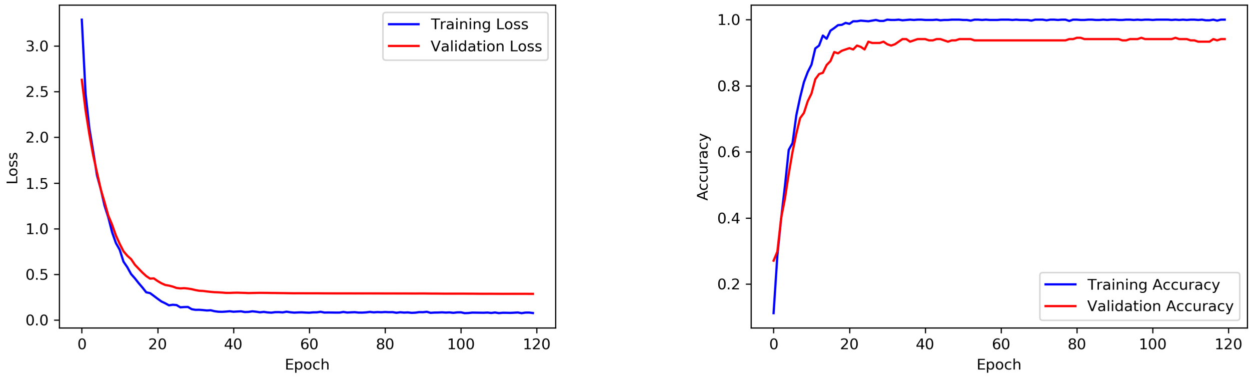
TABLE 3. Average recognition performance of proposed deep learning neural network with different activation functions and optimization methods on GaitBiometry dataset.

Proposed features + neural network	Accuracy	Precision	Recall	F-score
Proposed DLNN + tanh + SGD	95.46	95.46	95.89	95.0
Proposed DLNN + ReLU + SGD	96.10	96.10	96.47	95.81
Maxout network	96.28	96.48	96.37	95.94
Proposed DLNN + ReLU + RMSProp	96.32	96.41	96.45	95.87
Proposed DLNN + tanh + RMSProp	96.44	96.61	96.93	96.23
Proposed DLNN + ReLU + Adam	97.53	97.39	97.48	97.15
Proposed DLNN + tanh + Adam	**98.08**	**98.0**	**98.26**	**97.81**

and Adam optimizer, but deep learning neural network with tanh activation optimized by Adam optimizer triumphs over the methods reported in Table 2. The average recognition performance of the proposed deep learning neural network with different activation functions and optimization methods on GaitBiometry dataset is reported in Table 3. The lowest recognition accuracy of 95.46%, the precision of 95.46%, the recall of 95.89%, and the F-score of 95.0% are achieved by the proposed deep learning neural network with tanh activation and SGD optimization method. On the other hand, the highest recognition performance is secured when the proposed deep learning neural network architecture with tanh activation is trained with the proposed features and optimized by Adam optimizer. The minimization of the objective function optimized by RMSProp is better than SGD optimizer whereas Adam optimizer performs the best among three optimization methods. The recognition accuracy of the proposed deep learning neural network is 95.46–98.08%; the precision is 95.46–98.0%; the recall is 95.89–98.26%; the F-score is 95.0–97.81%.

The analysis of the average Kinect-based gait recognition performance of the proposed deep learning neural network with different activation functions and optimization methods can be summarized as follows. First, the hyperbolic tangent activation function is better than rectified linear unit activation function. Second, the lowest loss of the categorical cross-entropy objective function is attained by the Adam optimizer. Third, the recognition accuracy, precision, recall, and F-score achieved by the proposed deep learning neural network with *tanh* activation function and Adam optimizer is the best on both datasets.

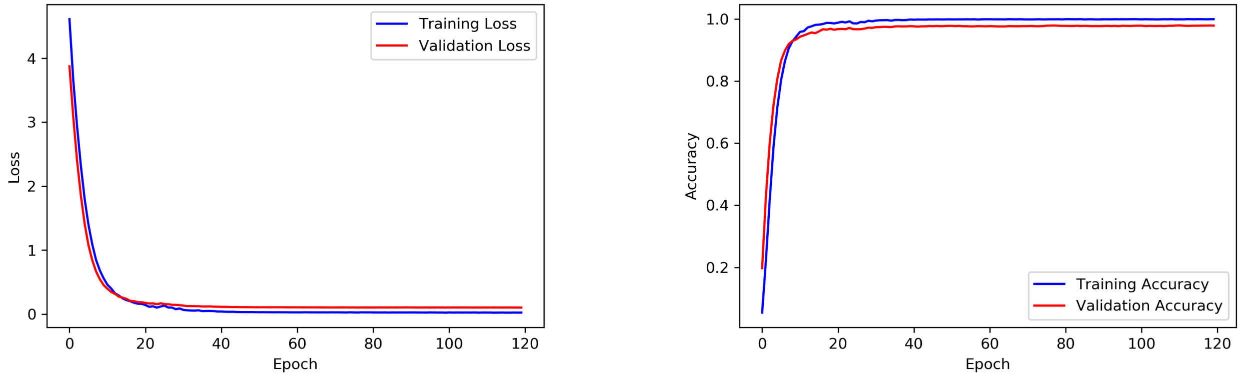
The learning curve demonstrates the performance of the neural network over the iteration while training. Another advantage of visualizing the learning curve is to identify the occurrence of the model overfitting. We generate the training loss for each of the training sets and validation loss for each of the validation sets over the epochs. Figure 8a shows the average training and validation loss on UPCV dataset. It can be observed from Figure 8a that the training loss starts from 3.2869 and the loss is gradually minimized to 0.0761. Although the validation loss decreases following the training loss, there is a difference of 0.21 between the training loss and the validation loss. It is worth mentioning that the more loss is decreased, the more the accuracy increases. The average training and validation accuracy on UPCV dataset is shown in Figure 8b. Although the training accuracy reaches to almost 100%, the highest validation accuracy is 95.30%. Furthermore, we visualize the learning curve of the proposed neural network architecture on GaitBiometry dataset. The average training and validation loss for the Kinect gait biometry dataset is shown in Figure 9a. The training loss starts from 4.6127 and the minimum training loss of 0.0206 is achieved. The validation loss almost reaches to the training loss in Figure 9a. the validation loss begins with 3.8743 and the minimum validation loss is 0.1001. The difference between training and validation loss is 0.0795. Since the validation loss is minimized to 0.1001, the validation accuracy reaches to 98.08% shown in Figure 9b. The summary of the analysis of the visualization of the learning curve of both dataset is as follows. First, although the validation loss does not decrease as training loss, the validation loss gradually decreases over iteration and does not increase anywhere. Second, it supports that our proposed neural network architecture is not biased to specific training and testing set. As a result, the model does not show overfitting to a specific set. Third, we expect to decrease the validation loss more on UPCV dataset if the samples are increased similar to GaitBiometry dataset.



(a) The average training and validation loss of the proposed deep learning neural network on UPCV dataset.

(b) The average training and validation accuracy of the proposed deep learning neural network on UPCV dataset.

FIGURE 8. **The average learning curve of the proposed deep learning neural network on UPCV dataset.**



(a) The average training and validation loss of the proposed deep learning neural network on GaitBiometry dataset.

(b) The average training and validation accuracy of the proposed deep learning neural network on GaitBiometry dataset.

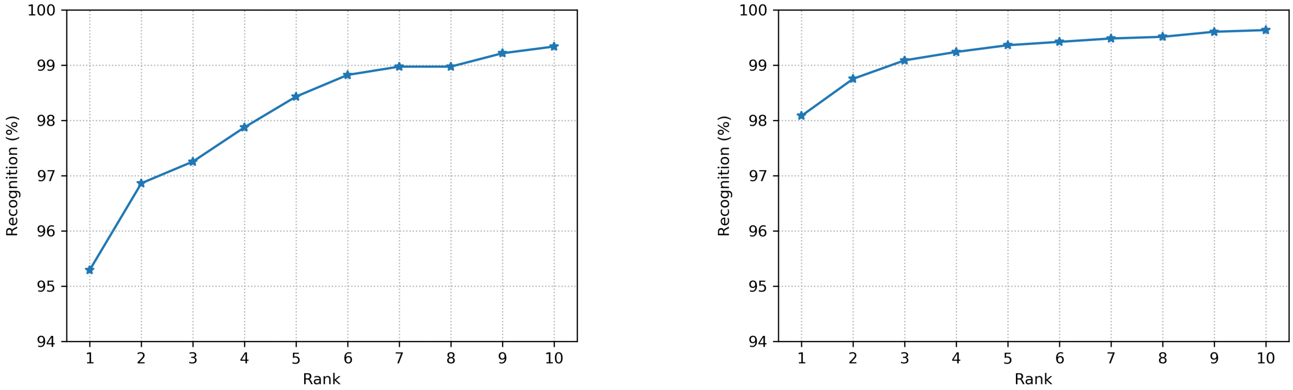
FIGURE 9. **The average learning curve of the proposed deep learning neural network on Kinect GaitBiometry dataset.**

The performance of the proposed Kinect-based gait recognition method is further evaluated using Cumulative Match Characteristic (CMC) curve. The purpose of the CMC curve is to plot the rank versus cumulative recognition percentage. CMC curves of the proposed method on UPCV and GaitBiometry datasets are shown in Figure 10a and Figure 10b respectively. On UPCV dataset, the rank-1 recognition accuracy is 95.30%. The recognition accuracy reaches to 99% with the Rank-7. Rank-10 recognition accuracy is 99.34%. On the other hand, rank-1 recognition accuracy on GaitBiometry dataset starts just above 98% and reaches over 99% within rank-3. Rank-10 recognition accuracy is 99.64%. Since the rank-10 recognition accuracy on UPCV dataset is 99.34% and 99.64% on GaitBiometry dataset, our proposed method is applicable for gait recognition in practical application in the future [13].

The running time of predicting a feature vector is calculated. The system configurations for determining the running time of the model prediction are Intel Core i7-8700 CPU of 3.20 GHz, 16 GB of Ram, and GPU of NVIDIA GeForce GTX 1080. The running time of predicting a feature vector of test set using the trained deep learning neural network is about 3.85×10^{-4} second.

VI. COMPARISONS WITH TRADITIONAL CLASSIFIERS

We also compare the results of neural network methods with the traditional machine learning (ML) methods, such as Naïve Bayes (NB) [45], Decision Tree (DTree) [46], K-Nearest Neighbors (KNN) [47], and Support Vector Machine (SVM) [48]. We re-implement NB, DTree, KNN, and SVM methods. The optimal values of the hyper-parameters of NB, DTree, KNN, and SVM are calculated using the exhaustive grid-search method so that we can obtain the highest recognition accuracy of NB, DTree, KNN, and SVM. Gaussian Naïve Bayes is used to fit the training samples. 5-fold cross-validation experiment is applied to search the optimal value of the hyper-parameter of traditional ML method. The hyper-parameter of maximum depth of the DTree classifier is searched within the range from 1 to 50. The optimal value of maximum depth is 31. To find the optimal value of the number of neighbors of KNN classifier, we set the search range of the number of neighbors from 1 to 70. The value of 1 of the parameter K shows the best performance for KNN classifier. In case of SVM classifier, the optimal values of three hyper-parameters—the penalty parameter (C), the kernel coefficient (γ), and the type of kernel—are searched exhaustively using the grid-search method. The list



(a) CMC curve of the proposed method on UPCV dataset.

(b) CMC curve of the proposed method on GaitBiometry dataset.

FIGURE 10. CMC curve of the proposed method on UPCV and GaitBiometry datasets.

TABLE 4. Average recognition performance of proposed features with the traditional ML methods on UPCV dataset.

Proposed features + traditional ML methods	Accuracy	Precision	Recall	F-score
Naïve Bayes	35.68	36.98	35.60	32.29
Decision Tree	52.57	47.58	48.11	44.51
KNN	80.39	77.19	77.83	75.22
SVM	90.98	90.19	90.41	89.40
Proposed features + Proposed DLNN + tanh + Adam	**95.30**	**94.40**	**94.02**	**93.27**

TABLE 5. Average recognition performance of proposed features with the traditional ML methods on GaitBiometry dataset.

Proposed features + traditional ML methods	Accuracy	Precision	Recall	F-Score
Decision Tree	40.04	43.65	41.78	39.03
Naïve Bayes	77.69	80.54	78.81	77.35
KNN	81.14	80.74	81.03	81.54
SVM	92.31	92.75	92.91	92.00
Proposed features + Proposed DLNN + tanh + Adam	**98.08**	**98.0**	**98.26**	**97.81**

of possible value for C is [1, 5, 10, 100, 1000]; for γ is [0.0001, 0.001, 0.01, 0.1, 1.0, 10.0]; for the type of kernel is [“linear”, “rbf”]. The best recognition accuracy is achieved using SVM classifier when C is equal to 1, γ is equal to 0.0001, and the type of kernel is equal to “linear”.

The result of recognition performance on UPCV dataset is shown in Table 4. Naïve Bayes performs the worst with the accuracy of 35.68%, precision of 36.98%, recall of 35.60%, and F-score of 32.29%. DTree achieves the accuracy of 52.57%, precision of 47.58%, recall of 48.11%, and F-score of 44.51%. KNN achieves the accuracy of 80.39% whereas the precision, recall, and F-score are below 80%. SVM shows the best accuracy, precision, recall, and F-score among the four traditional ML methods considered in this study by achieving accuracy of 90.98%, precision of 90.19%, recall of 90.41%, and F-score of 89.40%.

The performance of the traditional ML methods trained with the proposed features on GaitBiometry dataset is shown in Table 5. DTree classifier performs the worst with the accuracy of 40.04%, precision of 43.65%, recall of 41.78%, and F-score of 39.03%. NB classifier achieves the accuracy of 77.69%, precision of 80.54%, recall of 78.81%, and F-score of 77.35%. KNN achieves the accuracy of 81.14% which is better than DTree and NB classifiers. SVM shows the best accuracy, precision, recall, and F-score among the four traditional ML methods considered in this study by achieving accuracy of 92.31%, precision of 92.75%, recall of 92.91%, and F-score of 92%.

The takeaway messages of the experimental results with the traditional ML methods are as follows. First, the performance of the DTree classifier is the worst. Second, although NB shows recognition result close to 80% on GaitBiometry dataset, the recognition accuracy is the worst on UPCV dataset. As a result, the performance of the NB is not consistent. Third, KNN and SVM classifiers consistently show the recognition accuracy above 80% and 90% respectively. Fourth, the proposed features and proposed deep learning neural network with the hyperbolic tangent activation and Adam optimizer triumphs over the traditional ML methods.

VII. COMPARISONS WITH RELATED WORKS

The recognition performance of the proposed gait recognition method is compared with the state-of-the-art methods to prove the effectiveness of our approach. Ball *et al.* [10]

TABLE 6. Performance comparison of gait recognition methods on UPCV dataset.

Gait recognition methods	Accuracy	Precision	Recall	F-Score
Ball et al. [10]	57.0	53.19	54.87	51.32
Preis et al. [11]	78.0	74.27	73.41	70.43
Sun et al. [9]	82.67	80.50	80.19	79.67
Yang et al. [13]	86.67	85.48	83.76	83.08
JRA + JRD [12] + Proposed DLNN + tanh + SGD	87.06	82.10	81.93	80.09
JRA + JRD [12] + Proposed DLNN + tanh + RMSProp	91.37	88.53	87.40	86.47
JRA + JRD [12] + Proposed DLNN + tanh + Adam	93.33	91.15	90.70	89.73
Proposed features + Proposed DLNN + tanh + Adam	**95.30**	**94.40**	**94.02**	**93.27**

TABLE 7. Performance comparison of gait recognition methods on GaitBiometry dataset.

Gait recognition methods	Accuracy	Precision	Recall	F-Score
Ball et al. [10]	37.55	37.84	38.11	34.25
Preis et al. [11]	75.46	77.70	75.34	73.71
Sun et al. [9]	79.76	80.12	79.32	75.74
JRA + JRD [12] + Proposed DLNN + tanh + SGD	91.28	91.07	92.15	90.19
JRA + JRD [12] + Proposed DLNN + tanh + RMSProp	91.93	91.09	91.71	90.06
Yang et al. [13]	94.88	94.67	95.02	93.92
JRA + JRD [12] + Proposed DLNN + tanh + Adam	95.62	95.92	95.94	95.14
Proposed features + Proposed DLNN + tanh + Adam	**98.08**	**98.0**	**98.26**	**97.81**

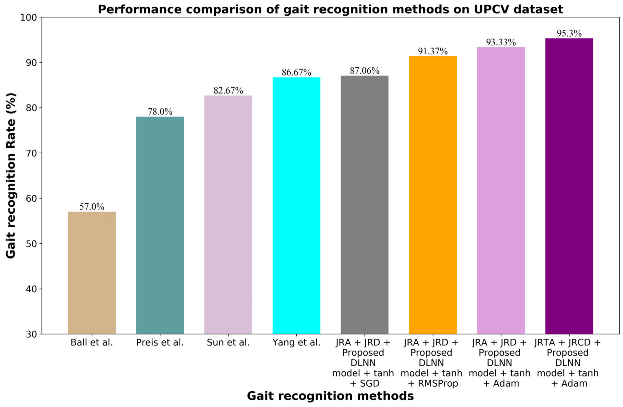


FIGURE 11. Comparion of rank-1 recognition accuracy (%) of gait recognition methods on UPCV dataset.

published Kinect-based gait recognition method analyzing gait sequences of four persons. The research work of Ahmed *et al.* [12] was on 20 persons whereas Sun *et al.* [9] researched on 52 participants. Since most of the state-of-the-art methods obtained the recognition results based on the private datasets, prior researches are re-implemented to compare the recognition result on publicly available Kinect-based benchmark gait datasets. All the comparative results are achieved using a 5-fold cross-validation experiment and the average result is reported.

Table 6 shows the comparison of rank-1 recognition performance of gait recognition methods on UPCV dataset. The proposed features and the proposed deep learning neural network with hyperbolic tangent activation and Adam optimization method achieve 95.30% recognition accuracy. The achieved recognition accuracy is 38.30% higher than Ball et al., 17.30% higher than Preis et al., and 12.63% higher than Sun et al. Our proposed method obtains 8.63% higher recognition accuracy than Yang et al. as well. The comparison of rank-1 recognition accuracy of gait recognition methods on UPCV dataset is shown in Figure 11. Furthermore, our proposed method not only achieves higher recognition accuracy but also secures higher precision, recall, and F-score on UPCV dataset.

The comparison of rank-1 recognition performance of gait recognition methods on GaitBiometry dataset is shown in Table 7. The recognition accuracy, precision, recall, and F-score of Ball *et al.* [10] is the lowest among the gait recognition method considered in this study. The method proposed by Sun *et al.* [9] achieves a better result than Ball *et al.* [10] and Preis *et al.* [11]. But when the proposed features of Ahmed *et al.* [12] is used to train our proposed deep learning neural network with tanh activation and SGD optimization method, it obtains 11.52% better accuracy than Sun et al. JRTA and JRCD features and the proposed deep learning neural network with hyperbolic tangent activation and Adam optimization method achieves the highest recognition accuracy of 98.08%. The achieved recognition accuracy is 60.53% higher than Ball et al., 22.62% higher than Preis et al., 18.32% higher than Sun et al., and 3.2% higher than Yang et al. The comparison of rank-1 recognition accuracy of gait recognition methods on GaitBiometry dataset is shown in Figure 12. The proposed gait recognition method secures the highest precision of 98.0%, recall of 98.26%, and F-score of 97.81% among the state-of-the-art methods considered in this study.

To demonstrate that the new features are more effective, we extract JRA and JRD features according to [12] and train proposed deep learning neural network with a hyperbolic tangent activation function. This neural network is optimized

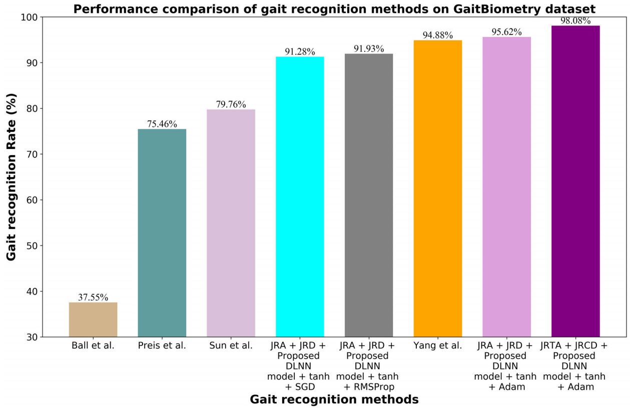


FIGURE 12. Comparion of rank-1 recognition accuracy (%) of gait recognition methods on GaitBiometry dataset.

by SGD, RMSProp, and Adam optimization methods. In Table 6 and 7, average recognition performance of JRA and JRD features is compared with the performance of proposed features on UPCV and GaitBiometry dataset respectively. It is evident that joint relative cosine dissimilarity and joint relative triangle area features provide better recognition accuracy, precision, recall, and F-score than JRA and JRD feature set. On UPCV dataset, the range of recognition accuracy by JRA and JRD features is around 87.06–93.33% whereas the range of recognition accuracy by the proposed features is around 95.30%. On GaitBiometry dataset, the highest recognition accuracy achieved by JRA and JRD features with our proposed deep learning neural network is 95.62% whereas our proposed features with our proposed deep learning neural network achieves 2.46% higher recognition accuracy of 98.08%. Higher precision, recall, and F-score are also achieved using our proposed features than JRA and JRD features on both datasets.

VIII. CONCLUSION AND FUTURE WORK

This research presents deep learning neural network architecture for Kinect-based human gait recognition. To increase the recognition performance, two unique geometric features: joint relative triangle area and joint relative cosine dissimilarity, are introduced. Data augmentation is performed by extracting multiple gait cycles from a gait sequence. Proposed features are extracted from each gait cycle. The performance of the proposed deep learning neural network is enhanced by the Adam optimization method. The proposed model is compared to the hyperbolic tangent activation function and Adam optimization method with the deep learning neural network with rectified linear unit activation function and SGD and RMSProp optimization methods. Moreover, traditional machine learning methods and other recently proposed gait recognition methods are compared with our proposed method. Two publicly available benchmark datasets are used to evaluate the proposed gait recognition method. The highest accuracy obtained by the proposed deep learning neural network, trained with the proposed geometric features, is around 95.30% on UPCV dataset and 98.08% on GaitBiometry dataset after 5-fold cross-validation experiment. This performance is the highest among traditional state-of-the-art classifiers considered in our paper for both datasets. The proposed features establish superior result than the prior researches also for both datasets. Furthermore, the proposed gait recognition method achieves higher accuracy, precision, recall, and F-score than previous state-of-the-art methods even in challenging scenarios having inconsistent gait sequences. Therefore, we believe that the proposed gait recognition architecture can be adapted to real-world environments. In the future, an evolutionary algorithm for optimizing the feature set can be studied in the context of the proposed neural network architecture. Moreover, recurrent neural network can be designed to avoid the necessity of resampling of feature vector to a vector of fixed size.

REFERENCES

[1] Y. Feng, Y. Li, and J. Luo, "Learning effective gait features using LSTM," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 325–330.

[2] V. O. Andersson and R. M. de Araújo, "Person identification using anthropometric and gait data from kinect sensor," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 425–431.

[3] C. BenAbdelkader, R. Cutler, and L. Davis, "Stride and cadence as a biometric in automatic person identification and verification," in *Proc. Fifth IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 372–377.

[4] D. Kastaniotis, I. Theodorakopoulos, G. Economou, and S. Fotopoulos, "Gait-based gender recognition using pose information for real time applications," in *Proc. 18th Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2013, pp. 1–6.

[5] M. Popa, A. K. Koc, L. J. M. Rothkrantz, C. Shan, and P. Wiggers, "Kinect sensing of shopping related actions," in *Proc. Int. Joint Conf. Ambient Intell.* Berlin, Germany: Springer, 2011, pp. 91–100.

[6] A. A. Chaaraoui, J. R. Padilla-López, and F. Flórez-Revuelta, "Abnormal gait detection with RGB-D devices using joint motion history features," in *Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, vol. 7, May 2015, pp. 1–6.

[7] Y.-J. Chang, S.-F. Chen, and J.-D. Huang, "A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities," *Res. Develop. Disab.*, vol. 32, no. 6, pp. 2566–2570, 2011.

[8] D. Gafurov, E. Snekkenes, and P. Bours, "Spoof attacks on gait authentication system," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 491–502, Sep. 2007.

[9] J. Sun, Y. Wang, J. Li, W. Wan, D. Cheng, and H. Zhang, "View-invariant gait recognition based on kinect skeleton feature," *Multimedia Tools Appl.*, vol. 77, no. 19, pp. 24909–24935, 2018.

[10] A. Ball, D. Rye, F. Ramos, and M. Velonaki, "Unsupervised clustering of people from 'skeleton' data," in *Proc. 7th ACM/IEEE Int. Conf. Hum.-Robot Interact.*, Mar. 2012, pp. 225–226.

[11] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien, "Gait recognition with kinect," in *Proc. 1st Int. Workshop Kinect Pervas. Comput.*, New Castle, U.K., 2012, pp. P1–P4.

[12] 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.

[13] 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.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[15] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Berlin, Germany: Springer, 2010, pp. 177–186.

[17] D. Kastaniotis, I. Theodorakopoulos, C. Theoharatos, G. Economou, and S. Fotopoulos, "A framework for gait-based recognition using Kinect," *Pattern Recognit. Lett.*, vol. 68, pp. 327–335, Dec. 2015.

[18] A. H. Bari and M. L. Gavrilova, "Multi-layer perceptron architecture for kinect-based gait recognition," in *Proc. Comput. Graph. Int. Conf.* Cham, Switzerland: Springer, 2019, pp. 356–363.

[19] Y. Makihara, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi, "Gait recognition using a view transformation model in the frequency domain," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 151–163.

[20] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.

[21] X. Li and Y. Chen, "Gait recognition based on structural gait energy image," *J. Comput. Inf. Syst.*, vol. 9, no. 1, pp. 121–126, 2013.

[22] E. Zhang, Y. Zhao, and W. Xiong, "Active energy image plus 2DLPP for gait recognition," *Signal Process.*, vol. 90, no. 7, pp. 2295–2302, 2010.

[23] T. H. W. Lam, K. H. Cheung, and J. N. K. Liu, "Gait flow image: A silhouette-based gait representation for human identification," *Pattern Recognit.*, vol. 44, no. 4, pp. 973–987, Apr. 2011.

[24] J. Wang, M. She, S. Nahavandi, and A. Kouzani, "A review of vision-based gait recognition methods for human identification," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl.*, Dec. 2010, pp. 320–327.

[25] C. Yam, M. Nixon, and J. N. Carter, "Automated person recognition by walking and running via model-based approaches," *Pattern Recognit.*, vol. 37, no. 5, pp. 1057–1072, 2004.

[26] C. Zhang, Y. Tian, and E. Capezuti, "Privacy preserving automatic fall detection for elderly using RGBD cameras," in *Proc. Int. Conf. Comput. Handicapped Persons*. Berlin, Germany: Springer, 2012, pp. 625–633.

[27] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 290–301, Jan. 2015.

[28] R. Wang, H. Zhang, and C. Leung, "Follow me: A personal robotic companion system for the elderly," *Int. J. Inf. Technol.*, vol. 21, no. 1, pp. 1–20, 2015.

[29] Y. Maret, D. Oberson, and M. Gavrilova, "Real-time embedded system for gesture recognition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 30–34.

[30] A. Al-Kaff, F. M. Moreno, A. de la Escalera, and J. M. Armingol, "Intelligent vehicle for search, rescue and transportation purposes," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 110–115.

[31] H. I. Osman, F. H. Hashim, W. M. D. W. Zaki, and A. B. Huddin, "Entryway detection algorithm using Kinect's depth camera for UAV application," in *Proc. IEEE 8th Control Syst. Graduate Res. Colloq. (ICSGRC)*, Aug. 2017, pp. 77–80.

[32] H. Wang, C. Zhang, Y. Song, and B. Pang, "Information-fusion based robot simultaneous localization and mapping adapted to search and rescue cluttered environment," in *Proc. 18th Int. Conf. Adv. Robot. (ICAR)*, Jul. 2017, pp. 511–517.

[33] R. Banerjee, A. Sinha, and K. Chakravarty, "Gait based people identification system using multiple switching kinects," in *Proc. 13th Int. Conf. Intell. Syst. Design Appl.*, Dec. 2013, pp. 182–187.

[34] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1297–1304.

[35] K. Khoshelham, "Accuracy analysis of kinect depth data," in *Proc. ISPRS Workshop Laser Scanning*, 2011, vol. 38, no. 1, pp. 133–138.

[36] A. Schmitz, M. Ye, R. Shapiro, R. Yang, and B. Noehren, "Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system," *J. Biomech.*, vol. 47, no. 2, pp. 587–591, 2014.

[37] F. Ahmed, A. S. M. H. Bari, B. Sieu, J. Sadeghi, J. Scholten, and M. L. Gavrilova, "Kalman filter-based noise reduction framework for posture estimation using depth sensor," in *Proc. 18th Int. Conf. Cogn. Inform. Cogn. Comput.*, Jul. 2019, pp. 150–158.

[38] C. Wang, M. Gavrilova, Y. Luo, and J. Rokne, "An efficient algorithm for fingerprint matching," in *Proc. 18th Int. Conf. Pattern Recognit.*, vol. 1, Aug. 2006, pp. 1034–1037.

[39] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 9–48.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[43] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[44] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," 2013, *arXiv:1302.4389*. [Online]. Available: https://arxiv.org/abs/1302.4389

[45] H. Zhang, "The optimality of naive Bayes," in *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2004, pp. 1–6.

[46] L. Rokach and O. Z. Maimon, *Data Mining With Decision Trees: Theory and Applications*, vol. 69. Singapore: World Scientific, 2008.

[47] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992.

[48] V. N. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.

A. S. M. HOSSAIN BARI received the B.Sc. degree in computer science and information technology from the Islamic University of Technology, in 2010. He is currently pursuing the M.Sc. degree in computer science with the University of Calgary, Canada, under the supervision of Prof. M. L. Gavrilova. He was with the Samsung R&D Institute Bangladesh (SRBD), from November 2010 to August 2018. He has authored over ten international journal and conference articles. He has secured U.S. patents while working in SRBD. His research interests include behavioral biometrics, computer vision, and machine learning.

MARINA L. GAVRILOVA is currently a Full Professor and an Associate Head of the CSPS Department, University of Calgary. She is an International Expert in the area of biometric security, machine learning, pattern recognition, data analytics, and information fusion. She is a Co-Founder of the Biometric Technologies Laboratory and the SPARCS Laboratory for interdisciplinary computational sciences research. She has published three coauthored books, over 30 books of conference proceedings, and over 200 peer-reviewed articles in machine learning, biometric security, and multimodal cognitive system architectures. Her professional excellence and international stature was recognized by Senior ACM and Senior IEEE membership statuses, as well as the prestigious Canada Foundation for Innovation, the Killam Foundation, and the University of Calgary U Make a Difference Awards. She is the Founding Editor-in-Chief of the *Transactions on Computational Sciences* (Springer). She serves on the Editorial Board of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SCIENCES, IEEE ACCESS, *The Visual Computer*, the *International Journal of Biometrics*, and the *International Journal of Cognitive Biometrics* and on the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE Steering Committee.