Thai Journal of Mathematics
Volume 17 (2019) Number 1 : 89–97

http://thaijmath.in.cmu.ac.th
ISSN 1686-0209

A Deep Neural Network Approach for Model-based Gait Recognition

Cholwich Nattee[1] **and Nirattaya Khamsemanan**

Sirindhorn International Institute of Technology, Thammasat University
Pathum Thani 12120, Thailand
e-mail : cholwich@siit.tu.ac.th (C. Nattee)
nirattaya@siit.tu.ac.th (N. Khamsemanan)

Abstract : Since the declaration of war on terrorism, human identification has gained more popularity throughout various research communities. One of the sought-after techniques that is used in the identification process is Gait recognition. Gait recognition is a biometric recognition technique that uses body measurements and movements during walks. This technique is non-invasive and can be done without an awareness of a subject. In this work, we develop identification models for gait information extracted from sequences of walking by Microsoft Kinect, from Andersson and Araujo, using a Deep Neural Networks (DNN) combined with Majority Vote. The results show our proposed model yields accuracy of 95.0%, which outperforms Support Vector Machine, k-Nearest Neighbor, Multi-Layer Perceptron and solely Deep Neural Networks techniques.

Keywords : deep neural network; human identification; gait recognition.
2010 Mathematics Subject Classification : 62J12; 68T05; 68T10.

1 Introduction

Because of the recent increase in terrorism activities, human identification techniques have gained more polarities from researchers in fields such as Mathematics, Computer Sciences etc. Gait recognition is a biometric human recognition technique. This technique uses the body's dimensions and movements to identify

[1]Corresponding author.

Copyright © 2019 by the Mathematical Association of Thailand. All rights reserved.

a subject. It can be done from a far and without a subject's alertness. Moreover, it is almost impossible to change gait data permanently and continuously.

Many works have been done on human identification techniques using gait data. The identification is performed by extracting gait features from a sequence of gait data. Then, a supervised machine learning technique is applied to create a classification model from a labeled examples. This classification model is a function that maps from an input gait data, usually represented in the form of a vector, to one of the collected human subjects. The techniques can be categorized into two groups according to the characteristics of gait data, i.e. model-free, and model-based techniques.

Model-free gait recognition techniques identify human subjects using sequences of images or videos captured while the subjects were walking. Most of the existing model-free techniques use silhouettes of subjects' bodies [1, 2] as raw data to extract gait features. A number of techniques have been proposed to subtract the background and extract a human body from captured images [3–6]. Gait Energy Image (GEI) [7] segments a human body from each silhouette, then combines a sequence of segmented silhouettes into an average image which is used as gait features in the recognition step.

Model-based techniques use human skeletal data extracted from captured gait images or videos to identify human subjects. Microsoft Kinect is one of the most widely used sensor to extract the data for gait recognition. It is a motion sensing device designed to capture human's body movement for video game playing. Microsoft Kinect SDK provides a number of functions to extract human skeletal data. A number of techniques have been proposed for model-based gait recognition [8–12]. Andersson and Araujo [13] propose a technique using temporal-spatial and kinematic parameters as gait features. Given a walking sequence, this technique extracts gait features, combines them into a single vector, and uses k nearest neighbor (kNN) technique to recognition human subjects.

In this work, we propose a gait recognition technique using a *Deep Neural Network* (DNN) along with *Majority Vote* of each walking sequence on the public skeleton dataset extracted by Microsoft Kinect and Microsoft Kinect SDK, and provided by Andersson and Araujo [13]. Our proposed technique achieves accuracy of 95% and outperforms other existing techniques using Support Vector Machine, k-Nearest Neighbor, and Multi-Layer Perceptron. It also yields higher accuracy than just using a Deep Neural Network alone.

2 Method

2.1 Gait Data Collection

Microsoft Kinect is a collection of cameras and sensors designed for capturing body movement of human players. It therefore allows the players to interact with the games by moving their body parts. Microsoft Kinect SDK allows researchers to extract various types of data streams from a Kinect. A number of applications

can then be developed from the extracted data. In this work, we utilize the skeletal data stream from Microsoft Kinect to develop a human identification system. From a walk, we can extract a sequence of 20 three-dimensional coordinates. Each coordinate represents a location of a user's body joint in the three-dimensional space where the location of the Kinect sensor is the origin $(0, 0, 0)$.

Definition 2.1. A frame $\mathbf{v}_j$ is a collection of skeletal data extracted from an image captured by Microsoft Kinect. Each frame is represented by a vector of 20 three-dimensional coordinates, i.e.

$$\mathbf{v}_j = [(x_{j,1}, y_{j,1}, z_{j,1}), (x_{j,2}, y_{j,2}, z_{j,2}), \ldots, (x_{j,20}, y_{j,20}, z_{j,20})]$$

where each coordinate represents the detected position of a body joint. The list of the 20 joints are `HEAD`, `SHOULDER_CENTER`, `SHOULDER_RIGHT`, `SHOULDER_LEFT`, `ELBOW_RIGHT`, `ELBOW_LEFT`, `WRIST_RIGHT`, `WRIST_LEFT`, `HAND_RIGHT`, `HAND_LEFT`, `SPINE`, `HIP_CENTER`, `HIP_RIGHT`, `HIP_LEFT`, `KNEE_RIGHT`, `KNEE_LEFT`, `ANKLE_RIGHT`, `ANKLE_LEFT`, `FOOT_RIGHT`, and `FOOT_LEFT`.

Definition 2.2. A walk $\mathbf{w}_i$ is a sequence of n_i frames, i.e.

$$\mathbf{w}_i = \langle \mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \ldots, \mathbf{v}_{i,n_i} \rangle.$$

The number of frames depends on various factors, e.g. body movement speed, walking speed, etc.

We use a public skeleton dataset provided by Andersson and Araujo [13] in this work. The dataset is a collection of skeletal data collected from 160 subjects. Each subject walked 5 times. Each walk is composed of 400–600 frames. Each frame is a collection of 20 joints. The data was captured by having each subject walks in front of a Kinect sensor from left to right along a semicircle. The Kinect sensor is placed on a spinning dish, and it is rotated to follow the position of the subject.

2.2 Gait Feature Extraction

Given a set of labeled walks,

$$\mathcal{D} = \{(\mathbf{w}_1, u_1), (\mathbf{w}_2, u_2), \ldots, (\mathbf{w}_n, u_n)\}$$

where each $\mathbf{w}_i$ is a walk defined in Definition 2.2, and $u_i \in \mathcal{U}$ denotes one of the human subjects that the walk belongs to.

As stated in Definition 2.2 that a walk consists of varying number of frames, the existing techniques propose methods to combine information from multiple frames into one fixed-length vector so that various existing machine learning algorithms can be applied to construct a classification model. Andersson and Araujo [13], for instance, propose to use two types of gait features: spatiotemporal and anthropometric features. Spatiotemporal features show how a human subject moves his/her

body while walking, e.g. *average stride length*. Anthropometric features capture the lengths of segments of human body. Each segment is represented by the mean and the standard deviation of the Euclidean distance between two adjacent joints. Gait features in this techniques are represented as 60-element vectors.

Different from the existing techniques, we propose a technique to use each frame of a walk as gait features without combining them into one single vector. This results in a classification model for identifying a human subject from an individual frame. In order to identify a subject from an entire walk, we need to combine frame-level predictive results into one prediction of the entire walk. This combination is done by *majority vote*, i.e.

$$u^* = \arg\max_{u \in \mathcal{U}} \sum_{k=1}^{m} \mathrm{I}(\hat{f}(v_k), u)$$

where $I(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$,

$\hat{f}$ is the frame-level classification model, and v_k is a frame in a walk to be classified.

2.3 Deep Neural Networks for Human Identification

A Deep Neural Networks (DNN) [14] is a supervised machine learning technique based on the idea of Artificial Neural Networks. This technique has recently become popular among AI researchers due to its capabilities on various domains of applications. Unlike shallow neural networks, DNNs typically compose of multiple hidden layers between the input and output layers. These additional hidden layers allow the DNNs to learn how to extract features, combine low-level features in high-level features, as well as perform a classification in one setting. It therefore does not need any additional efforts for handcrafting features. A classification model denoted by a DNN can be written as

$$f(\mathbf{x}) = f^{(p)}(\cdots(f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))))$$

where f is a composite classification function constructed from combination of multiple functions, i.e. $f^{(1)}$, $f^{(2)}$, $\ldots$, $f^{(p)}$. Each of these functions is a layer which implicitly consists of a number of weights and parameters. To accurately conduct predictions, these weights and parameters must be updated according to the provided training examples. A number of optimization algorithms have been proposed for training the DNNs, e.g., stochastic gradient descent, ADAM [15].

To identify a human subject from a walk, we need a classification model that can capture the body parts as well as the posture of each human subject. Since each frame of the sequence represents a posture, we construct a DNN that accepts a frame as an input, and predict the human subject as an output. As stated previously, the majority vote technique is applied to combine the outputs of the DNN from the same walk into a final prediction. Figure 1 shows the overall processes to identify a human subject from a walk. From the figure, each frame in

a walk is fed as an input to the DNN in order to make a prediction. The predicted classes from all frames are then summarized by the majority vote technique to yield a final prediction result.

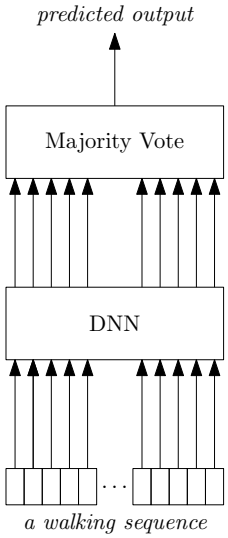

Figure 1: The overall processes for human identification

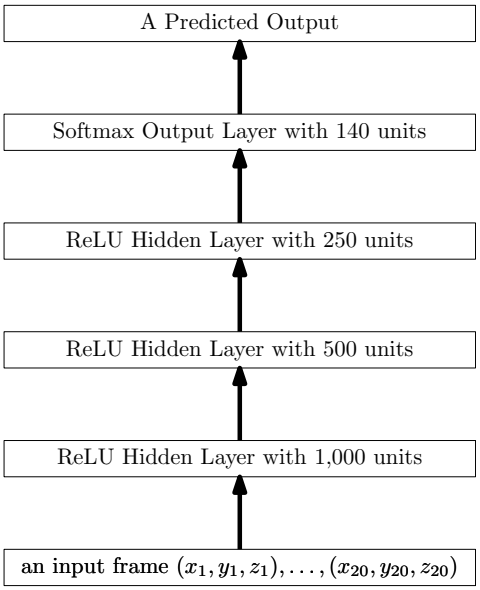

Figure 2: The architecture of DNN for frame classification

Figure 2 shows the architecture of the DNN used in this work. The input of the DNN is a frame directly obtained from the Kinect sensor. Each frame is a collection of 20 three-dimensional coordinates. We consider each value of the coordinate separatedly. Thus, the input of the DNN in this work is a 60-dimensional vector without any additional feature extraction step.

The input layer is fully connected to a hidden layer. This hidden layer composes of 1,000 rectifier linear units. It is then fully connected to two consecutive ReLU hidden layers with 500 and 250 units, respectively. The last hidden layer is fully connected to the output layer which composes of 160 softmax units. The number of units in the output layer is set to the number of human subjects that we want to identify. The output of the DNN is therefore a vector with 160 elements. Each element represents one of the 160 subjects in the dataset. The value of an element shows the probability that the input is classified to its corresponding subject.

The output of the DNN is computed from a frame which is only a small part of the entire walking sequence. To confidently make a prediction for the entire sequence, we combine all the outputs in the same sequence into a final prediction by selecting the human subject with the highest frequency among the outputs from the DNN.

3 Experiments

To evaluate the proposed technique, we implement the proposed DNN architecture in Python using a deep learning library called Keras (`https://keras.io`). The DNN is trained with the ADAM optimizer using the multiclass logloss as the objective function. The experiment is conducted using 5-fold cross validation. Based on this setting, the dataset is randomly split into 5 folds and the experiment is conducted 5 iterations. In each iteration, one fold is used as the test set, while the rest 4 folds are combined into the training set. To avoid overfitting, 10% of the training set is used as the validation set. The training algorithm is set to stop the network training process when the validation loss is not improved for 10 rounds.

The result of the proposed techniques is compared to those of the existing techniques presented in [13], i.e. Support Vector Machines (SVM), k Nearest Neighbors (kNN), and Multi-Layer Perceptron (MLP) which is a shallow neural network. Different from the proposed technique, the input of these techniques are a handcrafted feature vector. Each feature is extracted from all the frame in the walking sequence, e.g. average stride length, average and standard deviation of Euclidean distances between adjacent joints. We also include the accuracy of the frame-level DNN outputs without majority vote.

4 Results and Discussion

Table 1 shows the experimental results. From the results, the technique combining DNN and majority vote performs much better than the existing techniques. It yields the highest accuracy of 95.0%. However, the technique using only DNN performs the worst with the accuracy of 64.6%. The results follows what we have explained in the previous section. A frame represents only one posture in the entire walking sequence. It therefore does not always contain sufficient information to correctly identify a human subject. Some postures of the subject may be very similar to those of the others. Combining outputs from all the frames in the walk allows us to identify the subject with higher confidence. This results in the drastic improvement of the accuracy when we apply the majority vote.

Comparing to the other learning algorithms using handcrafted features, the proposed technique is capable to transform its input in form of raw frames into a meaningful set of features and construct a classifier based on the learned set at the same time. Thus, it results the classifier with high prediction accuracy.

Table 1: Prediction accuracies comparing four techniques: the proposed DNN with majority vote technique, SVM, kNN and MLP

Technique	Accuracy
SVM	86.3%
kNN	87.7%
MLP	84.7%
DNN	64.6%
DNN+Majority Vote	95.0%

5 Conclusion

We proposed a technique to identify human subjects using skeletal data stream from Microsoft Kinect sensor. This technique accepts a sequence of 20 three-dimensional coordinates as its input, and predicts one of the human subjects. Its classification model is based on the combination of DNN and majority vote algorithms. The DNN is designed to predict a human subject from each individual frame. The predicted outputs in the same walking sequence are then combined and the human subject with the highest frequency is chosen as the prediction of the sequence. We evaluate the proposed technique using the real-world walk datasets. The results show that the technique outperforms the existing techniques using other supervised learning algorithms including SVM, kNN, and MLP. The proposed tehcnique also outperforms the technique using only the DNN without majority vote.

Acknowledgements : The authors would like to thank the referee(s) for his/her comments and suggestions on the manuscript. This work was supported by the Thailand Research Fund (contract number MRG5280151).

References

[1] A.M. Baumberg, D.C. Hogg, Learning Spatiotemporal Models from Training Examples, University of Leeds, School of Computer Studies, 1995.

[2] S.H. Shaikh, K. Saeed, N. Chaki, Gait recognition using partial silhouette-based approach, Proceedings of the International Conference on Signal Processing and Integrated Networks (SPIN) (2014) 101-106.

[3] E. Sudderth, E. Hunter, K.K. Delgado, P.H. Kelly, R. Jain, Adaptive video segmentation: theory and real-time implementation, Image Understanding Workshop 1 (1998) 177-181.

[4] T. Horprasert, D. Harwood, L.S. Davis, A statistical approach for real-time robust background subtraction and shadow detection, Proceedings of IEEE International Conference on Computer Vision 99 (1999) 1-19.

[5] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2 (1999).

[6] O. Javed, K. Shafique, M. Shah, A hierarchical approach to robust background subtraction using color and gradient information, Proceedings of Workshop on Motion and Video Computing (2002) 22-27.

[7] J. Han, B. Bhanu, Individual recognition using gait energy image, IEEE Transactions on Pattern Analysis & Machine Intelligence 28 (2) 316-322.

[8] J. Preis, M. Kessel, M. Werner, C. Linnhoff-Popien, Gait recognition with Kinect, 1st International Workshop on Kinect in Pervasive Computing, New Castle, UK (2012) 1-4.

[9] N. Saitong-in, K. Assantachai, N. Khamsemanan, C. Nattee, Human identification from gait analysis using Microsoft Kinect, Proceedings of ICT International Senior Project Conference (ICT-ISPC) (2013).

[10] A. Cheewakidakarn, N. Khamsemanan, C. Nattee, View independent human identification by gait analysis using skeletal data and dynamic time warping, Proceeding of the 14th International Symposium on Advanced Intelligent Systems (ISIS) (2013).

[11] N.V. Boulgouris, K.N. Plataniotis, D. Hatzinakos, Gait recognition using dynamic time warping, Proceedings of the 6th IEEE International Workshop on Multimedia Signal Processing (2004) 263-266.

[12] M. Milovanovic, M. Minovic, D. Starcevic, Walking in colors: human gait recognition using Kinect and CBIR, IEEE MultiMedia 20 (4) 28-36.

[13] V.O. Andersson, R.M. Araujo, Person identification using anthropometric and gait data from Kinect Sensor, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015) 425-431.

[14] Y. Bengio, Learning deep architectures for AI, Foundations and Trends ® in Machine Learning 2 (1) (2009) 1-127.

[15] D. Kingma, J. Ba, Adam: A method for stochastic optimization, ArXiv e-prints (2014) https://arxiv.org/abs/1412.6980.

(Received 3 December 2018)
(Accepted 4 March 2019)