HUMAN GAIT ANALYSIS USING HYBRID CONVOLUTIONAL NEURAL NETWORKS

KHANG NGUYEN¹, VIET V. NGUYEN², NGA T. MAI³, AN H. NGUYEN⁴, ANH V. NGUYEN⁵,

¹Graduate University of Science and Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam
²Faculty of Information Technology, Hanoi University of Science & Technology, Ha Noi, Viet Nam
³Faculty of Mathematics and Information Technology, Thang Long University, Ha Noi, Viet Nam
⁴R&D Department, PetroVietnam Exploration Production Corporation, Ha Noi, Viet Nam
⁵Institute of Information Technology, Vietnam Academy of Science and Technology, Ha Noi, Viet Nam

Abstract. Human gait analysis is a promising method of researching on human activities like walking or sitting. It reflects the habits of one person and can be observed in any activity that person performs. The patterns in human movements are influenced by many factors, including physiology, social, psychological, and health factors. Differences in limb movements help identify gait patterns, which are often measured using inertial measurement unit sensors (IMU) like gyroscopes and accelerometers placed in various locations throughout the body.

This paper analyses the combination of IMU sensors and electromyography sensors (EMG) to improve the identification accuracy of human movements. We propose the hybrid convolutional neural network (CNN) and long short-term memory neuron network (LSTM) for the human gait analysis problem to achieve an accuracy of 0.9418, better than other models including pure CNN models. By using CNN’s image classification advancements, we analyse multivariate time series sensor signals by using a sliding window to transform sensor data into image representation and principal component analysis (PCA) to reduce the data dimensionality. To tackle the dataset imbalance issue, we re-weight our model loss by the inverse effective number of samples in each class. We use the human gait HuGaDB dataset with unique characteristics, for gait analysis.

Keywords. Human gait analysis; Wearable IoT devices; Time-series analysis; Deep learning; PCA; CNN; HuGaDB.

1. INTRODUCTION

The movement of human limbs is referred to as a human gait. There are two related but distinct concepts about human gait, which are human gait analysis and human gait

*Corresponding author. E-mail addresses: khang_nt@yahoo.com (Khang Nguyen); vietnv bk.it@gmail.com (V.V. Nguyen); ngamt@thanglong.edu.vn (N.T. Mai); anhh1@pvep.com.vn (A.H. Nguyen); anhnv@ioit.ac.vn (A.V. Nguyen).

© 2023 Vietnam Academy of Science & Technology
recognition even though both involve the measurement and analysis of walking patterns, they have different goals and techniques. The goal of gait analysis is to diagnose and monitor abnormality in normal gait patterns caused by injury, disease, aging, or improper exercise, and to evaluate the effectiveness of interventions such as surgery, physical therapy, and assistive devices. The goal of gait recognition is typically to identify individuals for security or surveillance purposes, such as in the context of law enforcement or border control [20].

Claudio Filipi et al. provide a comprehensive survey about gait recognition in deep learning [12]. This survey covers gait recognition through deep learning-based approaches using CNN and RNN (recurrent neural networks) with its variant LSTM and GRU and a few other models like Autoencoder, Deep Brief Network, etc. This research also describes useful information about multiple gait datasets. Although this survey focuses on gait recognition, it can be a good reference for other research about gait analysis.

Regarding human gait analysis, it is an important technique for assessing pattern movement and diagnosing gait-related conditions. However, gait analysis is a complex process that requires the integration of multiple disciplines and technologies. In this paper, we review the literature on the challenges and the proposed approach associated with gait analysis, including data variability, equipment choices, interpretation challenges, etc.

There are different methods for human gait analysis, including sensor-based techniques, and vision-based techniques. The sensor-based technique involves the use of various sensors (IMU or EMG) to capture and measure specific parameters of a person’s gait pattern [21,26,36]. The vision-based technique uses cameras and reflective markers to track the movement of various body segments during gait, providing detailed information about joint angles and movement patterns [19,24,39].

This research paper aims to present an approach to analyzing human gait by utilizing a dataset based on sensors to distinguish between different activities, such as walking, running, and sitting. This falls under the category of “human activity recognition” (HAR), which is just one component of human gait analysis. The findings of this study have a wide range of applications, including the analysis of abnormal gait patterns, prevention of patient falls, patient rehabilitation, and investigation of neurological system diseases.

The main contributions of this paper are as follows:

- We investigate a typical human gait sensor-based dataset in the time series format and apply a sliding window technique to represent time series data as images. This motivates us to use the state-of-the-art deep learning model for image classification using CNN.

- The dataset consists of many sensors’ spatial signals from the left and right legs, which contain redundant information. We perform principal component analysis (PCA), an unsupervised learning model, on this dataset to reduce the dimensionality and improve the training process.

- This problem is multi-class classification, and we have to address an imbalanced dataset issue. We propose to use the class-weight technique to emphasise the learning in the minority class which ensures the model will learn equally from all classes. This is a different and easier way to balance the data set than by using data sampling methods.
We provide theoretical analyses to show the performance analysis of pure deep learning models using CNN versus hybrid CNN models with LSTM and GRU. We achieve outstanding performance with a suitable combination of deep learning models and image representation of time series data.

2. RELATED WORKS

Researchers have been using machine learning for human gait recognition for several years. Since 2002, Lee and Grimson described concepts for using gait appearance to identify people [28]. This traditional method uses simple features such as moments extracted from machine vision of human walking motion, videos, or a sequence of images. This vision-based technique is simple, but effective as extracted features contain sufficient information for accurate prediction of human identification.

In 2008, Chen et al. proposed a sensor-based technique for abnormal gait detection using a dataset collected by inertial sensors in a shoe-integrated device [6]. Chen’s method used PCA for feature extraction and Support vector machine (SVM) for multi-pattern classification. Additionally, Nguyen et al. proposed a biometric recognition technique using an enhanced $k$-NN algorithm based on deep images of gloved hands [31]. This method extracted finger dimensions to demonstrate that it may be viable in settings where gloves are worn, such as in a hospital.

The trend towards increased use of wearables began in 2015, Chattopadhyay and Nandy looked at human gait analysis using wearable devices to predict abnormal gait patterns, such as hemiplegia and horseshoe gait [5]. They suggested the use of the hidden Markov model (HMM) and Symbolic aggregate approximation (SAX) method for generating observation sequences obtained from sample gait cycles. The detection of abnormal gait patterns is based on the maximum log-likelihood of an observed sequence, generated from a gait movement. The experimental results demonstrated that the HMM can detect gait abnormality in gait data.

In 2017, Alotaibi and Mahmood enhanced gait recognition using a deep CNN on the large CASIA-B gait dataset [2], their proposed model was not sensitive to some cases of the common variations and occlusions that impact gait recognition results. In 2019, Gao et al. proposed a hybrid CNN model with LSTM to detect abnormal gait patterns [13], including tiptoe, hemiplegic, and cross-threshold gait. The study collected two normal gait patterns (normal walking and fast walking) and these three simulated abnormal gait patterns from 25 participants. Their experiments showed that hybrid CNN-LSTM achieved better performance than pure CNN or LSTM for abnormal gait classification, achieving 93.11% accuracy. Similarly, Wang et al. also studied gait analysis using CNN and LSTM (Conv-LSTM) to solve the problems related to cross-view gait recognition [41].

Mehmood et al. introduced a new deep learning-based framework for human gait analysis [29]. The method involved four main steps: Pre-processing video frames, modifying pre-trained deep learning models, selecting the best features using a firefly algorithm, and classification. The study also used feature fusion to enhance the feature representation. Experiments were conducted on three angles of the CASIA-B dataset: 18, 36, and 54. The resulting accuracies for each angle were 94.3%, 93.8%, and 94.7%, respectively.

Abhishek Tarun and Anup Nandy applied the vision-based technique to extract the gait
signature from the gait energy image (GEI) using CNN [39]. The researchers proposed using SVM, Random forest, and LSTM classifiers on their own dataset (recorded by Microsoft Kinect sensor using Color Depth video) and the CASIA-A dataset. Their proposed approach using the random forest classifier achieved an accuracy of 98.71% (their report showed SVM at 93.58% and LSTM at 94.93%), which was higher than the other method using DeepCNN, which achieved only 97.58%.

Latisha Konz et al. presented a spatiotemporal deep learning model (ST-DeepGait) [24], to feature spatiotemporal co-movement patterns of human joints, and accordingly classify such patterns to enable human gait recognition. They contributed their gait dataset captured with an RGB-D sensor containing approximately 30 video samples of each subject for 100 subjects. The multi-layer RNN architecture was employed to induce a sequential notion of gait cycles in the model and achieved a recognition accuracy of over 90%.

3. METHODS

This paper proposes a novel model based on deep neural networks to recognize multiple signals captured from sensor-based movements, which is multivariate time series data. CNN is a cutting-edge deep learning architecture framework for computer vision applications [35]. To empower CNN in the field of image classification [9], we apply image encoding techniques to represent the multivariate time series in image formats. We conduct simulations using various instances of CNNs, including pure CNNs and hybrid CNNs, by combining CNNs with LSTM and GRU to evaluate the performance [34] of each variant instance. To reduce dimensionality and learn characteristics from multivariate time series data, we investigated using an unsupervised training step with PCA [11]. With this combination of CNN and PCA, there is a great advantage because the model can learn an imaging representation of the time series data faster. To define the baseline result, we execute machine learning models, including Random forest and SVM, on the raw dataset to record outputs. The output from the baseline model is used to evaluate and compare other deep learning models that we develop.

This section is divided into three parts. The first part describes the materials used for the research. The second part presents various techniques for preparing the data, including reducing data dimensionality using PCA, framing time series data into windowed segments, and encoding these segments into images. The final part focuses on model development, it includes presenting a proposed model as well as techniques for dealing with imbalanced datasets and minimizing loss during optimization.

3.1. Material overview

Since smart wearable devices have become popular, many datasets have been made freely available to address the HAR problem using sensor-based techniques, including MAREA [23], OU-ISIR [40], and HuGaDB [7]. This paper uses the HuGaDB dataset, published by Chereshnev and Kertesz-Farkas in 2017, considered to be the most comprehensive dataset as it records both static and dynamic activities to provide rich data for gait classification research. The HuGaDB dataset uses a time series format for recordings of participants’ movements from wearable body sensors with specific information described in Figure 1.
Participants

- 18 participants with attributes: ID, Weight, Height, Age, Sex

Sensors

- 6 inertial sensors, each produces 3 acceleration and 3 gyroscope data on x, y, z axes
- 2 EMG sensors, each produces 1 signal
- Total 28 signals from 8 sensors

Activity Datafiles

- Dataset was recorded continuously for 12 activities of the 18 participants. The data is time-series format at about 58 samples per second
- Each activity datafile for a participant recorded from 1 second to 6 minutes
- 2,111,962 samples totaling 10 hours, traced in 657 activity datafiles

Figure 1: HuGaDB dataset overview [7]

In this experiment, participants wore eight sensors mounted on both legs and performed combined activities as shown in Table 1. During movement, multiple signals from the sensors sampled, recorded, and labeled the data for each activity performance into respective data files. During the data collection process, a 3-axis accelerometer (ACC), a 3-axis gyroscope (GYRO), and electromyography (EMG) sensors were used to produce data files using a total of 38 features described in Table 2. All of these features are in a time series format.

Table 1: Gait activities in HuGaDB

<table>
<thead>
<tr>
<th>ID</th>
<th>Activity</th>
<th>Time(s)</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Walk at various speeds</td>
<td>11,544</td>
<td>679,073</td>
</tr>
<tr>
<td>2</td>
<td>Run at various speeds</td>
<td>1,218</td>
<td>71,653</td>
</tr>
<tr>
<td>3</td>
<td>Stair-up</td>
<td>2,237</td>
<td>131,604</td>
</tr>
<tr>
<td>4</td>
<td>Stair-down</td>
<td>1,982</td>
<td>116,637</td>
</tr>
<tr>
<td>5</td>
<td>Sit on a chair</td>
<td>4,111</td>
<td>241,849</td>
</tr>
<tr>
<td>6</td>
<td>Sit down on a chair</td>
<td>409</td>
<td>24,112</td>
</tr>
<tr>
<td>7</td>
<td>Stand up from a chair</td>
<td>380</td>
<td>22,373</td>
</tr>
<tr>
<td>8</td>
<td>Stand on a solid surface</td>
<td>5,587</td>
<td>328,655</td>
</tr>
<tr>
<td>9</td>
<td>Bicycle</td>
<td>2,661</td>
<td>156,560</td>
</tr>
<tr>
<td>10</td>
<td>Stand in a lift up</td>
<td>1,515</td>
<td>89,144</td>
</tr>
<tr>
<td>11</td>
<td>Stand in a lift down</td>
<td>1,185</td>
<td>69,729</td>
</tr>
<tr>
<td>12</td>
<td>Sit on car</td>
<td>3,069</td>
<td>180,573</td>
</tr>
<tr>
<td>-</td>
<td>Total</td>
<td>35,903</td>
<td>2,111,962</td>
</tr>
</tbody>
</table>

3.2. Data preparation

In this section, we propose to use a time series analysis method [4, 11] to learn the structure and dependence of the time series dataset. Firstly, we apply a sliding window technique to frame the time series into segments in a matrix format, and then in the second step, we encode these segments into an image representation. In parallel, we use an alternate branch to execute PCA for dimensionality reduction to investigate another option for the classifier. The data flow of sequential steps to analyze and transform the time series dataset
Table 2: Signal inputs from sensors

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Inputs</th>
<th>Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Right Foot</td>
<td>1-3</td>
<td>ACC (3-axis xyz)</td>
</tr>
<tr>
<td></td>
<td>4-6</td>
<td>GYRO (3-axis xyz)</td>
</tr>
<tr>
<td>2 - Right Shin</td>
<td>7-9</td>
<td>ACC</td>
</tr>
<tr>
<td></td>
<td>10-12</td>
<td>GYRO</td>
</tr>
<tr>
<td>3 - Right Thigh</td>
<td>13-15</td>
<td>ACC</td>
</tr>
<tr>
<td></td>
<td>16-18</td>
<td>GYRO</td>
</tr>
<tr>
<td>4 - Left Foot</td>
<td>19-21</td>
<td>ACC</td>
</tr>
<tr>
<td></td>
<td>22-24</td>
<td>GYRO</td>
</tr>
<tr>
<td>5 - Left Shin</td>
<td>25-27</td>
<td>ACC</td>
</tr>
<tr>
<td></td>
<td>28-30</td>
<td>GYRO</td>
</tr>
<tr>
<td>6 - Left Thigh</td>
<td>31-33</td>
<td>ACC</td>
</tr>
<tr>
<td></td>
<td>34-36</td>
<td>GYRO</td>
</tr>
<tr>
<td>7 - Right Thigh</td>
<td>37</td>
<td>EMG (1 signal)</td>
</tr>
<tr>
<td>8 - Left Thigh</td>
<td>38</td>
<td>EMG (1 signal)</td>
</tr>
</tbody>
</table>

into image representations is illustrated in Figure 2, and details of each step are described in the following subsections.

![Figure 2: Overall flow for data preparation](image)

3.2.1. Data dimensionality reduction using PCA

A time series \( X = \{x_1(t), x_2(t), \ldots, x_d(t)\} \) is a sequence of \( d \)-dimensional observations in time order. Most of these observations are sampled at the same discrete-time intervals. It could be a univariate time series if there was only one variable (signal); however, the HuGaDB is a multivariate time series as there are up to 38 signals observed from eight sensors.

PCA [1] is a powerful unsupervised learning method for the dimensionality reduction of multivariate time series data. It uses an orthogonal transformation to transform a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. The objective of PCA is to find the best \( k \) – dimensional approximation to each observation in a \( d \) – dimensional dataset, where \( d > k \), while still retaining as much information as possible. With this approach, PCA will reduce the dimensions from \( d \) to \( k \) by extracting the most important features, which are sufficient to cover the variations in the data and can help reduce the data size to enhance processing in the next steps. The PCA algorithm works by finding the covariance matrix of the dataset. It then decomposes the
matrix to obtain the eigenvectors and eigenvalues. The eigenvectors are referred to as the principal components (PC) of the dataset and help in the selection of the most suitable $k$ value for the new dimensional space. Figure 3 illustrates the importance of $PC_x$ from the eigenvectors after executing PCA on the HuGaDB dataset.

![Figure 3: Principal components of the HuGaDB dataset](image)

From this diagram, we decide to select $k = 14$ for the new feature space, which is equivalent to twelve signals from six inertial sensors and two other signals from two EMG sensors. Thus, we accept some information loss after the PCA transformation with the benefit of reducing the dimensionality of the HuGaDB dataset from 38 to 14 dimensions.

### 3.2.2. Data framing using sliding window

A method that uses an individual sample of the time series as training data input for the classifier will not achieve good accuracy as it does not consider the temporal dimension in the series of data samples. Therefore, we propose to use the sliding window method to frame a time series dataset. A 2-D sliding window is used to frame a multivariate time series into 2-D segments in the matrix data format. The frame size is 100 samples per window, which is equivalent to nearly 2 seconds of the participant’s movement. We do not have to handle the multi-class window problem because HuGaDB records all the samples in one datafile using the same activity label. We slide the window file-by-file to create 2-D segments with the respective labeled activity and we do not apply an overlap between two consequence windows.

The sliding window process [38] is illustrated in Figure 4, which shows six signals from ACC sensors in the right foot and right shin.

In the actual experiments, we utilized a sliding window approach for three distinct data inputs. These inputs included raw data with 38 signals, the 3-dimensional coordinates (i.e., xyz) of the sensors, and a third option utilizing PCA-transformed data with $k = 14$ signals. As a result of this process, we obtained three distinct outcomes, each consisting of segments with dimensions of $100 \times 38$, $100 \times 12 \times 3$, and $100 \times 14$. These outcomes were then used as inputs for the subsequent step of encoding the segments into images.
3.2.3. Encoding time series to images

Each segmented time series forms a 2-D matrix where there is a relationship between matrices and images. Each pixel of an image is represented by a matrix element, either a raw value by pixel plotting or a transformed value using another specific algorithm, such as Fourier transform (FT), recurrent plotting (RP), Markov transition field (MTF), or Gramian angular field (GAF) [42]. This approach is motivated by recent successes of deep learning in computer vision [10], especially the deep CNN model that we present in the next section.

Using the three outcomes from the sliding window process, we generate three corresponding alternate output images. Notably that we apply a 3D plot to the segmented dataset with dimensions of $100 \times 12 \times 3$ and the PCA-transformed dataset is utilized to investigate the impact of applying PCA to a multivariate time series. Figure 5 depicts the three interim image datasets, which will be used as the input layer for deep learning models during the training stage.

As a result of sliding a 100-length window over the 637 data files and 2,111,962 samples, of which data files with less than 100 samples are ignored from the execution. The whole multivariate time series dataset is converted to 20,432 segments and then encoded into image representations. The output contains three alternate sets of images as inputs to train the model to evaluate which method will achieve the best result. The image representations are also labeled with corresponding activities from the raw datafile, as illustrated by the distribution in Figure 6.

From this distribution diagram, we see that the dataset is imbalanced. Learning from an imbalanced dataset can be very difficult as most learners will predict bias towards the majority class, even in some extreme cases, ignoring the minority class. There are two common methods to address the multiple class imbalanced data problem. There are two common methods for handling class imbalance in machine learning: (1) data-level techniques that try to balance the data classes by sampling methods and (2) algorithm-level methods that implement class-weight to adjust the learning to reduce bias or customize the loss.
This paper proposes to use a class-weight method during the training process to handle the imbalanced dataset [15,37] which is described in the subsection “Model training strategy”.

3.3. Proposed models

3.3.1. Model development

Human gait datasets normally comprise a time series format from wearable sensors, and we propose an approach to transform the multivariate time series into image representations. CNN is proposed by Yann LeCun [27] as an evolved architecture of deep neural networks being used popularly for image classification problems. The innovative idea of CNN is that it can efficiently scan an image with a small window using a weight matrix to learn the pixel
values in that local region of the image. This slides the window throughout the whole image and produces another image. This step is called the convolution step. The produced image is scanned again in the next layer repeatedly, so the CNN consists of multiple convolution layers. The CNN’s benefits are that having fewer parameters but greatly enhances the learning time and reduces the amount of data required to train the model. In addition to the visual imagery analyses by CNN, we expect the model to be capable of recognizing patterns in sequences of data like multivariate times-series. Therefore, we propose the combination of CNN and recurrent neural networks (RNNs) is suitable for handling this requirement [34]. RNNs are another type of DNN and are often used with sequential data types. There are two common variants of RNNs: LSTM and GRU [32], as shown in Figure 7.

![Figure 7: Overview of LSTM and GRU models](image)

LSTM was proposed by Hochreiter and Schmidhuber to overcome the disadvantages of traditional RNNs by combining short-term with long-term memory through controllable gates [16,17]. LSTM has input and output gates and a forget gate to filter out less important information. LSTM maintains cell memory using cell state. GRU was proposed by Cho et al. [8] and is a special type of RNN based on a simplified form of LSTM. GRU combines the input and forget gates into a single update gate. GRU does not maintain the cell state and does not have an output gate. As a simplified version of LTESM, GRU has fewer training parameters than LSTM and it is expected to learn faster.

### 3.3.2. Model training strategy

**Class weight for multi-class imbalanced learning**

Multi-class learning is considered to be a challenging task for classification models as it impacts the performance lower than binary cases [25], this issue becomes even harder when faced with imbalanced data. Imbalanced data refers to a classification problem where data per class is not equally distributed; a majority class contains a large amount of data for that class, and a minority class contains a small amount of data [18]. If we had not managed this problem, the machine learning model would have been subject to a frequency bias toward the majority class.
There are two common ways to manage the class imbalance [14,18,43]: (1) under- or over-sampling to balance the data at the data preparation stage, and (2) specify class weights for classes at the training stage. We use the second method to address the class imbalance by providing a weight for each class; this places more emphasis on the minority classes to enable a classifier to learn fairly from both classes. Different from the first method, we do not have to manipulate data at the data preparation stage. The class-weight calculation is shown in Equation (1)

\[
\text{class_weight}_i = \frac{N^s}{N^c} \times N_i,
\]

where, \(N^s\) the total number of samples in the whole dataset; \(N^c\) the number of classes; \(N_i\) the number of samples of the class \(i^{th}\); \(\text{class_weight}_i\) the weight of the class \(i^{th}\).

### Min-loss optimization

The loss function “Categorical Crossentropy” is used during model training, and we adopt the min-loss technique as the optimization strategy to measure the accuracy of the model. This technique is defined as an early stop condition when training a complex network model. The log-loss function is shown in Equation (2)

\[
L(y, \hat{y}) = -\sum_{j=0}^{M} \sum_{i=0}^{N} (y_{ij} \ast \log(\hat{y}_{ij})).
\]

### 4. RESULTS

This section is structured into three parts. The first part proposes an evaluation approach, which details how the train, test, and validation datasets are proposed, as well as the performance metrics used for evaluation. The second part presents the experimental results, and the last part covers the discussion.

#### 4.1. Model evaluation

##### 4.1.1. Validation approach

Initially, we conducted the validation by both dividing the train and test sets at a 70/30 ratio and k-fold cross validation but the results were not good and suitable for this type of dataset, then we separated the training, validation, and test data by person. The validation set contains one person, and the test set contains two persons, randomly selected from 18 persons, and the training set contains the rest of 15 persons.

The activity data distribution by person is shown in Figure 8. From this figure, we can see that the activity sit on car and bicycle are only available for Person 1. Thus, we exclude this person for testing purposes, as doing so would not produce an accurate result.

##### 4.1.2. Performance metrics

The performance evaluation used a confusion matrix, and the following metrics are calculated:

- **Accuracy** gives the percentage of correctly predicted results.
- **Precision** gives performance information concerning false positive.
Recall gives the performance information concerning false negative. F1-Score is a mean of Precision and Recall.

4.2. Experimental results

In addition to the baseline models that use Random forest and SVM on the raw dataset, our study also evaluates other deep learning models. This includes pure CNN models with different composite layers and hyper-parameters, as well as hybrid CNN models with LSTM and GRU. We conducted multiple simulations to compare the performance and stability of the candidate model. Each simulation was executed on all three interim image datasets that we transformed from multivariate time series datasets to determine which candidate model achieved optimal performance. We defined the baseline model using SVM without any data processing tasks and then compared the accuracy of the baseline model with various instances of CNN models to evaluate the improvements.

Table 3: Experimental results

<table>
<thead>
<tr>
<th>Model</th>
<th>Pre-processing</th>
<th>Val. ACC</th>
<th>Test ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>No (Raw Data)</td>
<td>0.9100</td>
<td>0.8603</td>
</tr>
<tr>
<td>CNN</td>
<td>2D Plot with Sliding Window</td>
<td>0.8189</td>
<td>0.8794</td>
</tr>
<tr>
<td>CNN</td>
<td>3D Plot with Sliding Window</td>
<td>0.7479</td>
<td>0.8815</td>
</tr>
<tr>
<td>CNN</td>
<td>PCA and 2D Plot</td>
<td>0.9200</td>
<td>0.9010</td>
</tr>
<tr>
<td>CNN-GRU</td>
<td>PCA and 2D Plot</td>
<td>0.9500</td>
<td>0.8800</td>
</tr>
<tr>
<td>CNN-LSTM</td>
<td>PCA and 2D Plot</td>
<td>0.9502</td>
<td>0.9418</td>
</tr>
</tbody>
</table>

The results in Table 3 show that pure CNN models yield only a slight improvement over the SVM baseline accuracy of 0.8603, whereas using 2D and 3D plots with CNN (0.8794 and 0.8815) and applying PCA (0.9010) perform better performance. The CNN with PCA reduces noise and produces independent, uncorrelated features with an optimal number of
principal components \((k = 14)\).

After conducting various experiments with the hybrid model of CNN and other composite layers, we found that the hybrid CNN-LSTM achieved the highest accuracy of 0.9418, whereas the hybrid CNN-GRU model did not perform well. We observed that the GRU model had fewer training parameters, leading to faster training, but it was not as effective as the LSTM model in handling time series data. The superior performance of the hybrid CNN-LSTM model can be attributed to its ability to capture both spatial and temporal features of the gait data, allowing for more accurate classification.

The optimal CNN-LSTM model is shown in Figure 9. The confusion matrix and classification report diagrams are in Figure 10, and the accuracy and log-loss diagrams are plotted in Figure 11.

From the confusion metric and classification report diagrams, we found that the final classifier using CNN and LSTM performed well in identifying activities that can be easily distinguished from each other, such as running, walking, and stair-up and -down. However, the classifier incorrectly identifies activities that are types of standing, especially stand in the lift-up and -down, where the F1-score and Recall are very low. It predicts incorrectly other types of the stand activity, such as Stand on a solid surface. There is an obvious note that all metrics are zero for the class Bicycle and Sit on car as there is only the Person 1 has these two activities and it is excluded from the test set (refer to the sub section Model evaluation), however there is one sample of the activity Stand on a solid surface is incorrectly classified as the activity Bicycle.

4.3. Discussion

There are few researches using this dataset \([3, 22, 30]\) for human gait analysis. Kececi et al. (2018) \([22]\) applied some machine learning models including random forest, decision tree, naive bayes, multi-layer perceptron, etc., but their study selected only four out of 12 activities and only 54 out of 637 data files from the dataset, making it incomparable to our study that utilized the full dataset with all 12 activities for classification. In a study conducted by V. Nastos et al. in 2022 \([30]\), they utilized Chi-square \([33]\) and PCA as two
Figure 10: The Confusion matrix and Classification report of the Hybrid CNN-LSTM feature reduction techniques, which were combined with five classification models: Decision tree, random forest, k-NN, SVM, and AdaBoost. According to their experiments, the number of features was reduced from 38 to 36, indicating that gyroscopes and EMG devices are the most crucial wearable sensors, and the extracted features from these sensors have a higher chance of being selected for model training. After the feature reduction, their features remained in a numeric format, which is different from our research that employed image representation techniques. The study’s findings showed that the random forest approach was the optimal choice for analyzing gait patterns, achieving an accuracy of 80% and an F1-score of 79%. Our research yielded better results in terms of accuracy when compared to their study.
5. CONCLUSION

Human gait analysis is a critical research area that has applications in several domains, such as rehabilitation and ergonomics. Recent developments in technology and biomechanics have provided a more in-depth understanding of the complex patterns of movement involved in gait, leading to enhanced treatment and preventive measures for gait-related disorders. The sensor-based technique involves recording accelerometry data of human movements through time-stamped accelerometer readings obtained from sensors attached to the body. To process the resulting multivariate time series data, we devised a data preparation method that utilizes the sliding window technique to segment the data and encode dataframes into various image formats. Additionally, we employed PCA as a spatial time series dataset to reduce data dimensionality by extracting relevant information while discarding unnecessary data as noise.

This paper proposes the use of CNNs, which have proven to be effective for image and time series data, along with two complementary neural network architectures, GRU and LSTM, for efficient human gait analysis. The simulation results and diagnostics conducted on the most comprehensive human gait database (HuGaDB) show that the hybrid CNN-LSTM model outperforms traditional machine learning models and pure CNN models. These findings suggest that the combination of CNNs with LSTM can provide a more accurate and robust approach for analyzing complex human movement patterns.

In conclusion, the combination of CNN and LSTM can result in a more reliable and precise system for human gait analysis. The CNN can extract relevant features from the data, while the LSTM can model the temporal dependencies between these features. By working together, these architectures can provide a more comprehensive understanding of the complex movement patterns involved in human gait.

REFERENCES


[33] K. Pearson, “X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 50, no. 302, pp. 157–175, 1900.


February 07, 2023
Accepted on May 08, 2023