Image-based Gramian angular field processing for pedestrian stride-length estimation using convolutional neural network

Pham Doan Tinh, Bui Huy Hoang, Nguyen Duc Cuong
Department of Electronics and Computer Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

Article Info

Article history:

Received Feb 20, 2021
Revised Aug 7, 2021
Accepted Aug 23, 2021

Keywords:

Convolutional neural network
Gramian angular field
Image-based
Pedestrian stride-length

ABSTRACT

In an age when people spend most of their time indoors and smartphones become a necessity, there is an increasing demand to navigate user absolute position in indoor environments. While global positioning system (GPSs) perform well outdoors, their inaccuracy can not be tolerated in places where GPS signal is weak or barely detected. This leads to a number of solutions which utilize smartphone inertial measurement unit (IMU) to track user location. Most IMU-based methods track the trajectory of a person by using stride-length and heading estimation. Thus, the accuracy of stride-length estimation plays a very important role in these methods. Inspired by recent success in the field of computer vision and machine learning, we proposed an image-based stride-length estimation method that employs Gramian angular field (GAF) in converting accelerometer data into images, and then feed them into a convolutional neural network (CNN) to predict the stride-length. We evaluate the performance of our proposed method by using a public dataset from Qu Wang in his GitHub repository (available at https://github.com/Archeries/StrideLengthEstimation). The result shows that our proposed method is superior in terms of accuracy in one stride and in large walking distance than others using only data collected from the accelerometer.

Corresponding Author:

Pham Doan Tinh
Department of Electronics and Computer Engineering
School of Electronics and Telecommunications
Hanoi University of Science and Technology
No. 1 Dai Co Viet street, Hanoi, Vietnam
Email: tinh.phamdoan@hust.edu.vn

1. INTRODUCTION

A method to estimate a person's location without the support of any external infrastructure in the environment is known as pedestrian dead reckoning (PDR). This technique only utilizes the inertial measurement unit (IMU) sensors (namely, accelerometer, gyroscope, and sometimes magnetometer), which are attached or carried by the users. To obtain user relative position, three important values must be extracted, which are step event, stride length, and heading. Among the mentioned tasks, stride length estimation (SLE) receives attraction from many researchers because this information is valuable not only in positioning but also in activity monitoring, and gait analyzing [1].

The simple SLE method assumes that people’s average stride length can be represented using a constant. This approach is of course not accurate because different people have different stride lengths. Many studies on SLE were done with advanced techniques and models, D´ıez in [2] did a survey and divided approaches into two classes: direct methods and indirect methods. In the scope of this paper, we examine three main approaches: the first is the biomechanical methods, the second is integration methods, and the final is adaptive methods.

Biomechanical methods utilize gait analysis in their models like Miyazaki's [3], in which he used a gyroscope attached to the subject's thigh to measure the angles created by lower limbs during the walking motion. When evaluating the step length, the author assumed that the length of the subject legs is already known and two steps in the same stride are equal. Potential errors from assumptions are corrected by taking advantage of the relationship between stride length and walking velocity. Zijiska in [4] came up with a method to calculate stride using lower limb length and the difference in height of the center of mass (COM). Another approach was proposed by Weinberg [5], in which he used vertical acceleration to estimate the stride length. An attempt to improve Weinberg’s equation like Kang in [6] where he set up another logarithm-based formula to combine with the original one with some added constraints.

The double integration model could be implemented as a strap-down inertial navigation system (INS). Li and Young in [7] used a 2-axis accelerometer and a 1-axis gyroscope placed on a subject's shank to collect movements. The walking motion is then segmented and converted into a world coordinate frame using the angle calculated from gyroscope readings. Kose and colleagues in [8] took an approach that used a wavelet-based decomposition method to detect and separate steps from each leg. Then they applied the Kalman filter and reverse integration to compute step length. Error in the method is compensated by removing the pelvic rotation from the model. Another implementation to correct the sensor error is using the null-velocity update point (ZUPT) to reset the integration which can be found in [9]-[12].

There are two types of models in the adaptive approach which are parametric and non-parametric models. Kim in [13] perform experiments to determine the correlation of stride length and the mean of accelerometer signal from that same stride. Considering the method proposed in [14], which focuses on the importance of frequency and its linear relationship with the stride length. A similar approach with more features added can also be seen in [15]. Methods utilized variance of accelerometer signals to use in their model can be found in [16]-[18]. Besides the linear model, Zihajehzadeh in [19] uses gaussian processed regression (GPR) to achieve a better result. Much recent research on using non-parametric models like the method in [20] took advantage of Neural Network using three different values computed from maxima and minima in each stride as features. Hannink in [21] also used CNN but the accelerometer and gyroscope signal are normalized to 256 samples per stride. Gu in [22] trained a Stack Autoencoder to learn important features from input data, then they are fed to a regression layer for stride estimation. Although much progress was made in estimating a person's stride length, existing methods still pose limitations. The drawback of the biomechanical methods is that some parameters are required to know beforehand, which might not be available. About double integration, the sensor position plays a major role, thus smartphones or other electronic devices may not be suitable. Finally, with adaptive models, feature selection is crucial because it has a great influence on the performance of the model. After having a relative position of a user, we may then combine with some indoor positioning methods to make absolute position prediction to be more accurate [23], [24].

We took a different approach to solve the mentioned problems and present a unique method to estimate stride length. First, we only use accelerometer data from the dataset collected by Wang in [25]. Second, it doesn’t require knowing any information about users' height or leg length. Third, to reduce the task of feature selection and determine their relationship, data is preprocessed and converted to images using the GAF algorithm [26], which has been successfully applied as a time series encoder in [27], [28]. Finally, for the task of learning, we used the CNN model due to its flexibility and accuracy.

2. RESEARCH METHOD

2.1. GAF algorithm

In our research, we focus on exploiting the accelerometer due to its ability to collect data related to user walking motion. The raw output of an accelerometer can be described as time series and its patterns can be extracted to estimate subject stride length. To retain the features of the data, we took a new approach to present information using the GAF algorithm proposed by Wang in [26]. Wang algorithm is suitable for converting one-dimensional time series data into a two-dimensional array, which can also be interpreted as an image. The method is briefly described as follows.

Suppose that our accelerometer data is in form of a time series $X = \{x_1,\ x_2, \ldots x_n\}$, where n is the size of X. First, we would have to rescale data in the range of [-1, 1] as (1).

$$\widetilde{x_i} = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)} \quad (1)$$

Where $\widetilde{x_i}$ is the normalized value of x_i; $\max(X)$, $\min(X)$ is maximum and minimum value of X, respectively. The rescale data can be expressed in a polar coordinate system by using the following transformation.

$$\begin{cases} \emptyset_i = \arccos(\,\widetilde{x_i}) \\ r_i = \frac{i}{N}\,,\ i \in N \end{cases} \tag{2}$$

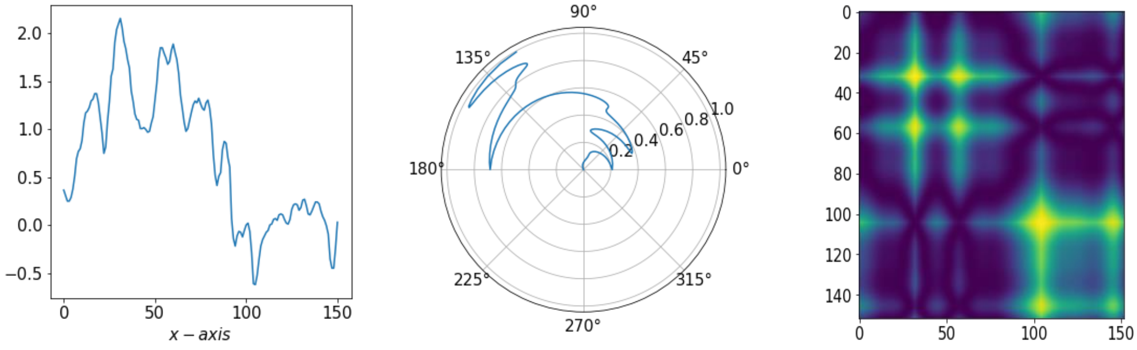
Where $\emptyset_i$ is the angle, r_i is the radius and N is the number of the data points. The cosine function would respond to input value in range of $[-1, 1]$ as $[0, \pi]$

This representation gave us another way to gain insights into time-series data. We can calculate the trigonometric sum/difference among sampling points to determine the time correlation between them. Gramian angular summation field (GASF) and Gramian angular difference field (GADF) are defined as (3), (4):

$$GASF = \begin{pmatrix} \cos(\emptyset_1+\emptyset_1) & \cdots & \cos(\emptyset_1+\emptyset_n) \\ \vdots & \ddots & \vdots \\ \cos(\emptyset_n+\emptyset_1) & \cdots & \cos(\emptyset_n+\emptyset_n) \end{pmatrix} \tag{3}$$

$$GADF = \begin{pmatrix} \sin(\emptyset_1+\emptyset_1) & \cdots & \sin(\emptyset_1+\emptyset_n) \\ \vdots & \ddots & \vdots \\ \sin(\emptyset_n+\emptyset_1) & \cdots & \sin(\emptyset_n+\emptyset_n) \end{pmatrix} \tag{4}$$

We utilize this algorithm to transform sensor data into images and the procedure is illustrated in Figure 1.



(a) (b) (c)

Figure 1. Illustration of GAF algorithm: (a) one dimensional time series data, (b) polar coordinate representation, (c) gasf representation

2.2. Proposed stride length estimation method

2.2.1. Overall architecture of the method

We proposed a method for stride length estimation, which consists of three phases as shown in Figure 2. The first phase is data preprocessing which handles raw data from the accelerometer sensor through filtering, segmentation and convert the signal to images. Inside data preprocessing we have a module called time series to image conversion. Its task is to rescale the data, represent data in polar coordinate, then construct a GASF or GADF matrix, the input to the CNN is normalized by resizing the GASF matrix to a fixed size (128x128). The second phase is training the CNN model using the images and labels extracted from the training dataset. Details of the model will be described in the latter section. After training, we use that model to predict value from the testing dataset.

2.2.2. Data preprocessing

Raw accelerometer sensor data is subject to noise from the shaking of user motion. To reduce the noise, we apply the Butterworth low-pass filter with a cutoff frequency equal to 5 Hz and an order equal to 5. After accelerometer readings are filtered, they need to be divided into smaller segments. Most of the time, this task is performed by a step detector or step counter. To simplify this requirement, we assume that the data was already divided, and each segment presents one stride as can be seen in Figure 3. After segmentation, filtered data from each axis will be converted to an image using the GAF algorithm mentioned in the previous section. The procedure can be seen in Figure 4.

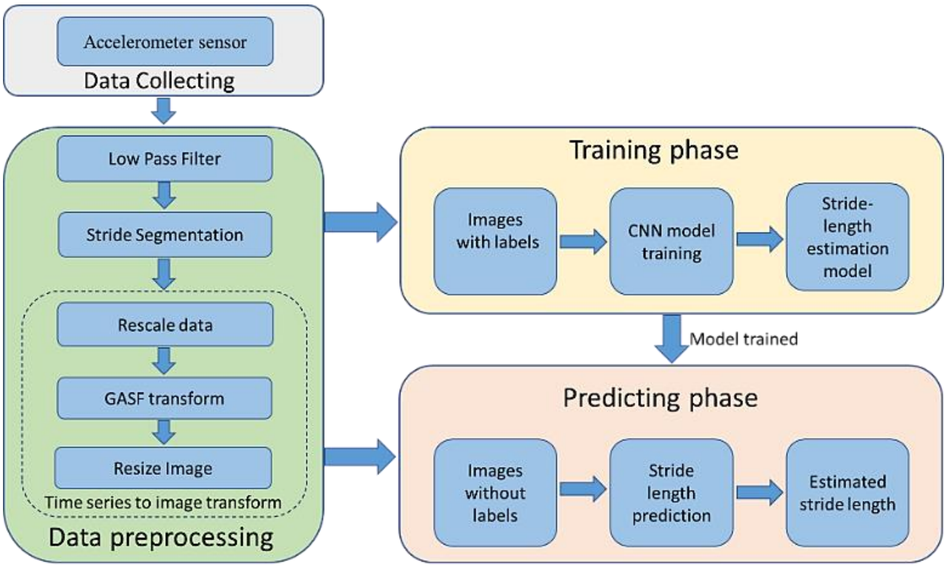


Figure 2. Proposed method architecture

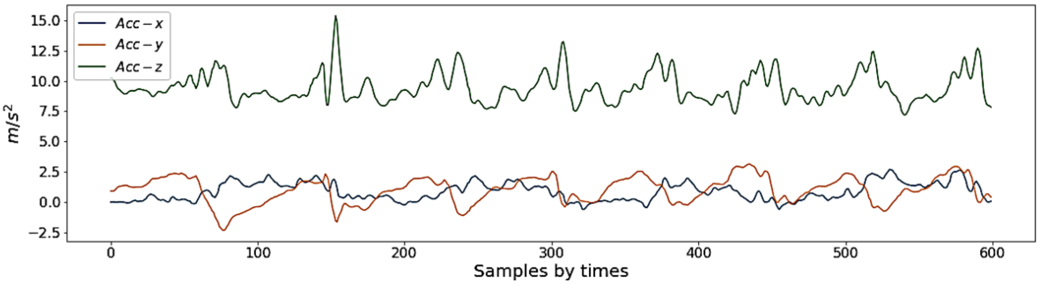
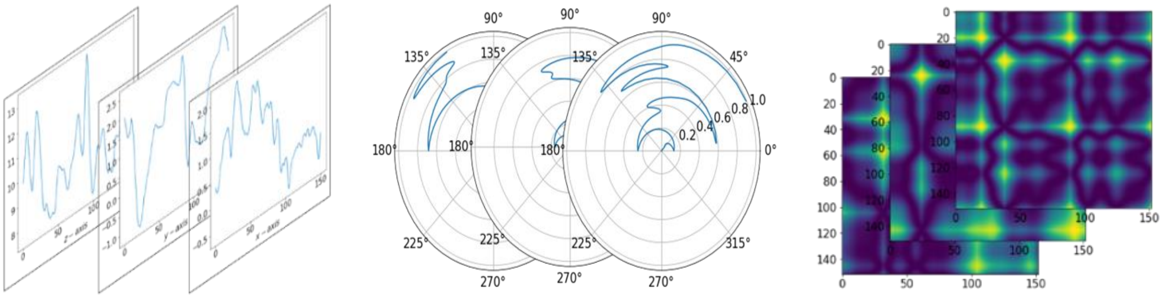


Figure 3. Raw data from accelerometer



(a) (b) (c)

Figure 4. Data to images conversion procedure: (a) Filtered data of 3 axes, (b) polar coordinate data of 3 axes, (c) GASFs

The CNN model requires input images to have a fixed size. However, each stride duration is different, which leads to different sizes of the image. Thus, we would have to resize the image to a particular dimension. As stated in the dataset [25], the sampling rate is 100 Hz and each stride contains about 120 samples, so we chose the size of the image in one axis to be 128×128 to retain the features inside.

2.2.3. CNN architecture

Most of the task involving CNN for images is classification. However, in our case we want the output to be the stride length so CNN would be treated as a regression model. We designed a simple CNN model that consists of 7 layers. First, we apply a convolutional layer (with ReLU activation) to create the feature map of the detected features from image input, then to prevent overfitting we use a dropout layer (with a rate of 0.3) before features are flattened, we normalize them using a BatchNormalization layer. After that 2 fully connected layers are used followed by a neuron that has a linear activation function at the end of the model. Except for the last neuron, all layers utilize rectified linear units as their activation function. For better illustration, the CNN architecture of each layer is shown in Figure 5.

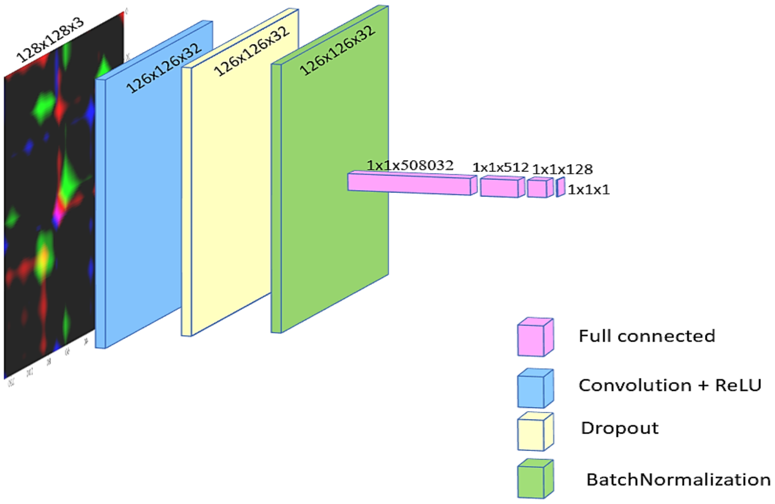


Figure 5. CNN model architecture

3. EXPERIMENTS AND EVALUATION

3.1. Distance estimation

To evaluate the performance when subject travel in large distance, we need to calculate the accumulated walking distance. The accumulative distance of the subject is computed as (5):

$$\widetilde{D} = \sum_{i=1}^{N} \tilde{s}_i \quad (5)$$

where $\widetilde{D}$ is the total traveled distance, $\tilde{s}_i$ is the estimation of i^{th} stride and N is the number of strides.

3.2. Error evaluation metrics

To keep consistence among the error metrics used for evaluating, we adopted the evaluation metrics from the dataset [25]. The relative stride error is calculated as (6):

$$E_s = \frac{1}{N}\sum_{i=1}^{N}\frac{|s_i - \tilde{s}_i|}{s_i} \times 100\ (\%) \quad (6)$$

where E_s denotes the stride length relative error; s_i and $\tilde{s}_i$ are the actual stride length and the estimated stride length of the i^{th} stride, respectively. The relative distance error is computed as (7):

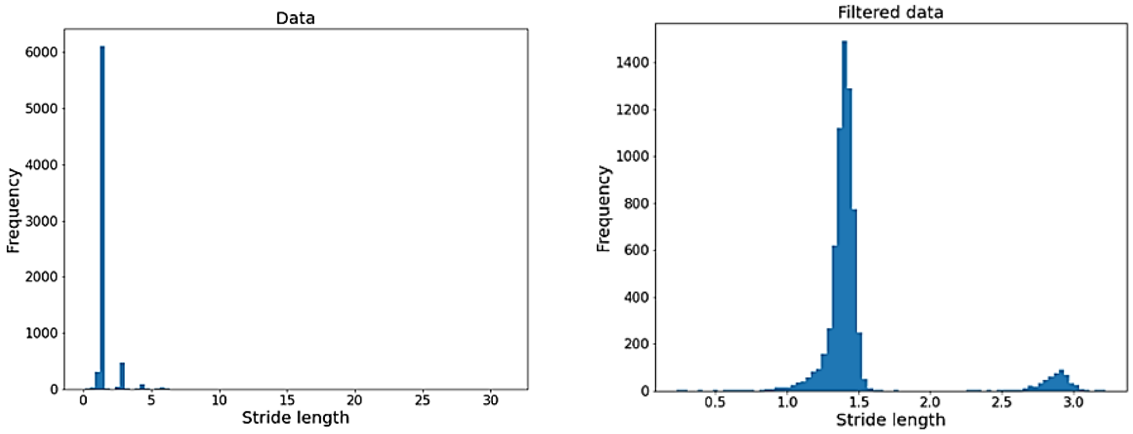
$$E_d = \frac{|\sum_{i=1}^{N}\tilde{s}_i - \sum_{i=1}^{N} s_i|}{\sum_{i=1}^{N} s_i} \times 100\ (\%) \quad (7)$$

where E_d denotes the walking distance relative error; $s_i, \tilde{s}_i$ are the actual stride length and the estimated stride length of the i^{th} stride, respectively.

3.3. Dataset

The dataset we chose for training and evaluation was created and presented by Qu Wang in [25]. In his dataset, 10000 strides and their parameters were recorded including readings from accelerometer, gyroscope, and magnetometer. To better illustrate the dataset, we analyzed the stride length-frequency distribution of the whole dataset in Figure 6(a).

From Figure 6(a), it is observed that most strides fall in the range from 0.2 meters and 3.5 meters, which is reasonable as the subject walking at different velocities. However, there exists a case when measured stride-length is above 3.5 meters and reaching nearly 30 meters. This happened because Wang's dataset covers several unique scenarios for example when users using escalators or elevators. If we keep those unusual data in the dataset, it could create a false pattern which can ruin the model. To prevent this, it is important to also implement an activity recognition algorithm to distinguish between different motion patterns and scenarios. However, the dataset does not provide us with the label of the movement type or subject walking environment, so it is not possible to classify subject unique cases. To simplify the problem that we studied, we filter out all the data that is not in the [0.2, 3.5] meter range. After filtering, the dataset has 7998 strides left and the distribution is shown in Figure 6(b).



(a) (b)

Figure 6. Dataset before and after being filtered, (a) before filter, (b) after filter

Next, we use the stride number to segment the dataset into a series of strides. This series is also labeled using the provided stride-length column in the dataset. As the accelerometer is our main concern, only signals from the accelerometer are used. To provide data for the training phase and evaluation phase, we split the data into the training set, validation set, and evaluation set. We use 5612 steps and 1403 steps for training and validating, respectively and the remains for evaluation.

3.4. Experimental result and analysis

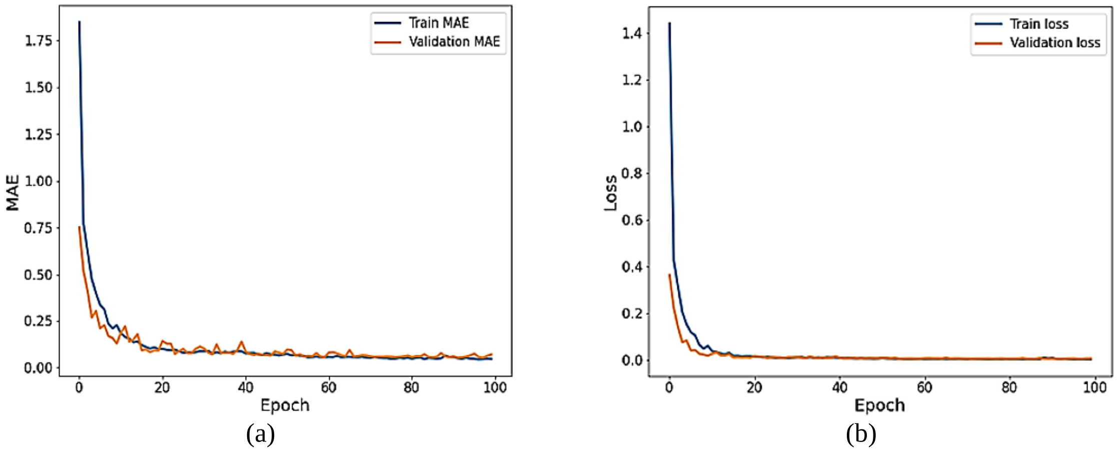
3.4.1. Model hyperparameters and the performance evaluation

Our model was built using Keras library. We use Huber as the loss function of the model because it is better to outlier than others. For the optimization task, we try several optimizers and found that Adam optimizer is the best fit for our model. Besides, to prevent overfitting the model, early stopping was utilized. The summary of a model hyperparameter is shown in Table 1.

Table 1. Proposed model hyperparameter

Parameter	Value and setting
Loss function	Huber
Optimizer	Adam
Learning rate	0.0001
Metrics	Mean Absolute Error (MAE)
Batch size	32
Epoch	100
Early Stopping	15

Figure 7 illustrates the mean absolute error (MAE) and Loss during the training process. The error and loss decrease rapidly in the first couple of epochs. Then as the iterations increase, the MAE became stable after 60 epochs while loss only needed 20 epochs to reach that state. The model gets the optimal performance after 92 epochs with the training loss, validation loss, training MAE, validation MAE values equal to 0.00381, 0.00141, 0.0515, and 0.05431 correspondingly.



(a) (b)

Figure 7. MAE and Loss of training and validation: (a) MAE, (b) loss

We evaluate the performance of our model using the prepared test set and plot the comparison of estimated stride length and the actual value in Figure 8. Figure 9 shows the result of some concrete strides from raw signals, to intermediate GAF images and corresponding stride-length prediction. From the figure, we can see that our proposed method gives the closest prediction value to the actual ones.

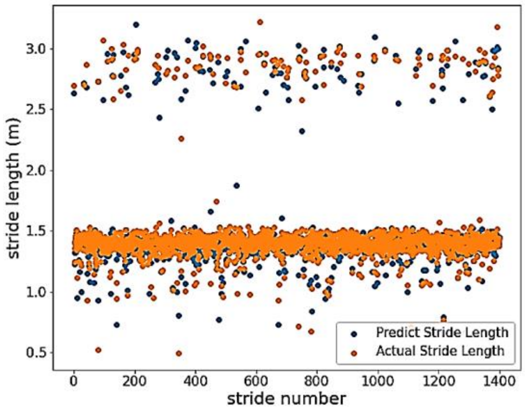


Figure 8. Comparison of estimated stride length in proposed method and actual stride length (with Validation data)

3.4.2. Comparison with other models

From raw signals, we calculate the root square of a_x, a_y, and a_z from the accelerometer sensor, and then apply low-pass filter to feed the signals to the GAF transformation.

$$A = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

For comparison, we implemented 4 models from Kim [13], Yao [16], Shin [17], and Weinberg [4]. These models can be briefly described as (8), (9), (10), (11),

$$SL_{Kim} = K.\sqrt[3]{\frac{\sum_{i=1}^{N}|A_i|}{N}} \tag{8}$$

$$SL_{Yao} = \alpha\sqrt[4]{A_{\max} - A_{min}} + \beta.f + \gamma.v \tag{9}$$

$$f_i = \frac{1}{t_i - t_{i-1}}$$

$$SL_{Shin} = \alpha.f + \beta.v + c \tag{10}$$

$$SL_{Weinberg} = K\sqrt[4]{A_{max} - A_{min}} \tag{11}$$

where $A_{max} and\ A_{min}$ denote the maximum and minimum acceleration values, respectively; A_i is the acceleration value at stride i^{th}; t_{i-1} and t_i are the starting and ending moments of time at step i^{th}; f is the stride frequency and v is the acceleration variance of the step. $K, \alpha, \beta, c, and\ \gamma$ are model coefficients identified during the training process.

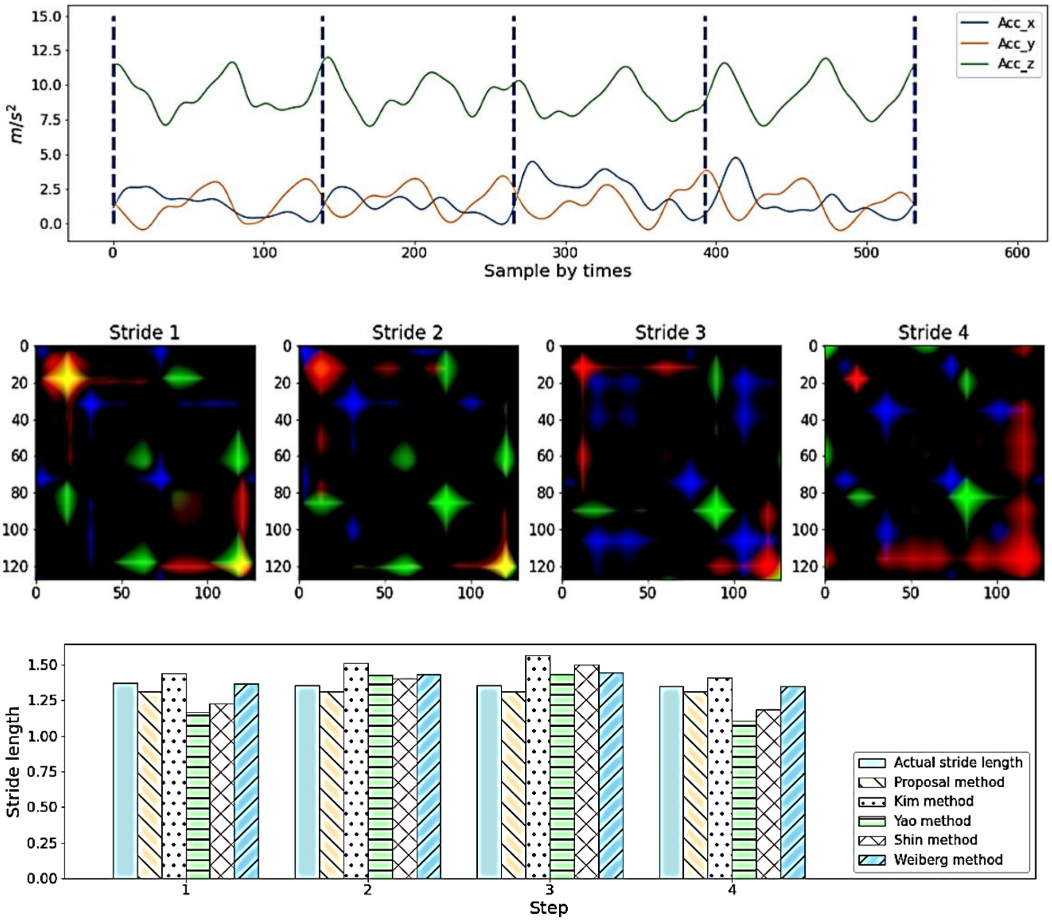
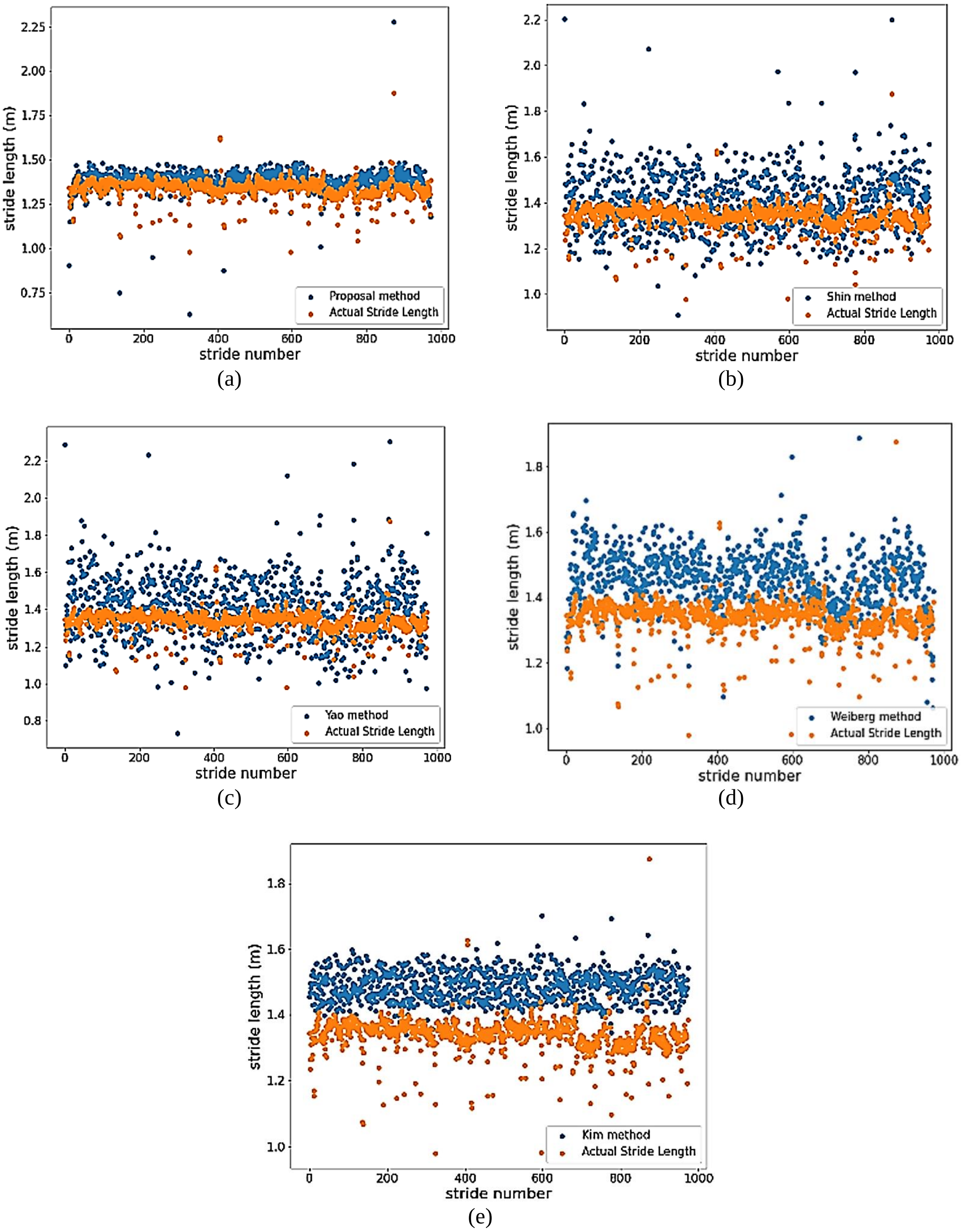


Figure 9. Some strides with raw signal and their image representation and estimated stride-length

The Shin, Yao, Weinberg, and Kim methods are evaluated using the testing dataset prepared earlier which consists of 973 steps. We can clearly see in Figure 10 that Shin and Yao's estimation is scattered around actual value while Weinberg and Kim's method tend to overestimate. Details of the error over the walking distance of our proposed method with others are shown in Table 2. Over the distance of 1300.5799 (m) our proposed method E_s and E_d are only 4.4378% and 3.1756%, which is the smallest among others. This indicates that our model has the best performance while evaluating the error of each stride and over a long distance. From Figure 11, it is obvious that our proposed method achieves 80% of strides with the estimation error under 0.08817 meters, while Kim, Yao, Shin, and Weinberg's are 0.20103, 0.2077, 0.17526, and 0.18179 meters, respectively.



(a)

(b)

(c)

(d)

(e)

Figure 10. Comparison of estimated stride length in other methods and actual stride length: (a) Our proposal method (b) Shin method, (c) Yao method, (d) Weinberg method, (e) Kim method

Table 2. Comparison of proposed method and others

	Real	Proposed method	Shin	Yao	Weinberg	Kim
Distance (m)	1300.5799	1341.8809	1361.4525	1366.5864	1414.1273	1447.1376
Error (m)	-	41.301	60.8726	66.0065	113.5474	146.5577
E_d (%)	-	3.1756%	4.6804%	5.0751%	8.7305%	11.2686%
Stride length error (m)	-	0.05827	0.11502	0.13454	0.12325	0.15202
E_s (%)	-	4.4378%	8.7553%	10.1992%	9.3031%	11.5474%

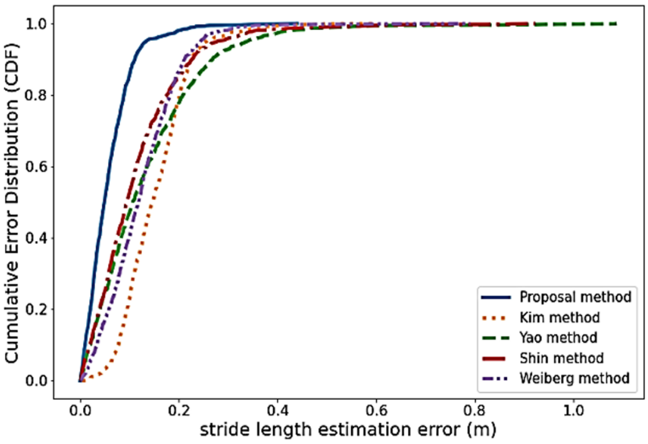


Figure 11. CDF of proposed method and others

4. DISCUSSION

From the indoor positioning perspective, having an accurate estimation of stride length and travel distance opens up new possibilities as many tracking systems rely on SLE. Using our proposed method combined with state-of-the-art techniques for step detection and heading estimation, we can minimize the error during the process and achieve a highly accurate position of the current users. Furthermore, this study could also be used in the field of gait analysis and health monitoring as the stride length of a person is a valuable parameter to predict an impaired gait. The main limitation of our proposed method is that it depends on heavy computation. As the accelerometer data is under the process of conversion from time series to image and passing through the CNN model, it would take a considerable amount of time. This leads to a problem that it is difficult for mobile devices' hardware to handle such an amount of work. A better idea is to place the system in a centralized server to harness the processing power and reduced the load for mobile devices.

In the future, studies can be done on how to reduce the computational time of the proposed method to support real-time tracking applications. The relationship between stride length and data from other sensors like gyroscope and magnetometer could be investigated to further improve accuracy. Finally, the lack of dataset labels for training should also be addressed since inaccurate data could result in the model learning false patterns. Thus, sensor data collecting procedures for stride length need to be rigorously examined so that with special moving patterns, the model can tell the difference between them.

5. CONCLUSION

In this paper, we have proposed a new method for stride length estimation. By utilizing the GAF algorithm, we were able to transform the accelerometer sensor time-series data into images. Then a CNN model was designed to estimate stride length given images as its input. We trained and evaluated the performance of our model using a public dataset created by Qu Wang. Although this dataset did not satisfy our requirements in labeling, it provided us an indicator of how the model performs. Experiments were conducted to measure the performance of our model compared to Kim, Yao, Shin, and Weinberg models. The experimental results show that the proposed method is better than others. Our model achieved 4.4378% in relative stride error and 3.1756% in relative distance error, which is superior compared to the closest methods which are 8.7553%, 4.6804%, respectively.

REFERENCES

[1] A. Salarian *et al.*, “Gait assessment in parkinson’s disease: Toward an ambulatory system for long-term monitoring,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 8, pp. 1434-1443, Aug. 2004, doi: 10.1109/TBME.2004.827933.

[2] L. E. Diez, A. Bahillo, J. Otegui, and T. Otim, “Step length estimation methods based on inertial sensors: A review,” *IEEE Sensors Journal,* vol. 18, no. 17, pp. 6908-6926, Sep. 2018, doi: 10.1109/JSEN.2018.2857502.

[3] S. Miyazaki, “Long-term unrestrained measurement of stride length and walking velocity utilizing a piezoelectric gyroscope,” *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 8, pp. 753-759, Aug. 1997, doi: 10.1109/10.605434.

[4] W. Zijlstra and A. L. Hof, “Assessment of spatio-temporal gait parameters from trunk accelerations during human walking,” *Gait Posture*, vol. 18, no. 2, pp. 1-10, Oct. 2003, doi: 10.1016/S0966-6362(02)00190-X.

[5] H. Weinberg, “Using the ADXL202 in pedometer and personal navigation applications,” *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1-6, 2002.

[6] W. Kang and Y. Han, “SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization,” *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2906-2916, May 2015, doi: 10.1109/JSEN.2014.2382568.

[7] Q. Li, M. Young, V. Naing, and J. M. Donelan, “Walking speed estimation using a shank-mounted inertial measurement unit,” *Journal of Biomechanics,* vol. 43, no. 8, pp. 1640-1643, May 2010, doi: 10.1016/j.jbiomech.2010.01.031.

[8] A. Kose, A. Cereatti, and U. D. Croce, “Bilateral step length estimation using a single inertial measurement unit attached to the pelvis,” *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, pp. 9, 2012, doi: 10.1186/1743-0003-9-9.

[9] J. R. Rebula, L. V. Ojeda, P. G. Adamczyk, and A. D. Kuo, “Measurement of foot placement and its variability with inertial sensors,” *Gait Posture*, vol. 38, no. 4, pp. 974-980, Sep. 2013, doi: 10.1016/j.gaitpost.2013.05.012.

[10] A. Ferrari, P. Ginis, M. Hardegger, F. Casamassima, L. Rocchi, and L. Chiari, “A mobile kalman-filter based solution for the real-time estimation of spatio-temporal gait parameters,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 24, no. 7, pp. 764-773, Jul. 2016, doi: 10.1109/TNSRE.2015.2457511.

[11] D. Trojaniello *et al.,* “Estimation of step-by-step spatio-temporal parameters of normal and impaired gait using shank-mounted magneto-inertial sensors: application to elderly, hemiparetic, parkinsonian and choreic gait,” *Journal of NeuroEngineering and Rehabilitation,* vol. 11, no. 1, pp. 152, 2014, doi: 10.1186/1743-0003-11-152.

[12] A. Rampp, J. Barth, S. Schuelein, K.-G. Gassmann, J. Klucken, and B. M. Eskofier, “Inertial sensor-based stride parameter calculation from gait sequences in geriatric patients,” *IEEE Transactions on Biomedical Engineering,* vol. 62, no. 4, pp. 1089-1097, Apr. 2015, doi: 10.1109/TBME.2014.2368211.

[13] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, “A step, stride and heading determination for the pedestrian navigation system,” *Journal of GPS,* vol. 3, no. 1 & 2, pp. 273-279, Dec. 2004, doi: 10.5081/jgps.3.1.273.

[14] L. Robert W., J. Thomas , “Dead reckoning navigational system using accelerometer to measure foot impacts” U.S.Patent 5 583 776A, Dec. 10, 1996

[15] Z. Sun, X. Mao, W. Tian, and X. Zhang, “Activity classification and dead reckoning for pedestrian navigation with wearable sensors,” *Measurement Science Technology,* vol. 20, no. 1, pp. 015203, Jan. 2009, doi: 10.1088/0957-0233/20/1/015203.

[16] Y. Yao, L. Pan, W. Fen, X. Xu, X. Liang, and X. Xu, “A robust step detection and stride length estimation for pedestrian dead reckoning using a smartphone,” *IEEE Sensors Journal,* vol. 20, no. 17, pp. 9685-9697, Sep. 2020, doi: 10.1109/JSEN.2020.2989865.

[17] S. H. Shin, C. G. Park, J. W. Kim, H. S. Hong, and J. M. Lee, “Adaptive step length estimation algorithm using low-cost MEMS inertial sensors,” in *2007 IEEE Sensors Applications Symposium*, San Diego, CA, USA, Feb. 2007, pp. 1–5. doi: 10.1109/SAS.2007.374406.

[18] Q. Ladetto, “On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering,” *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000),* Salt Lake City, UT, September 2000, pp. 1735-1740

[19] S. Zihajehzadeh and E. J. Park, “A Gaussian process regression model for walking speed estimation using a head-worn IMU,” in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC),* Seogwipo, Jul. 2017, pp. 2345-2348. doi: 10.1109/EMBC.2017.8037326.

[20] Y. Song, S. Shin, S. Kim, D. Lee, and K. H. Lee, “Speed estimation from a tri-axial accelerometer using neural networks,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* Lyon, France, Aug. 2007, pp. 3224-3227. doi: 10.1109/IEMBS.2007.4353016.

[21] J. Hannink *et al.,* “Mobile stride length estimation with deep convolutional neural networks,” *IEEE J. Biomed. Health Inform.*, vol. 22, no. 2, pp. 354-362, Mar. 2018, doi: 10.1109/JBHI.2017.2679486.

[22] F. Gu, K. Khoshelham, C. Yu, and J. Shang, “Accurate step length estimation for pedestrian dead reckoning localization using stacked autoencoders,” *IEEE Transactions on Instrumentation and Measurement,* vol. 68, no. 8, pp. 2705-2713, Aug. 2019, doi: 10.1109/TIM.2018.2871808.

[23] P. D. Tinh and T. T. N. Mai, “Mobile indoor positioning system utilizing WiFi RSSI and motion data,” in *2020 4th International Conference on Recent Advances in Signal Processing, Telecommunications and Computing (SigTelCom),* Hanoi, Vietnam, Aug. 2020, pp. 98-102. doi: 10.1109/SigTelCom49868.2020.9199021.

[24] D. T. Pham and T. T. Ngoc Mai, “Ensemble learning model for Wifi indoor positioning systems,” *IJ-AI*, vol. 10, no. 1, pp. 200, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp200-206.

[25] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and Y. Huang, “Pedestrian stride-length estimation based on LSTM and denoising autoencoders,” *Sensors*, vol. 19, no. 4, pp. 840, Feb. 2019, doi: 10.3390/s19040840.

[26] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation,” arXiv:1506.00327 [cs, stat], May 2015, Accessed: Aug. 06, 2021. [Online]. Available: http://arxiv.org/abs/1506.00327.

[27] H. Lee, K. Yang, N. Kim, and C. R. Ahn, “Detecting excessive load-carrying tasks using a deep learning network with a gramian angular field,” *Automation in Construction*, vol. 120, Dec. 2020, Art. No. 103390 doi: 10.1016/j.autcon.2020.103390.

[28] H. Xu *et al*., “Human activity recognition based on gramian angular field and deep convolutional neural network,” *IEEE Access*, vol. 8, pp. 199393-199405, 2020, doi: 10.1109/ACCESS.2020.3032699.

BIOGRAPHIES OF AUTHORS

Pham Doan Tinh is a Lecturer at the Department of Electronics and Computer Engineering, School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam. In 2011, he earned Doctor of Engineering Degree from Ritsumeikan University, Japan. He also obtained Bachelor and Master Degree in Electronics and Computer Engineering from Hanoi University of Science and Technology (Vietnam) in 1996 and 2000, respectively.

Bui Huy Hoang is a Student in Electronics and Computer Engineering Major of Hanoi University of Science and Technology (Vietnam). His research interests are in fields of Machine Learning Applications and Signal Processing.

Nguyen Duc Cuong is a Student in Electronics and Computer Engineering Major of Hanoi University of Science and Technology (Vietnam). His research interests are in fields of Machine Learning Applications and Image Processing.