International Journal of Innovative Computing, Information and Control
Volume **16**, Number **2**, April **2020**
ICIC International ©2020 ISSN 1349-4198
pp. **609–619**

CLASSIFICATION OF HUMAN GAIT ACCELERATION DATA USING CONVOLUTIONAL NEURAL NETWORKS

Daniel Kreuter[1,2], Hirotaka Takahashi[1], Yuto Omae[3]
Takuma Akiduki[4] and Zhong Zhang[4]

[1]Department of Information and Management Systems Engineering
Nagaoka University of Technology
1603-1, Kamitomioka, Nagaoka, Niigata 940-2188, Japan
hirotaka@kjs.nagaokaut.ac.jp

[2]Department of Physics
Technische Universität Darmstadt
Hochschulstraße 12, 64289 Darmstadt, Germany
daniel.kreuter@stud.tu-darmstadt.de

[3]Department of Industrial Engineering and Management
College of Industrial Technology
Nihon University
1-2-1, Izumi, Narashino, Chiba 275-8575, Japan

[4]Department of Mechanical Engineering
Toyohashi University of Technology
1-1, Hibarigaoka, Tenpakucho, Toyohashi, Aichi 441-8580, Japan

Received August 2019; revised December 2019

Abstract. *The human motion analysis using wearable sensors such as accelerometers and gyroscopes is one of the important issues in ubiquitous and wearable computing. Inspired by a paper by Akiduki et al. that was released in 2018 concerning the classification of human gait motion accelerometer data, this paper attempts to classify that same data using a convolutional neural network. In the original 2018 paper, a high degree of separation was found between the data of the 13 recorded test subjects, suggesting that classification purely by looking at the motion data is possible. For the purpose of classification using the neural network, the given time series data is converted into three matrices (equivalent to image data with three channels per pixel). Using these images as input for a convolutional neural network, an accuracy of 100% was achieved in classifying the subject number from previously unseen motion data.*
Keywords: Time series data, Human motion, Machine learning, Time series imaging

1. Introduction.

The human motion analysis generally includes the classification and/or characterization of movements of any particular individual from the data. Akiduki et al. [1, 2] have discussed the method of the classification and characterization of human motion data. In the paper, accelerometer data from human gait motion is recorded by placing four acceleration sensors on each of their 13 test subjects. The four sensors are worn on the right lower leg, left thigh, lower back and left forearm on a subject, respectively.

The method of Singular Value Decomposition (SVD) was used for investigation of the data (see also Mishima et al. [3] and Kamio et al. [4]). In this method, linear algebra is used to construct a data matrix out of the collected data and to extract intrinsic features of that matrix. This was done in order to separate features that are similar between

DOI: 10.24507/ijicic.16.02.609

each subject and features that differentiate the subjects. Although insightful information could be gained from this method, nowadays a more reliable and convenient way of classifying labeled data is the use of neural networks. Neural networks and specifically convolutional neural networks use machine learning algorithms to learn intrinsic features of a labeled dataset and carry over that learned knowledge in order to make predictions about previously unseen data [5]. The signal obtained from one axis of the accelerometer sensors from the human gait experiments can be seen as a series of one dimensional values recorded at different points in time, a so-called time series. While classification algorithms for time series data do exist (for example, various forms of dynamic time warping [6, 7]), a particular successful use for neural networks has been found in image/video recognition [8].

In this paper, a method for transforming time series data into images, proposed by Wang and Oates [9] in 2015, is employed in order to extract inherent features of the data and harness the power of modern computer vision techniques. This method involves converting the data into three separate matrices: the Gramian Angular Sum Field (GASF), the Gramian Angular Difference Field (GADF) and the Markov Transition Field (MTF). The GASF and GADF matrices provide deeper insight into temporal relations between data points while the MTF matrix obtains information about the dynamic properties of the data (i.e., transition probabilities). These images can then be used to classify the human gait acceleration data using a convolutional neural network. The aim of the network is to be able to assign previously unseen motion data to its subject of origin with high accuracy. After reliably assigning motion data to its subject of origin, future motivation for this research is to be able to identify the differences between each subject's walking motion. The ultimate goal of human motion data analysis should be to investigate how humans move and what about a person's movement characterizes them and differentiates them from someone else.

The paper is structured into a mathematical introduction to the GASF, GADF and MTF methodology followed by an overview of the data collection process and a discussion of the neural network implemented to classify the human gait acceleration data. A discussion of the results followed by a concluding summary closes the paper.

2. **Time Series Imaging Approach.**

2.1. **Gramian Angular Field.**

The concept behind the Gramian Angular Field as proposed by Wang and Oates [9] is motivated by the Gramian matrix and a bijective polar coordinate transformation.

For a bilinear form $\langle\cdot,\cdot\rangle$ on a Euclidian vector space V with vectors $(v_1,\dots,v_n)\in V$ the Gramian matrix is given by

$$\begin{aligned} &G_{ij} = \langle v_i, v_j\rangle \\ \Rightarrow\ &G = \begin{pmatrix} \langle v_1, v_1\rangle & \cdots & \langle v_1, v_n\rangle \\ \vdots & \ddots & \vdots \\ \langle v_n, v_1\rangle & \cdots & \langle v_n, v_n\rangle \end{pmatrix}. \end{aligned} \tag{1}$$

The Gramian matrix provides information about the linear dependency of the v_i since $\det G(v_1,\dots,v_n) > 1$ implies that the vectors are linearly independent [10].

Given time series data with time series of the form $X = \{x_1, x_2, \dots, x_n\}$, one can rescale the time series to the interval $[0,1]$ by

$$\tilde{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)}, \tag{2}$$

and bijectively map it onto the polar coordinate space by

$$\begin{cases} \phi_i = \arccos(\tilde{x}_i) \text{ for } \tilde{x}_i \in \tilde{X}, \\ r_i = \dfrac{t_i}{N} \text{ for } t_i \in \mathbb{N}, \end{cases} \tag{3}$$

where $N \in \mathbb{R}$ is a normalization constant and t_i are the time stamps of the time series. This map is bijective due to the monotony of the inverse cosine function on the interval $[0, 1]$, i.e., the function values $\left[0, \frac{\pi}{2}\right]$. Inspired by the Gramian matrix of a bilinear form two Gram-like matrices, the Gramian Angular Summation Field and the Gramian Angular Difference Field, with an angle-operation on the newly obtained polar coordinate representation of the data are proposed:

$$\mathrm{GASF}_{ij} = \cos(\phi_i + \phi_j) = \langle \tilde{x}_i, \tilde{x}_j \rangle - \sqrt{1 - \tilde{x}_i^2} \times \sqrt{1 - \tilde{x}_j^2}, \tag{4}$$

$$\mathrm{GADF}_{ij} = \sin(\phi_i - \phi_j) = \tilde{x}_j \sqrt{1 - \tilde{x}_i^2} - \tilde{x}_i \sqrt{1 - \tilde{x}_j^2}. \tag{5}$$

From the representation in Cartesian coordinates, it is evident that the two sum and difference operations do not make bilinear forms since they do not fulfill the linearity condition. However, the proposed matrices do preserve temporal relations of the data and increase sparsity compared to using a regular Gramian matrix with the dot product. To control the size of the matrices, Piecewise Aggregation Approximation [11] is applied to the time series before the conversion.

2.2. **Markov Transition Field.** Besides the two Gram-like matrices, a third matrix representation for time series data is proposed. In order to represent the dynamic features of the time series data and yet still preserve temporal dependencies, an extension to a Markov matrix is presented: the Markov Transition Field. By splitting a time series up into quantile bins (the number of bins shall be denoted as Q), transitions between bins can be counted and used as a measure for the dynamical movement of the data over time.

Thus, in a time series X, each $x_i \in X$ can be assigned into one quantile bin q_j with $j \in [1, Q]$. By treating the time series as a first order Markov chain, a $Q \times Q$ transition probability matrix between the bins can be constructed:

$$W = \begin{pmatrix} w_{11} & \cdots & w_{1Q} \\ \vdots & \ddots & \vdots \\ w_{Q1} & \cdots & w_{QQ} \end{pmatrix}, \tag{6}$$

where w_{nm} denotes the transition frequency from bin q_n to bin q_m. After normalization by $\sum_j w_{ij} = 1$, W is called the Markov Transition Matrix. To preserve temporal dependencies in the time series, these transition frequencies are noted between every entry in the time series, thus resulting in the Markov Transition Field:

$$\mathrm{MTF} = \begin{pmatrix} w_{ij}|_{x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij}|_{x_1 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij}|_{x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij}|_{x_n \in q_i, x_n \in q_j} \end{pmatrix}. \tag{7}$$

The MTF includes the transition probabilities between every time series entry to itself (main diagonal) and to every other entry. To control the size of this matrix, the blurring kernel $\left(\frac{1}{m^2}\right)_{m \times m}$ for an $m \times m$ patch of pixels is used.

3. **Overview of the Data Collection and Preparation.** The accelerometer data was collected from 13 subjects including 10 men and 3 women aged (24.3 ± 4.3) years. The

contents of the experiment were explained to the subjects and consent from each subject to use the obtained data for research purposes was obtained.

Each subject wore a motion capture suit with 4 accelerometer modules (WAA-010, ATR-Promotions Inc.) to measure the walking motion data attached to it. The positions of these sensors as well as the used coordinate system are shown in Figure 1. The positions of the four sensors are as follows: S1 at the right lower leg, S2 at the left thigh, S3 at the lower back and S4 at the left forearm. The subjects were instructed to walk along a 15m straight line. The test was performed 5 times with varying conditions. The subjects were instructed to walk the course twice at normal speed (cond. $N_{1\mathrm{st}}, N_{2\mathrm{nd}}$), then once at slow speed (cond. S) followed by once at fast speed (cond. F) and finally at normal speed again (cond. $N_{3\mathrm{rd}}$). In this paper only data from the normal speed conditions (i.e., $N_{1\mathrm{st}}, N_{2\mathrm{nd}}, N_{3\mathrm{rd}}$) is used.

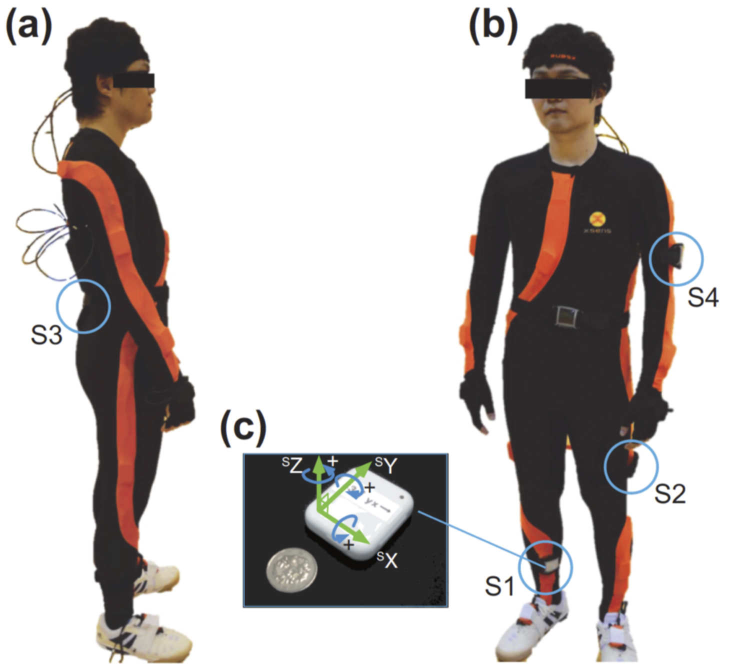


FIGURE 1. Wearable accelerometer modules. (a) and (b) show sensor setting on a subject's body, and (c) shows the accelerometer module and its coordinate system. Description and image are taken from [1].

The motions of a subject can be collected as acceleration along with three-axis of local coordinate on the sensor module: $\left({}^{S}X,\ {}^{S}Y,\ {}^{S}Z\right)$ shown in Figure 1(c). All signals from the modules were sampled at 100 Hz and sent to the host computer from each sensor module via Bluetooth. To identify human motion based on acceleration data, it is known that it is necessary to examine the motion up to around 10 Hz, i.e., it suggests that higher frequency bands are not necessarily required [12]. Then, to remove high-frequency noise, the signals were filtered by 3rd-order Butterworth LP filter with a cut-off frequency of 12.5 Hz [13]. Finally, time series data corresponding to each gait cycle was clipped from the walking data. The time series for about 20 gait cycles per each subject (Max. 27 cycles, Min. 18 cycles) were clipped from the whole walking data of each subject. Since the length of one gait cycle was different for each subject, the clipped time series have even lengths between by processing through an interpolation algorithm. In this paper, the cubic spline algorithm was used. Moreover, we rescaled the time series to the interval $[-1, 1]$. As the result, we obtained the (4 segments) $\times$ (3 axis) accelerometer data and

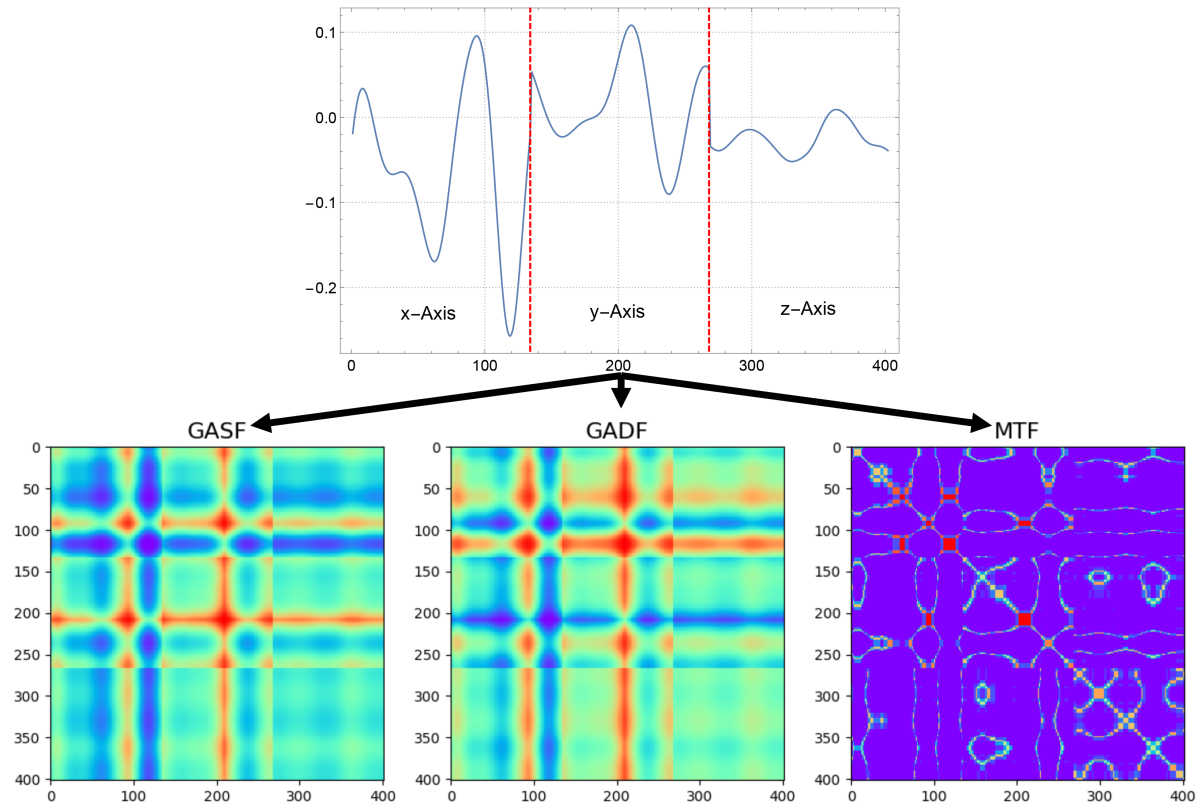


FIGURE 2. Example of a gait cycle signal from sensor S2 being converted into the GASF, GADF and MTF matrices. The gait cycle signal includes measurements from all axes of the sensor in a row. The matrix size is the full resolution (402×402). The quantile size is $Q = 20$.

the length of clipped time series was 134. The number of gait cycles of all subject was 281. An example of one gait cycle data from S2 is shown in the top of Figure 2.

More details of the preparation of the data were explained in [1].

4. **Convolutional Neural Network and Training Setup.** In order to classify the acceleration data by test subject, a convolutional neural network was used. The classification classes in this case were to be the 13 different test subjects whose motion data was recorded.

The motion data is given as a $281 \times (134 \times 4 \times 3)$ matrix D [1]. Each column in the matrix D is called a *gait frame vector* [1]. Each gait frame vector includes acceleration data of one gait cycle (134 data entries) for every sensor (4 sensors) and axis (3 axes per sensor). Hence the length of one such gait frame vector results to $134 \times 4 \times 3 = 1608$.

For the {GASF, GADF, MTF} approach, it was decided that higher resolution matrices would be beneficial for classifying the given motion data. A higher resolution matrix contains more information for the neural network to learn. Similarly, since one gait frame vector contains the signals of all sensors and all axes in a row (12 separate measured datasets joined in one time series signal), reducing the matrix size would entail a blurring over the "edge" of multiple independent sets of data and effectively distorting the true information that was measured. Since computation time too was a decisive topic to think about, it was suggested to use the data of only one sensor instead of all 4 for classification. This reduces the length of one considered gait frame vector from 1608 to 402. For this paper, sensor S2 was chosen since its position on the thigh of the subject results in large movement during one gait cycle (see Figure 1).

The conversion from the one gait cycle data to the GASF, GADF and MTF matrices is shown in Figure 2.

Using only the S2 data, the dimensions of the input for the convolutional neural network become $(281 \times 402 \times 402 \times 3)$.

The structure of the neural network was largely adopted from a GitHub repository by user *kiss90* [14]. Changes include a different input size, introduction of batch normalization, introduction of dropout layers and optimizations such as changing the pooling layers and the optimizer. A schematic showing the layering structure of the network is displayed in Figure 3. The structure resembles a fairly regular convolutional neural network. It employs three convolution blocks consisting of a convolution layer, a batch normalization layer, a maximum pooling layer and a ReLU-activation layer. In addition, one fully-connected layer is placed before the final class identification layer. Dropout layers after the first convolution block and just before the identification layer have been added as well in order to improve network generalization. For information on the layers' parameters, please refer to Figure 3. The network was constructed and trained using the Keras deep learning library [15] running on top of TensorFlow [16] in Python. Since the project by GitHub user *kiss90* was entirely written in Caffe [17], only the structure was used as an inspiration for the network while none of the code was taken.

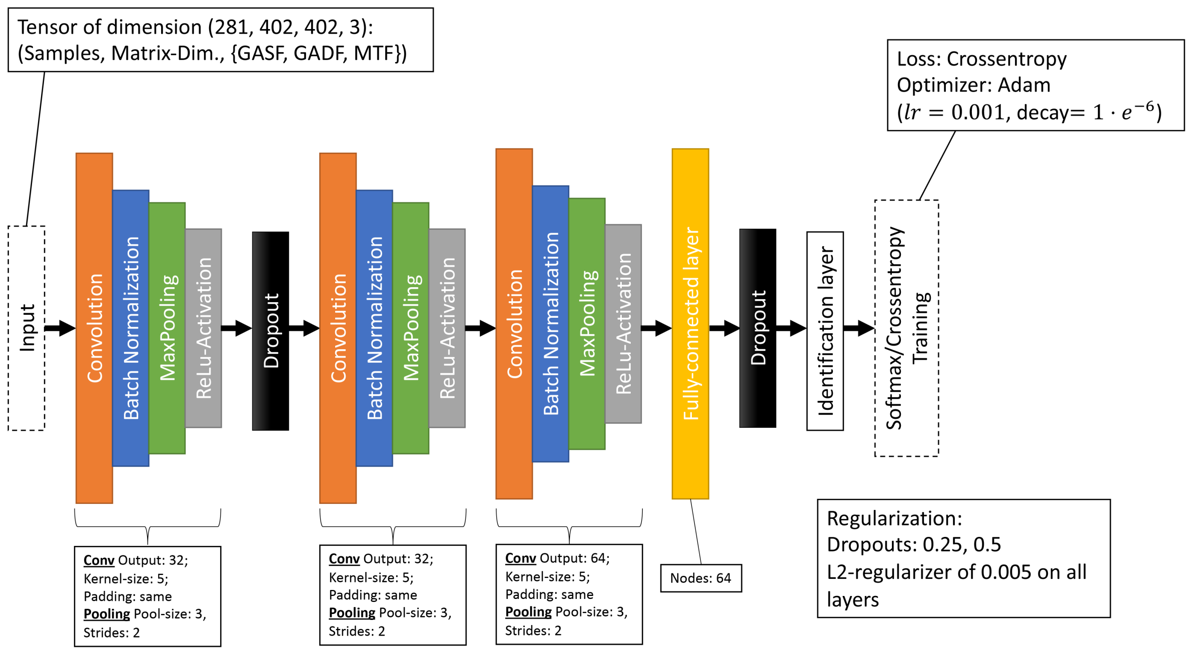
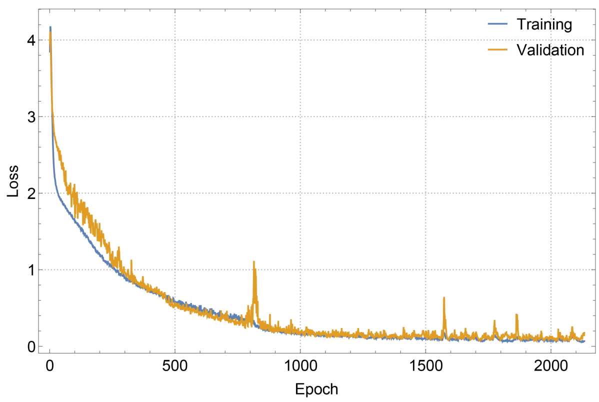


FIGURE 3. Structure of the convolutional neural network used for classification of the acceleration data. The network was constructed using the Keras [15] API with the TensorFlow backend [16].

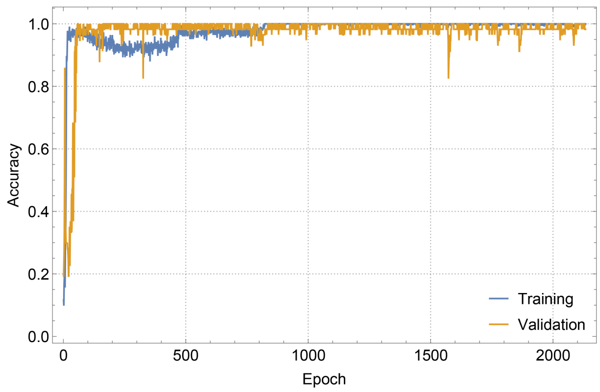
To train the final network configuration, the data was randomly split into training and validation sets with a ratio of $4 : 1$, resulting in 224 training samples and 57 validation samples. Adjustable hyperparameters for this network (learning rate, pooling strides, etc.) were chosen using 5-fold cross validation. An interesting hyperparameter of this particular network is the quantile size Q for the MTF matrix. Different sizes $Q \in \{5, 10, 20, 40, 60\}$ were tested to see its effect on network performance. Higher values for Q make for an

MTF matrix with very little structure while very low values for Q end up making the important features of the network stand out less. After some testing it was decided to use $Q = 20$.

An example of the training progress can be seen in Figure 4. Since the two curves remain very closely together in both graphs, the network seems to learn intrinsic features of the data rather than overfitting. The maximum epochs of the training are set to 10,000 with an earlystop mechanism built in which stops the training should the validation-loss not improve after 200 epochs. After the earlystop triggers, the best weights (i.e., those with the lowest validation-loss score) are restored.



(a) Training loss vs. validation loss



(b) Training accuracy vs. validation accuracy

FIGURE 4. Curves displaying the progression of the two metrics, loss function value and accuracy, during training of the neural network. The fitting to the training data is shown in the light blue curves (thin lines) while the evaluated performance on the validation data is shown in the orange curves (thick lines). The loss function is categorical cross-entropy [18].

During this training the neural network learns how to classify the given data by itself without providing intuitive information as to how this feat is achieved. Because of this, the classification criteria for each data point are far less transparent than in papers using the SVD approach [1, 4].

5. **Results and Discussion.** The neural network fitting function is set to show five different metrics of network performance: Loss function value, Accuracy, Precision, Recall and F1-Score (for more explanation see Chinchor [19]). This is done to have a wider set of information on the network's performance. The actual training on the network is of course set to minimizing only the Loss function value. The network's performance after training for roughly 2000 epochs on the motion data is shown in Table 1. For comparison, we also show the results of using GASF, GADF and MTF only as input data in Table 1.

As shown by Figure 4(b) as well as Table 1, the convolutional neural network achieves great classification results on the available data. It also seems to not make much of a difference if all three matrices (GASF, GADF and MTF) are used simultaneously or on their own, although using all three matrices should be preferred as they include more information of the data. It should also be noted that while we believe the imaging approach to be an effective way of encoding time series features, other methods of direct time series data input also exist. Particularly in a 2017 paper by Wang et al. [7] other neural network methods which use the direct time series as input without pre-processing seem to be proving effective also. In a quick test of the ResNet used in said paper the

Table 1. Network performance on unseen validation data after training for around 2000 epochs. For comparison, performance of using all three matrices simultaneously as well as alone has been recorded. The metric values are given up to 3 decimal places.

Input	Loss function value	Accuracy	Precision	Recall	F1-Score
{GASF, GADF, MTF}	0.170	1.0	1.0	1.0	1.0
GASF only	0.102	1.0	1.0	1.0	1.0
GADF only	0.090	0.991	0.999	0.799	0.888
MTF only	0.104	0.991	0.999	0.999	0.999

same high accuracy as with our imaging approach was achieved, although it is difficult to judge which one of the two has a higher performance from a test on this one dataset alone.

In order to check whether this was merely a "lucky" validation split multiple tests with different random splits have been performed and showed the same result. This validates the initial hypothesis that data from sensor S2 on the left thigh (see Figure 1) alone is enough to distinguish different test subjects by their accelerometer motion data. This result is also consistent with the conclusion of Akiduki et al. [1, 20] and Kamio et al. [4] that differences between subjects can be found purely from their gait accelerometer data. In [1], the highest degree of class separation was found for sensor S3 (positioned on the lower back of the subject), suggesting that sensor S3 most clearly differentiates the gait motions of the subjects. For this paper, data from sensor S2 (positioned on the left thigh of the subject) was used in order to classify the data by subject. Since the results already successfully classify the data, attempts using data of other sensors were not made. The position of both sensor S2 and sensor S3 suggests however that the core muscle area of the body (abdominal, pectoral, back and gluteal muscles) plays a crucial role in motion identification.

To compare further with the previously mentioned papers, Figure 5 shows the filter output of the network after the first convolution layer. Since the GASF, GADF and MTF images themselves are already encoded the filtered images too do not show an intuitive pattern for humans. However, it is noticeable that the network seems to develop filters that either focus on the Gramian matrices (e.g., row 1, image 3) or on the MTF matrix (e.g., row 3, image 2). This is in line with the notion that the Gramian matrices encode time series features together, whereas the MTF matrix encodes dynamic time series features by itself. Secondly, Figure 5 also contains some filters that are blank (e.g., row 1, image 2). This suggests that the network ended up not needing these filters at all and that a network that is not as deep would have also sufficed to classify the data.

The methodology proposed in this paper also has some limitations. The original measurement of the subjects' walking motion recorded three different walking speeds per subject: normal speed, slow speed and fast speed. Like in the previous papers only "normal speed" data is used. Whether the neural network's performance of this paper could be replicated if all speed data was included is unknown as it would introduce vastly different time lengths per gait frame. Other than changing the walking speed other environmental variables of the experiments could also be introduced (e.g., rocky path or stairs instead of flat path, sports shoes vs. hiking boots, slope of the path). These variables' influence on network performance is unknown as no such experiments were conducted. The original purpose of the experiment was to obtain accelerometer motion data from humans in order to investigate the obtained signals [2]. Many papers have since been written using data from this same experiment (see for example the aforementioned Akiduki et al. [1]

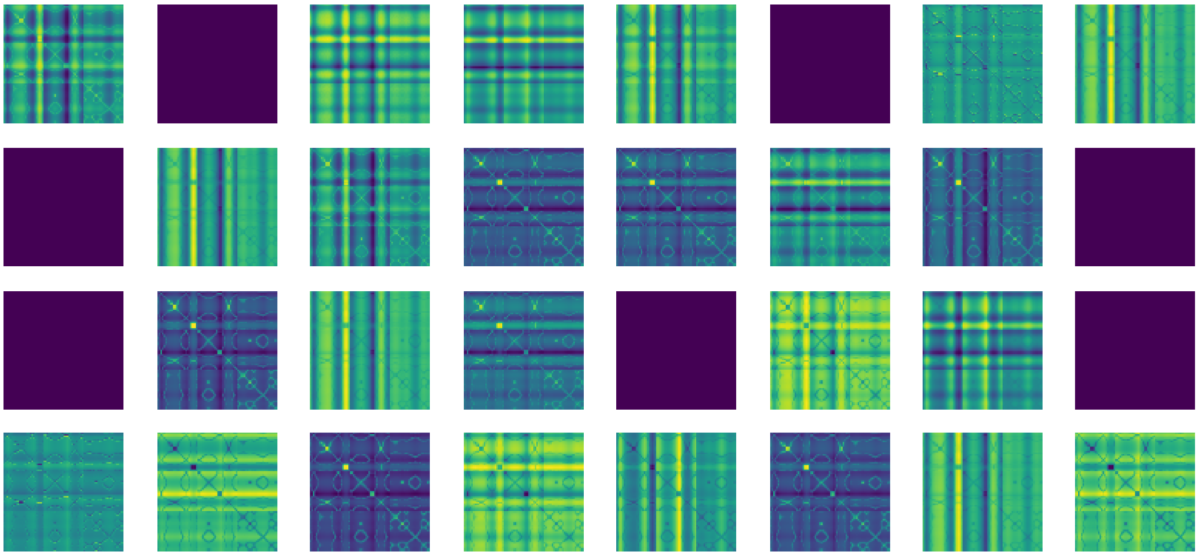


FIGURE 5. Output images of a gait-frame after going through the 32 filters of the first convolution layer. GASF, GADF and MTF are shown as three color-channels of the respective images.

or Kamio et al. [4]). It is however safe to assume that introducing more variables would make classification more difficult. On the other hand, more intrinsic information about the subject's movement features could be extracted and a truer classification might be achieved. Another limitation of the method is that only the 13 subjects on which the network was trained can be identified by their walking data. Previously unseen subjects are not incorporated into the network structure. For this to be achieved the number of subjects (i.e., the number of classes of the neural network) would have to be increased or subjects would have to be represented by body features such as age or physique rather than a single ID-digit. Following this paper, future work could attempt such a body-features based classification. If instead the number of people were to be increased for the current network configuration, (e.g., to 50, 100, 1000) the required parameters of the deep neural network would need to have hyper-linear growth with respect to the class number [21]. This in turn would tremendously increase computation time and make this option not realistically feasible. Extension of this method to other types of movement (e.g., swimming, jumping, tennis) could also be attempted.

6. **Summary.** The goal of this paper was to construct a convolutional neural network in order to classify the human gait acceleration data used in papers such as [1]. The classification classes in this case were to be the 13 different test subjects whose motion data was recorded. In order to classify the data using the neural network, some data pre-processing proposed by Wang and Oates [9] was applied.

From the time series data, three matrices were created: the Gramian Angular Sum Field, the Gramian Angular Difference Field and the Markov Transition Field. The GASF and GADF matrices rely on Gramian matrix properties together with a polar coordinate transformation of the data. The MTF matrix encodes dynamic properties of the data with each entry representing transition probabilities from one quantile bin to another. The size of the matrix was chosen to be the full resolution (402×402) so as to not blur over edges of independent accelerometer sensor datas.

Inspired by a similar GitHub project written in Caffe code a convolutional neural network was constructed using Keras in Python [14]. The network was adapted and some changes were made in order to improve network performance. The hyperparameters of the network were chosen using 5-fold cross validation. Methods against overfitting include Dropout layers, L2-regularization and an earlystop mechanism during training which restores the weights at which the validation-loss was at its minimum.

The performance of the network after training gave outstanding classification results of 100% accuracy on the 57 randomly selected validation samples. It is thus consistent with the conclusions of Akiduki et al. [1] that the degree of separation between the motion data is high enough for different subjects to be identified by their motion data alone. Limitations of this approach such as comparatively small sample size and considering only one walking speed were discussed. While neural networks excel at learning to classify data using gradient descent, the way how the input is classified is relatively nontransparent since it is hidden behind the vast weight-matrix of the network. The network offers concrete classification output; however, it does not provide intrinsic information about the data like the singular value decomposition [1, 4] or the attractor trajectory method [20] can.

The $(402 \times 402 \times 3)$ size of every data input makes the calculation and training of the network somewhat lengthy (on the employed server with two GPUs (GeForce RTX 2080 Ti) around 2 hours). Blurring each individual time series data set (one axis of one sensor of one gait frame vector) could resolve the aforementioned "blurring over edges"-problem and result in smaller, more manageable matrix sizes.

Acknowledgment. This work was supported in part by JSPS Grant-in-Aid for Scientific Research (C) (Grant No. 17K05437; H. Takahashi and Grant No. 16K06156; T. Akiduki). This work was also supported in part by JSPS Grant-in-Aid for Young Scientists (B) (Grant No. 17K13179 Y. Omae) and Young Scientists (Grant No. 19K20062; Y. Omae and Grant No. 19K14924; T. Akiduki).

REFERENCES

[1] T. Akiduki, K. Kawamura, Z. Zhang and H. Takahashi, Extraction and classification of human gait features from acceleration data, *International Journal of Innovative Computing, Information and Control*, vol.14, no.4, pp.1361-1370, 2018.

[2] T. Akiduki, A. Uchida, Z. Zhang, T. Imamura and H. Takahashi, Extraction of human gait feature from acceleration data, *ICIC Express Letters, Part B: Applications*, vol.7, no.3, pp.649-656, 2016.

[3] K. Mishima, S. Kanata, H. Nakanishi, T. Sawaragi and Y. Horiguchi, Extraction of similarities and differences in human behavior using singular value decomposition, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol.J94-A, no.4, pp.293-302, 2011.

[4] I. Kamio, H. Takahashi, T. Akiduki and Z. Zhang, Study of individual characteristics in human motion by using acceleration data, *ICIC Express Letters, Part B: Applications*, vol.7, no.10, pp.2225-2232, 2016.

[5] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, vol.521, pp.436-444, 2015.

[6] Y. S. Jeong, M. K. Jeong and O. A. Omitaomu, Weighted dynamic time warping for time series classification, *Pattern Recognition*, vol.44, no.9, pp.2231-2240, 2011.

[7] Z. Wang, W. Yan and T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, *Proc. of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp.1578-1585, 2017.

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, vol.60, no.6, pp.84-90, 2017.

[9] Z. Wang and T. Oates, Imaging time-series to improve classification and imputation, *Proc. of the 24th International Conference on Artificial Intelligence*, pp.3939-3945, 2015.

[10] G. Fischer, *Lineare Algebra*, Springer Fachmedien Wiesbaden, 2014.

[11] E. J. Keogh and M. J. Pazzani, Scaling up dynamic time warping for datamining applications, *Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.285-289, 2000.

[12] C. V. C. Bouten, K. T. M. Koekkoek, M. Verduin, R. Kodde and J. D. Janssen, A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity, *IEEE Trans. Biomedical Engineering*, vol.4, no.3, pp.136-147, 1997.
[13] M. Tada, F. Naya, R. Ohmura, M. Okada, H. Noma, T. Toriyama and K. Kogure, A method for measuring and analyzing driving behavior using wireless accelerometers, *IEICE Trans. Information and Systems*, vol.J91-D, no.4, pp.1115-1129, 2008.
[14] *"Kiss90", Time-Series-Classification*, https://github.com/kiss90/time-series-classification, Accessed on August 28, 2019.
[15] F. Chollet et al., *Keras*, https://keras.io, Accessed on August 28, 2019.
[16] A. Martin et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, https://www.tensorflow.org/, Accessed on August 28, 2019.
[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev. J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional architecture for fast feature embedding, *arXiv Preprint*, arXiv:1408.5093, 2014.
[18] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
[19] N. Chinchor, MUC-4 evaluation metrics, *Proc. of the 4th Conference on Message Understanding*, pp.22-29, 1992.
[20] T. Akiduki, Z. Zhang and H. Takahashi, Feature extraction for gait identification by using trajectory attractors, *ICIC Express Letters*, vol.13, no.6, pp.529-538, 2019.
[21] Q. Zhang, K.-C. Lee, H. Bao, Y. You, W. Li and D. Guo, Large scale classification in deep neural network with label mapping, *Proc. of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp.1134-1143, 2018.