



Research article

Features gradient-based signals selection algorithm of linear complexity for convolutional neural networks

Yuto Omae^{1,*}, Yusuke Sakai² and Hirotaka Takahashi²

¹ College of Industrial Technology, Nihon University, 1-2-1, Izumi, Narashino, Chiba 275-8575, Japan

² Research Center for Space Science, Advanced Research Laboratories and Department of Design and Data Science, Tokyo City University, Kanagawa 224-8551, Japan

* **Correspondence:** Email: oomae.yuuto@nihon-u.ac.jp.

Abstract: Recently, convolutional neural networks (CNNs) for classification by time domain data of multi-signals have been developed. Although some signals are important for correct classification, others are not. The calculation, memory, and data collection costs increase when data that include unimportant signals for classification are taken as the CNN input layer. Therefore, identifying and eliminating non-important signals from the input layer are important. In this study, we proposed a features gradient-based signals selection algorithm (FG-SSA), which can be used for finding and removing non-important signals for classification by utilizing features gradient obtained by the process of gradient-weighted class activation mapping (grad-CAM). When we defined n_s as the number of signals, the computational complexity of FG-SSA is the linear time $\mathcal{O}(n_s)$ (i.e., it has a low calculation cost). We verified the effectiveness of the algorithm using the OPPORTUNITY dataset, which is an open dataset comprising of acceleration signals of human activities. In addition, we checked the average of 6.55 signals from a total of 15 signals (five triaxial sensors) that were removed by FG-SSA while maintaining high generalization scores of classification. Therefore, FG-SSA can find and remove signals that are not important for CNN-based classification. In the process of FG-SSA, the degree of influence of each signal on each class estimation is quantified. Therefore, it is possible to visually determine which signal is effective and which is not for class estimation. FG-SSA is a white-box signal selection algorithm because it can understand why the signal was selected. The existing method, Bayesian optimization, was also able to find superior signal sets, but the computational cost was approximately three times greater than that of FG-SSA. We consider FG-SSA to be a low-computational-cost algorithm.

Keywords: convolutional neural network; signal importance; signal selection algorithm; time-directional Grad-CAM

Mathematics Subject Classification: 68T01, 68T07, 68T20

1. Introduction

Recently, many convolutional neural networks (CNNs) have been developed for classification in the time domain of signals, e.g., recognizing movement-intention using electroencephalography (EEG) signals [1], human activity recognition using inertial signals [2, 3], and swimming stroke detection using acceleration signals [4]. Although these studies used all signals, signals that are not important for correct classification may be included in the CNN input layer. These signals worsen the accuracy of the classification and increase calculation costs, required memory, and data collection costs. Therefore, finding and removing non-important signals from the CNN input layer and creating a classification model for the minimum signals possible are crucial. However, many CNNs use all signals [4–6] or manually select signals [7, 8].

We can visually find unimportant regions using gradient-weighted class activation mapping (grad-CAM) embedded in CNNs. Grad-CAM [9] uses the gradient of the feature maps output from the convolutional layer immediately before the fully connected layer to visualize which regions on the image influenced class estimation in the form of a heat map. For example, Figure 1 (A) is the generation process of grad-CAM when the image of 2 is input to CNN. First, the gradient of the output vector with respect to the feature maps is computed. Then, the feature maps are weighted by the sum of the gradients and they are stacked. A heat map is generated by passing the weighted feature map through the ReLU function. Considering this, we can understand that the CNN used the upper side of the image to estimate “2”. Grad-CAM was proposed by improving CAM [10]. CAM is the white-box method for only CNNs, which do not have fully connected layers. In contrast, Grad-CAM is applicable to various CNN model-families [9]. The method is primarily used to input image data, e.g., chest CT images [11], 3D positron emission tomography images [12] and chest X-ray images [13–15].

Cases of applying grad-CAM to CNNs that input time domain data of signals, as shown in Figure 1 (B), are continuously increasing. For example, the classification of sleep stages [16], prognostication of comatose patients after cardiac arrest [17], classification of schizophrenia and healthy participants [18], and classification of motor imagery [19] are performed using EEG signal(s). Electrocardiogram (ECG) signals are used to detect nocturnal hypoglycemia [20] and predict 1-year mortality [21]. Acceleration signals have been used to detect hemiplegic gait [22] and human activity recognition [23].

From these studies, we visually find unimportant signals for correct classification by applying grad-CAM to CNN, inputting the time domain of the signals. In the example shown in Figure 1 (B), signals s_1 and s_n are strongly activated, but signal s_2 is not. Therefore, s_2 can be interpreted as not important for class estimation. However, observing all of them and finding unimportant signals is extremely difficult because one grad-CAM for one input data is generated. Therefore, we propose features gradient-based signals selection algorithm (FG-SSA), which can be used to find and remove unimportant signals from a CNN input layer by utilizing features gradient obtained by the calculation process of grad-CAM. The algorithm provides a signal subset consisting of only important signals. The computational complexity of the proposed algorithm is linear order $O(n_s)$ (i.e., FG-SSA has a low calculation cost) when we define n_s as the number of all signals:

The academic contributions of this study are twofold, which are specified as follows.

- We propose a method for quantifying the effect of signals on classes estimation.
- We propose an algorithm to remove signals of low importance.

This algorithm can remove unnecessary signals and enable CNNs and datasets to be smaller in size.

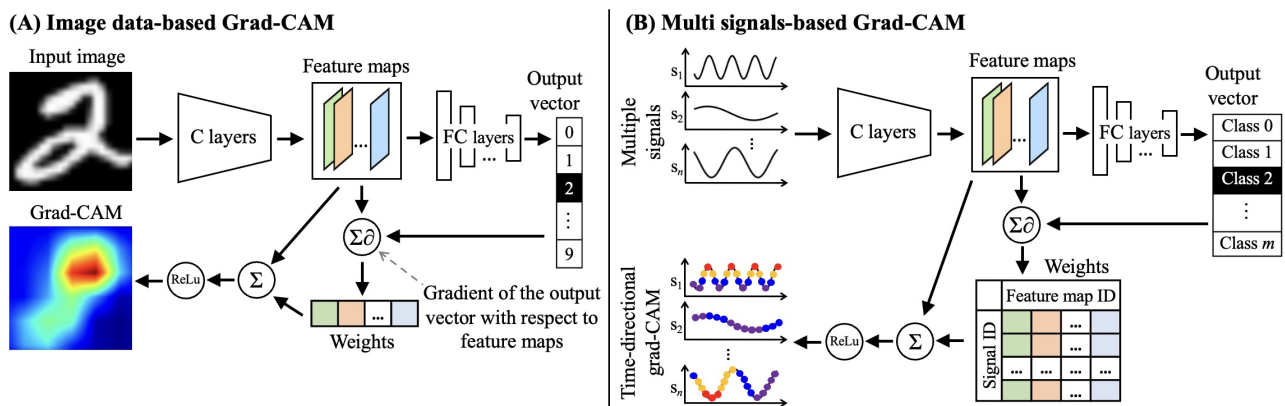


Figure 1. Schematic diagram of (A) image data-based grad-CAM and (B) multi signals-based grad-CAM. “C” and “FC” mean “convolution” and “fully-connected”, respectively. The MNIST dataset is used for the figure (A). The red color represents high activation and the dark blue color represents low activation.

2. Related works

The proposed algorithm, FG-SSA, is applied to CNNs that estimate class labels. It selects signals to be used in the input layer. This is similar to the feature selection problem of which features to use among all the features. The large number of patterns in feature combinations makes it difficult to find a globally optimal solution that maximizes verification performance for a subset of features. For example, a combination of 10 out of 100 features exceeds 10^{14} patterns. Therefore, the feature selection problem is known to be NP-hard [24, 25]. For NP-hard problems, finding a good feasible solution is better than finding a global optimal solution [26]. Therefore, many methods for finding approximate solutions to the feature selection problem have been proposed, e.g., within-class variance between-class variance ratio [27, 28], classification and regression tree [29], Lasso [30], ReliefF [31, 32], out-of-bag error [33], and others. Recently, some studies have used Bayesian optimization for feature selection [34–36], which expresses whether to use a feature as a binary variable and evaluates the combination of features. These feature selection methods reduce the computation time of machine learning and improve the prediction accuracy [37, 38].

The solution to the signal selection problem in CNNs also has a large number of patterns, which is obtained through several signal combinations. However, the feature selection algorithm described above cannot be used for signal selection problems and is not suitable for CNNs. Compared to other machine learning methods (support vector machine, decision tree, k-nearest neighbor, and others), CNNs require more computation time to construct a model. In addition, CNNs have many hyperparameters (number of layers, size of kernel filters, etc.), which also require computational resources for tuning. Therefore, instead of searching for an appropriate signals set, all of the prepared signals are often used such as [39, 40]. Hence, a method that can find an approximate solution at a small computational cost is needed for the feature selection and signal selection problems.

Several studies have attempted to address the signal selection problem. Some studies applied Bayesian optimization to signal selection [41, 42]. Because Bayesian optimization is a method for finding optimal solutions in black box functions, it can be applied to both feature selection and signal selection. However, it is not suitable for CNNs. Another approach is signal selection by group

Lasso [43]. These signal selection algorithms are important research, but they are difficult to visually interpret because they do not utilize Grad-CAM. To the best of the authors' knowledge, only these previous studies involved signal selection. We feel that developing signal selection algorithms for CNNs is an important issue but has not been sufficiently studied. Therefore, we propose a signal selection algorithm, FG-SSA, that provides a visual interpretation of which signals contribute to which class estimation and is computationally inexpensive.

3. Proposed method

3.1. Initial signals set

We consider a situation in which the task is to estimate a class c belonging to class set \mathcal{C} defined by

$$\mathcal{C} = \{c_i \mid i = 1, \dots, n_{\text{cm}}\}, \quad (3.1)$$

where n_{cm} is the number of classes. The measurement data of the initial signal set is defined by

$$\mathcal{S} = \{s_i \mid i = 1, \dots, n_s\}, \quad (3.2)$$

where n_s denotes the number of signals and s_i denotes the signal identification names. Some of the n_s signals are important for classification, whereas others are not. Therefore, finding a signal subset \mathcal{S}_{use} that has removed the unimportant signals from all signal sets \mathcal{S} is crucial. In this study, we provide an algorithm for finding such a subset of signals $\mathcal{S}_{\text{use}} \subseteq \mathcal{S}$. The applicable targets of the proposed method are all tasks of solving classification problems using CNNs by inputting the time domain of multisignals.

3.2. Class estimation

An example of CNN with multiple signals as input is shown in Figure 2. This figure shows the CNN solving the problem of classifying 10 human motion labels from 15 acceleration signals mentioned in Subsection 4.2; however, the proposed algorithm can be applied to various problems. The data for the input layer are in the form of a matrix $w \times n_s$. w is the window length of the sliding window method [44] and $n_s = |\mathcal{S}|$ is the number of signals.

Subsequently, the input data are convoluted using kernel filters of the time-directional convolution size s_c . The number of generated feature maps is n_f because the number of filters is n_f . The reason for adopting only time-directional convolution is to avoid mixing a signal and others in the convolution process. The input data are convoluted by these n_c convolution layers, and the CNN generates the feature maps shown in "extracted feature maps" in Figure 2. Subsequently, the CNN generates the output vector using fully connected layers and the SoftMax function. We define the output vector \mathbf{y} as

$$\mathbf{y} = [y_c] \in \mathbb{R}^{|\mathcal{C}|}, \quad c \in \mathcal{C}, \quad \sum_{c \in \mathcal{C}} y_c = 1. \quad (3.3)$$

The estimation class c' is

$$c' = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \{y_c \mid c \in \mathcal{C}\}. \quad (3.4)$$

Herein, we only define the output vector \mathbf{y} because it appears in the grad-CAM definition. We explain grad-CAM for time domain signals in the next subsection.

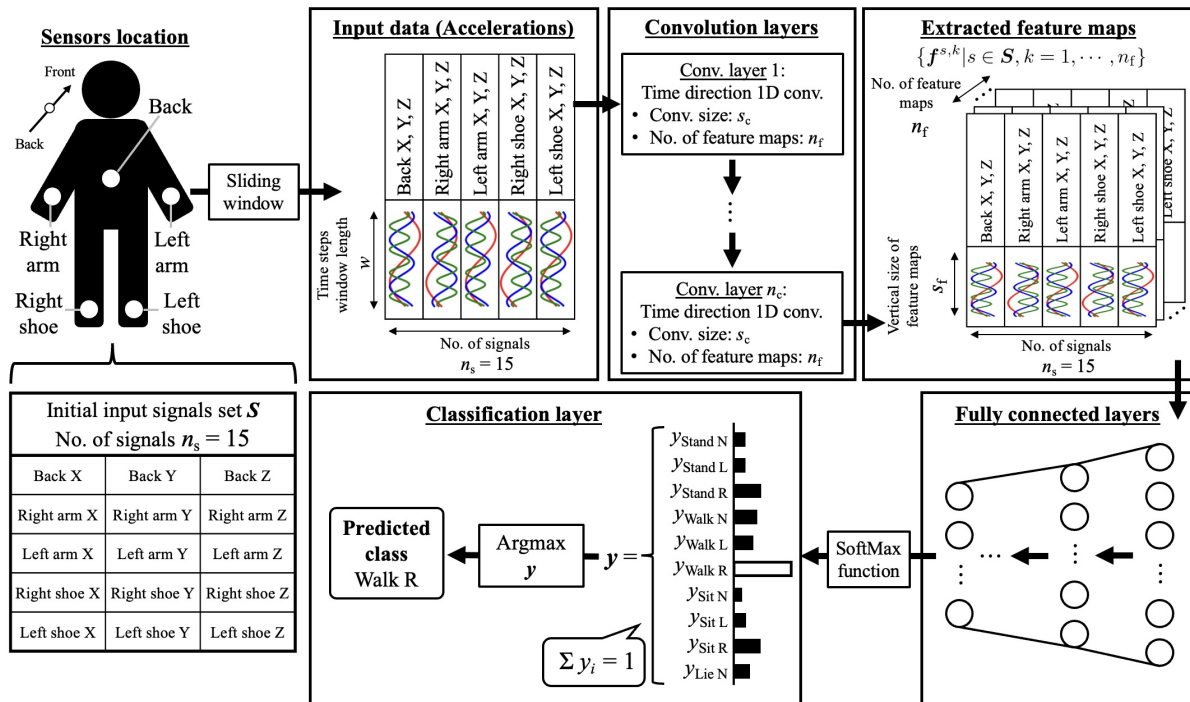


Figure 2. An example of CNN with multiple signals as input. This is the CNN using the initial signals set \mathcal{S} and classes set \mathcal{C} defined by Equations (4.1) and (4.2) described in Subsection 4.2. In other words, the CNN structure for 10 human activities recognition from 15 acceleration signals.

3.3. Time-directional grad-CAM

We defined the vertical size of the feature map before the fully connected layer as s_f , as shown in Figure 2. Let us denote the k th feature map of signal s as

$$\mathbf{f}^{s,k} = [f_1^{s,k} \ f_2^{s,k} \ \dots \ f_{s_f}^{s,k}]^T \in \mathbb{R}^{s_f}, \quad s \in \mathcal{S}, \quad k \in \{1, \dots, n_f\}. \tag{3.5}$$

In the case of CNNs for image-data-based classification, the form of feature maps is a matrix. However, in the case of CNNs consisting of time-directional convolution layers, the form of the feature maps is a vector. We defined the effect of the feature map $\mathbf{f}^{s,k}$ on the estimation class c' as

$$\alpha_{c'}^{s,k} = \frac{1}{s_f} \sum_{j=1}^{s_f} \frac{\partial y_{c'}}{\partial f_j^{s,k}}. \tag{3.6}$$

Then, we applied the grad-CAM of signal s to estimation class c' in the vector form of

$$\mathbf{Z}_{c'}^s = \text{ReLU}\left(\frac{1}{n_f} \sum_{k=1}^{n_f} \alpha_{c'}^{s,k} \mathbf{f}^{s,k}\right), \quad \mathbf{Z}_{c'}^s \in \mathbb{R}_{\geq 0}^{s_f}. \tag{3.7}$$

The activated region of each signal can be understood by calculating $\mathbf{Z}_{c'}^s$ for all signals $\forall s \in \mathcal{S}$.

In this study, we refer to $Z_{c'}^s$ as “time-directional grad-CAM” because it is calculated by the summing partial differentiations of the time direction. This was defined by a minor change in the basic grad-CAM for image data [9]. In the case of the basic grad-CAM [9], the CNN generates one grad-CAM for one input data. In contrast, in the case of time-directional grad-CAM, CNN generates grad-CAMs as many as the number of signals n_s for one input data.

Figure 3 shows the examples of $Z_{c'}^s$ calculated using the CNN presented in Figure 2 and the dataset described in Subsection 4.2. From top to bottom, these results correspond to $c' = \text{Stand N, Stand L, Stand R, and Walk N}$. Notable, an estimation class c' is an element of \mathcal{C} . We can determine the signals that are important by viewing the time-directional grad-CAM shown in Figure 3. For example, signals from the left arm and left shoe are not used for estimating the Stand N class. Moreover, left arm signals are not used for the estimation of the Stand R class. Therefore, the time-directional grad-CAM is effective for finding unimportant signals.

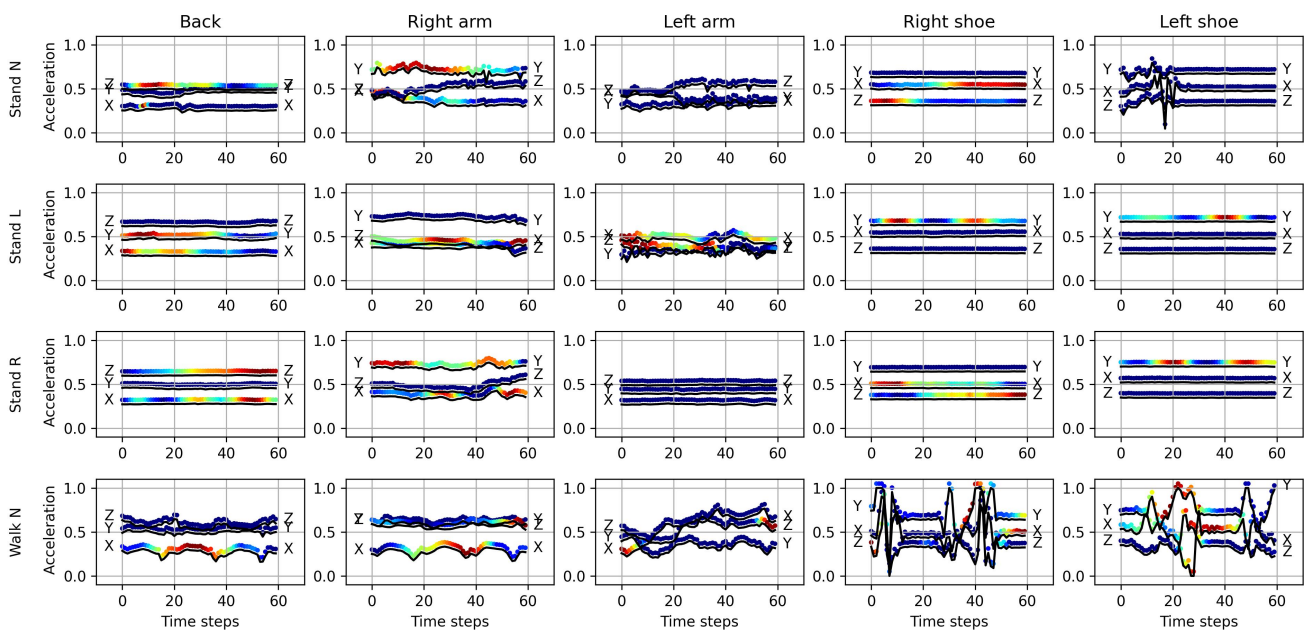


Figure 3. Examples of time-directional grad-CAMs of 15 acceleration signals for four estimation classes (Stand N, Stand L, Stand R, and Walk N) calculated by the CNN illustrated in Figure 2 using the dataset described in Subsection 4.2. The red color represents high activation and the dark blue color represents low activation.

3.4. Signals importance index

Although we can find unimportant signals by viewing time-directional grad-CAM, the result varies for each input data. Therefore, viewing all the grad-CAMs is difficult when the data size is large. Herein, we quantify the importance of signal s for classification based on $\alpha_{c'}^{s,k}$ defined in Eq (3.6).

Here, we denote the input dataset of size n_{dat} as

$$\mathbf{X} = \{\mathbf{X}_i \mid i = 1, \dots, n_{\text{dat}}\}, \quad \mathbf{X}_i \in \mathbb{R}^{w \times n_s}. \quad (3.8)$$

The size of i th input data \mathbf{X}_i is $w \times n_s$, which is the window length w and number of signals n_s . When

we define the input dataset of estimation class $c' \in C$ as $\mathbf{X}_{c'}$, we can represent the input dataset \mathbf{X} as

$$\mathbf{X} = \bigcup_{c' \in C} \mathbf{X}_{c'}. \quad (3.9)$$

Using the set $\mathbf{X}_{c'}$, we define the importance of the signal $s \in S$ to the estimation class $c' \in C$ as

$$L^s(\mathbf{X}_{c'}) = \frac{1}{|\mathbf{X}_{c'}|} \sum_{\mathbf{X}_i \in \mathbf{X}_{c'}} g_{c'}^s(\mathbf{X}_i), \quad (3.10)$$

where

$$g_{c'}^s(\mathbf{X}_i) = \frac{1}{n_f} \sum_{k=1}^{n_f} \beta_{c'}^{s,k}(\mathbf{X}_i), \quad \beta_{c'}^{s,k}(\mathbf{X}_i) = \begin{cases} \alpha_{c'}^{s,k}(\mathbf{X}_i), & \text{if } \alpha_{c'}^{s,k}(\mathbf{X}_i) \geq 0 \\ 0, & \text{otherwise} \end{cases}, \quad \mathbf{X}_i \in \mathbf{X}_{c'}. \quad (3.11)$$

In addition, $\alpha_{c'}^{s,k}(\mathbf{X}_i)$ is $\alpha_{c'}^{s,k}$ of the input data \mathbf{X}_i to a CNN, and c' is the estimated class. Therefore, $L^s(\mathbf{X}_{c'})$ represents the importance of signal s to class c' based on the grad-CAM. Notably, we ignore terms with negative partial derivatives to extract the positive effect on classification, as shown in Eq (3.11). Moreover, using $L^s(\mathbf{X}_{c'})$, we can obtain the matrix

$$\begin{aligned} \mathbf{I}_{\text{mat}}(\mathbf{X}_{c_1}, \mathbf{X}_{c_2}, \dots, \mathbf{X}_{c_{n_{\text{cm}}}}) &= \begin{bmatrix} L^{s_1}(\mathbf{X}_{c_1}) & L^{s_1}(\mathbf{X}_{c_2}) & \dots & L^{s_1}(\mathbf{X}_{c_{n_{\text{cm}}}}) \\ \vdots & \vdots & \ddots & \vdots \\ L^{s_{n_s}}(\mathbf{X}_{c_1}) & L^{s_{n_s}}(\mathbf{X}_{c_2}) & \dots & L^{s_{n_s}}(\mathbf{X}_{c_{n_{\text{cm}}}}) \end{bmatrix} \\ &= [L^s(\mathbf{X}_{c'})] \in \mathbb{R}_{\geq 0}^{|\mathbf{S}| \times |\mathbf{C}|}, \quad s \in \mathbf{S}, c' \in \mathbf{C}. \end{aligned} \quad (3.12)$$

As shown in Eq (3.9), since $\mathbf{X} = \mathbf{X}_{c_1} \cup \mathbf{X}_{c_2} \cup \dots \cup \mathbf{X}_{c_{n_{\text{cm}}}}$ is satisfied, we define

$$\mathbf{I}_{\text{mat}}(\mathbf{X}) \stackrel{\text{def}}{=} \mathbf{I}_{\text{mat}}(\mathbf{X}_{c_1}, \mathbf{X}_{c_2}, \dots, \mathbf{X}_{c_{n_{\text{cm}}}}). \quad (3.13)$$

We refer to $\mathbf{I}_{\text{mat}}(\mathbf{X})$ as the ‘‘signals importance matrix (SIM)’’ because the matrix comprises the importance of all signals and classes using the input dataset \mathbf{X} . The effect of each signal to all classes can be understood by calculating and viewing SIM $\mathbf{I}_{\text{mat}}(\mathbf{X})$.

Although $\mathbf{I}_{\text{mat}}(\mathbf{X})$ includes important information, summarizing the SIM is necessary to find signals that are unimportant to all classes. Therefore, by calculating the summation of the row values of SIM, we denote the importance of signal s as

$$I^s(\mathbf{X}) = \frac{1}{|\mathbf{C}|} \sum_{c' \in \mathbf{C}} I_{\text{mat}}^{s,c'}(\mathbf{X}), \quad (3.14)$$

where $I_{\text{mat}}^{s,c'}(\mathbf{X})$ is the value of row s and column c' of $\mathbf{I}_{\text{mat}}(\mathbf{X})$. By calculating $I^s(\mathbf{X})$ for all signals s , we obtain the vector as

$$\mathbf{I}_{\text{vec}}(\mathbf{X}) = [I^s(\mathbf{X})] \in \mathbb{R}_{\geq 0}^{|\mathbf{S}|}. \quad (3.15)$$

We refer to $\mathbf{I}_{\text{vec}}(\mathbf{X})$ as the ‘‘signals importance vector (SIV)’’ because it is the vector that consists of the signal importance.

The calculated examples of SIM and SIV are shown in Figure 4. This is the effect of the each individual signal on the class estimation by the CNN. These values were calculated using the CNN for Condition A described in Section 5. The result of SIM (columns 1 to 10 in Figure 4) includes various important data. First, we can confirm that the Right arm X signal is used for right hand movement (Stand R, Walk R, and Sit R) estimation. In the case of the left hand movement (Stand L, Walk L) estimation, the signals from the Back sensor responds strongly. Since the Back sensor is attached to the body trunk, it is assumed that the body trunk movement is used to estimate the movement of the left hand. The Left arm X responds to another left hand movement (Sit L) estimation. In addition, the Left shoe Y and Right shoe Y respond strongly to the motion estimation of Walk N. Based on these results, the SIM is considered reasonable. However, since some results are not comfortable, it is important to remove unnecessary signals by FG-SSA.

Moreover, we can identify the signals that are important for all classes by viewing SIV (the column 11 in Figure 4). From SIV, we denote the minimum importance signal s_{\min} and maximum signal s_{\max} as

$$(s_{\min}, s_{\max}) = \left(\underset{s \in S}{\operatorname{argmin}} I^s(\mathbf{X}), \underset{s \in S}{\operatorname{argmax}} I^s(\mathbf{X}) \right). \tag{3.16}$$

We expect to maintain the estimation accuracy even when removing s_{\min} because it is a minimum importance signal. In contrast, we expect the accuracy to decrease when we remove s_{\max} and re-learn the CNN because it is the most important signal.



Figure 4. Example of signals importance to all classes expressed by SIM $I_{\text{mat}}(\mathbf{X})$ and SIV $I_{\text{vec}}(\mathbf{X})$ calculated to CNNs of Condition A described in Section 5. Columns 1 to 10 are SIM and column 11 is SIV (“All classes”). The values are standardized from zero to one in each column. The greater the cell’s value, the greater is the relevance of signal s on estimation class c' .

Algorithm 1 Features gradient-based signals selection algorithm (FG-SSA)

Input:

- Training dataset $\mathbf{X}_{\text{train}}$
- Validation dataset $\mathbf{X}_{\text{valid}}$
- Initial signals set \mathcal{S}
- Maximum number of using signals γ

Output: Using signals set \mathcal{S}_{use}

- 1: Initialization of a signals set $\mathcal{S}_0 \leftarrow \mathcal{S}$
- 2: **for** $t = 0$ to $|\mathcal{S}| - 1$ **do**
- 3: Learning a CNN by the training dataset $\mathbf{X}_{\text{train}}^{\mathcal{S}_t}$
- 4: Calculating validation accuracy $A(\mathbf{X}_{\text{valid}}^{\mathcal{S}_t})$
- 5: Finding a minimum importance signal $s_{\min} \leftarrow \underset{s \in \mathcal{S}_t}{\operatorname{argmin}} I^s(\mathbf{X}_{\text{valid}}^{\mathcal{S}_t})$, see Equation (3.16)
- 6: Removing a signal: $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \setminus s_{\min}$
- 7: **end for**
- 8: $\mathcal{S}_{\text{use}} \leftarrow \underset{\mathcal{S}' \in \{\mathcal{S}_0, \dots, \mathcal{S}_{|\mathcal{S}|-1}\}}{\operatorname{argmax}} A(\mathbf{X}_{\text{valid}}^{\mathcal{S}'}), \text{ s.t., } |\mathcal{S}_{\text{use}}| \leq \gamma$
- 9: **return** \mathcal{S}_{use}

Notably, $\mathbf{X}_{\text{train}}^{\mathcal{S}_t}$ and $\mathbf{X}_{\text{valid}}^{\mathcal{S}_t}$ are all the input data belonging to the training and validation datasets using the signal set \mathcal{S}_t , respectively.

3.5. Signals selection algorithm

In this study, we proposed an algorithm to find a desirable signal subset $\mathcal{S}_{\text{use}} \subseteq \mathcal{S}$ by removing unimportant signals. The proposed method is presented in Algorithm 1. The main inputs to the algorithm are the training dataset $\mathbf{X}_{\text{train}}$, validation dataset $\mathbf{X}_{\text{valid}}$, and input initial signals set \mathcal{S} . To avoid data leakage, the test dataset was not used in the algorithm. The elements of set \mathcal{S} are the signal identification names defined in Eq (3.2). The first procedure is to create the initial signal set \mathcal{S}_0 , which consists of all signals (line 1). The next step is to develop a CNN using the training dataset $\mathbf{X}_{\text{train}}^{\mathcal{S}_0}$, consisting of \mathcal{S}_0 (line 3). Subsequently, we measure the validation accuracy $A(\mathbf{X}_{\text{valid}}^{\mathcal{S}_0})$ using the validation dataset $\mathbf{X}_{\text{valid}}^{\mathcal{S}_0}$ (line 4). Next, we determine the most unimportant signal $s_{\min} \in \mathcal{S}_0$ based on Eq (3.16) (line 5). Finally, we obtain the next signal set \mathcal{S}_1 by removing s_{\min} from \mathcal{S}_0 (line 6).

This procedure is repeated until the number of signals reaches one (i.e., $t = |\mathcal{S}| - 1$), and the validation accuracies are recorded. The algorithm returns the signal subset \mathcal{S}_{use} leading to maximum validation accuracy (lines 8 and 9). Notably, the case wherein the signal subset leading to the maximum accuracy is the initial signal set can occur (i.e., $\mathcal{S}_{\text{use}} = \mathcal{S}$). In cases wherein the main purpose is to achieve maximum accuracy, adopting the initial signal set as an optimal subset is a better option, if $\mathcal{S}_{\text{use}} = \mathcal{S}$ occurs. However, some cases exist that require a decrease in the number of signals. Therefore, we prepared constraints wherein the number of adopted signal sizes is γ or less (i.e., $|\mathcal{S}_{\text{use}}| \leq \gamma$). This is a hyperparameter of the proposed algorithm.

When the size of the initial signal set \mathcal{S} is n_s , the number of developing CNNs in Algorithm 1 is as follows:

$$T(n_s) = n_s - 1$$

$$\sim n_s. \quad (3.17)$$

That is, the computational complexity of Algorithm 1 is $O(n_s)$. This means that even if the signal size n_s increases, the signal subset is returned in realistic time.

Generally, the total pattern for choosing m from n signals is ${}_n C_m$. Because m , which is the number of signals leading to the maximum validation accuracy, is not known, the total number of combinations of input layers $T_{bs}(n_s)$ is similar to

$$\begin{aligned} T_{bs}(n_s) &= \sum_{m=1}^{n_s} {}_{n_s} C_m \\ &\sim \max_m {}_{n_s} C_m \\ &= {}_{n_s} C_{\lfloor n_s/2 \rfloor}. \end{aligned} \quad (3.18)$$

Therefore, finding the optimal signal subset using a brute-force search is difficult. The computation time tends to be large because the CNN includes other hyperparameters. From this viewpoint, a fast algorithm such as the proposed method is important.

4. Experiment 1: Relationship between validation accuracy and the number of deleted signals

4.1. Objective and outline

Herein, we examine the reliability of $s_{\min, \max}$ defined in Eq (3.16). We expect to maintain estimation accuracy even if the signal s_{\min} is removed because it is the most unimportant signal. In contrast, because signal s_{\max} is the most important signal, the estimation accuracy may decrease by removing it. We gradually removed signal $s_{\min, \max}$ using the processes of lines 2 to 7 of Algorithm 1 to verify the aforementioned hypothesis. In addition, we repeatedly developed CNNs and recorded their validation accuracies. We performed verification using a total of 100 seeds because the training process of CNN depends on randomness. The adopted layer structure of the CNN is explained in the Appendix section. The number of epochs was 300.

4.2. Dataset

Herein, we define the initial signals set S and the classes set C described in Subsection 3.1 concretely with real data. We used the dataset ‘‘OPPORTUNITY Activity Recognition’’. This is the dataset for activity recognition, and it is used in ‘‘Activity Recognition Challenge’’ held by IEEE in 2011 [45, 46]. The dataset is regarded as reliable because it is used for the performance evaluation of machine learning in some studies [47, 48]. The dataset contains data on multiple inertial measurement units, 3D acceleration sensors, ambient sensors, and 3D localization information. Four participants performed a natural execution of daily activities. The activity annotations include locomotions (e.g., sitting, standing, and walking) and left- and right-hand actions (e.g., reach, grasp, release, and etc.). Details of the dataset are described in [45, 46].

In this study, we used five triaxial acceleration sensors from all the measurement devices (sampling frequency: 32 [Hz]). Attachment points of sensors on the human body are ‘‘back,’’ ‘‘right arm,’’ ‘‘left arm,’’ ‘‘right shoe,’’ and ‘‘left shoe.’’ The total number of signals was 15 because we adopted five triaxial

sensors ($5 \times 3 = 15$). We adopted data splitting using the sliding window method with window length $w = 60$ and sliding length 60. One signal length was approximately 2 s since the sampling frequency was 32 [Hz]. Searching for the optimal window length size is important because it is a hyperparameter that affects the estimation accuracy [44, 49, 50]. However, we did not tune the window length size because the aim of this study was to provide a signal-selection algorithm.

Subsequently, we labeled the motion class label for each dataset based on human activity. The class labels are combinations of locomotions (“Stand,” “Walk,” “Sit,” and “Lie”) and three hands activity (“R”: moving right hand, “L”: moving left hand, and “N”: not moving hands). For example, the class label “Sit R” refers to the sitting and right-hand motions.

Table 1 lists the results of applying the described procedures to all the data. This indicates that the data size of the left-hand motion is small. Nearly all participants were right-handed (notably, we could not find a description of the participants’ dominant arms in the explanation of the OPPORTUNITY dataset). Moreover, data belonging to Lie R and L are absent. Therefore, these classes were removed from the estimation task (i.e., the total number of classes was 10).

Table 1. Samples size of each class label.

Labels	Samples	Rates [%]
Stand N	1070	20.68
Stand L	193	3.73
Stand R	642	12.41
Walk N	1405	27.16
Walk L	63	1.22
Walk R	150	2.90
Sit N	571	11.04
Sit L	130	2.51
Sit R	536	10.36
Lie N	414	8.00
Lie L	0	0.00
Lie R	0	0.00
Total	5174	100.00

Subsequently, we randomly split all the data into a training dataset (80%) and a test dataset (20%). Moreover, we assigned 20% of the training dataset to the validation dataset. The training, validation, and test datasets were independent because we adopted the sliding window method for the same window and slide length (60 steps). We defined the signal set \mathcal{S} , which has 15 elements, and the class set \mathcal{C} , which has 10 elements, as follows:

$$\mathcal{S} = \{\text{Back X, Back Y, Back Z, Right arm X, Right arm Y, Right arm Z, Left arm X, Left arm Y, Left arm Z, Right shoe X, Right shoe Y, Right shoe Z, Left shoe X, Left shoe Y, Left shoe Z}\}, \quad (4.1)$$

$$\mathcal{C} = \{\text{Stand N, Stand L, Stand R, Walk N, Walk L, Walk R, Sit N, Sit L, Sit R, Lie N}\}. \quad (4.2)$$

In other words, CNNs solve 10 classification problems from 15 multisignals. Moreover, an algorithm for removing unimportant signals was provided while maintaining the estimation accuracy. Although the sets \mathcal{S} and \mathcal{C} represent acceleration signals and human activities, respectively, the proposed algorithm can be used for other diverse signals, such as EEG and ECG.

4.3. Result and discussion

First, we indicate the validation accuracies when the most unimportant signal, s_{\min} , is gradually removed, as shown in Figure 5 (A). From left to right in Figure 5 (A), the number of deleted signals increases (i.e., the number of signals in the CNN input layer decreases). The leftmost and rightmost results are obtained using all signals and only one signal as the CNN input layer, respectively. The results show that even if six signals were deleted, the validation accuracy did not decrease. Moreover, the accuracy significantly decreases by removing seven or more signals. In this case, although CNNs estimate class labels from 15 signals in the set \mathcal{S} , six signals are unnecessary.

We calculated the average removed timings of the 15 signals to determine the unnecessary signals. Figure 5 (B) shows the results. This means that the lower the value, the earlier the signal is removed in the procedure of Algorithm 1. The results indicate that the signals of the right and left shoes are removed earlier than those of the other sensors. In contrast, some signals from the back, right arm, and left arm did not disappear early. Therefore, we can regard shoe signals as unimportant for classification. In addition, shoe sensors are important for walk motion classification. However, even if shoe sensors disappear, we consider that the back sensor attached to centroids of the human body contributes to the classification of walk motions because these motions are periodic.

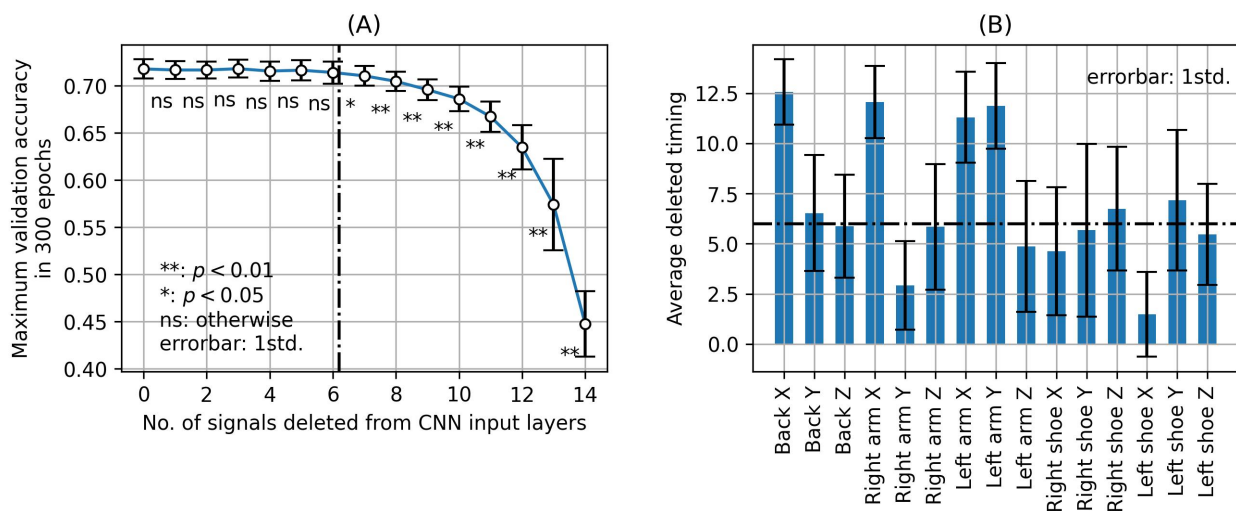


Figure 5. (A) Maximum validation accuracy in 300 epochs when the most unimportant signal s_{\min} is gradually removed and CNNs are re-learned. (B) Average removed timings of each signal. It means the smaller the timing, the earlier the signal is removed. The results of both (A) and (B) are average values of 100 seeds, and the error bars are standard deviations. The p values in (A) represent the results of two-sided t test, and the dashed line represents the timing of statistically decreasing validation accuracy.

Then, we indicate the validation accuracies when the most important signal, s_{\max} , is gradually re-

moved in Figure 6 (A). We can verify that the validation accuracy statistically decreases by removing one of the most important signals. Therefore, removing the most important signal, s_{\max} , leads to a worse accuracy. The average timing of the signal removal is shown in Figure 6 (B). From this figure, we confirm that the signals of back X, right arm X, and left arm Y are removed early. Moreover, we confirm that shoe sensor signals are not removed early. These tendencies are in contrast compared with the case of removing the most unimportant signal s_{\min} .

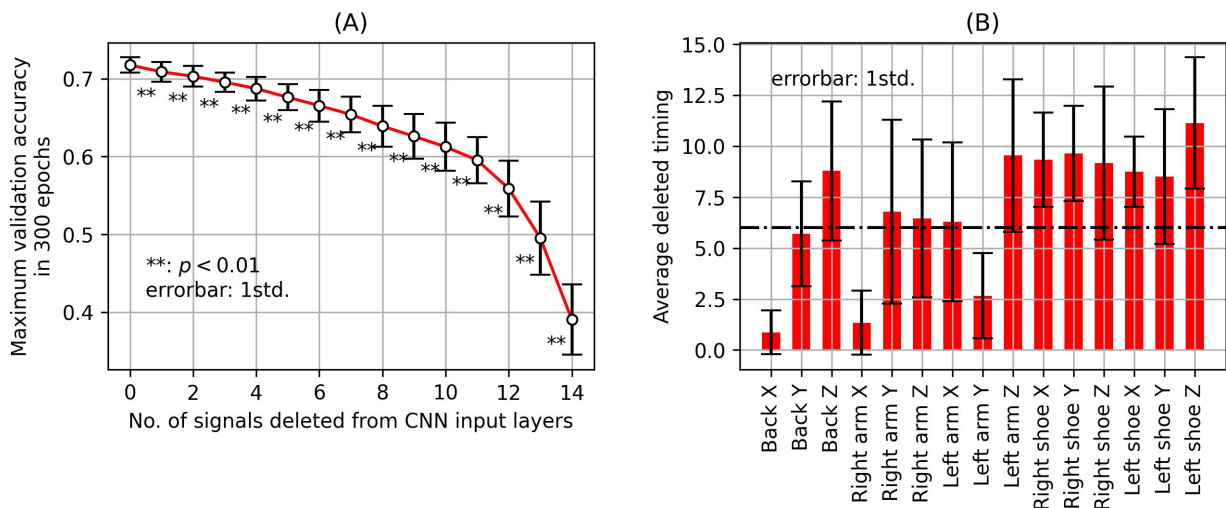


Figure 6. (A) Maximum validation accuracy in 300 epochs when the most important signal s_{\max} is gradually removed and CNNs are re-learned. (B) Average removed timings of each signal. In contrast to that shown in Figure 5, the most important signals are removed.

Clearly, (1) even if we remove the most unimportant signal s_{\min} , the estimation performance remains, and (2) the performance decreases when we remove the most important signal s_{\max} . Therefore, the signal importance $I^s(\mathbf{X})$ defined in Eq (3.14) is reliable.

5. Experiment 2: Effectiveness of FG-SSA on generalization scores

5.1. Objective and outline

Herein, we confirmed the effect of FG-SSA indicated in Algorithm 1 on the generalization performance of the classification. Therefore, we developed the following three conditions for CNNs:

- Condition A: CNN using all signals (i.e., FG-SSA is not used).
- Condition B: CNN for applied FG-SSA of $\gamma = n_s$.
- Condition C: CNN for applied FG-SSA of $\gamma = 9$.

Condition A means that the CNN does not remove signals but uses all signal sets \mathcal{S} . Condition B refers to the CNN using the signal subset \mathcal{S}_{use} obtained by the FG-SSA, given n_s as the constraint parameter γ . Under this condition, we allow the algorithm to return \mathcal{S} as \mathcal{S}_{use} when the maximum validation accuracy is achieved using all signal sets \mathcal{S} . In other words, a case wherein no signals are removed can occur. Condition C implies that the number of adopted signals is nine or less for a CNN input layer (i.e., $|\mathcal{S}_{\text{use}}| \leq 9$). In other words, six or more signals were deleted because the initial number of signals

n_s was 15. The value of hyper-parameter $\gamma = 9$ was determined by referring to the result shown in Figure 5 (A).

We developed CNNs using the signal set, which was determined by FG-SSA. Moreover, the optimal epochs leading to a maximum validation accuracy were adopted. The search range of epochs was from 1 to 300. Subsequently, the generalization performance is measured using the test dataset. The test dataset was used only at this time (i.e., it was not used for the parameter search). We developed CNNs for Conditions A, B, and C with a total of 100 seeds to remove the randomness effect.

5.2. Result and discussion

By developing CNNs under Conditions A, B, and C on a total of 100 seeds, the average number of signals used was 15.00, 11.94, and 8.35, respectively. In other words, an average of 3.06 and 6.65 signals were removed by FG-SSA in the cases of Conditions B and C, respectively. The generalization performance (F score, precision, and recall) measured by the test dataset for each condition is shown in Figure 7. These are histograms of 100 CNNs developed using 100 random seeds.

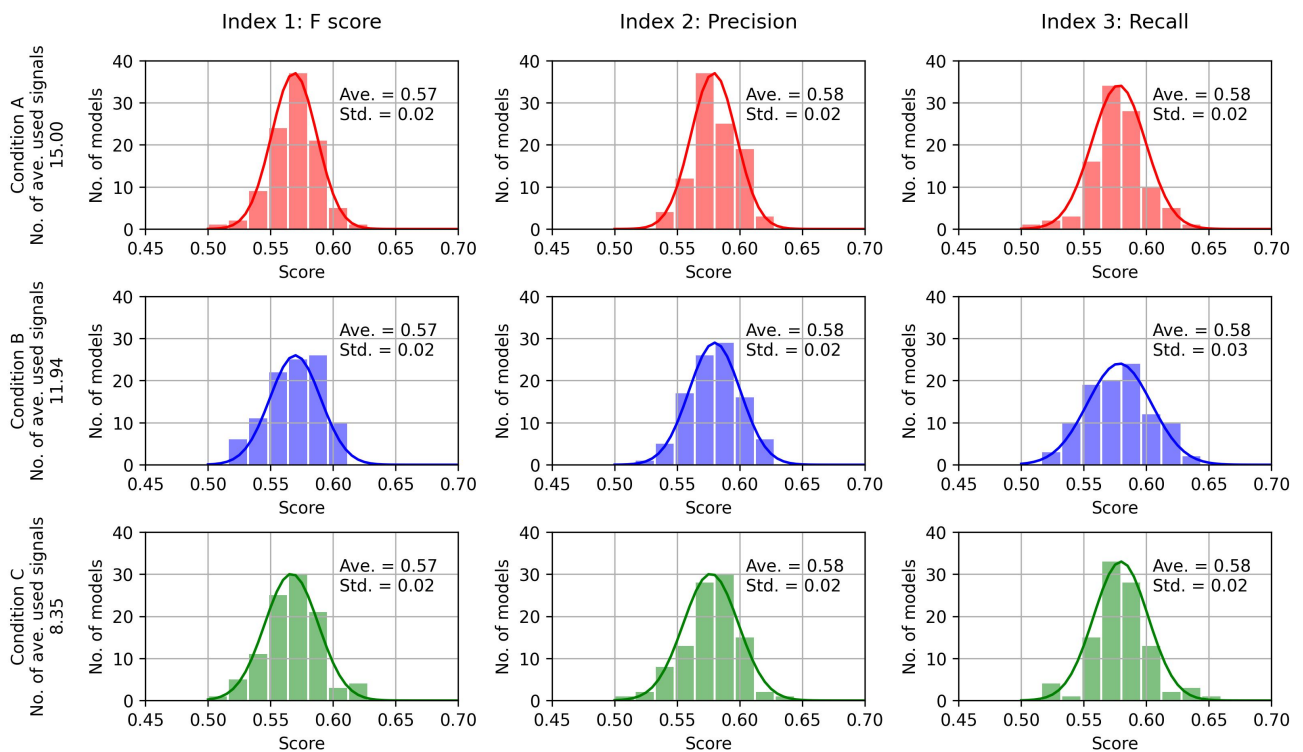


Figure 7. Histograms of estimation score using the test dataset (total 100 seeds). The macro averages of 10 classes of F score, precision, and recall from left to right, and Conditions A, B, and C from top to bottom.

The results indicated that the generalization scores were nearly the same for Conditions A, B, and C. Moreover, p values of the two-sided t -test were not statistically significant. The confusion matrices obtained by CNNs for each condition are shown in Figure 8. The values were averaged by 100 seed results and standardized, where the summation in each row was 1. Consequently, the confusion matrices were nearly the same. Although the proposed algorithm removed some signals, generalization

errors did not increase. Therefore, we consider that FG-SSA can find and remove signals that are not important for CNN-based classification.

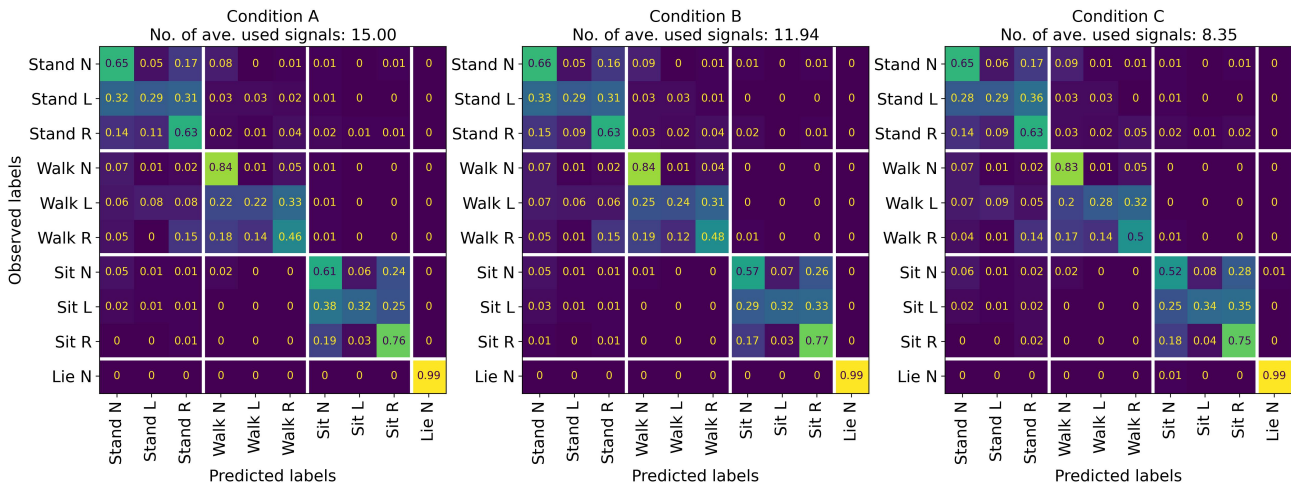


Figure 8. Confusion matrices of the Conditions A, B, and C using the test dataset. The values were averaged by 100 seed results and standardized, where the summation in each row was 1.

From another viewpoint, the number of correct classifications of left hand motions (Stand L, Walk L, and Sit L) was small in all conditions. We adopted weighted cross-entropy-based learning for CNNs because the original data size of the left hand motions is small, as shown in Table 1. However, we consider that an appropriate classification cannot be performed because the data diversity of these motions is low. Although we believe that nearly all participants are right-handed, the explanation of participant's dominant arm in the OPPORTUNITY dataset is insufficient to the best of our knowledge.

Note that this study also validates FG-SSA on a dataset other than OPPORTUNITY. This dataset is for an experiment that classifies swimming styles with signals measured from a single inertial sensor. The details of this experiment and the results obtained are presented in Appendix B. The experimental results also indicate that FG-SSA is effective at signal selection.

6. Comparison of FG-SSA and other methods

6.1. Objective and outline

From the described results, it was confirmed that signals can be removed without degrading the generalization performance using FG-SSA. In this section, we compare the generalization performance obtained by signal subsets selected by Condition C mentioned in Section 5 and signal subsets obtained by other signal selection methods. Then, we evaluate the effectiveness of the proposed method. Since Condition C has a problem of selecting a maximum of nine signals from 15 signals, this condition was also adopted in existing methods.

Bayesian optimization (BO) and random search (RS) were adopted as existing methods. BO is a search algorithm that solves combinatorial optimization problems and is sometimes applied to hyperparameter tuning in machine learning and deep learning [51, 52]. In this approach, the next hyperparameters to be verified are determined based on validation accuracies obtained with the previously set

hyperparameters. This can be applied to feature selection problems in classification problems [34–36]. Feature selection is a method of selecting features to be used for classification from all features prepared in advance. This is similar to the method of choosing signal subset from all signals. Therefore, BO can also be applied to signal selection and has been applied in several studies [41, 42]. Therefore, in this study, we adopted BO as an existing method for comparison with FG-SSA.

As shown in Eq (3.18), when selecting some signals from n_s signals, the number of selectable combinations becomes enormous. In contrast, as shown in Algorithm 1, FG-SSA can obtain the signal subset to be used only by constructing n_s CNNs. On the other hand, the greater the number of iterations for BO, the higher the possibility of discovering a desirable signal subset. However, in comparing FG-SSA and BO, it is fairer to set the number of CNN constructions to be the same. Therefore, we set the number of iterations of BO to n_s . Note that n_s is 15 in this experiment, as mentioned in Section 5. We adopted tree-structured Parzen estimator (TPE) [53] as a specific method of BO and Optuna [54] as a framework for implementation. TPE has the property of trying to obtain the desired solution with a small number of iterations [55].

Moreover, RS was adopted as a baseline. This is a method that repeats the process of randomly selecting signals and constructing a CNN for a specified number of times and uses the signal subset with the highest validation accuracy. This time, we decided to repeat the process of selecting up to nine signals from 15 signals n_s times. In other words, the numbers of constructing CNNs for RS, BO, and FG-SSA are the same.

The details of the process of existing methods are described in Algorithm 2. The inputs for this algorithm are the training dataset $\mathbf{X}_{\text{train}}$, validation dataset $\mathbf{X}_{\text{valid}}$, initial signals set \mathbf{S} defined in Eq (4.1), maximum number of selecting signals S_{max} , number of iterations T , and searching method $M \in \{\text{BO}, \text{RS}\}$. BO means the TPE-based Bayesian optimization and RS means random search. In Algorithm 2, the signal subset \mathbf{S}_t consisting of S_{max} signals is extracted by “SignalsSelection” function (line 2 in Algorithm 2, BO or RS method). We developed a CNN using the training dataset $\mathbf{X}_{\text{train}}^{\mathbf{S}_t}$ consisting of the signals subset \mathbf{S}_t and measured the validation accuracy $A(\mathbf{X}_{\text{valid}}^{\mathbf{S}_t})$ using the validation dataset $\mathbf{X}_{\text{valid}}^{\mathbf{S}_t}$. We repeated it from $t = 1$ to $t = T$ (i.e., T times) and selected the signal subset leading to the maximum validation accuracy. This subset is $\mathbf{S}_{\text{use}} \subseteq \mathbf{S}$ and it is returned by Algorithm 2 to users. Then, we developed a CNN using the signal subset \mathbf{S}_{use} and measured the generalization scores (F score, precision, and recall) using the test dataset. The layer structure and learning condition of the CNN are the same as the previous experiment described in Section 5.

In general, for both BO and RS, better signal subsets can be found as the number of iterations T increases. However, CNN requires a long time to build; thus, it is not desirable to increase T . As shown in Eq (3.17), the strength of FG-SSA is that it is a fast algorithm. Therefore, we decided to make the number of iterations T of BO and RS and the maximum number of selected signals S_{max} the same as those of FG-SSA (Section 5, Condition C). In other words, we set $T = 15, S_{\text{max}} = 9$. We ran Algorithm 2 100 times while changing the random seed to remove the effect of randomness (FG-SSA in Condition C is also the same).

Algorithm 2 Existing signals selection algorithm

Input:

Training dataset $\mathbf{X}_{\text{train}}$
 Validation dataset $\mathbf{X}_{\text{valid}}$
 Initial signals set \mathcal{S}
 Maximum number of using signals S_{max}
 Number of iterations T
 Search method $M \in \{\text{BO}, \text{RS}\}$

Output: Using signals set \mathcal{S}_{use}

```

1: for  $t = 1$  to  $T$  do
2:    $\mathcal{S}_t \leftarrow \text{SignalsSelection}(\mathcal{S}, S_{\text{max}}, M)$ 
3:   Learning a CNN by the training dataset  $\mathbf{X}_{\text{train}}^{\mathcal{S}_t}$ 
4:   Calculating validation accuracy  $A(\mathbf{X}_{\text{valid}}^{\mathcal{S}_t})$ 
5: end for
6:  $\mathcal{S}_{\text{use}} \leftarrow \underset{\mathcal{S}' \in \{\mathcal{S}_1, \dots, \mathcal{S}_T\}}{\text{argmax}} A(\mathbf{X}_{\text{valid}}^{\mathcal{S}'})$ 
7: return  $\mathcal{S}_{\text{use}}$ 
  
```

Note: “ $\text{SignalsSelection}(\mathcal{S}, S_{\text{max}}, M)$ ” selects and returns subset $\mathcal{S}_t \subseteq \mathcal{S}$ consisting of S_{max} signals from the initial signals set \mathcal{S} based on method M . When M is “Bayesian optimization (BO)”, tree-structured Parzen estimator is used for signals selection. When M is “random search (RS)”, signals are randomly selected.

6.2. Result and discussion

The generalization scores of FG-SSA, BO, and RS measured by the test dataset are shown in Table 2. The results consist of F score, precision, and recall of 10 classes belonging to the class set \mathcal{C} defined by Eq (4.2). The bold and underlined numbers indicate the highest and lowest scores, respectively. The values in the cells indicate the average scores of 100 runs, and p values are the result of the two-sided t test between FG-SSA and BO.

Based on Table 2, the baseline RS has the lowest score. A possible reason for this is that random signal selection is not strategic in terms of improving the generalization performance. In addition, FG-SSA has obtained better scores in more classes than BO. Although some results have better scores with BO, there is no significant difference in comparison with FG-SSA. Additionally, looking at the macro averages, we confirmed that FG-SSA scores higher than BO and RS in all three indicators. Considering this, it can be interpreted that FG-SSA can discover signal subsets leading to higher generalization performance than BO and RS. Therefore, FG-SSA is an excellent method from the viewpoint of performing good signal selection with a small number of iterations.

Since the number of CNN constructions in FG-SSA is 15, the number of Bayesian optimization iterations was set to 15 in the previous analyses. As shown in Table 2, FG-SSA demonstrated higher estimation performance than BO. However, by increasing the number of iterations of BO, it may be possible to find a signal set that yields scores similar to those of FG-SSA. To test this hypothesis, the generalization performance was measured when the number of BO iterations was 15, 30, and 45 (BO15, BO30, and BO45). Table 3 shows the obtained results. It can be seen that the generalization performance increases slightly as the number of BO iterations increases. When comparing FG-SSA

and BO15, FG-SSA performed significantly higher; however, when comparing FG-SSA and BO45, no significant differences were observed for all indices. Therefore, the two signal sets obtained with BO45 and FG-SSA could be similar. Conversely, the computation time for BO45 was approximately three times longer than that for FG-SSA. Therefore, FG-SSA can find a good signals set at a lower computational cost than BO, which is an existing method. Note that the computational environment used in this experiment was OS: Ubuntu 20.04 LTS, CPU: Xeon W-2225 (4.1 GHz), RAM: 32 GB DDR4-2933, GPU: NVIDIA RTX A6000 (48 GB).

Table 2. Estimation scores of FG-SSA, BO, and RS obtained by the test dataset (average results of 100 seeds). p values mean the two-sided t test between FG-SSA and BO. The bold numbers mean the highest values and the underlined numbers mean the lowest values.

Class	F score				Precision				Recall			
	FG-SSA	BO	RS	p	FG-SSA	BO	RS	p	FG-SSA	BO	RS	p
Stand N	.671	.668	<u>.642</u>	n.s.	.721	.702	<u>.701</u>	*	.635	.643	<u>.600</u>	n.s.
Stand L	.281	.260	<u>.235</u>	*	.282	.280	<u>.244</u>	n.s.	.299	.259	<u>.252</u>	**
Stand R	.597	.585	<u>.530</u>	*	.560	.549	<u>.503</u>	n.s.	.653	.638	<u>.577</u>	n.s.
Walk N	.859	.852	<u>.838</u>	**	.882	.880	<u>.847</u>	n.s.	.840	<u>.828</u>	.832	*
Walk L	.247	.233	<u>.184</u>	n.s.	.259	.234	<u>.192</u>	n.s.	.254	.244	<u>.195</u>	n.s.
Walk R	.402	.380	<u>.289</u>	*	.345	.322	<u>.257</u>	**	.501	.483	<u>.357</u>	n.s.
Sit N	.586	.594	<u>.534</u>	n.s.	.679	.686	<u>.644</u>	n.s.	.531	.536	<u>.476</u>	n.s.
Sit L	.349	.352	<u>.312</u>	n.s.	.426	.427	<u>.373</u>	n.s.	.322	.319	<u>.302</u>	n.s.
Sit R	.686	.688	<u>.657</u>	n.s.	.627	.619	<u>.593</u>	n.s.	.772	.785	<u>.756</u>	n.s.
Lie N	.989	.983	<u>.975</u>	**	.988	.981	<u>.973</u>	**	.990	.986	<u>.976</u>	**
Macro ave.	.567	.560	<u>.520</u>	*	.577	.568	<u>.533</u>	**	.580	.572	<u>.532</u>	*

** $p < .01$, * $p < .05$, n.s.: otherwise (FG-SSA vs. BO)

Table 3. Estimation scores of FG-SSA, BO15, BO30, and BO45 obtained by the test dataset and the corresponding calculation times (average and standard deviation of 100 seeds). The bold numbers indicate the highest values, and the underlined numbers indicate the lowest values. p values are obtained from the two-sided t test between FG-SSA and BO.

	Macro F score		Macro precision		Macro recall		Calculation time [sec]
	Ave.	p	Ave.	p	Ave.	p	
FG-SSA	.567 ± .021	-	.577 ± .022	-	.580 ± .022	-	2179.53 ± 8.37
BO15	<u>.560 ± .023</u>	*	<u>.568 ± .024</u>	**	<u>.572 ± .025</u>	*	2024.54 ± 9.27
BO30	.563 ± .020	n.s.	.572 ± .022	n.s.	.573 ± .022	*	4101.60 ± 29.97
BO45	.564 ± .020	n.s.	.572 ± .022	n.s.	.574 ± .021	n.s.	6174.26 ± 41.47

** $p < .01$, * $p < .05$, n.s.: otherwise (FG-SSA vs. other methods)

7. Conclusion, limitations, and future works

In this study, we described the following topics to find and remove unimportant signals for CNN-based classification:

- (1) The signal importance indices $\text{SIM } I_{\text{mat}}(\mathbf{X})$ and $\text{SIV } I_{\text{vec}}(\mathbf{X})$ are explained in Subsection 3.4.
- (2) The algorithm of the linear complexity $\mathcal{O}(n_s)$ for obtaining the signals subset \mathbf{S}_{use} from the initial signals set \mathbf{S} by finding and removing unimportant signals (see Algorithm 1).

Although the proposed algorithm performed well in the case of the OPPORTUNITY dataset, it had some limitations. This is explained in the following sections. Future work will confirm these findings.

- (A) The results described in this paper were obtained from limited cases. Although we assume that the algorithm can be applied to data other than acceleration signals (e.g., EEG and ECG), it is not validated.
- (B) In this experiment, the initial signals set size was 15. The number of signals may be much higher depending on some situations. It is important to determine the relationship between the number of unimportant signals and the effectiveness of FG-SSA.
- (C) We assume that the algorithm may extract the signals' subset weighted specific target class by providing class weights to the signal importance $I^s(\mathbf{X})$ defined in Eq (3.14). In other words, we denote the importance of signal s weighted classes as

$$I^s(\mathbf{X}; \mathbf{w}) = \frac{1}{|\mathcal{C}|} \sum_{c' \in \mathcal{C}} w^{c'} I_{\text{mat}}^{s,c'}(\mathbf{X}), \quad (7.1)$$

where $w^{c'}$ is the weight of class c' and \mathbf{w} is the class weight vector, which is defined as

$$\mathbf{w} = [w^{c'}] \in \mathbb{R}^{|\mathcal{C}|}, \text{ s.t., } \sum_{c' \in \mathcal{C}} w^{c'} = 1. \quad (7.2)$$

We note that the algorithm based on $I^s(\mathbf{X}; \mathbf{w})$ can return the signals' subset weighted specific classes. However, this effect is not confirmed in this study. This will be studied in the future.

A. Appendix: Layers structure of CNNs

The CNN's layer structure is illustrated in Figure 2. First, an input layer exists for inputting $w \times n_s$ size data (w : window length and n_s : the number of signals). Then, $n_c = 3$ convolution layers exist to generate n_f feature maps using kernel filters of convolution size s_c . The convolutions are only in the direction of time. Subsequently, the generated feature maps are transformed into a vector form by the flattened layer. Vector dimensions are gradually reduced to 200, 100, and 50. Finally, an output vector \mathbf{y} of 10 dimensions is generated, and classification is performed. The total number of CNN layers is nine (one input layer, three convolution layers, four dense layers, and one output layer). The activation function of each layer is ReLU.

We performed a hyperparameter search for convolution size s_c , the number of kernel filters n_f , and learning rate r . Particularly, CNNs were trained using the training and validation datasets described in Subsection 3.1 and 300 epochs. We adopted a weighted cross-entropy-based loss function based on the inverse values of class sample sizes because the sample sizes of each class are imbalanced, as shown in Table 1.

The parameter candidates are as follows:

$$s_c \in \{5, 10, 15\}, \quad n_f \in \{5, 10\}, \quad r \in \{10^{-3}, 10^{-4}\}. \quad (\text{A.1})$$

The combination of hyperparameters leading to the maximum validation accuracy was $(s_c, n_f, r) = (10, 10, 10^{-4})$ (accuracy: 0.726). Therefore, the adopted CNN architecture is as follows:

- Layer 1: Conv. layer, the kernel filter: $(s_c, n_f) = (10, 10)$
- Layer 2: Conv. layer, the kernel filter: $(s_c, n_f) = (10, 10)$
- Layer 3: Conv. layer, the kernel filter: $(s_c, n_f) = (10, 10)$
- Layer 4: Flatten layer based on the final conv. layer's output size
- Layer 5: Dense layer of 200 dimensions
- Layer 6: Dense layer of 100 dimensions
- Layer 7: Dense layer of 50 dimensions
- Layer 8: Classification layer
- Note: the activation function for all layers except the classification layer is ReLU.

We adopted this architecture for all CNNs used in this study. Although CNNs have many other hyperparameters, we avoided excessive tuning because the aim of this study was to provide a solution to the signal selection problem.

B. Appendix: Effectiveness of FG-SSA on swimming style classification

B.1. Outline

The effectiveness of the proposed method should be validated by multiple datasets, not only by a single dataset. Therefore, in this study, FG-SSA was applied to the task of automatically classifying swimming style (backstroke, breaststroke, butterfly, and front crawl) based on signals measured by an inertial sensor and CNN. Swimming style classification with inertial sensors is a typical research topic in sports engineering (e.g., [56, 57]). The subjects were 16 Japanese university students who selected two swimming styles from four swimming styles. The pool had a length of 25 m, and each swimmer made one round trip of swimming (total 50 m). A sensor was attached to the lower back of the swimmers, and it measured the tri-axial acceleration and angular velocity (six signals in total). The sampling frequency of the inertial sensor was 100 Hz, the acceleration range was ± 5 G, and the angular velocity range was ± 1500 dps. All signals were standardized to a minimum of 0 and a maximum of 1. To verify whether FG-SSA can eliminate unnecessary signals and select only important signals, nine meaningless signals generated by uniform random numbers in the 0 to 1 range were prepared. Considering the nine meaningless signals, the total number of signals was $3 + 3 + 9 = 15$.

B.2. Preprocessing

Of the 32 swims (16 subjects \times 2 swims), there was a measurement error (sensor fell off while swims) in 5 swims. Therefore, usable data were available for 27 swimings. The sliding window method was applied to the data with the window length and sliding length set to 1 sec. Since window length and sliding length are of the same size, the cut waveforms did not overlap. This process resulted in sample sizes of 91, 145, 332, and 205 for backstroke, breaststroke, butterfly, and front crawl, respectively. Of the available data, 80 % was used as training data and 20 % as test data. In addition, 20 % of the training data was used as validation data. Since the data splitting is random, the performance may be higher (or lower) than the average tendency due to random effects. To obtain reliable results, the experiment was run with a total of 100 random seeds.

B.3. Result and discussion

Figure 9 (a) shows the maximum validation accuracies in 300 epochs when FG-SSA is applied to the aforementioned task and unnecessary signals are removed one by one. In the initial half of FG-SSA, the estimation performance improves as more unnecessary signals are deleted, reaching a maximum when the number of deletions is 9. Then, in the later half of FG-SSA, the estimation performance degrades. This indicates that the estimation performance is worse when there are unnecessary signals in the input layer or when important signals are removed. Figure 9 (b) shows the deletion timing of each signal. Randomly generated meaningless signals are removed early in the algorithm, and the tri-axial acceleration and angular velocity signals are removed later. This result indicates that FG-SSA can efficiently find and remove only meaningless signals while keeping important signals.

The number of signals deleted by processing line 8 of Algorithm 1 was 9.66 ± 1.88 signals (average and standard deviation of the results of 100 seeds). In other words, the number of signals selected as the input layer of the CNN by FG-SSA was $15 - 9.66 = 5.34$. Next, the generalization performance was measured by the test dataset for CNNs using all signals (Condition A, 15 signals) and for CNNs using the signal set selected by FG-SSA (Condition B, average 5.34 signals). The result is shown in Figure 10. When all signals were used (Condition A), confusion in the estimation of “breaststroke” and “butterfly” was observed. On the other hand, when unnecessary signals were removed by FG-SSA (Condition B), misclassification was greatly reduced. Therefore, it can be concluded that signal reduction by FG-SSA is beneficial for improving the generalization performance.

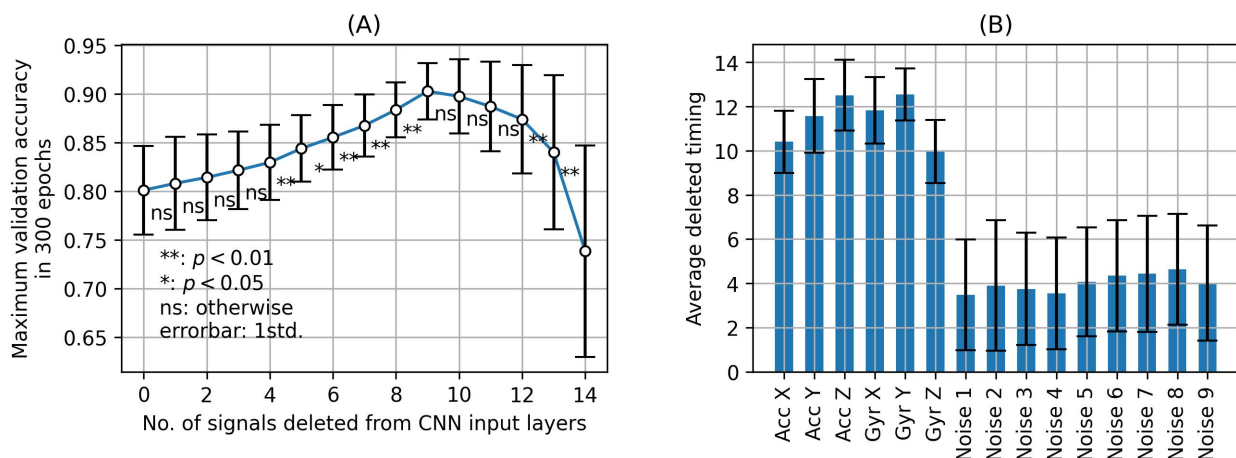


Figure 9. (A) Maximum validation accuracy in 300 epochs when the most unimportant signal s_{\min} is gradually removed by FG-SSA and CNNs are re-learned. (B) Average removed timings of each signal. The results of both (A) and (B) are average values of 100 seeds, and the error bars present standard deviations. The p values in (A) represent the results of two-sided t test.

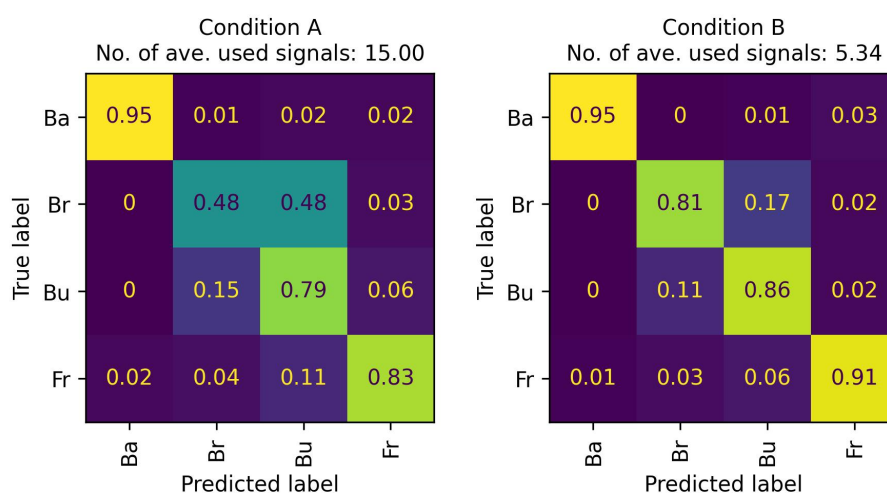


Figure 10. Confusion matrices of the Conditions A and B using the test dataset (“Ba”: backstroke, “Br”: breaststroke, “Bu”: butterfly, and “Fr”: front crawl). The values were averaged over 100 seed results and standardized, where the summation in each row was 1.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by JSPS Grant-in-Aid for Scientific Research (C) (Grant Nos. 21K04535 and 23K11310), and JSPS Grant-in-Aid for Young Scientists (Grant No. 19K20062). This work was also supported by the Tokyo City University Prioritized Studies.

Conflict of interest

The authors declare no conflicts of interest.

References

1. N. Shahini, Z. Bahrami, S. Sheykhivand, S. Marandi, M. Danishvar, S. Danishvar, et al., Automatically identified EEG signals of movement intention based on CNN network (end-to-end), *Electronics*, **11** (2022), 3297. <https://doi.org/10.3390/electronics11203297>
2. T. Zebin, P. J. Scully, K. B. Ozanyan, Human activity recognition with inertial sensors using a deep learning approach, *Proceedings IEEE Sensors*, (2017), 1–3. <https://doi.org/10.1109/ICSENS.2016.7808590>
3. W. Xu, Y. Pang, Y. Yang, Y. Liu, Human activity recognition based on convolutional neural network, *Proceedings of the International Conference on Pattern Recognition*, (2018), 165–170. <https://doi.org/10.1109/ICPR.2018.8545435>

4. Y. Omae, M. Kobayashi, K. Sakai, T. Akiduki, A. Shionoya, H. Takahashi, Detection of swimming stroke start timing by deep learning from an inertial sensor, *ICIC Express Letters Part B: Applications ICIC International*, **11** (2020), 245–251. <https://doi.org/10.24507/icicelb.11.03.245>
5. D. Sagga, A. Echioui, R. Khemakhem, M. Ghorbel, Epileptic seizure detection using EEG signals based on 1D-CNN approach, *Proceedings of the 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering*, (2020), 51–56. <https://doi.org/10.1109/STA50679.2020.9329321>
6. N. Dua, S. N. Singh, V. B. Semwal, Multi-input CNN-GRU based human activity recognition using wearable sensors, *Computing*, **103** (2021), 1461–1478. <https://doi.org/10.1007/s00607-021-00928-8>
7. Y. H. Yeh, D. P. Wong, C. T. Lee, P. H. Chou, Deep learning-based real-time activity recognition with multiple inertial sensors, *Proceedings of the 2022 4th International Conference on Image, Video and Signal Processing*, (2022), 92–99. <https://doi.org/10.1145/3531232.3531245>
8. J. P. Wolff, F. Grützmacher, A. Wellnitz, C. Haubelt, Activity recognition using head worn inertial sensors, *Proceedings of the 5th International Workshop on Sensor-based Activity Recognition and Interaction*, (2018), 1–7. <https://doi.org/10.1145/3266157.3266218>
9. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, *Int. J. Comput. Vision*, **128** (2016), 336–359. <https://doi.org/10.1109/ICCV.2017.74>
10. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 2921–2929.
11. M. Kara, Z. Öztürk, S. Akpek, A. A. Turupcu, P. Su, Y. Shen, COVID-19 diagnosis from chest ct scans: A weakly supervised CNN-LSTM approach, *AI*, **2** (2021), 330–341. <https://doi.org/10.3390/ai2030020>
12. M. Kavitha, N. Yudistira, T. Kurita, Multi instance learning via deep CNN for multi-class recognition of Alzheimer’s disease, *2019 IEEE 11th International Workshop on Computational Intelligence and Applications*, (2019), 89–94. <https://doi.org/10.1109/IWCIA47330.2019.8955006>
13. J. G. Nam, J. Kim, K. Noh, H. Choi, D. S. Kim, S. J. Yoo, et al., Automatic prediction of left cardiac chamber enlargement from chest radiographs using convolutional neural network, *Eur. Radiol.*, **31** (2021), 8130–8140. <https://doi.org/10.1007/s00330-021-07963-1>
14. T. Matsumoto, S. Kodera, H. Shinohara, H. Ieki, T. Yamaguchi, Y. Higashikuni, et al., Diagnosing heart failure from chest X-ray images using deep learning, *Int. Heart J.*, **61** (2020), 781–786. <https://doi.org/10.1536/ihj.19-714>
15. Y. Hirata, K. Kusunose, T. Tsuji, K. Fujimori, J. Kotoku, M. Sata, Deep learning for detection of elevated pulmonary artery wedge pressure using standard chest X-ray, *Can. J. Cardiol.*, **37** (2021), 1198–1206. <https://doi.org/10.1016/j.cjca.2021.02.007>
16. M. Dutt, S. Redhu, M. Goodwin, C. W. Omlin, SleepXAI: An explainable deep learning approach for multi-class sleep stage identification, *Appl. Intell.*, **53** (2023), 16830–16843. <https://doi.org/10.1007/s10489-022-04357-8>

17. S. Jonas, A. O. Rossetti, M. Oddo, S. Jenni, P. Favaro, F. Zubler, EEG-based outcome prediction after cardiac arrest with convolutional neural networks: Performance and visualization of discriminative features, *Human Brain Mapp.*, **40** (2019), 4606–4617. <https://doi.org/10.1002/hbm.24724>
18. C. Barros, B. Roach, J. M. Ford, A. P. Pinheiro, C. A. Silva, From sound perception to automatic detection of schizophrenia: An EEG-based deep learning approach, *Front. Psychiatry*, **12** (2022), 813460. <https://doi.org/10.3389/fpsyt.2021.813460>
19. Y. Yan, H. Zhou, L. Huang, X. Cheng, S. Kuang, A novel two-stage refine filtering method for EEG-based motor imagery classification, *Front. Neurosci.*, **15** (2021), 657540. <https://doi.org/10.3389/fnins.2021.657540>
20. M. Porumb, S. Stranges, A. Pescapè, L. Pecchia, Precision medicine and artificial intelligence: A pilot study on deep learning for hypoglycemic events detection based on ECG, *Sci. Rep-UK.*, **10** (2020), 170. <https://doi.org/10.1038/s41598-019-56927-5>
21. S. Raghunath, A. E. U. Cerna, L. Jing, D. P. vanMaanen, J. Stough, D. N. Hartzel, et al., Prediction of mortality from 12-lead electrocardiogram voltage data using a deep neural network, *Nat. Med.*, **26** (2020), 886–891. <https://doi.org/10.1038/s41591-020-0870-z>
22. H. Shin, Deep convolutional neural network-based hemiplegic gait detection using an inertial sensor located freely in a pocket, *Sensors*, **22** (2022), 1920. <https://doi.org/10.3390/s22051920>
23. G. Aquino, M. G. Costa, C. F. C. Filho, Explaining one-dimensional convolutional models in human activity recognition and biometric identification tasks, *Sensors*, **22** (2022), 5644. <https://doi.org/10.3390/s22155644>
24. R. Ge, M. Zhou, Y. Luo, Q. Meng, G. Mai, D. Ma, et al., Mctwo: A two-step feature selection algorithm based on maximal information coefficient, *BMC Bioinformatics*, **17** (2016), 142. <https://doi.org/10.1186/s12859-016-0990-0>
25. T. Naghibi, S. Hoffmann, B. Pfister, Convex approximation of the NP-hard search problem in feature subset selection, *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (2013), 3273–3277. <https://doi.org/10.1109/ICASSP.2013.6638263>
26. D. S. Hochba, Approximation algorithms for NP-hard problems, *ACM SIGACT News*, **28** (1997), 40–52. <https://doi.org/10.1145/261342.571216>
27. C. Yun, J. Yang, Experimental comparison of feature subset selection methods, *Seventh IEEE International Conference on Data Mining Workshops*, (2007), 367–372. <https://doi.org/10.1109/ICDMW.2007.77>
28. W. C. Lin, Experimental study of information measure and inter-intra class distance ratios on feature selection and orderings, *IEEE T. Syst. Man Cy-S*, **3** (1973), 172–181. <https://doi.org/10.1109/TSMC.1973.5408500>
29. W. Y. Loh, Classification and regression trees, *Data Mining and Knowledge Discovery*, **1** (2011), 14–23. <https://doi.org/10.1002/widm.8>
30. M. R. Osborne, B. Presnell, B. A. Turlach, On the lasso and its dual, *J. Comput. Graph. Stat.*, **9** (2000), 319–337. <https://doi.org/10.1080/10618600.2000.10474883>
31. R. J. Palma-Mendoza, D. Rodriguez, L. de Marcos, Distributed Relieff-based feature selection in spark, *Knowl. Inf. Syst.*, **57** (2018), 1–20. <https://doi.org/10.1007/s10115-017-1145-y>

32. Y. Huang, P. J. McCullagh, N. D. Black, An optimization of Relieff for classification in large datasets, *Data Knowl. Eng.*, **68** (2009), 1348–1356. <https://doi.org/10.1016/j.datak.2009.07.011>
33. R. Yao, J. Li, M. Hui, L. Bai, Q. Wu, Feature selection based on random forest for partial discharges characteristic set, *IEEE Access*, **8** (2020), 159151–159161. <https://doi.org/10.1109/ACCESS.2020.3019377>
34. M. Mori, R. G. Flores, Y. Suzuki, K. Nukazawa, T. Hiraoka, H. Nonaka, Prediction of Microcystis occurrences and analysis using machine learning in high-dimension, low-sample-size and imbalanced water quality data, *Harmful Algae*, **117** (2022), 102273. <https://doi.org/10.1016/j.hal.2022.102273>
35. Y. Omae, M. Mori, E2H distance-weighted minimum reference set for numerical and categorical mixture data and a Bayesian swap feature selection algorithm, *Mach. Learn. Know. Extr.*, **5** (2023), 109–127. <https://doi.org/10.3390/make5010007>
36. R. Garriga, J. Mas, S. Abraha, J. Nolan, O. Harrison, G. Tadros, et al., Machine learning model to predict mental health crises from electronic health records, *Nat. Med.*, **28** (2022), 1240–1248. <https://doi.org/10.1038/s41591-022-01811-5>
37. G. Chandrashekar, F. Sahin, A survey on feature selection methods, *Comput. Electr. Eng.*, **40** (2014), 16–28. <https://doi.org/10.1016/j.compeleceng.2013.11.024>
38. N. Gopika, M. Kowshalaya, Correlation based feature selection algorithm for machine learning, *Proceedings of the 3rd International Conference on Communication and Electronics Systems*, (2018), 692–695. <https://doi.org/10.1109/CESYS.2018.8723980>
39. L. Fu, B. Lu, B. Nie, Z. Peng, H. Liu, X. Pi, Hybrid network with attention mechanism for detection and location of myocardial infarction based on 12-lead electrocardiogram signals, *Sensors*, **20** (2020), 1020. <https://doi.org/10.3390/s20041020>
40. F. M. Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, M. T. Hompel, Convolutional neural networks for human activity recognition using body-worn sensors, *Informatics*, **5** (2018), 26. <https://doi.org/10.3390/informatics5020026>
41. T. Thenmozhi, R. Helen, Feature selection using extreme gradient boosting bayesian optimization to upgrade the classification performance of motor imagery signals for BCI, *J. Neurosci. Meth.*, **366** (2022), 109425. <https://doi.org/10.1016/j.jneumeth.2021.109425>
42. R. Garnett, M. A. Osborne, S. J. Roberts, Bayesian optimization for sensor set selection, *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, (2019), 209–219. <https://doi.org/10.1145/1791212.1791238>
43. E. Kim, Interpretable and accurate convolutional neural networks for human activity recognition, *IEEE T. Ind. Inform.*, **16** (2020), 7190–7198. <https://doi.org/10.1109/TII.2020.2972628>
44. M. Jaén-Vargas, K. M. R. Leiva, F. Fernandes, S. B. Goncalves, M. T. Silva, D. S. Lopes, et al., Effects of sliding window variation in the performance of acceleration-based human activity recognition using deep learning models, *PeerJ Comput. Sci.*, **8** (2022), e1052. <https://doi.org/10.7717/peerj-cs.1052>
45. R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, et al., The opportunity challenge: A benchmark database for on-body sensor-based activity recognition, *Pattern Recogn. Lett.*, **34** (2013), 2033–2042. <https://doi.org/10.1016/j.patrec.2012.12.014>

46. H. Sagha, S. T. Digumarti, J. D. R. Millán, R. Chavarriaga, A. Calatroni, D. Roggen, et al., Benchmarking classification techniques using the opportunity human activity dataset, *2011 IEEE International Conference on Systems, Man and Cybernetics*, (2011), 36–40. doi: 10.1109/ICSMC.2011.6083628
47. A. Murad, J. Y. Pyun, Deep recurrent neural networks for human activity recognition, *Sensors*, **17** (2017), 2556. <https://doi.org/10.3390/s17112556>
48. J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, (2015), 3995–4001.
49. O. Banos, J. M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, *Sensors*, **14** (2014), 6474–6499. <https://doi.org/10.3390/s140406474>
50. T. Tanaka, I. Nambu, Y. Maruyama, Y. Wada, Sliding-window normalization to improve the performance of machine-learning models for real-time motion prediction using electromyography, *Sensors*, **22** (2022), 5005. <https://doi.org/10.3390/s22135005>
51. J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, S. H. Deng, Hyperparameter optimization for machine learning models based on bayesian optimization, *J. Electron. Sci. Technol.*, **17** (2019), 26–40. <https://doi.org/10.11989/JEST.1674-862X.80904120>
52. P. Doke, D. Shrivastava, C. Pan, Q. Zhou, Y. D. Zhang, Using CNN with bayesian optimization to identify cerebral micro-bleeds, *Mach. Vision Appl.*, **31** (2020), 1–14. <https://doi.org/10.1007/s00138-020-01087-0>
53. J. Bergstra, R. Bardenet, Y. Bengio, B. Kegl, Algorithms for hyper-parameter optimization, *Adv. Neural Inf. Process. Syst.*, **24** (2011), 2546–2554.
54. T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, (2019), 2623–2631, <https://optuna.readthedocs.io/en/stable/>. doi: 10.1145/3292500.3330701
55. H. Makino, E. Kita, Stochastic schemata exploiter-based AutoML, *2021 IEEE International Conference on Data Mining Workshops*, (2021), 238–245. <https://doi.org/10.1109/ICDMW53433.2021.00037>
56. P. Siirtola, P. Laurinen, J. Roning and H. Kinnunen, Efficient accelerometer-based swimming exercise tracking, *IEEE SSCI 2011: Symposium Series on Computational Intelligence*, (2011), 156–161. <https://doi.org/10.1109/CIDM.2011.5949430>
57. G. Brunner, D. Melnyk, B. Sigfússon, R. Wattenhofer, Swimming style recognition and lap counting using a smartwatch and deep learning, *2019 International Symposium on Wearable Computers*, (2019), 23–31. <https://doi.org/10.1145/3341163.3347719>

