fNIRS-GANs: data augmentation using generative adversarial networks for classifying motor tasks from functional near-infrared spectroscopy

Tomoyuki Nagasawa¹, Takanori Sato¹, Isao Nambu¹,³ and Yasuhiro Wada¹

¹ Graduate School of Engineering, Nagaoka University of Technology, Nagaoka, Japan
² National Institute of Technology, Akita College, Akita, Japan
³ Author to whom any correspondence should be addressed.
E-mail: inambu@vos.nagaokaut.ac.jp

Keywords: data augmentation, functional near-infrared spectroscopy, brain–computer interface, Wasserstein generative adversarial networks, support vector machine, neural networks

Abstract

Objective. Functional near-infrared spectroscopy (fNIRS) is expected to be applied to brain–computer interface (BCI) technologies. Since lengthy fNIRS measurements are uncomfortable for participants, it is difficult to obtain enough data to train classification models; hence, the fNIRS–BCI accuracy decreases. Approach. In this study, to improve the fNIRS–BCI accuracy, we examined an fNIRS data augmentation method using Wasserstein generative adversarial networks (WGANs). Using fNIRS data during hand-grasping tasks, we evaluated whether the proposed data augmentation method could generate artificial fNIRS data and improve the classification performance using support vector machines and simple neural networks. Main results. Trial-averaged temporal profiles of WGAN-generated fNIRS data were similar to those of the measured data except that they contained an extra noise component. By augmenting the generated data to training data, the accuracies for classifying four different task types were improved irrespective of the classifiers. Significance. This result suggests that the artificial fNIRS data generated by the proposed data augmentation method is useful for improving BCI performance.

1. Introduction

Functional near-infrared spectroscopy (fNIRS) is a neuroimaging technique that measures human brain activity in a non-invasive manner. Using near-infrared lights, fNIRS detects relative concentration changes in oxygenated and deoxygenated hemoglobin (ΔOxy-Hb and ΔDeoxy-Hb, respectively) in the cerebral cortex [1, 2]. Compared with other neuroimaging methods, fNIRS has the advantages of being portable and easy-to-use and is thus used in many areas, including the monitoring of cognitive ability and neonatal brain function, daily life situations, and rehabilitation. One of the applications of fNIRS is brain-computer interfaces (BCIs). BCIs are technologies that enable humans to operate external devices using their brain activity observed with electroencephalograms (EEG) or electrocorticograms as well as fNIRS, which measures hemodynamic activity [3]. Several studies have demonstrated that BCIs can utilize fNIRS [3–9]. Nevertheless, higher classification accuracy is desirable for reliable and useful applications.

A common approach to improve BCI accuracy is to find a better classification model. Therefore, previous studies have tested several classification algorithms and developed methods to achieve higher classification accuracy [4–6]. While these previous studies focused on algorithms, the number of datasets is also an important aspect for achieving high classification accuracy. In general, BCI classification models must be trained using a large amount of data measured by neuroimaging, e.g. fNIRS; however, collecting such large amounts of data is usually difficult, because neuroimaging data measurements are lengthy and burdensome for participants even though fNIRS is easy to use. Therefore, only a limited amount of data can be...
acquired. As a result, classification models cannot be sufficiently trained, which leads to reduced BCI accuracy.

One solution to this problem is data augmentation. Data augmentation is a common approach used in the field of image recognition, where simple manipulations of the original data such as flipping, changing color, and adding noise are known to be effective [10]. Similarly, in BCI research using EEG, several augmentation methods have been proposed to improve the accuracy or reduce the calibration time [11–15].

Here, we aimed to improve fNIRS-BCI accuracy by augmenting fNIRS data using generative models. Generative models that use neural networks, e.g. generative adversarial networks (GANs) [16], are being actively applied to a wide range of computer science research, especially image generation [16–18]. In medical imaging analysis, one previous study succeeded in improving liver detection using GANs [19]. Several data augmentation methods using GANs have also been tested for neuroimaging using EEG [20–23]. A recent study also applied GANs to fNIRS simulation data [24]. However, GAN training is unstable, and the applicability of GANs for real fNIRS data has not been tested yet.

In this study, we propose a fNIRS data augmentation method using Wasserstein GANs (WGANs) [25]. A WGAN is a variant of a GAN and consists of a generator and a critic (figure 1(a)). The generator provides synthetic fNIRS generated data from the random latent variable $z$ so as to become similar to the measured (original) fNIRS data. The critic evaluates the generated data and measures the original data. By repeating the training of each weight for the generator and the critic, the generator can generate similar data to the original data (for details, see section 2.1). This method has been shown to be useful for data augmentation using EEG [20, 21]. Here, we applied fNIRS-GANs to the measured fNIRS datasets [8, 9]. Specifically, the GAN generator was trained using measured datasets (Training process, figure 1(a)), and then we tested whether our proposed data augmentation method could generate artificial fNIRS measurement data and thus improve classification accuracy.

2. Methods

2.1. Data augmentation using WGANs

2.1.1. Augmentation using GANs

A GAN [16] generates artificial data by training two neural networks, the discriminator $D(\cdot)$ and the generator $G(\cdot)$, with an adversarial procedure. The GAN value function $V(D, G)$ is expressed as:

$$V(D, G) = \mathbb{E} \left[ \log D(x) \right] + \mathbb{E} \left[ \log \left( 1 - D\left( G(z) \right) \right) \right],$$

(1)

where $x$ is the training data in the discriminator, $z$ is the latent variable that represents the input noise in the generator, and $\mathbb{E} [\cdot]$ denotes expectation function. In GANs, the discriminator is trained by maximizing $V(D, G)$ and the generator is trained by minimizing $V(D, G)$.

The discriminator, with the parameter $\theta_D$, is updated by increasing its gradient, $g_D$, which is calculated using the training data $\{ x^{(1)}, \ldots, x^{(N)} \}$ and latent variable $\{ z^{(1)}, \ldots, z^{(N)} \}$:

$$g_D = - \nabla_{\theta_D} \frac{1}{N} \sum_{i=1}^{N} \left[ \log D\left( x^{(i)} \right) - \log \left( 1 - D\left( G\left( z^{(i)} \right) \right) \right) \right],$$

(2)

where $N$ is the number of training data samples. On the other hand, the generator, with the parameter $\theta_G$, is updated by decreasing its gradient, $g_G$, which is calculated using the noise $\{ z^{(1)}, \ldots, z^{(N)} \}$:

$$g_G = - \nabla_{\theta_G} \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 - D\left( G\left( z^{(i)} \right) \right) \right).$$

(3)

In other words, the discriminator is trained to maximize the probability of assigning the correct label to both the training data and the generated data. In contrast, the generator is trained to minimize the probability of assigning the correct label to the generated data.

By updating $\theta_D$ several times before updating $\theta_G$, minimizing the value function (1) corresponds to minimizing the Jensen–Shannon divergence. However, the problem is that $g_G$ vanishes because the Jensen–Shannon divergence is not continuous with respect to $\theta_G$. This could stop generator learning. Also, GANs present the problem of mode collapse, as the generator is likely to produce the most frequent value.

2.1.2. Wasserstein GANs

WGANs have been proposed to solve the problem of gradient vanishing [25]. Instead of the Jensen–Shannon divergence, the WGAN minimizes the Wasserstein distance that is continuous and differentiable with respect to $\theta_G$. Also, the WGANs prevent mode collapse. In the WGAN, the discriminator $D$ is referred to as the critic because it does not discriminate the data. The value function $V(D, G)$ is expressed as:

$$V(D, G) = \mathbb{E} \left[ D(x) \right] - \mathbb{E} [D(G(z))].$$

(4)

The critic, with the parameter $\theta_D$, is updated by increasing its gradient, $g_D$, which is calculated using the training data $\{ x^{(1)}, \ldots, x^{(N)} \}$ and latent variable $\{ z^{(1)}, \ldots, z^{(N)} \}$:

$$g_D = - \nabla_{\theta_D} \frac{1}{N} \sum_{i=1}^{N} \left[ D\left( x^{(i)} \right) - D\left( G\left( z^{(i)} \right) \right) \right].$$

(5)

The generator is updated by decreasing $g_G$:

$$g_G = - \nabla_{\theta_G} \frac{1}{N} \sum_{i=1}^{N} D\left( G\left( z^{(i)} \right) \right).$$

(6)

Here, the discriminator is required to take $\theta_D$ within a small range of $[-c, c]$ due to the Lipschitz constraint.
The learning of WGANs is slow compared with that of GANs. Therefore, Gulrajani et al proposed a method that introduces a penalty term (referred to as gradient penalty, GP) to constrain the gradient norm with respect to the output of the critic [26]. By incorporating GP, the value function (4) is:

\[
V(D, G) = \mathbb{E}[D(x)] - \mathbb{E}[D(G(z))] - \lambda \mathbb{E}[\left\| \nabla_x D(\tilde{x}) \right\|_2 - 1]^2
\]

(7)

and the gradient of \( \theta_D \) is written as follows:

\[
\nabla \theta_D \frac{1}{N} \sum_{i=1}^{N} \left[ D(x^{(i)}) - D(\tilde{x}^{(i)}) \right] - \lambda \left[ \left\| \nabla_x D(\tilde{x}^{(i)}) \right\|_2 - 1 \right]^2.
\]

(8)

Here, \( \tilde{x}^{(i)} \) is expressed as:

\[
\tilde{x}^{(i)} = \epsilon x^{(i)} - (1 - \epsilon) G(z^{(i)}),
\]

(9)

and \( \lambda = 10 \) is the gradient penalty coefficient and \( \epsilon \) is a random variable that follows uniform distribution \( U(0, 1) \). We used WGANs that incorporate the penalty (WGAN-GP) [26] to augment the fNIRS data.

2.1.3. The proposed fNIRS data augmentation method using GANs

In this study, both the discriminator and the critic in the WGANs were configured as fully connected

---

Table 1. Network structure dimensions.

| Layer           | Critic | Generator |
|-----------------|--------|-----------|
| Input           | \( N_{\text{time}} \times N_{\text{ch}} + 1 \) | \( N_{\text{ch}} + 1 \) |
| Hidden          | \( N_{\text{time}} \times 1 \) | \( N_{\text{time}} + 1 \) |
| Output          | 1      | \( N_{\text{time}} \times N_{\text{ch}} \) |

Figure 1. Framework of GAN and data augmentation. (a) The training process of GAN. The generator creates the data from the random variables \( z \), and the critic evaluates the generated and original (measured) data. (b) A schematic view of data augmentation. After the training process, the data generated by the generator (referred to as generated data) are combined with the original fNIRS data as augmented data.
feedforward neural networks with three layers (figure 1). Time series data in all fNIRS channels were given as input vectors $x$ to the critic. Random numbers sampled from a uniform distribution $U(-1, 1)$ represented input to the generator $z$ that was a vector with $N_z$ dimension ($=100$). WGANs were constructed for each task condition. In other words, multiple WGANs were used to generate synthetic data (generated data) for each task. Table 1 shows the network structures for each layer in the critic and the generator. $N_{\text{time}}$ and $N_{\text{ch}}$ are the numbers of the time samples ($=300$) and channels ($=41$), respectively. A bias term was added to the input and hidden layers. The activation function for the input and output layers in the critic was the identity function, $f(x) = x$. A sparse gradient causes a vanishing gradient in the critic, and thus a leaky rectified linear unit (ReLU):

$$f(x) = \begin{cases} x & (x \geq 0) \\ 0.2x & (x < 0) \end{cases}$$ (10)

was selected as the activation function for the hidden layer. For the generator, the sigmoid function:

$$f(x) = 1/(1 - \exp(-x))$$ (11)

was selected for the output layer, while the identity function and leaky ReLU were used in the input and hidden layers.

Initial values for each unit in the generator and the critic were random variables from the normal distribution $\mathcal{N}(0, \sigma^2)$ with a mean 0 and variance $\sigma^2$ [27], where $\sigma^2$ is denoted as $\sigma = m^{-1/2}$ using the number of units $m$. The weights between units in the critic and the generator were updated by adaptive moment estimation (Adam) [28]. The recommended parameter values for Adam are $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$ [28]. In addition, the WGAN is trained by updating the weights in the critic five times after updating the weights in the generator once. This WGAN training was repeated 10000 times, and the fNIRS data for all the trials with the training data were used for a single training session.

2.2. Datasets

In this study, we used the data from nine participants that were collected in a previous study [9]. The experiment was approved by the ethical committee of the Nagaoka University of Technology. Data were measured for the nine participants (one left-handed female and eight right-handed males; 21–26 years old) according to the declaration of Helsinki. All the participants submitted a document regarding informed consent. The participants performed a total of 80 trials, conducted in five sessions with 16 trials per session. One trial consisted of a simple block design, including a 5 s pre-rest, 10 s task, and 15 s post-rest. During the task period, the participant either rested or conducted a left-, right-, or bimanual motor task (figure 2(a)). Thus, four different task conditions were set.

The brain activity was measured using a multi-channel fNIRS system (OMM-3000, Shimadzu Corp., Kyoto, Japan) and its sampling period was set to 100 ms. The measured 41 channels covered the sensorimotor areas in both hemispheres (figure 2(b)) so that the center of the source probe surrounded by CH19, 25, 26, and 32 was located at Cz of the international 10–20 system. We also recorded the position of the probes and reference points (nasion, left ear and right ear) with a stylus marker (FASTRAK; Polhemus, Colchester, VT, USA). From these data, we confirmed that the measured channels covered sensorimotor areas (see appendix for details). Note that actual measurements were performed for 45 channels including 4 short-distance channel probes (short channels) [8, 9]; however, we only used 41 normal fNIRS channels. In the previous study, the short channels were used to reduce the effects of artifacts in the superficial layers or scalp [29]. However, to focus on the effects of augmentation, the short channels and reduction of scalp artifacts were not used in the present study.

2.3. Preprocessing

Based on the procedure used in previous studies [8, 9], the measured fNIRS data were preprocessed as follows. First, we performed detrending based on a discrete cosine transform (cut-off period: 60 s). Second, baseline correction was conducted so that the average during the 5 s pre-rest period became 0. Finally, smoothing using a moving average (time window: 3 s) was conducted. The preprocessed time series data were divided into trial data and used as inputs to the classifiers to examine the effects of data augmentation on classification accuracy. The analysis was conducted using custom Python scripts with Chainer (https://chainer.org), Scikit-learn (https://scikit-learn.org), and SciPy (https://www.scipy.org) for the data augmentation and the classification.

2.4. Classification

Next, the effects of data augmentation on classification accuracy in fNIRS-BCIs were evaluated. To see if the effects of data augmentation were dependent on the type of classifier, we used support vector machines and neural networks [30] to classify the following four types of fNIRS data: left-hand grasping, right-hand grasping, bimanual grasping, and resting.

2.4.1. Cross-validation

The accuracy was defined as the number of correctly classified trials divided by the number of trials. This was evaluated by ten-fold cross-validation where all the data were randomly divided into ten sets and one set (10% of the data) was used as test data, while classifiers were trained using eight sets (80% of the data, referred to as the training data), and hyper-parameters (kernel types, penalty parameters for support vector machine (SVM), and the number of epochs for neural networks (NN); see below) were determined with the
The generated data were synthesized from only the training data in the cross-validation set. Using the generator of fNIRS-GANs, 100 trials of the generated data for each task were synthesized (400 trials in total). These generated data were combined with the original training data and used as augmented data (figure 1(b)). To see the effects of data augmentation, we compared the accuracy obtained from the classifiers trained with only the original fNIRS data and with the augmented data. The statistical significance was examined by a paired t-test with the threshold set to 5% ($p = 0.05$). Note that for both original and augmented data, the test data was identical but the training data was changed.

2.4.2. Support vector machines
For SVM classification, we combined binary classifiers into a binary decision tree [31, 32] used in previous studies [8, 9] and performed a four-class classification. The combination of kernel types (linear or Gaussian), penalty parameters, and Gaussian parameters were determined by a grid search. For the linear SVM, a penalty parameter C was searched from $2^{-20}$ to $2^{0}$ with intervals of $2^{0.5}$. For the Gaussian SVM, the penalty parameter $C$ was searched from $2^{0}$ to $2^{10}$ with intervals of $2^{0.5}$, and the Gaussian parameter from $2^{0}$ to $2^{10}$ with intervals of $2^{0.5}$. The parameters that showed the highest accuracy for the validation data were selected as the optimal parameters, which were thus used for test data.

2.4.3. Neural networks
In addition to SVMs, we used fully connected feedforward NNs and calculated their accuracies. We tested NNs with one hidden layer, and multi-class classification was adopted as the classification model. For the activation function of each unit, the sigmoid, identity, and softmax functions were selected for the hidden, input, and output layers, respectively.

Figure 2. Experimental protocol (a) and channel locations (b). The source probe outlined in black was positioned at Cz in the international 10–20 system. Note that short channels (green squares) are not used in this study.
The number of units in the hidden layers was set to 300 (+1 for bias) per layer. The initial value for each weight between units was set to a random variable sampled from the normal distribution, which was the same as used in the WGAN (see 2.1.3). In the training epoch, the cross-entropy error function was defined as follows:

\[
E(w) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_k(n) \log y_k(n),
\]

where \(N\) is the number of the training samples (trials), \(K\) is the number of classes, \(t_k\) is the target (label) data, and \(y_k\) is the output of the NN. The weight \(w\) was updated so that the cross-entropy error function was minimized using the gradient descent:

\[
w^{(t+1)} = w^{(t)} - \eta g,
\]

where \(t\) is the time step (i.e. epoch), \(\eta\) is the learning coefficient set to 0.01, and \(g\) is the gradient of the weight.

During training, the NN iterated epochs by gradient descent using all the training data, and the classification model was evaluated by the validation data. The training iteration was stopped when the cross-entropy error function for the validation data remained the same for ten successive epochs or when 10000 iterations were finished.

3. Results

3.1. Comparison of trial-averaged waveforms

Figure 3 shows the averaged waveforms across trials of the measured (original) and generated data for the left-hemisphere (CH24) and right-hemisphere (CH27) of a representative participant (participant A). To compare both data, we used 20 sets of generated data from each task and averaged across 20 trials for each dataset. The generated data showed a similar temporal form compared to the original fNIRS data. For example, increased activities for the channel contralateral to the moving left hand were observed (CH27 for the left-hand task). Small but similar activity patterns were observed at CH24 for the right-hand task. In addition, the amplitude was large for the difficult bimanual task. On the other hand, the generated data included extra noise components as indicated by the large variances.

3.2. Classification accuracy

Next, we conducted four-class classification analysis of both the measured and generated data using SVMs and NNs. Figure 4 shows the mean classification accuracy across participants calculated by ten-fold cross-validation. The number of generated datasets was 100 for each task, while 16 original datasets for...
each task were used. The classification accuracy greatly improved after data augmentation for both SVMs (0.733) and NNs (0.746), whereas the accuracies for the original data were lower (approximately 0.4; figure 4). The training data using our proposed WGANs increased to approximately 0.7 for both classification methods. The differences between accuracies using both classification methods were comparable but slightly larger for NN than SVM. This suggests that useful data can be acquired using the proposed data augmentation method.

3.3. Effects of the number of augmented data
We examined the effects of the number of augmented data on improving accuracy by changing the number of the augmented data used for training. Figure 5 shows the accuracy changes with increasing numbers of augmented data for each task (class) for both classification methods. In NNs, we found that the accuracy improved largely with increases of 20 samples, and gradual accuracy improvements were observed when the numbers of augmented samples were larger than 20. The highest accuracy improvement was found when the number of augmented samples was 60. While, on average, the accuracy of SVMs gradually increased with increasing numbers of augmented samples, accuracies for all participants seemed to improve by adding 40–60 samples.

4. Discussion
In this study, we aimed to improve the accuracy of fNIRS-BCIs and proposed a fNIRS data augmentation method using WGANs. The classification accuracy significantly improved after data augmentation for real fNIRS data irrespective of the classifiers (SVM or NN), which indicates that the data generated by the proposed method might be useful.
4.1. WGAN, GAN, and other augmentation methods

In this study, we used the WGAN-GP method for data augmentation [26]. WGAN-GP is known to have better convergence properties compared to normal GAN. In fact, we found that parameter tuning was very difficult when using GANs for fNIRS data [24]. Therefore, incorporating WGAN-GP is necessary for stable data augmentation. We would like to note that we have not tested other data augmentation methods other than GAN and WGAN-GP. Simple data augmentation methods such as flipping, changing color, and adding noise are effective for image analysis (e.g. see [10]). In EEG analysis, similar simple data augmentation [13], concatenation of the segments from different trials [11, 12], and empirical mode decomposition [14, 15] has showed improvements of BCI accuracies. Moreover, the GAN-based method has been proposed for time series data [33]. However, it is unclear what kind of manipulation is useful because fNIRS signal properties differ from the image properties and EEG data; therefore, we should examine the effects of simple augmentation techniques as well as more sophisticated methods on fNIRS signal properties in the future.

4.2. Number of samples

One of the most important aspects for performance improvement of data augmentation for fNIRS-BCI is the number of augmented samples for classifier training. It is expected that GANs generate similar data and that overfitting of the training data occurs if little original training data is available. The accuracy improvement is highest when the sample number is approximately 60 for NNs (figure 5(b)) and 80 for SVMs (figure 5(a)), which suggests a limitation for the improvement for our data (16 original training samples for each class). Thus, like the previous reports of EEG data augmentation [13, 15, 21], we confirmed that the augmented data quality and accuracy improvement depend on the number of samples. It should be noted that there is a possibility that GAN-based data augmentation can exacerbate bias present in the original training data [34]. We believe that there was no bias for the class, individuals, gender, or handedness in our study because our data augmentation was done for each individual subject data and each class. Nevertheless, the relationship between data quality, sample number, and the effects of biases contained in the data must be explored further.

4.3. Future work

Currently, we only tested data obtained from a previous study. Applying this method to additional fNIRS data is necessary to elucidate the characteristics of the data augmentation method and fNIRS signals, as well as the process of superficial-layer artifact removal, which is crucial for fNIRS data analysis [29, 35]. In addition, other classifiers and features should be tested in the future. The classifiers used in this study were simple NNs and SVMs, and although other classifiers should be tested in the future, this is beyond the scope of the present study. However, we postulate that other types of complex neural networks (e.g. convolutional neural networks [6]) can be combined with our proposed data augmentation. Similar to the selection of the classifiers, many previous studies have tested different

---

Table 2. Position of each channel in MNI space.

| CH # | X     | Y     | Z     | X     | Y     | Z     |
|------|-------|-------|-------|-------|-------|-------|
| 1    | −30.9 | 4.1   | 35.8  | 4.9   | 49.1  | 2.5   |
| 2    | −10.3 | 2.4   | 38.3  | 2.9   | 57.8  | 1.8   |
| 3    | 14.5  | 2.3   | 38.1  | 2.3   | 57.9  | 2.1   |
| 4    | 35.6  | 3.4   | 35.0  | 2.6   | 49.0  | 3.5   |
| 5    | −43.0 | 2.9   | 22.4  | 5.0   | 51.0  | 3.2   |
| 6    | −21.4 | 2.2   | 25.3  | 3.0   | 62.9  | 1.7   |
| 7    | 3.2   | 3.4   | 25.5  | 2.5   | 63.1  | 1.1   |
| 8    | 25.1  | 2.8   | 25.0  | 1.9   | 62.2  | 2.1   |
| 9    | 46.2  | 2.5   | 22.4  | 2.4   | 50.4  | 3.2   |
| 10   | −51.2 | 2.6   | 9.4   | 4.6   | 47.9  | 3.1   |
| 11   | −35.4 | 1.9   | 13.0  | 2.4   | 62.9  | 1.3   |
| 12   | −13.0 | 0.9   | 13.9  | 1.8   | 70.9  | 0.6   |
| 13   | 15.1  | 1.5   | 13.0  | 1.8   | 71.2  | 0.8   |
| 14   | 37.2  | 2.9   | 12.2  | 2.2   | 62.7  | 1.5   |
| 15   | 55.2  | 3.2   | 7.6   | 3.0   | 47.0  | 3.5   |
| 16   | −61.5 | 2.0   | −8.0  | 4.4   | 41.8  | 3.6   |
| 17   | −45.1 | 2.3   | −3.5  | 5.1   | 59.9  | 1.9   |
| 18   | −23.2 | 1.5   | −0.9  | 2.3   | 72.7  | 0.9   |
| 19   | 0.3   | 2.1   | −2.1  | 1.0   | 72.5  | 0.7   |
| 20   | 24.0  | 2.4   | −2.4  | 2.4   | 73.0  | 1.2   |
| 21   | 48.0  | 2.7   | −5.6  | 3.4   | 60.6  | 2.2   |
| 22   | 64.6  | 2.3   | −9.3  | 3.3   | 41.1  | 4.1   |
| 23   | −55.6 | 1.9   | −20.4 | 3.9   | 56.0  | 2.9   |
| 24   | −36.1 | 1.5   | −17.1 | 2.5   | 71.9  | 1.0   |
| 25   | −14.0 | 0.5   | −14.6 | 0.9   | 78.4  | 0.5   |
| 26   | 13.8  | 1.1   | −16.3 | 1.8   | 78.1  | 0.2   |
| 27   | 37.4  | 2.4   | −18.4 | 2.2   | 72.1  | 1.3   |
| 28   | 58.1  | 2.4   | −22.4 | 3.6   | 55.8  | 2.6   |
| 29   | −64.0 | 2.2   | −36.7 | 4.4   | 44.8  | 4.0   |
| 30   | −48.0 | 2.1   | −31.2 | 2.8   | 65.0  | 1.7   |
| 31   | −25.3 | 1.7   | −30.0 | 1.9   | 74.7  | 0.7   |
| 32   | −1.0  | 3.0   | −31.6 | 1.0   | 77.0  | 1.1   |
| 33   | 24.5  | 2.8   | −32.3 | 2.6   | 76.6  | 1.1   |
| 34   | 48.1  | 2.4   | −33.4 | 3.5   | 64.2  | 3.0   |
| 35   | 65.6  | 1.5   | −38.5 | 4.7   | 44.6  | 4.7   |
| 36   | −57.0 | 2.7   | −48.4 | 3.1   | 51.9  | 3.0   |
| 37   | −37.2 | 2.2   | −45.9 | 2.7   | 69.7  | 1.4   |
| 38   | −14.7 | 1.4   | −44.6 | 1.3   | 79.2  | 0.5   |
| 39   | 13.4  | 1.8   | −44.9 | 2.2   | 79.6  | 0.4   |
| 40   | 35.8  | 3.0   | −48.7 | 4.4   | 70.6  | 1.8   |
| 41   | 56.8  | 2.8   | −51.4 | 5.5   | 52.0  | 3.7   |

CH: channel.  
SD: standard deviation across subjects.
types of features to be classified, including mean, slope [36], and variances [37] (for review, see [3]), while we used time-series data [4, 8, 9] to investigate the effect of data augmentation. These should also be examined in future work.

5. Conclusion

Here, we have proposed a GAN-based data augmentation method for fNIRS-BCIs. Our results have demonstrated that the accuracy of fNIRS-BCIs can be improved using our data augmentation method. While there are many issues to be resolved, including parameter settings, GAN structures, and a more sophisticated classification algorithm efficacy, this study indicates the feasibility of a GAN-based method for BCIs.

Acknowledgments

This study was partly supported by the JSPS KAKENHI (15K12597 and 18H04109), Nagaoka University of Technology Presidential Research Grant, and the Nagai N-S Promotion for Science of Perception. We would like to thank Editage (www.editage.jp) for English language editing.

Appendix

The channel position in MNI space is listed in table 2. To obtain channel positions, we first calculated the midpoint of the source and detector probes measured by the stylus maker (FASTRAK; Polhemus, Colchester, VT, USA). Then we obtained the position of each channel in Montreal Neurological Institute (MNI) space using NFRI tool software [38] and confirmed that the measured channels covered sensorimotor areas (table 2).

ORCID iDs

Isao Nambu https://orcid.org/0000-0002-1705-6268

References

[1] Leff D R, Orihuela-Espina F, Ebwell C E, Athanasiou T, Delpy D T, Darzi A W and Yang G Z 2011 Assessment of the cerebral cortex during motor task behaviours in adults: a systematic review of functional near infrared spectroscopy (fNIRS) studies NeuroImage 54 2922–36
[2] Ferrari M and Quaresima V 2012 A brief review on the history of human functional near-infrared spectroscopy (fNIRS) development and fields of application NeuroImage 63 921–35
[3] Nasier N and Hong K S 2015 fNIRS-based brain–computer interfaces: a review Frontiers Hum. Neurosci. 9 3
[4] Sitaram R, Zhang H, Guan C, Thulasidas M, Hoshi Y, Ishikawa A, Shimizu K and Birbaumer N 2007 Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain–computer interface NeuroImage 34 1416–27
[5] Henrich J, Herff C, Heger D and Schultz T 2015 Investigating deep learning for fNIRS-based BCI Proc. of the Annual Int. Conf. of the IEEE Eng. Med. Biol. Soc. pp 2846–7
[6] Trakaoulwaivan T, Behboodi B, Lee J, Kim K and Choy J-W 2017 Convolutional neural network for high-accuracy functional near-infrared spectroscopy in a brain–computer interface: three-class classification of rest, right-, and left-hand motor execution Neurophotonics 5 011008
[7] Erdogan S B, Oztasrati F, Dilek B, Kadak K S, Hanoglu I and Akin A 2019 Classification of motor imagery and execution signals with population-level feature sets: Implications for probe design in fNIRS based BCI J. Neural Eng. 16 026029
[8] Sato T, Sugai K, Nambu I and Wada Y 2014 Classification of functional near-infrared spectroscopy signals applying reduction of scalp hemodynamic artifact 2014 UKACC Int. Conf. on Control 708–13
[9] Sato T, Nambu I and Wada Y 2017 Effectiveness of scalp-hemodynamics reduction to brain–computer interfaces by functional near-infrared spectroscopy IEEE Trans. Electron. Inf. Syst. 137 717–23
[10] Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet classification with deep convolutional neural networks Adv. Neural Inf. Process. Syst. 1097–105
[11] Lotte F 2011 Generating Artificial EEG Signals To Reduce BCI Calibration Time Proc. 5th Int. Brain–Computer Interface pp 176–9
[12] Lotte F 2015 Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces Proc. IEEE 103 871–90
[13] Freer D and Yang G 2019 Data augmentation for self-paced motor imagery classification with C-LSTM J. Neural Eng. 17 016041
[14] Dinares-Ferran J, Ortner R, Guger C and Solé-Casals J 2018 A new method to generate artificial frames using the Empirical Mode Decomposition for an EEG-based motor imagery BCI Front. Neurosci. 12 308
[15] Zhang Z, Duan F, Sole-Casals J, Dinares-Ferran J, Cichocki A, Yang Z and Sun Z 2019 A novel deep learning approach with data augmentation to classify motor imagery signals IEEE Access 7 19945–54
[16] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets Adv. Neural Inf. Process. Syst. 2672–80
[17] Radford A, Metz L and Chintala S 2016 Unsupervised representation learning with deep convolutional GANs (arXiv: 1511.06434)
[18] Mirza M and Osindero S 2014 Conditional generative adversarial nets (arXiv: 1411.1784)
[19] Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J and Greenspan H 2018 GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification Neurocomputing 321 321–31
[20] Hartmann K G, Schirrmeister R T and Ball T 2018 EEG-GAN: generative adversarial networks for electroencephalographic (EEG) brain signals (arXiv: 1806.01875)
[21] Luo Y and Lu B L 2018 EEG data augmentation for emotion recognition using a conditional wasserstein GAN 2018 Proc. of Ann. Int. Conf. of the IEEE Eng. Med. Biol. Soc. 2535–8
[22] Wang J, Zhong S H, Peng J, Jiang J and Liu Y 2018 Data augmentation for eeg-based emotion recognition with deep convolutional neural networks Lecture Notes in Computer Science 10705 82–93 (Berlin: Springer)
[23] Zhang Q and Liu Y 2018 Improving brain computer interface performance by data augmentation with conditional deep convolutional generative adversarial networks (arXiv: 1806.07108)
[24] Nagasawa T, Sato T, Nambu I and Wada Y 2019 Improving fNIRS-BCI accuracy using GAN-based data augmentation Int. IEEE/EMBS Conf. on Neural Eng. pp 1208–11
[25] Arijovsky M, Chintala S and Bottou L 2017 Wasserstein generative adversarial networks 34th Int. Conf. on Machine Learning, ICML 2017 214–23
[26] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V and Courville A. 2017 Improved training of wasserstein GANs. Montreal institute for learning algorithms. *Adv. Neural Inf. Process. Syst.* 3767–77

[27] LeCun Y A, Bottou L, Orr G B and Müller K R. 2012 Efficient backprop *Neural networks: Tricks of the trade* 9–48 (Berlin: Springer)

[28] Kingma D P and Ba J. 2014 Adam: a method for stochastic optimization (arXiv: 1412.6980)

[29] Sato T, et al. 2016 Reduction of global interference of scalp-hemodynamics in functional near-infrared spectroscopy using short distance probes *Neural Image* 141 120–32

[30] Bishop CM. 2006 *Pattern Recognition and Machine Learning* (New York: Springer)

[31] Fei B and Liu J. 2006 Binary tree of SVM: a new fast multiclass training and classification algorithm *IEEE Trans. Neural Netw.* 17 696–704

[32] Madzarov G, Gjorgjevikj D and Chorbev I. 2009 A multi-class SVM classifier utilizing binary decision tree *Informatica* 33 233–41

[33] Esteban C, Hyland S L and Rätsch G. 2017 Real-valued (Medical) time series generation with recurrent conditional GANs (arXiv: 1706.02833)

[34] Jain N, Manikonda L, Hernandez A O, Sengupta S and Kambhampati S. 2018 Imagining an engineer: on GAN-based data augmentation perpetuating biases (arXiv: 1811.03751)

[35] Tachtsidis I and Scholkmann F. 2016 False positives and false negatives in functional near-infrared spectroscopy: issues, challenges, and the way forward *Neurophotonics* 3 031405

[36] Naseer N and Hong K S. 2013 Classification of functional near-infrared spectroscopy signals corresponding to the right- and left-wrist motor imagery for development of a brain-computer interface *Neurosci. Lett.* 553 84–9

[37] Holper L and Wolf M. 2011 Single-trial classification of motor imagery differing in task complexity: a functional near-infrared spectroscopy study *J. Neuroeng. Rehabil.* 8 1–13

[38] Singh A K, Okamoto M, Dan H, Jurcak V and Dan I. 2005 Spatial registration of multichannel multi-subject fNIRS data to MNI space without MRI *Neural Image* 27 842–51