Convolutional Neural Network Architecture and Input Volume Matrix Design for ERP Classifications in a Tactile P300–based Brain–Computer Interface

Takumi Kodama[1] and Shoji Makino[1]

***Abstract*— In the presented study we conduct the off–line ERP classification using the *convolutional neural network* (CNN) classifier for somatosensory ERP intervals acquired in the full–body tactile P300–based Brain–Computer Interface paradigm (fbBCI). The main objective of the study is to enhance fbBCI stimulus pattern classification accuracies by applying the CNN classifier. A 60×60 squared input volume transformed by one–dimensional somatosensory ERP intervals in each electrode channel is input to the convolutional architecture for a filter training. The flattened activation maps are evaluated by a multilayer perceptron with one–hidden–layer in order to calculate classification accuracy results. The proposed method reveals that the CNN classifier model can achieve a non–personal–training ERP classification with the fbBCI paradigm, scoring 100 $\%$ classification accuracy results for all the participated ten users.**

I. INTRODUCTION

The *brain–computer interface* (BCI) is a human computer–interaction technique that allows communications only using brain activities [1]. In the past decade, BCI have been actively developed to assist the amyotrophic lateral sclerosis (ALS) patients, who have difficulty expressing their intentions or thoughts due to neuro–motor disabilities, since the neurotechnology does not require any muscle movements as inputs. However, BCI has a huge issue for performance results, such as low stimulus pattern classification accuracies. Practical BCI users have been awaited the completion of a high performance BCI paradigm.

To solve the issue, we apply the *convolutional neural network* (CNN) for classifying ERP intervals which taken in a P300–based BCI paradigm to decide the presence or absence of P300 responses in response to external stimuli. The CNN has been recognized as one of the most effective method to resolve the computer vision tasks [2]. The pixel elements of an input volume are convolved with filters in several layers, then *neural networks* (NN) are applied to the output vectors. In the presented study, we optimize the input volume design and convolutional architecture of the CNN classifier aiming to improve a classification performance of the P300–based BCI paradigm.

The dataset applied in the current study is the full–body tactile P300–based BCI (fbBCI) paradigm [3]. The P300–based BCI using the sense of touch (tactile) stimuli has been recognized as one of the alternative paradigms, as it allows us to communicate with the locked–in syndrome (LIS) patients who loses their vision or audition as a late symptom of the ALS disease. So far, however, the tactile BCIs have been considered as a low performance paradigm due to a complexity of stimulus pattern discriminations comparing to visual or auditory BCIs. Though the fbBCI paradigm has adopted spatial stimulus patterns (longer distance in each pattern for the better discriminations), the best classification accuracy in the previous study is 59.83 $\%$ using the non–linear SVM classifier with personal training [4].

Accordingly, the main objective of the presented study is to improve the fbBCI stimulus pattern classification accuracies using the CNN classifier which optimized for the somatosensory ERP intervals. Our hypothesis is that the *somatosensory evoked potentials* (SEP) especially acquired in the fbBCI paradigm would utilize the characteristic of the CNN algorithm because of the strong and contrastive brainwave responses [2], [3]. Consequently, the validities of both proposed input volume design and convolutional architecture of the CNN classifier will be demonstrated with the improved fbBCI classification accuracy results at the end of the paper.

II. METHODS

The fbBCI *electroencephalogram* (EEG) experiment was carried out with ten BCI naive users. Both five healthy males and females participated in the experiment with a mean age of 21.9 years old (SD: 1.45). All the experiments were conducted in a sound proof room at the Life Science Center of TARA, University of Tsukuba, Japan.

The fbBCI paradigm was a six-command tactile BCI. Dayton Audio TT25-16 vibrotactile transducers were placed on a Japanese–style mattress to create six stimulus patterns, which were given to the left arm (pattern number #1), right arm (#2), shoulder (#3), waist (#4), left leg (#5) and right leg (#6) of the user, respectively [3]. Users took the experiments having their body lying down on the mattress and transducers. For easier discriminations of the patterns, each transducer was deployed with enough distances, which were wider than the previous tactile BCI paradigms applying transducers to usual hand [5] or facial area [6].

The stimulus carrier frequencies of the transducers were set at 40 Hz. The stimulus durations were set to 100 ms, whereas inter–stimulus–intervals (ISI) were randomly adjusted from 400 ms to 430 ms to break continuous pattern stimulations. The acquisition duration of ERP intervals was set at 800 ms long after the vibrotactile stimulus onsets. In the fbBCI experiment, users were asked to concentrate on one of the stimulus patterns while the random stimulations in order to evoke a somatosensory P300 response [3]. User's EEG signals were captured by a bio–signal amplifier system g.USBamp from g.tec Medical Engineering GmbH, Austria,

[1]Life Science Center of TARA, University of Tsukuba, Tsukuba, Japan

978-1-5090-2809-2/17/$31.00 ©2017 IEEE

where the sampling frequency was set at 512 Hz. Active wet electrodes were attached to channel Cz, Pz, C3, C4, P3, P4, CP5 and CP6 following the 10/10 extended international system, covering the primary somatosensory and parietal cortices on the scalp.

In the current study, all classifications were conducted in an off–line environment. The ERP intervals recorded in the online experiment were signal pre–processed before generating input volumes for the CNN classifier. The pre–processing began with a bandpass filtering, the passband of which was set at $0.1 \sim 30$ Hz range, to limit interference signals of the vibrotactile transducers. In order to assess an effectiveness of removing background noises, the non epoch averaging and simple moving averaging (SMA) settings were applied to the filtered ERP intervals. SMA was a method to conduct epoch averagings without reducing the number of ERP intervals. The ERP interval length ERP_{len} in each electrode channel was 410, as calculated by $ERP_{len} = \lceil ERP_{dur} \cdot f_s \rceil$, where ERP_{dur} stood for the duration length of the ERP interval (800 ms) in seconds (0.8 s in this study), f_s for the sampling frequency (512 Hz) and $\lceil \cdot \rceil$ denoted integer ceiling function.

After the signal pre–processing, the ERP intervals generated in each electrode channel were transformed into squared matrices to create input volumes for CNN classifier. The input volume consisted of the squared matrices of both eight electrode channels and mean value of all electrodes. The method of generating the input volume was described below:

1) The initial 10 elements of each ERP interval were eliminated to create squared (n × n) matrices. Namely, ERP_{len} was converted from 410 to 400 length.
2) Every 20 elements of each ERP interval were deployed in vertical direction. For instance, the first 20 elements (1∼20) were deployed on the first column, then the subsequent 20 elements (21∼40) were on the second column. Consequently, the 400 elements of each ERP interval were transformed into squared 20 rows × 20 columns matrices.
3) Finally each 20 × 20 matrix was concatenated in a 3 × 3 grid pattern to create 60 rows × 60 columns input volumes. The order of the matrices was Cz, Pz, P3, P4, mean of all electrodes, C3, C4, CP5, CP6 from the top left.

The reason why the ERP elements were deployed in vertical direction to generate an squared matrix was that the filter in the convolution layer could be trained as responding to the vertical and gradient edge patterns [7]. The processes to convert ERP intervals into input volumes were also described in Figure 1.

The concatenated 60 × 60 input volumes were employed for filter trainings in the convolutional architecture of the CNN classifier. The default number of input volumes which generated in an experiment was 60 for targets and 300 for non–targets, though, the non–target input volumes were randomly selected as many as the number of target input volumes (60) to keep a class equivalence. Since the fbBCI conducted six EEG experiments for each ten participated user, $60 \times 6 \times 10 = 3600$ target and 3600 non–target input

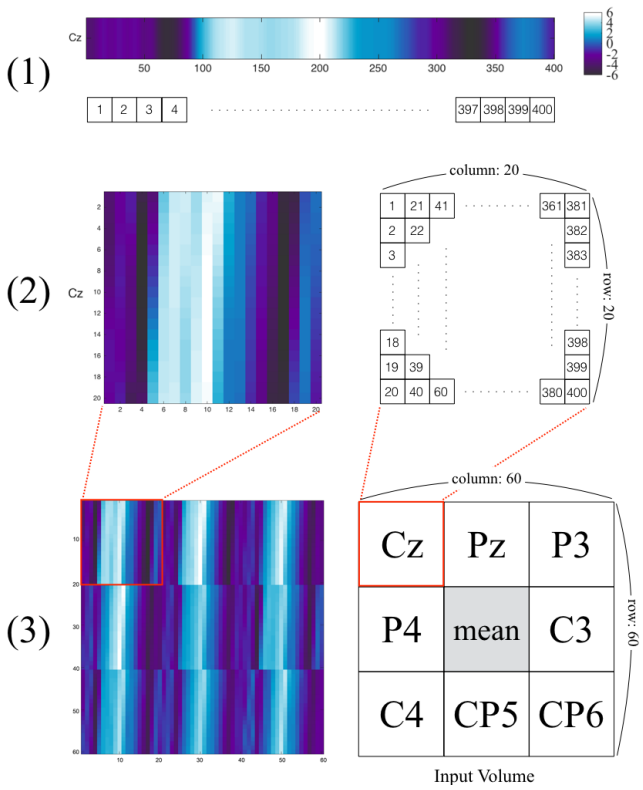

Fig. 1. The generating process of the input volume for the CNN classifier from the signal pre–processed somatosensory ERP intervals in the current study. The each element of the ERP intervals denoted the electrical potential (μV), the value range of which was shown in the heat map. The sample heat maps in the presented figure signified the mean ERP intervals of target stimulus, so the P300 responses were confirmed around elements number 200 (white colored area), namely around $300 \sim 400$ ms ($200/f_s \simeq 0.39$ s). The numbers in boxes described in beside of the heat maps were associated with the elements number. (1) Original length of the ERP interval (ERP_{len}) was 410, but the first 10 elements were removed to create n × n squared matrices. (2) The 400 ERP elements were deployed in 20 × 20. Every 20 elements were placed in vertical direction from top left as shown in the boxes. (3) The 20 × 20 matrices generated in eight electrode channel and mean of all the electrodes were concatenated in a 3 × 3 grid pattern for creating 60 × 60 input volume.

volumes were collected altogether. In order to achieve a non–personal–training ERP classification, the CNN classifier carried out a cross validation between the participated users. For instance, user No.1's input volumes (360 targets vs. 360 non–targets) were evaluated by the classifier model which trained by leftover user No.2 $\sim$ No.10's input volumes ($360 \times 9 = 3240$ targets vs. 3240 non–targets altogether). Accordingly, using the CNN classifier, the fbBCI stimulus pattern classification accuracies could be calculated without any user–specific classifier models.

The convolutional architecture of the CNN classifier which comprised of both two convolution and two pooling layers to classify input volumes was implemented based on LeNet [7]. MXNet [8] was employed as a calculation library. All the input volumes, filters and activation maps were limited to two dimensional in each layer. The original size of input volumes (I_x, I_y) was settled on $(60, 60)$. The amount of zero padding

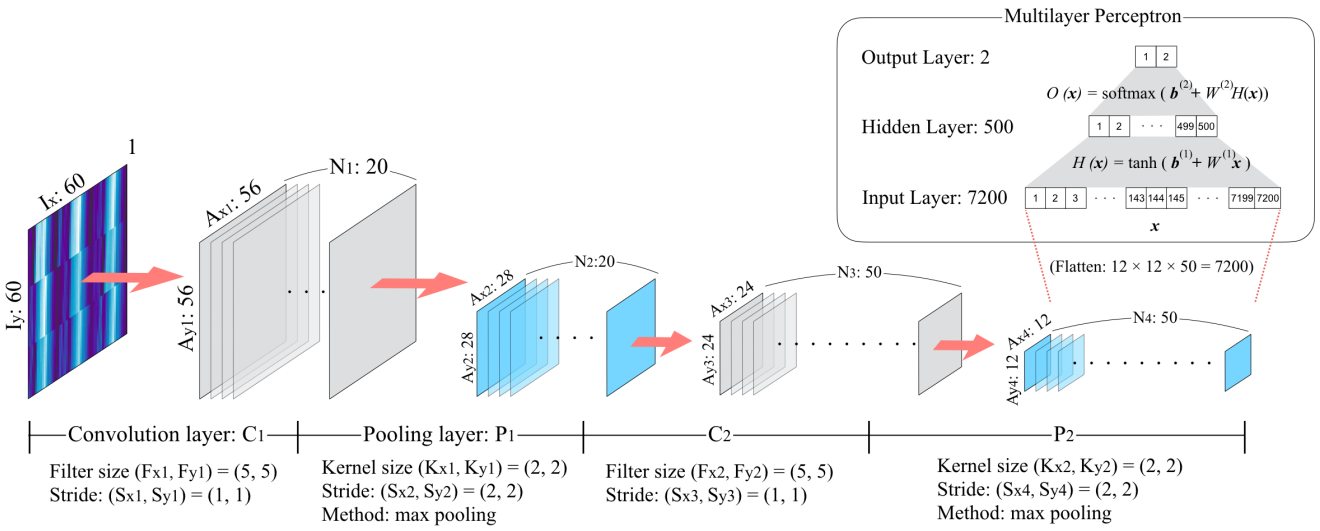

Fig. 2. The overview of the convolutional architecture employed in the fbBCI paradigm. The first convolution layer C_1 generated 20 activation maps $(A_{x1}, A_{y1}) = (56, 56)$ from the input volume matrices $(I_x, I_y) = (60, 60)$. The consequent pooling layer P_1 conducted a max pooling to the previous activation maps, creating new 20 activation maps $(A_{x2}, A_{y2}) = (28, 28)$. At the second convolution layer C_2, 50 activation maps $(A_{x3}, A_{y3}) = (24, 24)$ were created in the same way as the first convolution layer. After that those activation maps were input to the second pooling layer P_2, generating new 50 activation maps $(A_{x4}, A_{y4}) = (12, 12)$. Finally, the activation maps generated in the last layer of the convolutional process were flattened into one–dimensional vectors $\boldsymbol{x}$ (L = 7200) to classify with the one–hidden–layer multilayer perceptron. The flattened vectors $\boldsymbol{x}$ were conducted two calculations $(H(\boldsymbol{x}), O(\boldsymbol{x}))$ in both input and hidden layers by transforming 7200 units into two units at the output layer.

was disabled in every layer. The convolutional architecture which optimized for the fbBCI paradigm was described as follows:

1) **First convolution layer** C_1: The first convolution was conducted in the convolution layer C_1. The number of filters was set at 20, the filter size (F_{x1}, F_{y1}) was $(5, 5)$ and the stride (S_{x1}, S_{y1}) was $(1, 1)$. The size of activation maps (A_{x1}, A_{y1}) was (56, 56) as calculated by $A_n = (I_n - F_n)/S_n + 1$. The number of generated activation maps N_1 was the same as the number of filters (20).
2) **First pooling layer** P_1: The first max pooling was conducted in the pooling layer P_1. The kernel size of max pooling filter (K_{x1}, K_{y1}) was set at $(2, 2)$ as well as the stride (S_{x2}, S_{y2}) at $(2, 2)$. The size of output activation maps as a result of the max pooling could be calculated by $A'_n = \lceil A_n/K_n \rceil$ in case of the kernel size equals to the stride size ($K_n = S_n$). In the pooling layer the number of activation maps were kept intact from the previous convolution layer ($N_1 = N_2$). Consequently, the size of activation maps (A_{x2}, A_{y2}) was $(28, 28)$, the number of which was $N_2 = 20$.
3) **Second convolution layer** C_2: In the second convolution layer, the number of filters were increased from 20 to 50. Since both size of filter (F_{x2}, F_{y2}) and size of stride (S_{x3}, S_{y3}) were same as the first convolution layer C_1, activation maps (A_{x3}, A_{y3}) were $(24, 24)$ as calculated by the same equation. The number of generated activation maps N_3 was raised to 50.
4) **Second pooling layer** P_2: The function of the second pooling layer was the same as the first pooling layer P_1. The number of activation maps N_4 was 50. Both (K_{x2}, K_{y2}) and (S_{x4}, S_{y4}) were adjusted to $(2, 2)$. As P_2 was the final layer of the convolutional process, the activation maps $(A_{x4}, A_{y4}) = (12, 12)$ were flattened into the one–dimensional vector $\boldsymbol{x}$. The length of the flatten vector was calculated by $A_{x4} \times A_{y4} \times N_4 = 12 \times 12 \times 50 = 7200$.
5) **Multilayer perceptron**: After the convolutional process was finished, the flattened vector $\boldsymbol{x}$ was input to the multilayer perceptron with one–hidden–layer. At the input layer, 7200 units were allocated to input the flattened vector $\boldsymbol{x}$ (L = 7200). Then 500 hidden units were deployed for the intermediate hidden layer. The output vector of the hidden layer $H(\boldsymbol{x})$ was obtained as

$$H(\boldsymbol{x}) = \tanh\left(\boldsymbol{b^{(1)}} + W^{(1)}\boldsymbol{x}\right), \tag{1}$$

where $\boldsymbol{b^{(1)}}$ denoted offset vectors and $W^{(1)}$ represented weight matrices between the input and hidden layer. The activation function $\tanh(x) = (\mathrm{e}^x - \mathrm{e}^{-x})/(\mathrm{e}^x + \mathrm{e}^{-x})$ was employed for $H(\boldsymbol{x})$ [2]. Finally two units were employed for the output layer, as the fbBCI paradigm adopted two–class classification (Target or Non–Target). The final output vector of the multilayer perceptron $(O(\boldsymbol{x}))$ was calculated by

$$O(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{b^{(2)}} + W^{(2)}H(\boldsymbol{x})), \tag{2}$$

where both offset vectors $\boldsymbol{b^{(2)}}$ and weight matrices $W^{(2)}$ stood for the valuables between the hidden layer and output layer [2]. $H(\boldsymbol{x})$ signified the output vector of the previous layer by Eqn.(1).

TABLE I

FBBCI PARTICIPATED USERS AND CLASSIFICATION ACCURACY RESULTS

User No.	Gender	Age	non–averaging	SMA
1	F	23	97.22 %	100 %
2	M	23	30.0 %	100 %
3	F	22	72.22 %	100 %
4	M	23	86.11 %	100 %
5	F	20	94.44 %	100 %
6	M	22	88.89 %	100 %
7	M	24	86.11 %	100 %
8	F	20	100 %	100 %
9	M	22	100 %	100 %
10	F	20	41.67 %	100 %
Average.	-	21.6	79.66 %	100 %

TABLE II

CONFUSION MATRIX OF CNN CLASS PROBABILITY RESULTS WITH NON EPOCH AVERAGING

		Predicted condition	
		Non–Target	Target
True condition	Non–Target	**13.5424** %	86.4576 %
	Target	2.5989 %	**97.4011** %

The details of the CNN architecture and multilayer perceptron employed in the fbBCI paradigm was summarized in Figure 2.

III. RESULTS

The fbBCI stimulus pattern classification accuracy results using the CNN classifier were acquired by selecting the highest target class probabilities among the six vibrotactile pattern candidates. The accuracy results of two signal pre–processing settings for generating input volumes, which were non epoch averaging and simple moving averaging (SMA), were summarized in Table I. As shown in the table, the mean classification accuracy of all participated users resulted in 79.66 % with non–averaging and 100 % with SMA, respectively.

The mean class probabilities for both class (Target or Non–Target) and marginal errors of fbBCI ten participated users were described as a confusion matrix with non epoch averaging in Table II and SMA in Table III, respectively. The probability results with non epoch averaging shown in Table II revealed that input volumes of the non-target stimulus pattern often misclassified as the target pattern (around 86 % of them) under the two–class classification. On the other hand, the probability results using SMA accurately classified both classes (almost 100 %) as shown in Table III. The results indicated that the background noises of somatosensory ERP intervals significantly affected the performance of the CNN classifier.

IV. DISCUSSIONS AND CONCLUSIONS

In the presented study the novel analysis of the CNN classifier application for a tactile P300-based BCI paradigm was conducted. The CNN architecture dedicated to classify proposed input volumes which transformed by the somatosensory ERP intervals was developed. Overall, the CNN classifier scored 79.66 % stimulus pattern classification

TABLE III

CONFUSION MATRIX OF CNN CLASS PROBABILITY RESULTS WITH SIMPLE MOVING AVERAGING

		Predicted condition	
		Non–Target	Target
True condition	Non–Target	**99.8756** %	0.1243 %
	Target	0.0565%	**99.9435** %

accuracy with non epoch averaging and 100 % with SMA as the mean of all the participated users without creating user–specific classifier models. The first objective of the study was achieved by the dramatically improved classification accuracies as the previous result was 59.83 % with personal training.

In the future study, to implement the proposed methods for the online experimental environment would be the primary task, as these ERP classifications were conducted under the off–line environments in the current study. ALS patients have expected the development of a high performance and user–friendly BCI paradigm in practical environments.

In conclusion, we could confirm effectivenesses of our methodology with high performance classification accuracy results without personal trainings. We expect that these findings will contribute to the development of P300–based BCI paradigms and also improve the life quality of ALS patients.

REFERENCES

[1] J. Wolpaw and E. W. Wolpaw, Eds., *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, 2012.

[2] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[3] T. Kodama, S. Makino, and T. M. Rutkowski, "Tactile brain-computer interface using classification of p300 responses evoked by full body spatial vibrotactile stimuli," in *Asia-Pacific Signal and Information Processing Association, 2016 Annual Summit and Conference (APSIPA ASC 2016)*, APSIPA. IEEE Press, December 2016, p. Article ID:176.

[4] T. Kodama, K. Shimizu, S. Makino, and T. M. Rutkowski, "Full–body tactile p300–based brain– computer interface accuracy refinement," in *Proceedings of the International Conference on Bio-engineering for Smart Technologies (BioSMART 2016)*, bioSMART. IEEE Press, December 2016, pp. 20–23.

[5] K. Shimizu, S. Makino, and T. M. Rutkowski, "Inter–stimulus interval study for the tactile point–pressure brain–computer interface," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE Engineering in Medicine and Biology Society. IEEE Press, August 25–29, 2015, pp. 1910–1913. [Online]. Available: http://arxiv.org/abs/1506.04458

[6] H. Mori, Y. Matsumoto, Z. R. Struzik, K. Mori, S. Makino, D. Mandic, and T. M. Rutkowski, "Multi-command tactile and auditory brain computer interface based on head position stimulation," in *Proceedings of the Fifth International Brain-Computer Interface Meeting 2013*. Asilomar Conference Center, Pacific Grove, CA USA: Graz University of Technology Publishing House, Austria, June 3-7, 2013, p. Article ID: 095. [Online]. Available: http://castor.tugraz.at/doku/BCIMeeting2013/095.pdf

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274*, 2015.