

INTRACORTICAL ACTIVITY DECODING OF MOTOR IMAGERY BASED ON DEEP CONVOLUTIONAL NEURAL NETWORK: A PILOT STUDY

Duo Chen¹, Rosa So², Yi Ding¹, Neethu Robinson¹, Cuntai Guan¹

¹School of Computer Science and Engineering, Nanyang Technological University, Singapore

²Institute for Infocomm Research, A*STAR, Singapore

E-mail: chenduo@ntu.edu.sg (Duo Chen)

ABSTRACT: An intracortical activity decoding algorithm based on convolutional neural network (CNN) is proposed for classifying the primate intracortical activity of motor imagery under a series of movement tasks. An intracortical brain-computer interface allows the subjects to continuously drive a mobile robot using its brain activity from the motor cortex. Under the leave-one-day-out cross validation, the CNN-based algorithm achieves reliable classification performance in all the four movement tasks, including, Forward, Left, Right, and Stop. Experimental results indicate that the Cohen's kappa of the two subjects are both above 0.4. Additionally, the comparison shows that the CNN-based method is significantly better than the linear discriminant analysis (LDA) which is a state-of-the-art decoding approach.

INTRODUCTION

Brain-computer interface (BCI), which can provide a new communication channel to humans, has received increasing attention in recent years [5]. BCI allows one to control external devices through direct brain activity recognition by a computer [1]. The use of invasive electrocorticographic (ECoG) signals for BCI applications draws numerous interests recently [13]. ECoG directly records signals originating from brain tissues beneath the electrode surface, thus it has much higher spatial resolution compared with EEG [18].

Hence, the motor imagery based BCI system is based on the fact that there will be a change of activation in certain areas of the brain when a subject imagines movement any part of their body [9]. For example, when a person imagines moving his/her right arm, there will be a desynchronization of neural activities in the primary motor cortex of the left brain [12].

In recent years, researchers have increasingly focused on BCI systems for different practical areas, e.g., brain-wave controlled robot [6]. Multi-array electrodes implanted in motor image area can record stable neural activity for achieving various kinds of objectives in BCI system [8]. With invasive or noninvasive recording method, the BCI uses the neural activity of the brain to control effectors such as robotic arm or wheel chair. The core components of a BCI system are brain signal acquisition, pre-

processing, feature extraction, classification, translation and feedback control of external devices [7].

Convolutional neural network (CNN) is a multilayer perceptron with a special topology and contains several hidden layers [12]. CNN has been employed to solve certain computer vision problem, e.g., medical image processing [15]. There is also increasing interest in using deep CNN for end-to-end neural signal analysis. Brain activity decoding can be improved by using batch normalization, dropout and exponential linear units implies which are generally used in deep learning [14].

In this work, we proposed a CNN-based decoding algorithm classify the intracortical activities in the motor cortex of primate under a series of movement tasks, including, "Forward", "Left", "Right", and "Stop". With the robust algorithm, the monkey can control the robot in 360 degree using its brain signal. The decoding problem is transferred into a 4-labels classification problem for annotating long-term activity automatically. Since the cortical activity may vary among subjects and the day-to-day variation may affect the algorithm performance [2], we constructed a robust subject-specific movement decoding algorithm based on CNN using the leave-one-day-out strategy.

METHOD

Data acquisition: The dataset used in this work was collected at A-Star I2R, Singapore, consists of intracortical activities from two healthy monkeys (*Macaca Fascicularis*) [6]. Figure 1 demonstrates the experimental procedure. A total of 96-channel multielectrode array was implanted into the hand/arm region of the left primary motor cortex. Then, the monkeys were first trained to control the platform using a joystick. Specific cues with food stimuli were then given, ordering the monkeys to move toward a certain direction, which is "Forward", "Left", "Right", or "Stop". If the monkey was successful in the trial, the food would be given as a reward. When the monkey can handle the robot skillfully to reach the given targets, the joystick would be disconnected from the platform. Since the joystick was disconnected, the monkey can only control the robotic platform using its intracortical activities. The recorded intracortical activities were transferred to the computational center for fur-

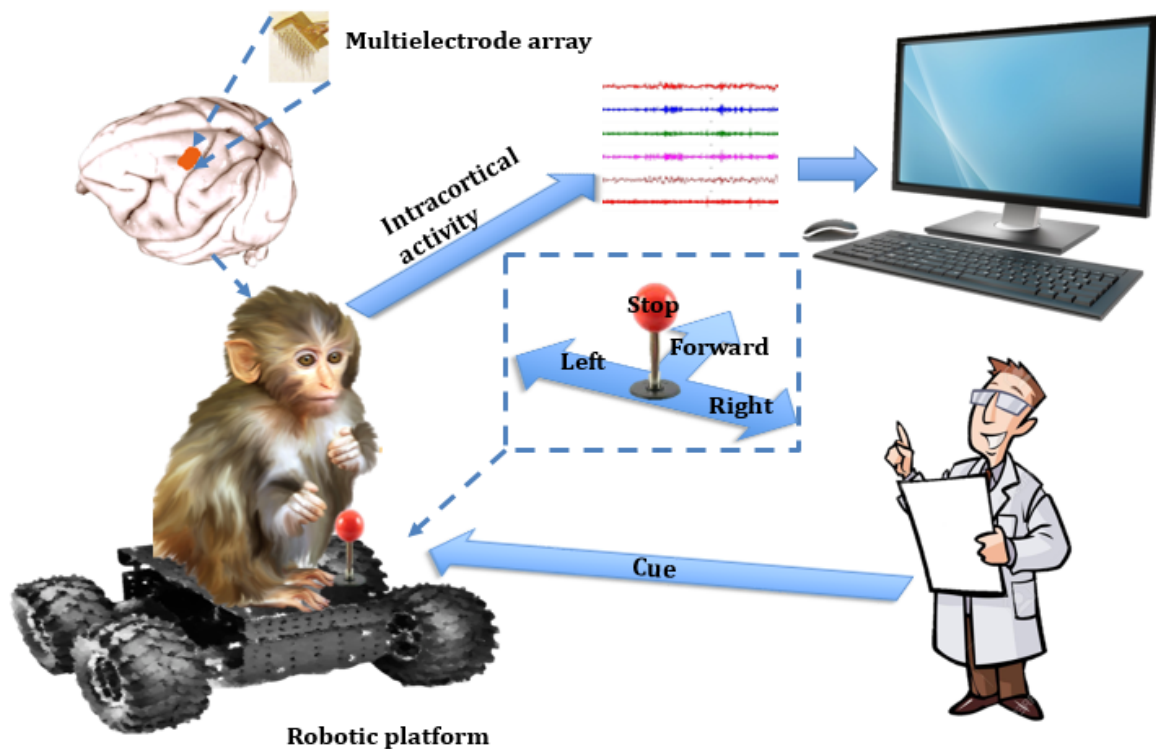


Figure 1: **Experiment Setup.** The implanted array in the hand/arm region records the intracortical activity during the movement tasks. If the monkey moves in the direction correctly, a food reward will be given. The disconnected joystick is treated as the ground truth of the moving willing from the monkey.

ther analysis. Only the data of joystick disconnected were used in the classification problem.

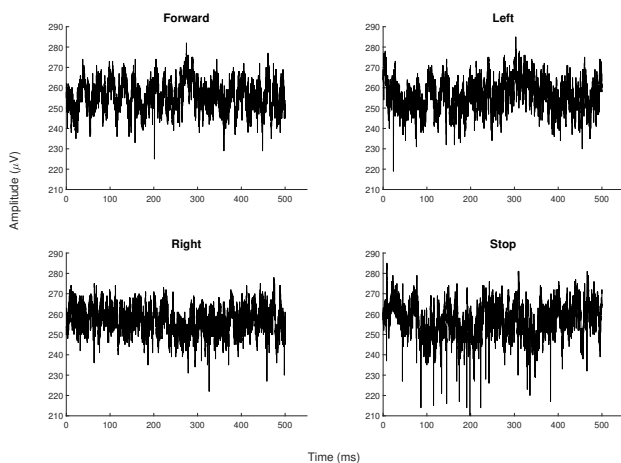


Figure 2: **Examples of the intracortical activities belong to the four movement tasks.**

The intracortical activities were sampled at 12987Hz . Monkey A provided 6 days' data with 1-3 sessions in each day over 10 weeks, depending on how well the monkey cooperated with the trainers. Similarly, Monkey B provided 9 days' data with 1-3 sessions in each day over 10 weeks. Each session lasted about 350 seconds containing 14-16 moving trials. Figure 2 demonstrates an ex-

ample of the intracortical activity segment for each of the four movement tasks. Each subfigure is a 500ms single-channel intracortical activity segment belongs to a certain movement task. For space sake, only one channel is selected for the illustration.

Preprocessing: The raw data were first filtered through two FIR filters. From a biophysics perspective, much of the cortical activity is known as the origins of the local field potential (LFP). In this work, the intracortical activity in a low frequency band is treated as the LFP for the feature extraction.

High-frequency signals have been proved as putative biomarkers of certain neurological disorder. In BCI system, firing rate of the high-frequency brain wave is usually used as an effective marker for decoding the neural activity. The intracortical activity in a high frequency band is used for extracting the firing rate as a component in the feature matrix.

The features from LFP and high-frequency band are extracted separately and combine together to construct the feature matrix. For the CNN-based classification, the feature matrix is used as the 2D input.

To extract the features from LFP, a 2048-order FIR filter was used with the pass band at $[0.1, 128]\text{Hz}$. To extract the features from high frequency band, a 256-order FIR filter was used with the pass band at $[300, 3000]\text{Hz}$.

After filtering the signal into low and high frequency bands, the intracortical activities were divided into multi-channel segments using a sliding window. The segmenta-

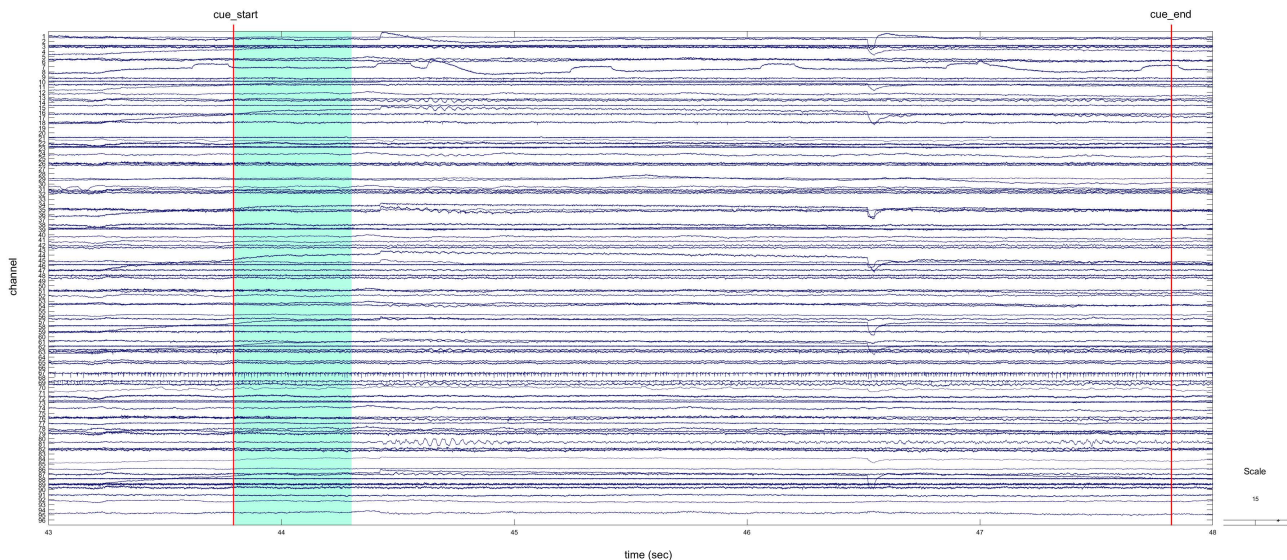


Figure 3: **Segments extracted from long-term multi-channel intracortical activities.** The illustration belongs to a “Right” trial. The time 43.79s is the start cue (cue_start), while the time 49.47s is the end cue of the same trial. The green block is a 500ms slide window with 400ms overlap. In each step, the slide window will move 100ms while the 96-channel signal inside will be treated as one intracortical activity segment.

tion process of one “Right” trial is illustrated in Figure 3. The time 43.79s is the start cue of a right movement task (cue_start), and the time 49.47s is the end cue of the same trial. The green block is a 500ms slide window with 400ms overlap. In each step, the slide window moves 100ms while the 96-channel signal inside is chopped as one cortical activity segment being labelled according to the movement task. With the same sliding window, all the trials of the four movement tasks are transferred into multi-channel intracortical activity segments.

Problem formulation: The purpose of this study is identifying intracortical activity in motor area triggered by various movement tasks, including, “Forward”, “Left”, “Right”, and “Stop”.

After the preprocessing, the research problem is transferred into a 4-label classification problem. Algorithm 1 illustrates the process of the intracortical activity decoding.

In the feature extraction, features based on wavelet transform and event-related desynchronization (ERD)/synchronization (ERS) are extracted from LFP while the firing rate is extracted from the high-frequency band. Then, the wavelet feature, ERD/ERS, and firing rate construct the feature matrix.

In the leave-one-day-out cross validation, the feature matrix is treated as a 2D input of the CNN.

Details of feature extraction and CNN-based classification will be given in the following subsections.

Wavelet feature: As a powerful time-frequency analysis toolbox, DWT decomposes a signal into a series of coefficients and features can be extracted from the coefficients to represent the properties of the signal. “sym2” is employed as the mother wavelet here. The decomposition level is set to 5 producing 5 detail bands and 1 approximation band. For each band, 9 features as listed below

are extracted to construct the wavelet component in the feature matrix.

1. Max: the maximum coefficient
2. Min: the minimum coefficient
3. Mean: the average of coefficients
4. STD: the standard deviation of coefficients
5. Skewness: the skewness of coefficients
6. Kurtosis: the Kurtosis of coefficients
7. Energy: the squared sum of coefficients
8. nSTD: normalized STD, $\frac{STD}{Max-Min}$
9. nEnergy: normalized energy, $\frac{Energy}{band\ length}$

Firing rate: The signals in [300, 3000]Hz are used for spike detection. Spikes are detected using an automated threshold-crossing criterion selected for each channel. The threshold (Thr) for spike detection follows the formula [11]:

$$Thr = 4\delta_n; \delta_n = median\left\{\frac{|x|}{0.6745}\right\} \quad (1)$$

, where x is the filtered signal, and δ is an estimate of the standard deviation of the background noise.

The spike detection are operated in the negative quadrants. Amplitude lower than $-Thr$ will be marked as a spike. For each channel, the number of spikes is recorded as the firing rate in the corresponding channel position.

ERD/ERS: An internally or externally paced event results not only in the generation of an event-related potential (ERP) but also in a change in the ongoing EEG/MEG

Algorithm 1 Pseudocode for CNN-based intracortical activity decoding

Input: Activities of n Days

Output: Cohen’s kappa, precision, recall, f1-score for each movement task

```

1: Initialization: Kappa = zeros((n,1)), precisionAll, recallAll, f1-scoreAll = zeros((n,4,3)), zeros((n,4,3)), zeros((n,4,3))
2: Read intracortical activity datasets
3: Filter data into low and high frequency bands
4: slide data in low and high frequency bands into 500 ms segments with 400ms overlap
5: Extract (wavelet, ERD/ERS, firing rate) features ▷ feature extraction
6: for day in (1 : n) do ▷ leave-one-day-out cross validation
7:   Set up  $X_{train}, Y_{train}, X_{test}, Y_{test}$  for the  $i$ th day
8:   batchSize = 128; inputRows, inputCols = 96, 56
9:   lossFun = “sparse_categorical_crossentropy”
10:  targetNames = [‘Forward’, ‘Right’, ‘Left’, ‘Stop’]
11:  model = Sequential()
12:  model.add(Conv2D(100, (3,3), strides = (1,1), input_shape = (inputRows, inputCols,1)))
13:  model.add(Activation(actiFun))
14:  for epoch in (1 : 10) do
15:    model.add(Conv2D(100, (3,3), strides = (1,1), padding = “same”))
16:    model.add(Activation(“ReLU”))
17:    model.add(MaxPooling2D(pool_size = (2, 2)))
18:    model.add(Dropout(0.5))
19:    model.add(Flatten())
20:    model.add(Dense(56))
21:    model.add(Activation(“ReLU”))
22:    model.add(Dropout(0.25))
23:    model.add(Dense(4))
24:    model.add(Activation(“softmax”))
25:    model.compile(loss = lossFun, optimizer = “sgd”, metrics = “accuracy”)
26:    model.fit(X_train, Y_train, batch_size = batchSize, epochs = 50)
27:    Y_predict = model.predict_classes(X_test)
28:    Kappa[day] = cohen_kappa_score(Y_test, Y_predict)
29:    precisionAll[day], recallAll[day], f1-scoreAll[day] = classification_report(Y_test, Y_predict, target_names = targetNames, output_dict = True)
30: Cohen’s kappa, precision, recall, f1-score = mean(Kappa), mean(precisionAll), mean(recallAll), mean(f1-scoreAll)
31: return Cohen’s kappa, precision, recall, f1-score

```

in form of an event-related desynchronization (ERD) or event-related synchronization (ERS). In this work, ERD/ERS are extracted from the intracortical activity as one component in the feature matrix for the classification. The ERD/ERS follows the definition in [10] as

$$ERD/ERS = \frac{A - R}{R} \times 100\% \quad (2)$$

, where A is the power of signal within the frequency band of interest, R is the power of signal as the preceding baseline in the reference period. Here the reference signal is selected from 2.2 to 0.2 second before each movement cue.

CNN structure: Deep learning is a promising avenue for big data analysis. Some state-of-the-art deep learning algorithms, such as deep neural networks and deep CNN have been successfully applied to image classification [4]. Recently, referenced from previous work in computer vision, CNN has been increasingly employed for EEG signal classification [14]. Here we use the deep CNN as the classifier to decode the intracortical activities under the 4-label movement tasks.

Keras 2.2.4 (<https://keras.io/>) is used to construct the CNN network. Details of the network structure are shown in Algorithm 1.

Leave-one-day-out cross validation: Previous work proved that motor intracortical spiking activity under different movement tasks are easily-classified in one day. However, the problem turns into tricky when facing a inter-day timeline [17]. The mental state, physiological state, individual difference, etc, may affect the motor cortical activity of the subject [2]. In the real practice, a re-calibration for classification on each day will be very time-consuming. Even if the re-calibration can guarantee the prediction performance, ignoring the pre-recorded signals will be quite a waste, abandoning valuable patterns may exist across days.

Considering the individual and the day-to-day differences, the subject-specific leave-one-day-out cross validation is employed for both monkeys. In each step of the cross validation, the feature matrix of one day is treated as the test set, while all the left days construct the training set.

Classification evaluation: Precision, recall, f1-score, and Cohen’s kappa are employed for the classification evaluation. Accuracy is not employed here since in daily real action, it is impossible to make a perfect data balance of the 4 movement tasks. The majority of one or more classes may produce a fake high or low accuracy. In contrast, Cohen’s kappa are much robust than simple percent agreement.

The definition of Cohen’s kappa is:

$$k = \frac{p_o - p_e}{1 - p_e} \quad (3)$$

, where p_o is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and p_e is the expected agreement when both annotators assign labels randomly. p_e is estimated using a

Table 1: Decoding Performance of CNN on the Two Monkeys.

	Monkey A ($k = 0.4077 \pm 0.1863$)			Monkey B ($k = 0.4754 \pm 0.1297$)		
	precision	recall	f1-score	precision	recall	f1-score
Forward	0.5471 ± 0.1508	0.5017 ± 0.2814	0.4916 ± 0.1972	0.5732 ± 0.1776	0.4640 ± 0.3280	0.4227 ± 0.2016
Left	0.5401 ± 0.2095	0.4065 ± 0.2863	0.4235 ± 0.2162	0.7499 ± 0.2032	0.6719 ± 0.1852	0.6685 ± 0.1376
Right	0.5962 ± 0.2308	0.6832 ± 0.1445	0.5805 ± 0.0966	0.7236 ± 0.1544	0.5442 ± 0.2799	0.5416 ± 0.2028
Stop	0.8271 ± 0.1564	0.5996 ± 0.3142	0.6025 ± 0.2524	0.7179 ± 0.1287	0.7287 ± 0.2235	0.6925 ± 0.1335
Average	0.6276 ± 0.1869	0.5477 ± 0.2566	0.5245 ± 0.1906	0.6912 ± 0.1660	0.6022 ± 0.2542	0.5813 ± 0.1689

Table 2: Decoding Performance of LDA on the Two Monkeys.

	Monkey A ($k = 0.3288 \pm 0.1374$)			Monkey B ($k = 0.4371 \pm 0.1281$)		
	precision	recall	f1-score	precision	recall	f1-score
Forward	0.3287 ± 0.1436	0.2596 ± 0.1803	0.2719 ± 0.1837	0.5632 ± 0.1372	0.3406 ± 0.2621	0.3646 ± 0.2177
Left	0.5830 ± 0.1743	0.5651 ± 0.3044	0.4780 ± 0.2045	0.7693 ± 0.1813	0.5895 ± 0.2846	0.5987 ± 0.2035
Right	0.6184 ± 0.1185	0.4666 ± 0.1869	0.5058 ± 0.1315	0.6614 ± 0.2130	0.5500 ± 0.2715	0.5032 ± 0.1952
Stop	0.6354 ± 0.1577	0.6732 ± 0.2363	0.5957 ± 0.1144	0.6264 ± 0.1178	0.8001 ± 0.2071	0.6751 ± 0.1044
Average	0.5414 ± 0.1485	0.4912 ± 0.2270	0.4629 ± 0.1585	0.6551 ± 0.1623	0.5700 ± 0.2538	0.5354 ± 0.1802

per-annotator empirical prior over the class labels [3]. If the raters are in complete agreement then $k = 1$. If there is no agreement among the raters other than what would be expected by chance, $k = 0$.

In this work, Cohen’s kappa is calculated across days in the leave-one-day-out cross validation. For each subject, the performance of CNN in each class is evaluated under precision, recall, and f1-score. Taking class “Forward” for instance, suppose all “Forward” segments are treated as “positive”, then, segments belong to “Left”, “Right”, and “Stop” are clustered into “negative”. Therefore, the classifier had 4 possible outcomes: 1. True positive (TP); 2. False positive (FP); 3. True negative (TN); 4. False negative (FN). The definitions of precision, recall, and f1-score are as follows:

1. $f1 - score = \frac{2TP}{2TP+FP+FN}$;
2. $precision = \frac{TP}{TP+FP}$;
3. $recall = \frac{TP}{TP+FN}$;

RESULTS AND DISCUSSION

Here we exhibit the algorithm outcomes and briefly give some heuristics of using the CNN-based intracortical activity decoding algorithm.

Table 1 lists the algorithm performance on the two monkeys. Cohen’s kappa is calculated across days in the leave-one-day-out cross validation. The *mean* \pm *std* of Cohen’s kappa is in the first line. Results indicate that the decoding algorithm produces Cohen’s kappa above 0.4 on both monkeys leading to a moderate agreement between the true and the predicted labels of the test data set. To zoom in each of the four movement tasks, for each subject, the performance of CNN is evaluated under precision, recall, and f1-score. The *mean* \pm *std* of the cross validation is listed corresponding to “Forward”, “Left”, “Right”, and “Stop”. The last line in Table 1 is the average values of precision, recall, and f1-score of the two monkeys.

All the recalls are much higher above the chance level (25.00%). In most cases, the precision, recall, and f1-score are close to or even above 50.00%. The recall in class “Left” of Monkey A gets the lowest value 40.65%, meaning the decoding algorithm more occasionally labelled the positive segments into negative than it performs in the other cases. The high FN also creates the relatively lower f1-score, compared with that in the other three classes. Similarly, the f1-score of class “Forward” on Monkey B is much lower than that in the other classes. This indicates that the individual difference may affect the performance of the CNN-based decoding algorithm. Furthermore, for each subject, the computational results in the four classes reflect the class difference when stripping one from the others. Taking Monkey B for example, the decoding algorithm obviously performs much better in class “Stop” than in “Forward”, “Left”, or “Right”. The precision and recall are both above 70.00% while the f1-score is the highest (69.25%) across all the cases. This suggests that it is relatively easier for the algorithm to make a distinction for class “Stop” than the other classes on Monkey B.

For space sake, we do not illustrate the algorithm performance of each day in the leave-one-day-out cross validation. The standard deviation can partially indicate the day-to-day difference. Both monkeys produce the standard deviation above 10.00% in all the four movement tasks, except the f1-score of class “Right” on Monkey A. This indicates the existence of high day-to-day difference of the intracortical activity on both monkeys.

For the comparison with the state-of-art method for motor imagery decoding, Table 2 illustrates the computational results under linear discriminant analysis (LDA) [16]. In the same feature space, the CNN-based decoding algorithm generates better overall output than LDA. For Monkey A, the average precision, recall, and f1-score got an improvement at .0862, .0565, and .0616, respectively. For Monkey B, the average precision, recall, and f1-score got an improvement at .0361, .0322, and .0459, respectively.

Overall, on both Monkey A and B, the CNN-based de-

coding algorithm produced stable classification results. The average recall is much higher than the chance level while the precision and f1-score are all above 50.00%. Since the method generates robust performance in overcoming the day-to-day and the individual differences, it might be useful in certain BCI system like, brain-wave controlled robot/wheelchair. It is worth to mention that the current precision and recall here are not sufficient for control of the actuator in real practise which leaves much space for decoding performance improvement. In our future work, we will modify the CNN structure and try other classifiers to improve the decoding accuracy.

CONCLUSION

Using multielectrode array implanted in the motor cortex, the BCI system allows the subject to continuously drive a mobile robot in four directions with intracortical activity from the motor cortex. In this work, we present a CNN-based decoding algorithm for classifying the intracortical activities from the motor cortex under a series of movement tasks, including, Forward, Left, Right, and Stop. Considering the individual and the day-to-day differences, we make use of the leave-one-day-out cross validation to attain more robust and thus generalizable results. Experimental results indicate the reliable performance of the CNN-based multi-label classifier. The Cohen's kappa of the two subjects are both above 0.4. The average precision, recall, and f1-score of the 4 classes are all above 50.00%, where the highest is 69.12% and the lowest is 52.45%. Compared with LDA, the new decoding algorithm yields better classification results. In our future work, we will modify the CNN structure and try other classifiers to further improve the decoding outcome.

References

- [1] Birbaumer Niels. Breaking the Silence: Brain-Computer Interfaces (BCI) for Communication and Motor Control. *Psychophysiology*. 2006;43(6):517–532.
- [2] Christensen James C., Estep Justin R., Wilson Glenn F., Russell Christopher A. The Effects of Day-to-Day Variability of Physiological Data on Operator Functional State Classification. *NeuroImage*. 2012;59(1):57–63.
- [3] Cohen Jacob. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*. 1960;20(1):37–46.
- [4] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep Residual Learning for Image Recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [5] Lebedev Mikhail A., Nicolelis Miguel A.L. Brain–Machine Interfaces: Past, Present and Future. *Trends in Neurosciences*. 2006;29(9):536–546.
- [6] Libedinsky Camilo et al. Independent Mobility Achieved through a Wireless Brain-Machine Interface. *PLOS ONE*. 2016;11(11):1–13.
- [7] McFarland Dennis J., Sarnacki William A., Vaughan Theresa M., Wolpaw Jonathan R. Brain-Computer Interface (BCI) Operation: Signal and Noise during Early Training Sessions. *Clinical Neurophysiology*. 2005;116(1):56–62.
- [8] Miller Kai J., Schalk Gerwin, Fetz Eberhard E., Nijssen Marcel den, Ojemann Jeffrey G., Rao Rajesh P.N. Cortical Activity During Motor Execution, Motor Imagery, and Imagery-Based Online Feedback. *Proceedings of the National Academy of Sciences*. 2010.
- [9] Park C., Looney D., Rehman N. ur, Ahrabian A., Mandic D. P. Classification of Motor Imagery BCI Using Multivariate Empirical Mode Decomposition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2013;21(1):10–22.
- [10] Pfurtscheller G., Neuper C. Motor Imagery and Direct Brain-Computer Communication. *Proceedings of the IEEE*. 2001;89(7):1123–1134.
- [11] Quiroga R. Quian, Nadasdy Z., Ben-Shaul Y. Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. *Neural Computation*. 2004;16(8):1661–1687.
- [12] Sakhavi S., Guan Cuntai, Yan S. Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*. 2018;1–11.
- [13] Schalk G., Leuthardt E. C. Brain-Computer Interfaces Using Electrocorticographic Signals. *IEEE Reviews in Biomedical Engineering*. 2011;4:140–154.
- [14] Schirmer Robin Tibor et al. Deep Learning with Convolutional Neural Networks for EEG Decoding and Visualization. *Human Brain Mapping*. 2017;38(11):5391–5420.
- [15] Shin H. et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*. 2016;35(5):1285–1298.
- [16] Vidaurre C., Kawanabe M., Büna P. von, Blankertz B., Müller K. R. Toward Unsupervised Adaptation of LDA for Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering*. 2011;58(3):587–597.
- [17] Xu Z., So Rosa, Toe K. K., Ang K. K., Guan Cuntai. On the Asynchronously Continuous Control of Mobile Robot Movement by Motor Cortical Spiking Activity. In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2014, 3049–3052.
- [18] Zhang Dan, Song Huaying, Xu Rui, Zhou Wenjing, Ling Zhipei, Hong Bo. Toward a Minimally Invasive Brain–Computer Interface Using a Single Subdural Channel: A Visual Speller Study. *NeuroImage*. 2013;71:30–41.