2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)
Mexico City, Mexico. November 11-13, 2020

Deep Learning Networks for Vowel Speech Imagery

José Manuel Macías-Macías
Laboratorio de Sistemas de Percepción Visual con Aplicaciones en Robótica
Tecnológico Nacional de México / I.T. Chihuahua
Chihuahua, México
jmmaciasm@itchihuahua.edu.mx

Juan Alberto Ramírez-Quintana
Laboratorio de Sistemas de Percepción Visual con Aplicaciones en Robótica
Tecnológico Nacional de México / I.T. Chihuahua
Chihuahua, México
jaramirez@itchihuahua.edu.mx

Graciela Ramírez-Alonso
Facultad de Ingeniería
Universidad Autónoma de Chihuahua
Chihuahua, México
galonso@uach.com

Mario Ignacio Chacón-Murguía
Laboratorio de Sistemas de Percepción Visual con Aplicaciones en Robótica
Tecnológico Nacional de México / I.T. Chihuahua
Chihuahua, México
mchacon@itchihuahua.edu.mx

***Abstract*— Speech Imagery (SI) is a successful alternative for communication systems based on Electroencephalographic (EEG) signals that do not need external stimuli like evoked potentials. A recent strategy for SI is to analyze Speech Related Potentials (SRP) features on EEG signals to recognize vowels. However, there is research to be done in the development of machine learning methods for vowel classification in SI signals based on SRP. Therefore, this paper proposes two neural networks to classify vowels in speech imagery signals using SRP: a Convolutional Neural Network called sCNN and a Capsule Neural Network called sCapsNet. The experiments were developed with the DaSalla dataset. According to the results, sCapsNet reports better performance than sCNN and Support Vector Machine (SVM). The average accuracy was 71.9%, 67.63%, and 71.33% respectively. Besides, the capsules of sCapsNet could model by vectors the SRP features regardless of time and differences of SI vowels by subjects.**

Keywords*— *Speech Imagery, EEG, Convolutional Neural Network, CapsNet.

I. Introduction

Speech imagery (SI) is a mental strategy that consists of imaging letters or words without producing any movements or sounds. This mental strategy is a viable alternative to develop communication systems based on Electroencephalographic (EEG) signals; therefore, the speech is a natural way to communicate and does not need an external stimulus as evoked potentials [1]–[3] or motor imagery [4]–[7]. Many datasets related to SI have been developed, e.g. Nguyen et al. [8] created a dataset from 15 subjects with an age range from 22 to 32 years old. They performed 100 trials per word or vowel in three sessions of 3 modalities of SI: 2 short words, 2 long words and 2 vowels. This dataset was employed to develop a method based on covariance matrix descriptors and relevance vector machines which achieves an accuracy of 70% to 95% with a high variance between subjects. Pressel et al. [9] presented a dataset of 15 subjects to perform tasks of SI of 5 vowels and 6 Spanish words. The method used to process this dataset uses the Discrete Wavelet Transform (DWT) and the classifier is based on Support Vector Machine (SVM) and Random Forest (RF) algorithms. This method reaches 25% and 20% accuracy for vowels and words classification, respectively. DaSalla et al. proposed in [10] a dataset for vowels SI classification from 3 subjects. They implemented a method based on Common Spatial Patterns (CSP) and SVM to process the SI signals while reaching an accuracy of 71.33%.

Vowel recognition in SI is more useful than other mental strategies to create communication systems because speech is a natural way to communicate [11]. Vowels can be recognized by Speech Related Potentials (SRP) because they include peaks in the EEG signal; they are yielded by the activity in the posterior and medial regions of the scalp [12]. These peaks can be used as features and can be processed with machine learning methods to create more accurate systems. However, there is research to be done in the development of processing methods that can extract features from EEG data and recognize if the SI signal contains information related to vowels. Therefore, this paper proposes two adaptations of Deep learning models for vowel classification in SI that evaluates their performance in this alternative communication. The proposed models are Convolutional Neural Network for SI (sCNN) and a Capsule Neural Network for SI (sCapsNet). The convolutional network was selected because it is a supervised model that has reported the best accuracies in different EEG signals processing tasks such as Evoked Potentials (EP) and disease diagnosis [2], [13]–[15]. Additionally, capsule networks were selected because they are an improvement of CNN models that can learn better with fewer samples compared to other deep learning networks [16]–[18].

The paper is organized in the following way: Section 2 presents the DaSalla dataset and the feature extraction of signals, Section 3 shows the contribution of this paper, which is the proposed networks, Section 4 report the experiments as well as the results, and finally, Section 5 discusses the conclusions.

II. Dataset

The Dasalla dataset consists of the EEG signals of three subjects (2 males and 1 female) with an age range from 26 to 29 years old. The signals were recorded by the BioSemi ActiveTwo system, which has 64 Ag/AgCl electrodes that are placed in the scalp according to the international 10-20 system and a sampling rate of 2048 Hz, which is downsampled at 256 Hz.

The EEG signals were obtained with a SI protocol where the subjects must perform three different tasks: imagine an /a/, /u/ or no action. These 2 vowels were selected from among (/a/, /e/, /i/, /o/, /u/) because the brain has a different way to express them compared to other vowels since it needs to send the signals to

978-1-7281-8987-1/20/$31.00 ©2020 IEEE

move the mouth muscles with a different pattern. For /a/, the mouths need to be opened, but for /u/, the mouth must be rounded. The other vowels (/e/, /i/, /o/), have similar brain and muscle patterns between them.

The imagination of the /a/ and the /u/ is based on the mouth movements and the base sound, therefore, the dataset considers each vowel as a class. The patterns generated by the vowels /u/ and /a/ are based on SRP [10]. The SRP peak for the /a/ vowel is presented at 379 ms after an imaginary task and 355 ms for the /u/. The no-action task is used as a control state where the subject must remain alert. In this state, there is no peak.

A cue visual interface was designed to generate the SI patterns in the EEG signals and this interface implements the protocol that consists of three steps, as shown in Figure 1. The protocol performs 150 trials with a duration of about 1 minute approximately. The trials begin with a "*beep*" sound that prepares the subject for the first step. After the sound, the subject stares a cross that appears in the monitor about 2 to 3 seconds. In other words, the time between the sound and the next step prevents any "*auditory event related potential*" (AERP) that could interfere and avoids any subject's predisposition.

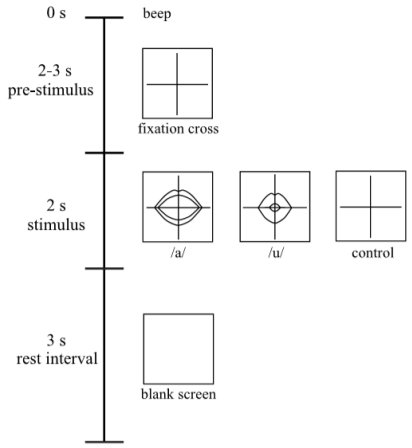

Figure 1. Experimental paradigm in the DaSalla dataset [10].

The SI activity that generates the SRPs is performed in the second step and it lasts 2 seconds. Three scenarios for each per imagination task can be presented in this step:

1. Open mouth silhouette so that the subject imagines the vowel /a/.
2. Round mouth silhouette so that the subject imagines the vowel /u/.
3. Cross to indicate an alert state where the subject has to be focused, avoiding any movement or sound waiting for the cue. This state is called "*control state"* and it is used for the algorithm so that it can recognize between SI activity and alert state.

The third step presents a blank screen for 3 seconds. The goal is that the subject takes a break and finishes the trial to continue with the next one.

Each subject performed 150 trials, 50 per task. There was an EEG signal *s(m)* recorded for each subject, where *m* is the time sample index. Different time indicators were reported on *s(m)* to know when each task was performed. The processing of *s(m)* consisted of a zero-phase bandpass filter with cutoff frequencies of 1Hz and 45Hz. After the filtering, a 3 seconds epoch was obtained per each task to get the information related to the imagery task (1 second from the pre-stimulus step and 2 seconds from the stimulus step). Each epoch will be denoted by the matrices E_g^i, where *i=1,…,50* is the epoch index, and *g=1,…,3* the task index. Rows and columns of the E_g^i epoch matrix correspond to the EEG channels and samples, respectively.

The feature extraction process presented by DaSalla et al. in [10], is based on CSP. The CSP algorithm is used to reduce the dimension of data but maximizing the variance between two classes. The processing of CSP initiates with the computation of the normalized covariance matrix of two of the three imagination tasks as follows:

$$\bar{C}_g = \frac{1}{n}\sum_{i=1}^{n}\frac{E_g^i\left(E_g^i\right)^T}{trace\left(E_g^i\left(E_g^i\right)^T\right)} \tag{1}$$

where n is the number of trials in each task. The resulting matrices $\bar{C}_g$ are used to find the composite covariance matrix C_c, which is factorized with its eigenvectors given as follows:

$$\begin{aligned} C_c &= \bar{C}_1 + \bar{C}_2 \\ C_c &= V_c \lambda_c V_c^T \end{aligned} \tag{2}$$

where V_c is the eigenvectors matrix, and λ_c is the diagonal matrix of eigenvalues, $\bar{C}_1$ and $\bar{C}_2$ correspond to two groups of three possible tasks [10]. Consequently, a whitening linear transformation is used to equalize the variances in the eigenspace, which is given by:

$$W = \sqrt{\lambda_c^{-1}} V_c \tag{3}$$

This transformation is applied to equalize the two covariance matrices as follows:

$$S_g = W\bar{C}_g Q W^T \tag{4}$$

$$S_1 = U\lambda_1 U^T \quad S_1 = U\lambda_2 U^T \tag{5}$$

Finally, a projection matrix is defined as $P = \left(U^T W\right)^T$ where the columns of P^{-1} are the CSP. This matrix is used to decompose each EEG epoch, as the following:

$$Z_g^i = PE_g^i \tag{6}$$

where Z_g^i are the feature vectors.

According to DaSalla et al. [10], the first and second spatial filters are enough to generate features to perform a classification between two classes because they represent the largest variance between other filters and them. They are the matrix Z_g^i, which has a dimension of *4x128* per each *ith* epoch and *gth* task. However, the size of Z_g^i is small for convolutional computations of the proposed networks. Then, Z_g^i was resized from *4x128* to

32x128 using a data augmentation based on duplicate consecutively each row eight times.

The signals available to download in DaSalla dataset consist on CSP features Z_g^i related to the *t=1,...,3* different classes to identify: /a/:control, /u/:control and /a/:/u/. These signals will be used as inputs in the neural network models that are being proposed with their corresponding classes as outputs in the classification tasks.

III. Artificial Neural Networks For Classification

In this work, two models are presented for the analysis of the vowel SI signals: sCNN based on CNN and sCapsNet based on capsule network modification. The following subsections detail their composition and algorithms.

A. Neural Network Model sCNN

Figure 2 shows the architecture of sCNN, which consists of five layers (*l = 1,...,5*): an input layer, two convolutional layers, two fully connected layers and one output layer. Each of them will be described next.

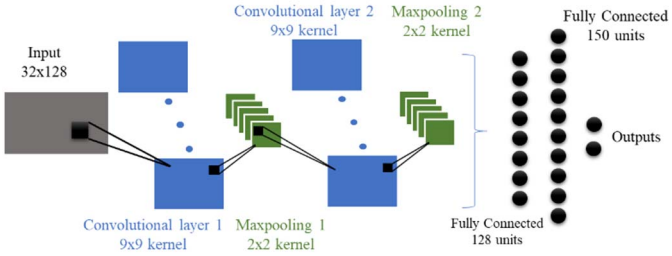

Figure 2. sCNN neural network model.

The input of the networks is the Z_g^i EEG signal projected by CSP. The layers *l=2* and *l=3* are convolutions given by:

$$d_{lk} = W_{lk} * Z_g^i + \beta_{lk}, \quad l = \{1,2\}$$
$$y_{lk} = f(d_{lk}) \tag{7}$$

where W_{lk} is a set of weights, β_{lk} represents the bias term, k is the index and d_{lk} is the result of the convolutional operation. Both convolutional layers have *256* filters of 9x9, a stride of 2, and a rectifier linear unit (ReLU) activation function f [19]. This operation gives the y_{lk} feature maps. The ReLU was selected because its derivative is always equal to one if the input is greater than zero [20] and it avoids the gradient vanishing problem [6]. After each ReLU function, there is a max-pooling process given by:

$$u_{ijkl} = \max_{p,q \in y_{lk}} z_{pqk} \tag{8}$$

where z_{pqk} is an elements window (kernel) per each y_{lk} with *HxH* size, p and q are indexes for each y_{lk} elements and u_{ijkl} is the max-pooling output.

The layers *l=4*, *l=5,* and *l=6* constitute a Multilayer Perceptron (MLP) of three layers, whose input is the u_{ijk} output of the second convolutional layer. The MLP is composed of 128 neurons in *l=4*, 150 neurons in *l=5* and finally, the output layer *l=6*, has 2 neurons to define the classification of one of the two imagination tasks presented in the input. All the layers of the MLP have also a ReLU activation function and the number of neurons was defined empirically.

B. Neural Network model sCapsNet

The CapsNet was proposed by Sabour et al. in 2017 [21] as a deep learning network for Optical Character Recognition (OCR). These networks have an encoder-decoder architecture, and their functionality is based on groups of neurons called capsules whose activity is related to an entity present in the input [21], e.g. a circle in the image, a trace in a digit or a peak in EEG signal. The output of each capsule is denominated activity vector formed by the output of each neuron in the capsule. CapsNet was considered in SI vowel classification because this type of model reported a good performance on EEG classification topics [16], [17] and they are an improvement of CNN models [22]. For this experiment, we proposed the model of Figure 3, which is called sCapsNet, and it is an adaptation of the original model of CapsNet proposed in [21]. This model has six layers, the first three are the encoder and the three remaining are the decoder.

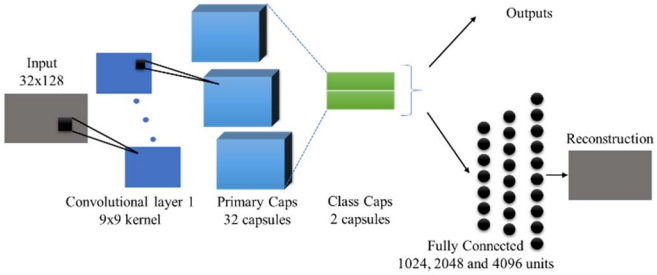

Figure 3. sCapsNet neural network model.

The first layer *l=1* is a traditional convolutional layer given by (7) which is used to detect basic features in the input. The second layer *l=2* is called "PrimaryCaps", and its function is to produce combinations of basic features to generate high level characteristics that represent entities of the input. This layer has 256 kernels with a size of 3x3 and a stride of 2 to deal with the feature map generated in the first layer. Every group of 8 scalars in the feature map constitutes the primary capsule *i* [23]. The third layer *l=3* is called "ClassCap", and it has two capsules of *1x4*. In this layer, each capsule represents each class.

The output of layers 2 and 3 are vectors given by the activity of each neuron inside the capsules. Formally, for all the layers of capsules, the total input is given by:

$$s_j = \sum_i c_{ij} \hat{u}_{j|i}, \quad \hat{u}_{j|i} = W_{ij} u_i \tag{9}$$

where s_j is the input of the *jth* capsule and it is given by the weighted sum of the prediction vectors $\hat{u}_{j|i}$ with the coupling coefficients c_{ij}, which are generated by the iterative Dynamic routing algorithm [21]. This algorithm will be explained in section *C*. u_i represents the output capsule of the previous layer and W_{ij} is the weight matrix with the function to couple the output of previous layers and the input of capsules of the next layers.

After computing (9), a "squashing" nonlinear function is used to normalize the magnitude and ensure that short vectors get shrunk to almost zero and long vectors get shrunk to a length slightly below 1 [21]. This function is defined by:

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \quad (10)$$

where v_j is the output vector of the *jth* capsule. This function is used because the length of the output vector of capsules in sCapsNet is used to represent the probability existence of an entity.

The decoder has three fully connected layers *l= {4,5,6}* with 1024, 2048 and 4096 neurons, respectively. The function of these layers is to rebuild the EEG signal based on the "ClassCaps" output. The result of these layers works as a regularizer on the training by minimizing the sum of squared differences between the regularizer unit and the network input and this process reduces overfitting cases. ClassCaps outputs are the classification output of the model.

C. Dynamic routing algorithm

This section explains how capsules are coupled through layers based on the scalar product. If the scalar product of the two output vectors of capsules *i* and *j* is high, then, their link will be maximized.

Let c_{ij} be the coupling coefficients between capsule *i* and all the capsules in the next layer *j* sum are equal to 1. The initial logits b_{ij} are the log probabilities that indicate how capsule *i* should be coupled with capsule *j* based on the routing algorithm showed in List 1. The log priors are learned discriminatively at the same time as all the other weights during training. The initial c_{ij} are iteratively refined by measuring the agreement between the capsule output v_j of (10) of each capsule *j,* and the prediction $\hat{u}_{j|i}$ made by capsule *i*. This agreement in the algorithm remains on the scalar product $a_{ij} = v_j \bullet \hat{u}_{j|i}$ which is the product of Euclidean magnitudes of two vectors and the cosine angle between them. The algorithm is shown in List 1, as the original pseudocode proposed by Sabour et al in [21]. The softmax function is defined as follows:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_t \exp(b_{it})} \quad (11)$$

List 1. Dynamic routing between capsules.

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{\boldsymbol{u}}_{j|i}$, r, l)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ softmax com
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$
7: for all capsule i in layer l and capsule j in layer $(l+1)$
return $\mathbf{v}_j$

If and only if an entity is present in the input, the top-level capsule for one class of *t* will have a long instantiation vector. To allow that multiple SI vowels can use a separate margin loss, L_t for each vowel capsule related with class *t* is computed by:

$$L_t = T_t \max\left(0, m^+ - \|v_t\|\right)^2 + \lambda\left(1-T_t\right)\max\left(0, \|v_t\| - m^-\right)^2 \quad (12)$$

where T_t=1 if and only if a vowel of class *t* is present. m^+= 0.9 and m^-= 0.1. The λ down-weighting coefficient of the loss for absent vowels or control state is defined to stop the initial learning from shrinking the lengths of the activity vectors in capsules. We used λ=0.5 as in the original paper, where it is introduced [21].

IV. Experiments and Results

The experiments were developed with a Dell Precision Tower 5810 workstation with NVIDIA GPU 970x, Intel E5-1603 processor and 8 Gb RAM of 2133 Mhz. The sCNN was trained with the stochastic gradient descent algorithm, 25 epochs, a batch size of 1, and a learning rate of 0.0025. sCapsNet was trained by the dynamic routing algorithm between capsules, 20 epochs with a batch size of 1, and α=0.0005 for the lost function. The training, test and validation sets of DaSalla were partitioned on 60%, 20% and 20% respectively. A five cross validation was performed for both models. Each model was trained by a specific subject for each pairwise classification.

A. Performance of sCNN and sCapsNet

Tables I and II show the pairwise classification performance of the validation set of sCNN and sCapsNet. According to Table I, sCNN achieved the best average results with subject 1, because the network presented the best accuracy in the classes /a/:control and /u/:control with this subject. Additionally, sCNN has the best classification results on subject 3, and class /a/:/u/. sCapsNet achieves a result of 89.3% with the /a/:/u/ classification. Table II demonstrates that it is the best classification result compared to /a/:control and /u/:control due to existence of similarities between the signals in each vowel class, while control class signals are different.

Table I. Accuracy results of sCNN.

sCNN	/a/:control	/u/:control	/a/:/u/	Average
Subject 1	78%	76%	66%	73.3%
Subject 2	64%	74%	59%	65.6%
Subject 3	58%	60%	74%	64%
Average	66.6%	70%	66.3%	**67.63%**

Table II. Accuracy results of sCapsNet.

sCapsnet	/a/:control	/u/:control	/a/:/u/	Average
Subject 1	61.47%	65%	89%	71.8%
Subject 2	60%	66%	91%	72.3%
Subject 3	64%	63%	88%	71.6%
Average	61.82%	64.6%	**89.3%**	**71.93%**

The SRP peak for the /a/ is presented 379 ms after the imaginary task and 355 ms for /u/. In the control state, there is not any peak generated. sCNN classifies the level of presence of the SRP peaks by convolutional operations and determinates if there is a vowel or not in the EEG signal. On the other hand, the capsules of sCapsNet model the presence of features through the vectors (number of neurons on capsules) using the direction and magnitude of the features. According to the experiments, we found that the number of neurons in the capsules is associated with the modeling of entities presented in the input. In the cases of /a/ or /u/ tasks, sCapsNet could clearly model the SRP vowel identities, using 2 capsules with 4 neurons for each one. The number of capsules corresponds to the number of classes to identify, which are 2 in this specific case. The number of neurons was defined empirically according to the experiments.

However, the control task generates a stochastic signal in each trial which does not have a defined pattern to be modeled by the capsule vectors. This can be clarified in Table III, where the average accuracy from /a/:/u/ is lower than vowels vs /control/.

B. Comparison with other works

sCNN and sCapsNet were compared with the SVM method that was used in [10] to recognize vowels in the DaSalla dataset. They used a two-class nonlinear SVM with a radial basis function kernel. The metric used in DaSalla is accuracy and according to Table III, sCapsNet obtained the best average results.

Additionally, the three subjects present similar results with sCapsNet, while SVM has good results just with subject 1. This behavior can be attributed to the capsules because the SRP features presented in the EEG signal are represented by the activity vector of the capsules. These features could be represented through vectors that characterize the SRP entities that are invariant to time translation. In other words, capsules can model classes based only on their entities. On the other hand, a simple SVM employs kernel tricks and maximal margin to perform classification with two hyperplanes for two classes (/a/:/control/ and /u/:/control/). Based on the fact that the SVM uses differences between two classes and its variance, the SVM reports in [10] better results in the vowel vs control state than two vowel patterns. This behavior is the opposite case of the sCapsNet results, which are shown in Table II. As can be seen in Table III, SVM obtained the best results with subject 1, this is because the kernel, the hyperplanes, and the processing method were designed considering the signals of this subject. Therefore, a bias of the proposed model for subject 1 can be considered. On the other hand, sCapsNet generates results with less accuracy than the SVM in subject 1, however, sCapsNet attained the best average accuracy, as illustrated in Table III.

Table III. Comparison with other literature works.

	sCNN	sCapsNet	SVM [10]
Subject 1	73.3%	71.8%	78%
Subject 2	65.6%	72.3%	68%
Subject 3	64%	71.6%	68%
Average	67.63%	71.9%	71.33%

V. Conclusions

This paper proposed two deep learning models designed to classify SI vowels: sCNN and sCapsNet. The signals were obtained from the DaSalla dataset, whose EEG signals were already processed with the CSP algorithm. The performance of sCNN and sCapsNet was compared against the SVM method presented in the DaSalla dataset [10]. According to the results, sCapsNet achieved the best performance since it reported the best accuracy results and the lowest variance between subjects. It was observed that the capsules of sCapsNet could model the SRP features of the EEG activity presented in the Z_g^i inputs with better results in the /a/:/u/ classification. sCNN used convolutional operations by spatial filters, that searches for features across the input signal. Consequently, through layers, these features are combined to define the vowel classification. However, it is necessary to stack several layers and adjust the kernel dimension or parameters to improve the performance. For SVM the performance results depended on the kernel used and the feature extraction. In this case, the analysis performed on the DaSalla dataset was focused on the signals of subject 1, it makes a positive inclination results for this subject.

Acknowledgment

This research was funded by Tecnológico Nacional de México (TecNM) under grant 7598.20-P.

References

[1] J. Ramírez-Quintana, J. Macias-Macias, A. Corral-Saenz, and M. Chacon-Murguia, "Novel SSVEP Processing Method Based on Correlation and Feedforward Neural Network for Embedded Brain Computer Interface," in *Mexican Conference on Pattern Recognition*, 2019, pp. 248–258.

[2] N. S. Kwak, K. R. Müller, and S. W. Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PLoS One*, vol. 12, no. 2, pp. 1–20, 2017, doi: 10.1371/journal.pone.0172578.

[3] Aya Rezeika, M. Benda, P. Stawicki, F. Gembler, A. Saboor, and I. Volosyak, "Brain – Computer Interface Spellers : A Review," *Brain Sci.*, vol. 8, no. 57, pp. 1–38, 2018, doi: 10.3390/brainsci8040057.

[4] Z. Tang, S. Sun, S. Zhang, Y. Chen, C. Li, and S. Chen, "A brain-machine interface based on ERD/ERS for an upper-limb exoskeleton control," *Sensors*, vol. 16, no. 12, pp. 1–14, 2016, doi: 10.3390/s16122050.

[5] N. Tomida, T. Tanaka, S. S. Ono, M. Yamagishi, and H. Higashi, "Active Data Selection for Motor Imagery," *IEEE Trans. Bio-Medical Eng.*, vol. 62, no. 2, pp. 458–467, 2015, doi: 10.1109/TBME.2014.2358536.

[6] J. Zhang, C. Yan, and X. Gong, "Deep convolutional neural network for decoding motor imagery based brain computer interface," *2017 IEEE Int. Conf. Signal Process. Commun. Comput. ICSPCC 2017*, vol. 2017-Janua, pp. 1–5, 2017, doi: 10.1109/ICSPCC.2017.8242581.

[7] M. Alansari, M. Kamel, B. Hakim, and Y. Kadah, "Study of wavelet-based performance enhancement for motor imagery brain-computer interface," *2018 6th Int. Conf. Brain-Computer Interface, BCI 2018*, vol. 2018-Janua, pp. 1–4, 2018, doi: 10.1109/IWW-BCI.2018.8311520.

[8] C. H. Nguyen, G. K. Karavas, and P. Artemiadis, "Inferring imagined speech using EEG signals: A New Approach using Riemannian Manifold Features," *J. Neural Eng.*, pp. 1–16, 2017.

[9] G. A. Pressel Coretto, I. E. Gareis, and H. L. Rufiner, "Open access database of EEG signals recorded during imagined speech," *12th Int. Symp. Med. Inf. Process. Anal.*, vol. 10160, no. December, 2017, doi: 10.1117/12.2255697.

[10] C. S. DaSalla, H. Kambara, M. Sato, and Y. Koike, "Single-trial classification of vowel speech imagery using common spatial patterns," *Neural Networks*, vol. 22, no. 9, pp. 1334–1339, 2009, doi: 10.1016/j.neunet.2009.05.008.

[11] R. A. Sharon, S. Narayanan, M. Sur, and H. A. Murthy, "An Empirical Study of Speech Processing in the Brain by Analyzing the Temporal Syllable Structure in Speech-input Induced EEG," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2019-May, pp. 4090–4094, 2019, doi: 10.1109/ICASSP.2019.8683572.

[12] R. I. Herning, R. T. Jones, and J. S. Hunt, "Speech event related potentials reflect linguistic content and processing level," *Brain Lang.*, vol. 30, no. 1, pp. 116–129, 1987, doi: 10.1016/0093-934X(87)90032-0.

[13] N. K. N. Aznan, S. Bonner, J. D. Connolly, N. Al Moubayed, and T. P. Breckon, "On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks," *IEEE Syst. J.*, 2018.

[14] T. Nguyen and W. Chung, "A Single-Channel SSVEP-Based BCI Speller Using Deep Learning," *IEEE Access*, vol. 7, pp. 1752–1763, 2019, doi: 10.1109/ACCESS.2018.2886759.

[15] S. L. Oh *et al.*, "A deep learning approach for Parkinson's disease diagnosis from EEG signals," *Neural Comput. Appl.*, vol. 5, 2018, doi: 10.1007/s00521-018-3689-5.

[16] K. W. Ha and J. W. Jeong, "Motor imagery EEG classification using capsule networks," *Sensors (Switzerland)*, vol. 19, no. 13, 2019, doi:

10.3390/s19132854.

[17] H. Chao, L. Dong, Y. Liu, and B. Lu, "Emotion recognition from multiband eeg signals using capsnet," *Sensors (Switzerland)*, vol. 19, no. 9, 2019, doi: 10.3390/s19092212.

[18] J. Guo, F. Fang, W. Wang, and F. Ren, "EEG Emotion Recognition Based on Granger Causality and CapsNet Neural Network," *2018 5th IEEE Int. Conf. Cloud Comput. Intell. Syst.*, pp. 47–52, 2018.

[19] M. Liu, W. Wu, Z. Gu, Z. Yu, F. F. Qi, and Y. Li, "Deep learning based on Batch Normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018, doi: 10.1016/j.neucom.2017.08.039.

[20] B. Xu *et al.*, "Wavelet Transform Time-Frequency Image and Convolutional Network-Based Motor Imagery EEG Classification," *IEEE Access*, vol. 7, no. MI, pp. 6084–6093, 2019, doi: 10.1109/ACCESS.2018.2889093.

[21] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 3857–3867, 2017.

[22] R. Mukhometzianov and J. Carrillo, "CapsNet comparative performance evaluation for image classification," pp. 1–14, 2018.

[23] B. Jia and Q. Huang, "DE-CapsNet: A diverse enhanced capsule network with disperse dynamic routing," *Appl. Sci.*, vol. 10, no. 3, 2020, doi: 10.3390/app10030884.