Deep Learning for EEG Motor Imagery Classification

Kejia Liu, Jia Li Ma, Shiv Patil
BMEN 4470 Deep Learning for Biomedical Signal Processing
December 20, 2021

***Abstract*—The present study seeks to investigate and compare EEG classification accuracy between the multi-layer convolutional neural network feature fusion (MCNN) model and EEG-TCNet model. The same dataset, BCI Competition IV-2a, was being used for model training and evaluation. The results showed that the EEG-TCNet model has higher classification accuracy, 65.9%, compared to the 32.3% for MCNN model. In addition, the EEG-TCNet model required much less trainable parameters, around 4700, compared to the 2.4 millions for just 1 layer of the MCNN model. Overall, the EEG-TCNet model shows better performance compared to the MCNN model in classifying EEG signals.**

Keywords—deep learning, motor-imagery, convolutional neural networks, Convolutional Neural Networks

I. Introduction

Brain-computer interface (BCI) helps build a connection between the human mind and electronic devices by detecting biological signals during brain events, allowing the transition of mental activities into signals that are recognizable by computer systems [1]. A widely discussed application of BCI is motor imaginary, a cognitive process of the imagination of certain movements without actually performing them. This system enables people with motor disabilities to interact with the environment using mental thoughts instead of muscle movement. The whole process is in the absence of external stimulus input by detecting neural signals of the human brain, regularly recorded with non-invasive electroencephalography (EEG). Non-invasive BCI typically place the sensors on the skull rather than under the skull to avoid health issues. On the other hand, unlike invasive BCI, sensors outside the head cannot possibly record the movement of single neurons or sample with a high temporal resolution. The BCI Competition IV-2a dataset [2] is a widely used dataset of motor imagery tasks for the imagination of movement in four different parts of the body. The EEG signals are sampled with 22 electrodes on the scalp in a non-invasive way.

According to these features of non-invasive BCI, there are three methods to further improve the performance of a BCI system: improving the quality of recorded data, proposing new features and improving the classifier [1]. For the first and second part, it is not the goal of this project. The third method requires capturing more information from EEG data during the feature extraction step, and designing a classifier with higher accuracy.

In non-invasive EEG based BCI, EEG signal classification significantly affects the overall performance of BCI. In general, according to the small size of data in the BCI Competition IV-2a dataset, there are limited options for the classifiers. Convolutional Neural Networks (CNNs) have been successfully used in computer vision and speech recognition [3], achieving a better result other than methods relying on manually extracted features. EEGNet and multi-layer CNN are two successfully applied CNN models used for extracting data from EEG signals. Another applicable model for EEG-based BCIs is temporal convolutional networks (TCNs). This network performs well in time series classification.

In this project, we implement both an EEG-TCNet and a multi-layer CNN net for motor imagery classification. By evaluating the two models on the BCI-IV 2a dataset, we test and discuss the difference in performance between the two frameworks.

II. Materials and Methods

A. Data Pre-processing

The BCI Competition IV-2a dataset [2] consists of EEG data from 9 subjects. Two sessions of each subject are recorded on different days. Each session consists of 6 runs separated by a break. Each run contains 12 trails for each of the four possible classes, yielding a total size of 288 trials per session per subject. The start time and duration of each event, including cues and trails, were recorded with specific values in the dataset. The signals were sampled at 250 Hz and filtered with a 0.5 Hz - 100 Hz bandpass filter. Each trail lasts for 7.5 seconds including the end of the direction cue for 0.25 seconds, shown in Fig 1. Particularly, trials containing artifacts were marked by an expert manually and are excluded during preprocessing.

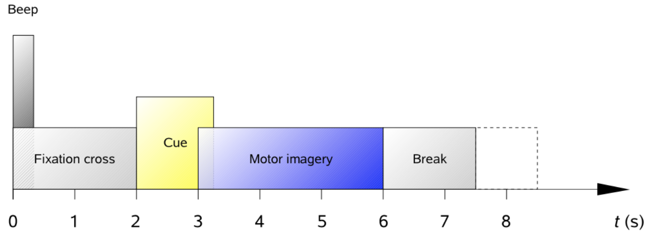


Fig. 1. Timing scheme of the BCI Competition IV-2a[2].

Data files are loaded by the open-source toolbox BioSig in MATLAB. The signal contains 25 channels, with EEG signals in the first 22 channels and EOG signals in the last 3 channels. According to the requirement of the BCI group, EOG signals are not used for classification. In order to minimize the influence of artifacts, a 4 Hz high pass filter was used on the dataset. Also, a high pass filter is a suggested way by the publisher of the dataset to remove eye artifacts.

The two sessions are originally designed for training and testing respectively. True labels for testing data are published after the competition. For each subject, we use the original 50%-50% split of training data and testing data. Normalization method is performed on signals by limiting the amplitude of the signal so that the peak magnitude is no larger than 1.

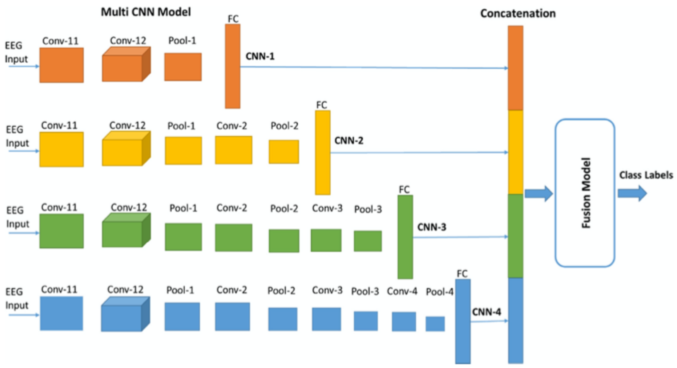


Fig. 2. Architecture of the multi-layer CNN feature fusion model [4].

B. Multi-layer CNN Feature Fusion

Fig 2 shows the architecture of the multi-layer CNN feature fusion model (MCNN), consisting of four CNNs, each with a different architecture, and a final multilayer perceptron (MLP). The main motivation for creating such a fusion model is that these different architectures may capture unique EEG features; thus, by fusing their outputs with an MLP, generic features for MI classification can be extracted [4].

First, the CNNs are trained on the MI dataset. The output features from each model are then concatenated and passed as input to the MLP. The output of the MLP passes through a softmax activation layer to get the probability scores for the MI classes.

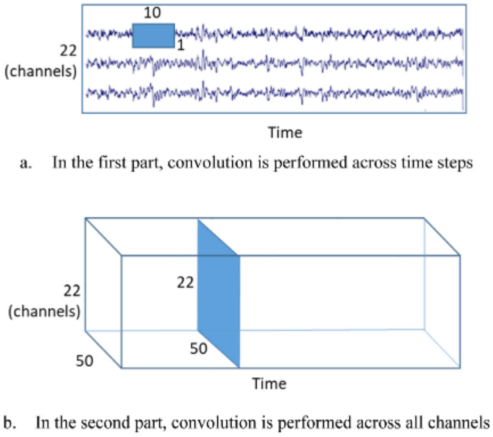
Each CNN model is based on the AlexNet architecture, a powerful model that has reached high accuracies for image classification purposes [5]. The models – CNN-1, CNN-2, CNN-3, and CNN-4 – consist of a different number of convolutional blocks and filters, as shown in Table 1.. These parameters were selected by Amin et al. to achieve optimal performance scores.

TABLE I. ARCHITECTURE OF EACH CNN USED FOR FEATURE FUSION [4]

CNN-1	CNN-2	CNN-3	CNN-4
Conv (30 × 1, 50 filters) Conv (1 × 22, 50 filters) Max Pool (3 × 1, stride 3) Dense (1024) Softmax (4 classes)	Conv (25 × 1, 50 filters) Conv (1 × 22, 50 filters) Max Pool (3 × 1, stride 3) Conv (10 × 1, 100 filters) Max Pool (3 × 1, stride 3) Dense (1024) Softmax (4 classes)	Conv (20 × 1, 50 filters) Conv (1 × 22, 50 filters) Max Pool (3 × 1, stride 3) Conv (10 × 1, 100 filters) Max Pool (3 × 1, stride 3) Conv (10 × 1, 100 filters) Max Pool (3 × 1, stride 3) Dense (1024) Softmax (4 classes)	Conv (10 × 1, 50 filters) Conv (1 × 22, 50 filters) Max Pool (3 × 1, stride 3) Conv (10 × 1, 100 filters) Max Pool (3 × 1, stride 3) Conv (10 × 1, 100 filters) Max Pool (3 × 1, stride 3) Conv (10 × 1, 200 filters) Max Pool (3 × 1, stride 3) Dense (1024) Softmax (4 classes)

For each model, the first convolutional block is split into two layers: in the first, each filter performs a convolution over time samples, and in the second, each filter performs spatial filtering for all channels with filters of the preceding temporal convolution. The result is a convolution across all input channels for a given sample. Fig 3 provides an illustration of these two convolution operations.

The convolutions are followed by an exponential linear unit (ELU) activation function, which has been shown to be faster and more accurate than ReLU. Max pooling and batch normalization techniques are also included. A dense linear layer is used as the output layer to produce 4 raw logits corresponding to the prediction probability of each MI category.



a. In the first part, convolution is performed across time steps

b. In the second part, convolution is performed across all channels

Fig. 3. (a) The first convolution is performed across time steps of an EEG channel. (b) The spatial convolution is performed for all channels simultaneously [4].

Depending on the number of convolutional blocks specified in the model, the output of the initial convolution passes through an additional number of convolutions (for a possible maximum of 3 convolutions). The convolution blocks additionally include max pooling, dropout, and batch normalization.

After training the multi-layer CNNs on the BCID dataset, the output features are concatenated and fed to an MLP. The MLP consists of two hidden layers with 50 nodes each. In addition, a 50% dropout rate is used to achieve generalization. The MLP is trained on the combined feature vector, and the output is passed to a softmax layer to get the probabilities for the MI classes.

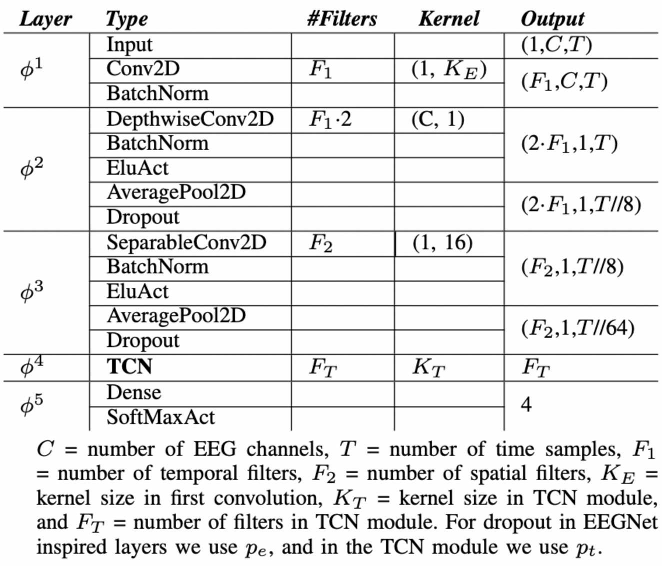
The final MCNN model was trained for 50 epochs with a batch size of 16 and Adam optimizer. The learning rate for the optimizer was set to 1e-4 with weight decay of 0.01. Negative log-likelihood loss (NLLLoss) was used to train the model to learn the softmax outputs of this multi-classification task. The entire architecture and training framework for MCNN was implemented in PyTorch.

C. EEG-TCNet

EEG-TCNet consists of two major components, which are an EEGNet, which is a compact convolutional neural network, and a TCN. The architecture of an EEG-TCNet is shown in Table II. This architecture is proposed by Thorir et al [6].

The EEGNet block in this architecture, it's inspired by the EEGNet architecture introduced by V J Lawhern et al [7], and it consists of 3 blocks. The first block uses 2D temporal convolution to learn frequency filters for the EEG, and the output of this 2D convolution gets normalized with Batchnormalization before feeding it to the second block, for depthwise convolution. The depthwise convolution learns spatial filters that are specific to the learned frequency filter from the 2D convolution layer. It is a combination of depthwise convolution and a pointwise convolution, where the depthwise convolution individually obtains the temporal summary of each feature map from the previous depthwise convolution, and the pointwise convolution applies a 1x1 kernel, with depth of the stacked up temporal summary, to mix these feature maps together. Both depthwise and separable convolution follows by Batchnormalization, ELU activation function, average 2D pooling with 1x10 kernel, and dropout of 0.2. The advantages of using depthwise convolution and separable convolution in the EEGNet is that it allows learning different features maps individually possible while reducing the number of trainable parameters in a deep network [7]. A 1x10 kernels was used for 2D pooling instead of the 1x8 kernels that the original paper used. The change was made to keep the number of residual block in the TCN model at 2.

TABLE II. EEG-TCNet Architecture [6]



Layer	*Type*	*#Filters*	*Kernel*	*Output*
ϕ^1	Input			$(1,C,T)$
	Conv2D	F_1	$(1, K_E)$	(F_1,C,T)
	BatchNorm			
ϕ^2	DepthwiseConv2D	$F_1 \cdot 2$	(C, 1)	$(2 \cdot F_1,1,T)$
	BatchNorm			
	EluAct			
	AveragePool2D			$(2 \cdot F_1,1,T//8)$
	Dropout			
ϕ^3	SeparableConv2D	F_2	(1, 16)	$(F_2,1,T//8)$
	BatchNorm			
	EluAct			
	AveragePool2D			$(F_2,1,T//64)$
	Dropout			
ϕ^4	**TCN**	F_T	K_T	F_T
ϕ^5	Dense			4
	SoftMaxAct			

C = number of EEG channels, T = number of time samples, F_1 = number of temporal filters, F_2 = number of spatial filters, K_E = kernel size in first convolution, K_T = kernel size in TCN module, and F_T = number of filters in TCN module. For dropout in EEGNet inspired layers we use p_e, and in the TCN module we use p_t.

After the convolutions in EEGNet, the remaining temporal information in the EEGNet output gets explored by the TCN block, which consists of 2 residual blocks, and each block includes dilated 1D convolution, Batchnormalization, and dropout of 0.3. The kernel size used in the residual block is 4. The receptive field TCN is calculated at 19 with the receptive field size (RSF) equation, $RSF\ = 1 + 2(K_T - 1)(2^L - 1)$, where KT is the kernel size, and L is the number of residual blocks [6]. The receptive field is greater than the available temporal information in the EEGNet output. This allows the TCN to capture all information available in the input from the EEGNet. The dilation factors used in the 2 residual blocks are 1 and 2, respectively. ELU activation function is used after the convolution output is being batch normalized. The output from the TCN block is fed to a fully connected softmax dense layer for classification.

The model was trained for 750 and 600 epochs with Adam optimizer at 1e-3 and 5e-4 learning rate, respectively. The batch size was 64 for 750 epoch and 32 for 600 epoch. The use of bias was eliminated in all EEGNet convolution layers as described in [7].

III. Results and Conclusions

A. Multi-layer CNN Feature Fusion Performance

The MCNN model has a total of 792,942 parameters after condensing the architecture described in the original paper. The size of the model was reduced to resolve GPU memory consumption issues in Google Colaboratory. Specifically, the number of nodes in the final dense layer was reduced from 1024 to 512 and dropout with a rate of 0.5 was added after convolutional blocks.

TABLE III. MCNN Prediction Accuracy for Each of 9 Subjects

Subject	**Test Accuracy**
1	39.86%
2	28.14%
3	35.57%
4	28.17%
5	32.00%
6	29.80%
7	35.29%
8	35.43%
9	26.00%
Average	32.25%

Table III lists the prediction accuracies of MCNN for each of the 9 subjects in the BCI IV-2a data set. The average test accuracy is 32.25%, notably lower than the 75.7% reported in [4]. The aforementioned modifications conducted to reduce the size of the model, such as halving the number of nodes in the dense layer, may have indeed reduced the performance of MCNN. Another important factor is that Amin et al. had pre-trained their model on the High Gamma data set (HGD), a large MI data set created under controlled recording conditions [8]. By use of pre-training and transfer learning, the authors could reach competitive accuracies without running the risk of overfitting their model on the relatively small BCID data set. It is likely that the MCNN model that was built in this project overfits the BCID data as training accuracy, which had approached 100% for all 9 subjects by the 100th epoch, is much higher than the prediction accuracy. Indeed, early stopping, a regularization method that requires that a validation data set is evaluated during training, is a promising strategy to avoid overfitting [9]. Learning rate and weight decay are additional parameters that may be adjusted in accordance with the model training to achieve optimal prediction performance [10,11].

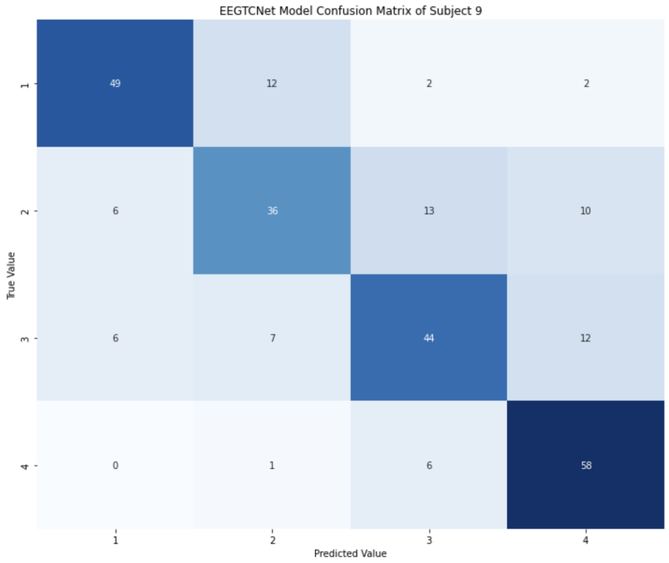
B. EEG-TCNet Performance

The EEG-TCNet model has a total of 4,272 parameters, and the trained EEG-TCNet model has an average prediction accuracy of 65.9% and 62.06% for 750 and 600 epochs, respectively. These are prediction accuracy is not as close to the original paper's prediction accuracy, which is 77.35%. There is a relative error of 14.8% and 18.8%. Each subject's training accuracy reaches high percentages (e.g. 90%) at a different rate. But, in general, most of the subjects reach 90% training accuracy around 400 epochs. The prediction accuracy for each subject ranges from 43.36% to 80.51% for 750 training epochs. For training with 600 training epochs, the prediction accuracy range is similar, which is from 41.86% to 84.24%. The prediction accuracy and number of epochs taken to reach 90% training accuracy is shown in table IV for each subject. The confusion matrix for subject 9 is shown in Fig 4 for model trained with 750 epochs. It shows that 70.6% of the predictions were true positives.

The difference between the lowest prediction accuracy and the highest prediction accuracy, for both trainings, with 750 epoch and 600 epoch, were almost around 40%. The reason for difference can be caused by the fixed parameters in the model [6]. To achieve better prediction accurate one approach to take is to vary the training parameter within the model since EEG signal is unique for each person [13]. But overall, the EEG-TCNet model still demonstrated a satisfactory EEG signal classificaiton.

TABLE IV. Prediction Accuracy with Number of Epochs taken to Reach 90% Training Accuracy under Different Rraining Parameters

Subject	**750 Epochs** **Adam (1e-3); Batch: 64**	**600 Epochs** **Adam(5e-4), Batch: 32**
1	73.31% | ~300 epoch	72.95% | ~360 epoch
2	54.42% | ~600 epoch	54.77% | ~400 epoch
3	74.73% | ~350 epoch	84.24% | ~300 epoch
4	59.21% | ~ 600 epoch	52.63% | ~ 600 epoch
5	63.41% | ~ 500 epoch	58.33% | ~ 450 epoch
6	43.26% | ~ 450 epoch	41.86% | ~ 550 epoch
7	80.51% | ~360 epoch	66.06% | ~380 epoch
8	73.43% | ~ 360 epoch	70.47% | ~ 370 epoch
9	70.83% | ~ 350 epoch	57.19% | ~ 400 epoch



	1	2	3	4
1	49	12	2	2
2	6	36	13	10
3	6	7	44	12
4	0	1	6	58

Fig. 4. Confusion matrix for subject 9 with model trained with 750 epochs.

References

[1] B. Graimann, B. Allison, and G. Pfurtscheller, “BrainComputer Interfaces: A Gentle Introduction,” 2009, pp. 1–27.

[2] C. Brunner, R. Leeb, G. R. Mu ¨ller-Putz, A. Schlo ¨gl, and G. Pfurtscheller, “BCI competition 2008 - Graz data set A.”

[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, 2015.

[4] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, and M. Shamim Hossain, “Deep learning for EEG motor imagery classification based on multi-layer cnns feature fusion,” *Future Generation Computer Systems*, vol. 101, pp. 542–554, 2019.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional Neural Networks,” *Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017*.

[6] T. M. Ingolfsson, M. Hersche, X. Wang, N. Kobayashi, L. Cavigelli, and L. Benini, “EEG-TCNet: An accurate temporal convolutional network for embedded motor-imagery brain–machine interfaces,” 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020.

[7] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, “EEGNet: a compact convolutional neural network for EEG-based braincomputer interfaces,” Journal of Neural Engineering, vol. 15, no. 5, p. 056013, 2018.

[8] R. T. Schirrmeister, J. T. Springenberg, L. D. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, “Deep learning with convolutional neural networks for EEG decoding and visualization,” *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[9] DeepAI, “Early stopping,” *DeepAI*, 17-May-2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/early-stopping-machine-learning. [Accessed: 24-Dec-2021].

[10] S. Bos and E. Chug, "Using weight decay to optimize the generalization ability of a perceptron," *Proceedings of International Conference on Neural Networks (ICNN'96)*, 1996, pp. 241-246 vol.1, doi: 10.1109/ICNN.1996.548898.

[11] L. Rice, E. Wong, and J. Z. Kolter, “Overfitting in adversarially robust deep learning,” *arXiv.org*, 04-Mar-2020. [Online]. Available: https://arxiv.org/abs/2002.11569. [Accessed: 24-Dec-2021].

[12] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] S. Marcel and J. R. Millan, “Person authentication using brainwaves (EEG) and maximum a posteriori model adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 743–752, 2007.