IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2020, accepted December 4, 2020, date of publication December 28, 2020, date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047678

Temporal Pyramid Pooling for Decoding Motor-Imagery EEG Signals

KWON-WOO HA AND JIN-WOO JEONG
Department of Computer Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea
Corresponding author: Jin-Woo Jeong (jinw.jeong@kumoh.ac.kr)

This work was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2019R1F1A1045329 and Grant 2020R1A4A1017775.

ABSTRACT Detecting a user's intentions is critical in human–computer interactions. Recently, brain–computer interfaces (BCIs) have been extensively studied to facilitate more accurate detection and prediction of the user's intentions. Specifically, various deep learning approaches have been applied to the BCIs for decoding the user's intent from motor-imagery electroencephalography (EEG) signals. However, their ability to capture the important features of an EEG signal remains limited, resulting in the deterioration of performance. In this paper, we propose a multi-layer temporal pyramid pooling approach to improve the performance of motor imagery-based BCIs. The proposed scheme introduces the application of multilayer multiscale pooling and fusion methods to capture various features of an EEG signal, which can be easily integrated into modern convolutional neural networks (CNNs). The experimental results based on the BCI competition IV dataset indicate that the CNN architectures with the proposed multilayer pyramid pooling method enhance classification performance compared to the original networks.

INDEX TERMS Brain–computer interface, deep learning, feature fusion, pyramid pooling.

I. INTRODUCTION

Detecting a user's intent correctly and providing them appropriate information or service on time is essential in human–computer interactions (HCIs). Recently, various interaction methods, such as eye tracking, gesture recognition, and brain signal-based approaches, have been proposed to detect a user's intentions more accurately, thereby improving the user experience in HCI. In particular, brain–computer interface (BCI) technology, that detects the user's intentions using brainwaves, has been considered increasingly in recent years.

The BCI technology is capable of efficiently aiding users with low communication skills or serious physical disabilities. Furthermore, it can effectively interact with machines or devices using the user's brain signals [1], [2]. BCI-based systems record the electrical activities of the human brain via various neuroimaging modalities such as functional magnetic resonance imaging (fMRI), electroencephalography (EEG), and positron emission tomography (PET) to detect a user's intentions. Among the various methods available for capturing brain activities, EEG is commonly used in BCI systems because of its high temporal resolution, portability, low cost, and non-invasiveness [1].

Sensorimotor rhythm (SMR) is a type of brainwave that can be observed after executing the movements or a motor imagery (MI) task. An SMR is strongly related to an MI task, which is generally defined as a mental process where an individual imagines himself/herself performing a specific action (such as, left or right hand or foot movement) without the actual activation of any muscles [3]. Therefore, successful decoding of an SMR can result in the generation of mental commands, which can remotely control a device. During the MI tasks, several regions of the brain (e.g., motor cortex, sensory areas, and prefrontal areas, etc.) are activated. Various studies have attempted to analyze the signals from these regions to determine a category for the executed MI tasks. For example, MI-based BCI systems can aid individuals with motor disabilities in controlling wheelchairs [4], [5], or robotic arms [6] without any physical interaction.

Over the past few decades, there have been various efforts to detect user intentions using MI signals, that can act as explicit or implicit feedback for the interaction design. Studies from the early stages [7] of MI-BCIs have focused

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

on extracting well-designed features and classifying the user's intentions based on the extracted features with various machine learning algorithms, such as linear discriminant analysis (LDA) [8], [9] and support vector machines (SVMs) [10]–[12]. For example, common spatial pattern (CSP) [13] and filter bank common spatial pattern (FBCSP) [14] algorithms are well-known feature extraction methods that contribute to the performance improvement of motor imagery EEG classification tasks.

Although previous BCI systems relied on well-designed and handcrafted features, they remain unsatisfactory with regard to classification accuracy. Conversely, deep learning-based approaches have been successful with respect to computer vision tasks, such as image classification, object detection, and recognition tasks, without extensive feature engineering [15]. Hence, many researchers in the BCI field have been inspired by this and have attempted to apply various deep learning approaches, such as convolutional neural networks (CNNs), to the EEG domain. A study by [16] proposed two types of CNN architectures (ShallowNet and DeepNet) capable of decoding a user's movement intent from the raw EEG signals. The authors of [16] reported the classification performance of CNN architectures, optimized with various hyper-parameters, normalizations, and activation functions. Experimental results revealed that CNN-based methods could successfully work without any feature engineering, and outperform the classic decoding methods [13], [14]. In [17], a CNN architecture termed as EEGNet was proposed to address the MI-based BCI tasks. This study aimed to produce a more compact network structure, reducing the number of parameters and the time required for training the networks, while preserving the overall performance. Sakhavi *et al.* [18] proposed an architecture built on the FBCSP method to produce a novel representation of the MI-EEG signals. The authors of [18] focused on lowering the feature dimension while preserving valuable temporal information. A more recent paper [19] attempted to learn the most discriminative and complementary spatial and temporal information for EEG-based brain computer interfaces.

Nevertheless, the current CNN-based approaches still exhibit limitations with respect to the classification accuracy due to the insufficient number of samples and the unstable nature of EEG signals. The EEG signals are inherently dynamic, unstable, inconsistent, and have a low signal-to-noise ratio. This makes it difficult for current deep learning architectures to automatically learn the important features from raw EEG signals. For example, the EEG signals measured from a single individual on the same day for the same task can exhibit different patterns. Therefore, a more robust method to learn the informative features from various perspectives is expected to improve the performance of EEG classification.

With respect to feature learning, recent studies on deep learning for computer vision have investigated the effects of multilevel feature extraction and fusion. Specifically, [20] proposed the application of a multilevel pooling layer immediately after the convolution layer, placed before the first fully connected layer in a CNN structure. The original goal of this layer was to handle the different scales, sizes, and aspect ratios of the input images by generating a fixed-length representation of features. However, the authors of [20] observed that the multilevel pooling strategy was also helpful in learning the various perspectives of the features when training the models for image classification and object detection tasks. The results from [20] validated that multilevel (or multiscale) feature extraction and fusion helps in extracting more informative data from the network. Furthermore, it contributes to improving the performance of the original network for various tasks. Hence, it is also expected that EEG classification can benefit from the multilevel pooling approach, which can learn features from various perspectives. However, studies on multilevel pooling for the EEG domain are limited. Therefore, this study aims to discuss the feasibility of a multilevel pooling approach to decode EEG signals for MI-based BCI applications. In this paper, we first discuss the basic concept of the spatial pyramid pooling method, successfully used in computer vision and other domains. Subsequently, we present the design and implementation of a novel type of pyramid pooling approach suitable for motor imagery EEG classification. Finally, we validate the feasibility of the proposed method.

The remainder of this paper is organized as follows. Section 2 reviews the related studies on motor imagery EEG classification. In Section 3, we briefly review the concept of pyramid pooling. Section 4 discusses the proposed method to apply the pyramid pooling approach to an EEG domain. In Section 5, we present and analyze the experimental results. Finally, we discuss the results and present our conclusions in Section 6.

II. RELATED WORK

In this section, we briefly describe the previous studies on motor imagery EEG classification.

A. FILTER BANK COMMON SPATIAL PATTERN

The filter bank common spatial pattern (FBCSP) [11] algorithm is a popular method for motor imagery EEG classification and was the best classification approach of the BCI competition IV [21]. The FBCSP overcomes the limitations of the common spatial patterns (CSP) algorithm [13], which is a spatial filter algorithm designed to effectively extract the discriminatory features from the motor imagery EEG signals. The CSP method has a limitation, wherein the classification accuracy decreases when the selected frequency range is inappropriate for the subject. To solve this problem, the FBCSP automatically selects the discriminative subject-specific frequency range.

Nevertheless, the FBCSP is limited due to its dependence on hand-crafted features, such as the selection of frequency band ranges and feature extraction methods.

B. MOTOR IMAGERY EEG CLASSIFICATION WITH CNNs

In the computer vision field, deep learning methods, such as the CNN, have succeeded in improving image understanding and classification performance based on their advanced characteristics, such as automatic feature extraction and learning [22]. Therefore, several recent studies have attempted to apply CNNs for understanding the motor imagery EEG and leverage the advantages of their architecture during classification tasks.

Schirrmeister *et al.* [16] presented two CNN architectures, ShallowNet and DeepNet, which decode the motor imagery from the raw EEG signals without any hand-crafted features. They reported performance improvements in classifying EEG signals using CNNs with various hyper-parameter set-ups, such as dropout, batch normalization, and activation functions. ShallowNet is a shallow network architecture consisting of two main blocks. The first block performs temporal and spatial convolution operations. Temporal convolution is performed with 40 kernels with a dimension of 1×25. Subsequently, a spatial convolution is conducted with 40 kernels with a dimension of $E \times 1$, where E denotes the number of electrodes. In the second block, a square nonlinearity, a logarithmic activation function, and an average pooling operation are applied. classification is then performed using the Softmax function. In contrast, DeepNet architecture consists of five main blocks. The first block performs temporal and spatial convolution operations in a manner similar to the ShallowNet architecture. The remaining four blocks are composed of a set of convolution and max pooling operations. All the layers, except the final fully connected layer, utilize an exponential linear unit (ELU) [23] as the activation function. Batch normalization and dropout are used to improve the performance of both models.

In [17], another CNN-based method, termed as EEGNet, was proposed for the classification and interpretation of MI EEG signals. The main idea of EEGNet was the adoption of depth-wise separable convolution operations [24] used in the recent CNN architectures for computer vision tasks. The advantage of depth-wise separable convolution operations in a CNN architecture includes reducing the number of parameters and simultaneously minimizing performance loss. The EEGNet presented a compact architecture, which could perform at par with deeper networks. The EEGNet consists of three main blocks. The first block performs temporal convolution and depth-wise convolution operations. Temporal convolution is performed via kernels with a size of 1×64. Depth-wise convolution is conducted with $E \times 1$, where E denotes the number of electrodes. In the second block, separable convolution is performed to combine the depth-wise convolution. Finally, the third block is a Softmax-based classification layer without a fully connected layer. In all the layers, with the exception of the third block, the ELU [23] activation function and average pooling are applied.

Conversely, the authors of [25] argued that the current CNNs exhibit limitations wherein the spatial relationships between the features of an object are not maintained while training a model. To address this problem, they proposed the capsule network (CapsNet), in which a group of neurons (i.e., Capsule), which represent the various parameters of an entity, forms the basic unit of training. The structure of the original CapsNet consists of a single convolution layer followed by Capsule layers. In CapsNet, the learning process is performed with a ''dynamic routing by agreement'' algorithm, which iteratively updates the values of low-level and high-level capsules in the Capsule layer. CapsNet has proven to be successful in automatically learning the various properties (e.g., rotation and thickness) of an object through the experimental results on the MNIST dataset, a large database of handwritten digits commonly used for training the various image processing systems [26]. Inspired by this, Ha *et al.* [27] proposed a method to apply CapsNet to classify motor imagery EEG signals. [27] used short-time Fourier transform (STFT) to convert 1D EEG signals to 2D time–frequency domain spectrogram images to be used as inputs for training and testing the Capsule networks. Accordingly, the configurations for the convolution layer and Capsule layers (e.g., kernel size, stride, and hyper-parameters for the dynamic routing algorithm) were optimized for the EEG domain. The authors of [27] showed that the CapsNet-based approach leads to competitive performance when compared to that of the CNN-based approaches.

As described above, various CNN-based techniques have been proposed for decoding the motor imagery EEG signals in recent years. However, their accuracy in the classification of signals is still not satisfactory. Therefore, studies on more robust methods of feature extraction, feature representation, training network architecture, and data augmentation are required. Among these, this study primarily focuses on the importance of feature extraction and representation in improving the performance of CNN architectures for motor imagery classification.

III. PYRAMID POOLING

In this section, we briefly review the concept of a pyramid pooling approach and its applications in various domains. The next section shows how it can be utilized in the EEG decoding domain.

The pyramid pooling method was proposed to address the problem of a fixed input size for deep neural networks in a computer vision field. For example, images are usually cropped or warped to have a fixed size and then fed into the deep neural network for training and inference. Crop and warp operators tend to either omit parts of the object or lead to geometric distortion [28]. These limitations can result in some information loss, which can decrease the recognition accuracy of deep neural networks. To overcome this issue, a novel pooling strategy, termed spatial pyramid pooling (SPP), was proposed by [20] based on the spatial pyramid matching model [29], [30]. The main contribution of the SPP layer is to generate a fixed length output, regardless

of the input size. The SPP layer is placed between the final convolution layer and the first fully connected layer to aggregate information from the previous feature maps. In the SPP layer, the feature maps are pooled to different levels as opposed to that in standard pooling operations (e.g., max or average-pooling). Given the feature maps with a size of $a \times a$ (e.g., 13×13), a pyramid level of $n \times n$ bins can be implemented as a sliding window pooling, where the window size $win = \lceil a/n \rceil$ and stride $str = \lfloor a/n \rfloor$, with $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denoting ceiling and floor operations [20]. The network structure can have a l-level spatial pyramid pooling layer. Hence, the first fully connected (FC) layer concatenates the l outputs from the spatial pyramid pooling layer. For example, a feature map (size of 13×13) before the SPP layer, which is composed of a 3-level spatial pyramid pooling where the pooling size and stride for each level are (5, 4), (7, 6), and (13, 13), will be converted to 3×3, 2×2, and 1×1 bins. Subsequently, the next FC layer will take these values as input.

The following points summarize the advantages of adopting an SPP layer. First, the SPP layer generates a fixed-length representation regardless of the input size. Second, the SPP layer can utilize features at multiple levels (multiple scales). Finally, based on these features, the SPP layer can increase the scale invariance of a network and suppress the overfitting problem [20], [31], [32]. Due to the advantages of SPP, several studies in various domains have attempted to adopt the SPP layer to develop their own CNNs. The studies from [31], [33]–[35] showed that an SPP-based CNN architecture could improve the performance of image classification tasks. A few other works have also reported advantages of adopting the SPP layer for object detection [36] and for biomedical tasks [37]. On the other hand, many studies also attempted to revise the underlying concept of the SPP mechanism and apply it to temporal domain tasks. Although the main idea was identical to that of the SPP method, the target of the task was in the form of time-series data. Several studies [38]–[41] presented how a temporal pyramid pooling (TPP) method can be utilized in action recognition tasks. Furthermore, the authors of [42] applied a TPP approach to identify music data.

Previous studies utilized the concept of the pyramid pooling mechanisms (i.e., spatial and temporal) to improve the performance of CNN architectures for various tasks. In this study, we focus on how to adopt and revise the concept of pyramid pooling for the classification of motor imagery EEG signals.

IV. PYRAMID POOLING FOR MI-EEG CLASSIFICATION

A. TEMPORAL PYRAMID POOLING LAYER

In this section, we describe the architecture of the proposed multilayer TPP approach. Typically, the current CNN architecture consists of several convolutions, pooling blocks, and fully-connected layers. Specifically, several CNNs for motor imagery EEG classification usually perform two types of convolution operations in its first building block [16], [17].

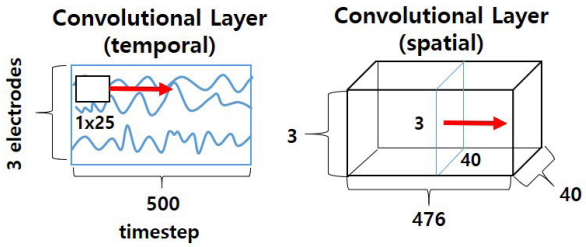

FIGURE 1. Convolution layers of CNNs for MI EEG classification.

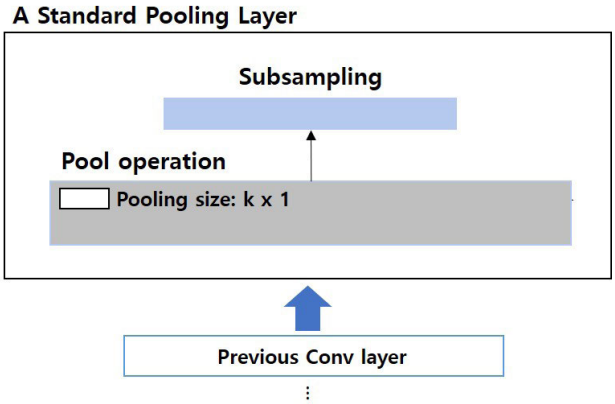

FIGURE 2. Illustration of standard pooling.

Figure 1 provides more details about temporal and spatial convolution operations for 2D EEG signals (i.e., time step $\times$ electrodes) sampled from three electrodes during 2 seconds at a sampling rate of 250Hz. The first convolution layer of a network is generally a temporal convolution layer in which a convolution operation is performed over time steps for each electrode (left side of Figure 1). In the case of Figure 1, a temporal convolution operation is performed with 40 convolution filters with a size of 25×1, and thereby resulting in a feature map with a size of $3 \times 476 \times 40$. Subsequently, a spatial convolution operation is performed with 3×1 convolution filters for all channels. The resultant feature maps are passed into the fully connected layer through the subsequent pooling and convolution layers.

Figure 2 illustrates the process of the standard pooling operations (e.g., max-pool and average-pool) on 2D EEG signals. It is similar to the standard pooling operation in a 2D image domain, wherein the pooling layers of the CNNs for the EEG domain also utilize 2D feature maps from the previous convolution layers as inputs. Afterwards, a standard pooling operation similar to max pooling or average pooling with a size of $k \times 1$ is performed. It is well-known that a standard pooling operation can act as a sub-sampling of the feature maps. However, a pooling operation with an inappropriate size can result in the loss of important information [25]. According to [25], traditional pooling operations can be useful to reduce the redundancy of representation and the network parameters; however, they often fail to consider the spatial hierarchies between objects. Similarly, a standard pooling operation applied to the EEG feature

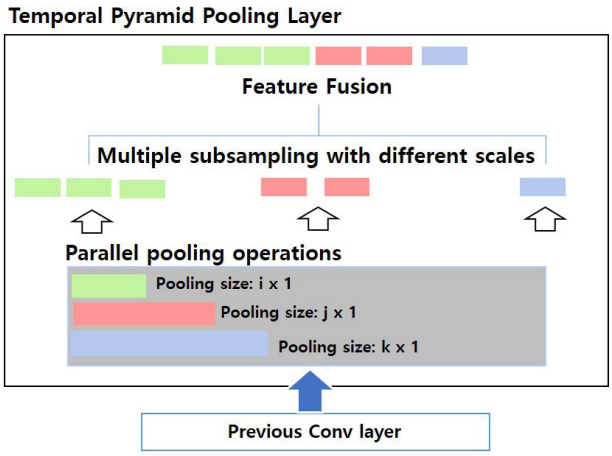

FIGURE 3. Illustration of temporal pyramid pooling.

map results in the temporal sub-sampling of feature maps. In particular, inappropriate sub-sampling of feature maps in the temporal domain may yield a discontinuous and discrete representation of data. Hence, this can lead to information loss which can make the training of classification models more difficult. In this work, we focus on the advantages of a pyramid pooling approach to address the aforementioned limitations of recent CNN-based architectures for EEG decoding. Among the benefits of adopting the pyramid pooling method, we expect that multi-scale feature extraction and fusion can be promising options to effectively represent EEG signals. Therefore, we propose to replace a standard pooling layer with a TPP layer to preserve informative data in the EEG decoding domain. The workflow of the TPP layer is illustrated in Figure 3. In contrast to the standard pooling layer illustrated in Figure 2, a TPP layer simultaneously performs multiple pooling operations with different pooling sizes. Figure 3 depicts that the primary objective of a pyramid pooling layer is to perform multiple temporal sub-sampling with different scales and then fuse the data collected to generate an integrated feature map. This is similar to an SPP layer used in 2D image classification and object detection tasks. As mentioned above, the SPP method used in computer vision tasks exploits multi-level spatial bins, and has been shown to be robust to the variance in object deformations and spatial layout [30]. It is well known that the EEG signals are inherently dynamic, unstable, inconsistent, and have a low signal-to-noise ratio. For example, the EEG signals measured from an individual on the same day for the same task can exhibit different patterns. Furthermore, There is also an inconsistency in the way each individual performs motor imagery tasks. For example, the timing and duration of each subject's motor imagination will be different even for the same motor command (e.g., move left-hand). Similar to the SPP method, therefore, the TPP approach studied in this paper exploits multi-level temporal bins to strengthen the original network against the variance in the stability of the EEG signals. Thus, instead of temporal subsampling of EEG signals with a fixed

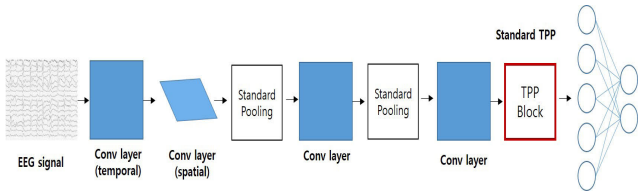

FIGURE 4. Architecture of CNNs with temporal pyramid pooling.

size, a TPP layer attempts to perform a multi-level subsampling operation such that various perspectives of the EEG signals are extracted during both training and testing phases. In this study, the number of pooling operations in each TPP layer was set to 3. For example, given the feature map with a size of 120×1 from the previous convolution layer, a TPP layer extracts three separate feature maps with a size of 40×1, 20×1, and 15×1 using three different pooling operations with different scales, i.e., 3×1, 6×1, 8×1, respectively. Finally, the feature maps from the three different pooling operations are concatenated into a single feature map with a size of 75×1. We expect that the CNN architectures can extract more informative feature maps using this new pooling operation in the EEG signals, thus, improving classification accuracy.

Originally, the main goal of the original SPP-layer [20] used in the image domain was to output a fixed-length feature representation from the input images with arbitrary scales. Therefore, an SPP layer is placed between the last convolution layer and the first fully-connected layer. Similarly, for the EEG domain, a TPP layer can be placed between the last convolution layer and the first fully-connected layer, as illustrated in Figure 4. In this case, we only replace the last standard pooling layer of the original networks with the proposed TPP layer. Similar to a conventional network architecture, the final dense layer is connected to the last feature map of the network, and the signals are classified as two-class or four-class motor imagery signals using a Softmax function.

B. MULTI-LAYER TEMPORAL PYRAMID POOLING

As mentioned above, time series data, such as EEG signals, can be discontinuous and discrete after passing through a set of convolutions and pooling operations, thereby resulting in unavoidable information loss. This can be more critical for deeper networks such as DeepNet [16] and EEGNet [17]. In the computer vision field, it is well known that each layer of a CNN has a different level of abstraction [43]. For example, lower layers in a CNN represent simple aspects of an image,such as the edges, whereas higher layers represent increasingly sophisticated aspects of an image, such as the shapes and patterns in the image. Based on this observation, we designed an extended version of the TPP layers, termed as multilayer temporal pyramid pooling architecture (MTPP), which attempts to extract multiscale temporal features from each abstraction layer and exploits them by fusion. The TPP layers placed at different levels of the network will attempt to decode different levels of semantics from the EEG data.

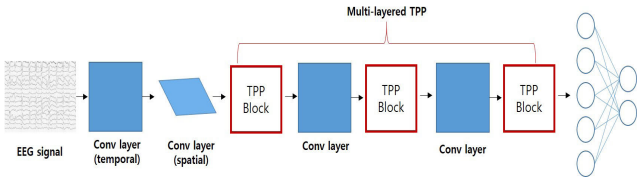

FIGURE 5. Architecture of CNNs with multilayered temporal pyramid pooling.

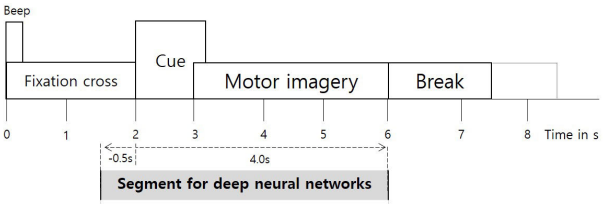

FIGURE 6. Experiment protocol for BCI Competition IV-2a.

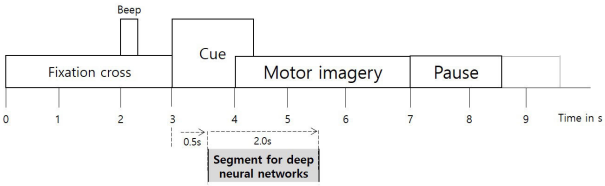

FIGURE 7. Experiment protocol for BCI Competition IV-2b.

Hence, using multilayered TPP, we expect that a variety of informative data can be extracted, preserved, and forwarded to subsequent layers of the network. To this end, in addition to the utilization of a single TPP layer before the first fully-connected layer, we also apply the proposed TPP mechanism to every pooling layer of the original network, as shown in Figure 5. In the MTPP approach, the intermediate feature maps extracted from each TPP layer are used as inputs for the subsequent conv/pooling layers. In this context, our feature fusion approach can be considered a feature-level self-augmentation since it tries to enrich a feature representation by generating multiple views from the feature itself. A similar approach, called Mosaic augmentation, was observed from the object detection task in the computer vision field [44], which picks a set of different images of different scales and ratios and then merges them into a single image. Even though there exist 1) no semantic relationships between the sub-images included in the merged one, and 2) unavoidable border lines between the sub-images, the experimental results showed the effectiveness of the mosaic augmentation technique. Conversely, the proposed MTPP architecture can be deemed a feature-level self-augmentation version of the mosaic technique customized for the EEG domain. Considering the success of a similar approach in the computer vision field, we also expect that the proposed approach can be used for both multi-scale feature representation and feature-level augmentation to improve the overall performance.

In our study, every TPP layer at different abstraction levels of the network shares the same configuration (e.g., pooling size and stride). In the next section, we discuss how the proposed TPP layers improve the performance of the original CNNs for classification of MI EEG signals. Additionally, we analyze how the changes in configuration of the TPP layers affect the overall performance.

V. EXPERIMENT

A. DATASET AND PREPROCESSING

To evaluate the effects of the proposed approach, we conducted an extensive experiment on BCI competition IV 2a and 2b datasets [21]. The BCI competition IV-2a and 2b datasets were obtained from 9 subjects via recording the EEG signals during 4/2-class motor imagery tasks, respectively. The EEG signals were recorded at a sampling frequency of 250 Hz. The signals were band-pass filtered between 0.5 Hz and 100 Hz, and a notch filter was applied at 50 Hz for noise removal. The dataset IV-2a included a set of EEG signals measured from 22 electrode positions, while the dataset IV-2b included EEG signals measured from 3 electrode positions. Figure 6 shows the protocol of the four-class (i.e., left-hand, right-hand, feet, and tongue) motor imagery classification task to collect the IV-2a dataset. Each trial begins with a fixation cross with an additional beep sound. After two seconds, a visual cue (an arrow pointing either to the left, right, down or up, corresponding to one of the four classes) appears on the screen for 1.25 seconds, followed by an MI task for a period of 3 seconds. Figure 7 illustrates a protocol of a two-class (i.e., left-hand and right-hand) motor imagery classification task to collect the IV-2b dataset. Each trial begins with a fixation cross with an additional beep sound. Three seconds later, a visual cue (an arrow pointing either to the left or right) is presented for 1.25 seconds. Then, a subject is supposed to imagine the corresponding hand movement over a period of 3 seconds [45]. The dataset IV-2a consists of two sessions for each subject. The first session consists of training data and the second session consists of test data. Each session includes 288 trials (i.e., 72 trials per class). The dataset IV-2b consists of five sessions for each subject. The first three sessions consist of training data and the remaining sessions are with test data. The first two sessions include 120 trials per session and the remaining sessions have 160 trials per session. On an average, the training data for each subject consists of 400 trials (120+120+160), and the test data consists of 320 trials (160+160) for each subject.

Typically, EEG signals can be divided into alpha (8–12 Hz), beta (12–31 Hz), gamma ($\geq$ 32 Hz), theta (4–7 Hz), delta (<4 Hz), and mu (8–13 Hz) bands. However, only a set of specific bands are selected and used to handle a specific BCI task. For example, mu and beta bands desynchronize over the sensorimotor cortex, which is contralateral to an imagined movement [17]. Therefore, these bands can be used for decoding motor imagery intents. A frequency range of 4–38 Hz was chosen in this study similar to that selected by [16] to cover the mu and beta bands. In addition to the frequency range, most of the setups for preprocessing

EEG signals, such as the length and timing of EEG segments, were inherited from those reported in the state-of-the-art CNN-based studies on decoding motor imagery EEG signals [16], [17]. The preprocessing steps used for our experiment are summarized as follows: 1) Raw EEG signals from datasets IV-2a and IV-2b were bandpass filtered between 4–38 Hz. 2) EEG segments of different lengths and timing were extracted from each dataset for analysis. For the IV-2a dataset, 4.5 seconds of an EEG segment from -0.5 to 4 seconds around the onset of the visual cue was extracted from each trial, as depicted in Figure 6. This is based on the finding of [16] that EEG segments starting from 500ms before the cue produced better performances for CNNs on the IV-2a dataset. For the IV-2b dataset, 2.0 seconds of an EEG segment from 0.5 to 2.5 seconds after the onset of the visual cue was extracted from each trial, as illustrated in Figure 7. As the optimal setup of the EEG segments for the CNNs on the IV-2b dataset was not reported in [16], [17], we applied the traditional setting reported by [11]. Finally, for training and testing networks with dataset IV-2a, a set of 2D EEG signals composed of 22 (the number of electrodes) $\times$ 1,125 (250Hz $\times$ 4.5s) values were prepared. On the other hand, for training and testing networks with dataset IV-2b, a set of 2D EEG signals composed of 3 (the number of channels) $\times$ 500 (250Hz $\times$ 2s) values were prepared. According to the evaluation protocol of [45], a classifier was trained and tested for each subject.

The state-of-the-art CNNs [16], [17] with and without the proposed MTPP architecture were implemented using the BrainDecode framework [16], which provides various features for EEG processing. All the experiments were conducted on a workstation PC equipped with 2 NVidia GeForce RTX 2080 Ti GPUs, 64GB RAM, and an Intel Core i9-7920x.

B. PERFORMANCE EVALUATION

To validate the effectiveness of the proposed method, we re-implemented the state-of-the-art CNN-based approaches, such as ShallowNet [16], DeepNet [16], and EEGNet [17], to replace the standard pooling layers used in the networks with the proposed TPP layers. The structures of ShallowNet, DeepNet, and EEGNet are depicted in Tables 13, 12, and 14, respectively, in the Appendix. As presented in Tables 13, 12, and 14, each network has a different number of pooling layers of different types. The DeepNet architecture adopts 4 max-pooling layers while ShallowNet and EEGNet adopt a single and two average-pooling layers, respectively. Therefore, DeepNet and EEGNet can utilize both the single standard TPP layer (Figure 4) and the multilayered TPP architecture (Figure 5), while ShallowNet can only utilize a single standard TPP layer (Figure 4). The revised version of the original networks (i.e., ShallowNet, DeepNet, and EEGNet) are abbreviated as Shallow++, Deep++, and EEGNet++ throughout the manuscript. The original (ShallowNet, DeepNet, and EEGNet) and the revised networks (Shallow++, Deep++, and EEGNet++) share the same configuration except for the types of pooling layers (i.e., standard pooling for the original networks and temporal pyramid pooling for the revised networks). In our experiment, we used a 3-level pyramid pooling (i.e., 3 pooling operations with different sizes) approach. The pooling sizes of each window for each network that are used to decode the dataset IV-2a are summarized in Table 1. Similarly, the parameters used to decode dataset IV-2b are summarized in Table 2. With this setup, we first compare the classification accuracy of each network with and without the proposed TPP layers.

TABLE 1. Pooling sizes of TPP layers for each network on dataset IV-2a.

	Shallow++	*Deep++*	*EEGNet++*
pool window #1	120×1	3×1	6×1
pool window #2	260×1	8×1	42×1
pool window #3	290×1	25×1	98×1

TABLE 2. Pooling sizes of TPP layers for each network on dataset IV-2b.

	Shallow++	*Deep++*	*EEGNet++*
pool window #1	40×1	3×1	8×1
pool window #2	200×1	6×1	64×1
pool window #3	250×1	19×1	74×1

Table 3 compares the average classification accuracy of the CNN architectures used on the BCI competition IV-2a dataset. Table 3 shows that the proposed MTPP layers improved the average classification accuracy of the original networks with some exceptions. First, in the case of the networks with a single TPP layer (i.e., TPP column), ShallowNet achieved a significant performance improvement of 5.06%p (73.57 to 78.63, $p<0.05$ with Wilcoxon signed-rank test) and DeepNet achieved a moderated performance improvement of 2.63%p (58.37 to 61.00, $p<0.1$ with Wilcoxon signed-rank test), respectively. However, EEGNet failed to realize any performance improvement from adopting the single TPP layer. Rather, it showed a decrease in the average classification accuracy (i.e., 71.95 to 71.80). Conversely, the performance of both DeepNet and EEGNet slightly improved when adopting the multilayered TPP architecture. DeepNet showed a moderate performance improvement of 2.7%p (58.37 to 61.07, $p=0.12$ with Wilcoxon signed-rank test) while EEGNet achieved a marginal performance improvement of 1.24%p (71.95 to 73.19, $p=0.21$ with Wilcoxon signed-rank test). However, their performance improvements were not statistically significant. In summary, among the various configurations, ShallowNet with a single TPP layer achieved the highest average classification accuracy (i.e., 78.63) and performance improvement compared to its original configuration (+5.06), and outperformed all the other methods with regard to the average classification accuracy. From this experiment, it is evident that ShallowNet, with only a single max-pooling layer, even significantly benefits from the TPP method. Furthermore, it was observed that a multilayered TPP approach results in a higher performance gain for deeper networks (i.e., DeepNet and EEGNet).

TABLE 3. Average classification accuracy of each network with and without TPP layers on dataset IV-2a. The numbers in parentheses denote the improvement in performance when compared to that of the original architecture. Stars indicate statistically significant differences compared to the original network (Wilcoxon test, p<0.1:*, P<0.05:**, P<0.01:***).

	Original	*TPP*	*MTPP*
ShallowNet	73.57	78.63** (+5.06)	N/A
DeepNet	58.37	61.00* (+2.63)	61.07 (+2.70)
EEGNet	71.95	71.80 (-0.15)	73.19 (+1.24)

TABLE 4. Average classification accuracy of each network with and without TPP layers on dataset IV-2b. The numbers in parentheses denote the improvement in performance when compared to that of the original architecture. Stars indicate statistically significant differences compared to the original network (Wilcoxon test, p<0.1:*, P<0.05:**, P<0.01:***).

	Original	*TPP*	*MTPP*
ShallowNet	76.02	78.00** (+1.98)	N/A
DeepNet	75.65	75.13 (-0.52)	78.05** (+2.40)
EEGNet	81.79	81.82 (+0.03)	82.83** (+1.04)

Table 4 compares the average classification accuracy of the CNN architectures applied on the BCI competition IV-2b dataset. Table 4 indicates that the results from the IV-2b dataset show a slightly different pattern when compared to those from the IV-2a dataset. For the IV-2b dataset, adopting a single TPP layer does not contribute to performance improvement of deeper networks (i.e., DeepNet and EEGNet). Only ShallowNet realized a moderate, statistically significant performance improvement of 1.98%p (76.02 to 78, p<0.05 with Wilcoxon signed-rank test). In the case of EEGNet, there were no significant changes (+0.03%p) in classification accuracy. Moreover, it was observed that the performance of DeepNet with a single TPP layer even decreased from 75.65 to 75.13 (−0.52%p). However, both DeepNet and EEGNet could benefit from the multilayer TPP architecture. In particular, DeepNet achieved a statistically significant improvement in performance, gaining 2.4%p compared to that of the original DeepNet architecture (75.65 to 78.05, p<0.05 with Wilcoxon signed-rank test). EEGNet also achieved a slight, but statistically significant performance improvement of 1.04%p (81.79 to 82.83, p<0.05 with Wilcoxon signed-rank test). To sum up, among the various configurations, EEGNet with the multilayer TPP approach achieved the highest average classification accuracy (i.e., 82.83) and outperformed all the other methods. DeepNet achieved the highest performance improvement (i.e., 2.40) with the multilayer TPP approach. Similar to the previous experiment, it was also observed that a multilayered TPP approach leads to higher performance gain for deeper networks (i.e., DeepNet and EEGNet).

The experimental results on BCI competition IV-2a and 2b datasets reveal the following implications. First, a single-layer TPP approach generally works better for shallow networks (i.e., ShallowNet). As we observed from Tables 3 and 4, the single TPP layer did not significantly contribute to the performance improvement, except for ShallowNet. Furthermore, both DeepNet and EEGNet even experienced a decrease in performance when using a single TPP layer. One of the main architectural differences between ShallowNet and deeper networks (i.e., DeepNet and EEGNet) is the number of convolutions and pooling layers applied. In case of the deeper networks, more convolution and pooling operations are applied to the feature maps. Therefore, we expect only limited data to remain at the TPP layer, placed at the end of the network, so that only a limited amount of performance improvement was achieved.

Second, deeper networks (i.e., DeepNet and EEGNet) experienced greater advantages from the proposed multilayer TPP approach than from the single-layer TPP approach. On dataset IV-2a, the multilayer TPP approach resulted in the performance improvement for both DeepNet (+2.7) and EEGNet (+1.24), even though statistical significance was not observed (p=0.18–20 with Wilcoxon signed-rank test). Conversely, on dataset IV-2b, both DeepNet and EEGNet benefited from the application of the multilayer TPP approach with a statistical significance (p<0.05 with Wilcoxon signed-rank test). This is also consistent with our expectation mentioned in Section IV-B that the MTPP can act as both multi-scale feature representation as well as feature-level self-augmentation. The multi-scale self-augmented feature generated by the TPP layer was more effective for the dataset IV-2b which has a set of EEG signals with relatively short length (i.e., 2.0s). This also implies that the proposed method can contribute to the reduction of inference delay which is important for the practical use of BCI applications. As mentioned in Section IV-B, all the standard pooling layers are replaced with TPP layers in our multilayer TPP approach. Instead of arbitrary subsampling of temporal information through standard pooling operations, we attempted to extract multiscale features and aggregate them for further processing. Therefore, the results of the multilayer TPP approach can be interpreted that various important temporal information were effectively captured by each TPP layer and then forwarded to the subsequent layers, thereby contributing to the performance improvement of deeper networks.

Finally, EEGNet achieved the lowest performance gain from using the TPP approaches. However, the classification accuracy of EEGNet was still comparable to that of the other architectures. The single TPP layer failed to improve the performance of EEGNet for both datasets. Additionally, the performance improvement by a multilayer TPP approach was less than 1.3%p. As mentioned above, the main idea of EEGNet is to adopt a depth-wise separable convolution layer, which reduces the number of parameters, thereby configuring a compact structure. The number of parameters of EEGNet trained for both datasets IV-2a and IV-2b are much lower than those of ShallowNet and DeepNet (reviewed in the next section), implying that the amount of available information that the TPP layers can extract is very limited. Thus, there was only little room for performance improvement. Hence, we expect relatively lower performance benefits of adopting

TABLE 5. Subject-level classification accuracy on BCI competition IV-2a dataset. Stars indicate statistically significant differences compared to the FBCSP algorithm (Wilcoxon test, p<0.1:*, P<0.05:**, P<0.01:***).

Subject	FBCSP	ShallowNet	Shallow++	DeepNet	Deep++	EEGNet	EEGNet++
#1	76.00	81.25	89.93	67.36	78.13	78.13	83.68
#2	56.50	54.86	56.25	42.01	39.93	57.99	49.31
#3	81.25	84.38	89.93	76.74	80.90	93.40	92.36
#4	61.00	82.99	80.90	56.25	56.94	63.54	61.46
#5	55.00	53.13	70.49	28.82	27.08	52.78	63.19
#6	42.25	51.39	58.68	33.68	39.93	52.43	54.86
#7	82.75	92.36	93.40	68.75	76.39	88.89	88.54
#8	81.25	81.60	83.33	75.69	72.57	79.17	80.90
#9	70.75	80.21	84.72	76.04	77.78	81.25	84.38
Average	67.75	73.57**	78.63***	58.37***	61.07**	71.95**	73.19**

TABLE 6. Subject-level classification accuracy on BCI competition IV-2b dataset. Stars indicate statistically significant differences compared to the FBCSP algorithm (Wilcoxon test, p<0.1:*, P<0.05:**, P<0.01:***).

Subject	FBCSP	ShallowNet	Shallow++	DeepNet	Deep++	EEGNet	EEGNet++
#1	73.50	70.00	76.25	68.75	75.31	78.44	78.75
#2	59.40	53.93	56.07	56.07	59.29	66.07	66.43
#3	61.90	53.13	55.63	53.75	55.00	64.06	67.50
#4	71.50	96.88	96.25	95.63	95.31	95.94	95.00
#5	61.40	85.94	89.06	82.81	89.38	91.88	94.38
#6	70.10	76.25	80.00	77.81	80.94	83.13	84.38
#7	69.60	77.19	77.81	75.63	75.94	82.81	85.31
#8	62.00	90.94	87.81	88.44	90.94	90.00	92.19
#9	75.50	80.00	83.13	81.88	80.31	83.75	81.56
Average	67.21	76.03*	78.00**	75.65*	78.05**	81.79***	82.83***

TPP layers on EEGNet to be attributed to the compactness of the EEGNet.

Next, we provide the classification accuracies for each subject from the classification models. In this experiment, we added a well-known classical classification method, FBCSP-based algorithm [14], as another baseline method. The reported scores in these experiments are from the best performing model (i.e., Deep++ and EEGNet++ indicate the revised version of DeepNet and EEGNet with a multi-layer TPP approach).

Table 5 presents the subject-level results of performance evaluation on BCI competition IV-2a dataset. First, most of the CNN-based approaches, except DeepNet (58.37) and Deep++ (61.07), outperformed the traditional FBCSP-based method (67.75). ShallowNet and Shallow++ showed performance improvements of 5.82%p ($p<0.05$ with Wilcoxon signed-rank test) and 10.88%p ($p<0.01$ with Wilcoxon signed-rank test) higher than those observed using FBCSP method, respectively. Similarly, EEGNet and EEGNet++ realized performance improvements of 4.2%p ($p<0.05$ with Wilcoxon signed-rank test) and 5.44%p ($p<0.05$ with Wilcoxon signed-rank test) higher than those observed using the FBCSP method, respectively. Contrarily, the proposed TPP layers contributed to improving the performance for most subjects. In particular, Shallow++ worked better than the original ShallowNet, except for subject No.4. More specifically, Shallow++ demonstrated a performance improvement of 5.96% on average. Deep++ achieved an average performance improvement of 5.21%, except for subjects No.2, No.5, and No.8, compared with that observed using the the original DeepNet architecture. EEGNet++ outperformed the original EEGNet architecture with an average performance improvement of 4.7%, except for subjects No.2, No.3, No.4, and No.7.

Table 6 summarizes the subject-level results of classification accuracy on BCI competition IV-2b dataset. In this dataset, the FBCSP-based method obtained an average classification accuracy of 67.21%, which was still lower than the worst performing network (i.e., DeepNet), which produced a classification accuracy of 75.65%. All the CNN-based approaches produced statistically significant performance improvements when compared to that resulting from the application of the FBCSP method. In particular, EEGNet and EEGNet++ observed the highest performance improvements when compared to the FBCSP method (i.e., +14.58%p and +15.62%p, $p<0.01$ with Wilcoxon signed-rank test). On the other hand, the proposed TPP layers contributed to the performance improvement for all the CNN-based approaches. However, the TPP layers negatively affected the classification performance for subject No.4. All the CNN approaches when adopted with the MTPP layers experienced an average decrease in performance of -0.62%p for subject No.4. Furthermore, performance degradation was observed in certain other cases. The performance of Shallow++ for subject No.8 decreased by 3.13%p compared to the performance of the original ShallowNet (90.94 to 87.81). Additionally,

in the case of subject No.9, the performance deteriorated for Deep++ (from 81.88 to 80.31) and EEGNet++ (from 83.75 to 81.56). These negative results imply that further optimizations are required to realize a more robust performance.

C. CHANGES IN PERFORMANCE BASED ON NETWORK PARAMETERS

In this section, we analyze the effects of different network configurations on the overall performance of the CNNs. Specifically, the effects of the window level of the TPP on the classification accuracy and the number of trainable parameters of each network were analyzed. As explained in Section III, an l-level pyramid pooling layer consists of l pooling operations wherein each operation has a different pooling size. If the window level of TPP is set to be 3, then 3 intermediate pooled features are extracted in a pooling layer and aggregated into a single feature representation.

Figure 8 shows the change in average classification accuracy based on the window level of the pyramid pooling applied on the IV-2a dataset. As shown in Figure 8, the performances of Shallow++ and EEGNet++ slightly improved as the window level of pyramid pooling increased. On the other hand, the Deep++ realized the best result (61.2) with 2-level pyramid pooling. However, this was not significantly different from that observed with the 3-level TPP (61.1). For this dataset, the average performance of a 2-level pooling was 70.4 while that with a 3-level pooling was 71.0. Similarly, Figure 9 shows the changes in average classification accuracy based on the window level of pyramid pooling applied on the IV-2b dataset. In this case, the performance of Deep++ slightly improved as the window level increased. Furthermore, the Deep++ experienced the highest improvement in performance corresponding to 0.8%p (77.3 to 78.1) while the Shallow++ and EEGNet++ architectures exhibited negligible improvements in performance (i.e., 77.9 to 78.0 for Shallow++ and 82.9 to 82.8 for EEGNet++). For this dataset, the average performance of a 2-level pooling was 79.3 while that applying a 3-level pooling was 79.6. Thus, as seen in Figures 8 and 9, we found that the overall average

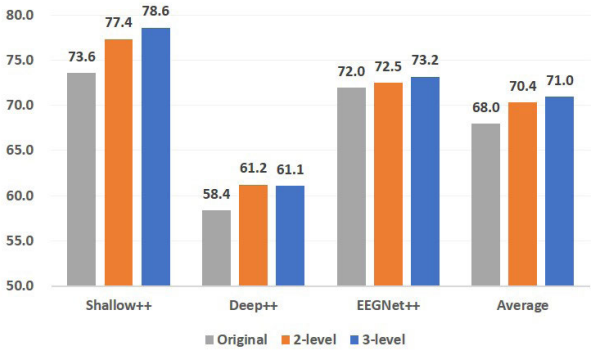

FIGURE 8. Change in accuracy based on the window level of pyramid pooling on dataset IV-2a.

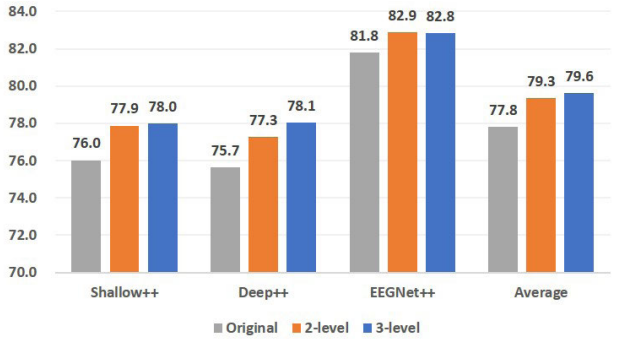

FIGURE 9. Change in accuracy based on the window level of pyramid pooling on dataset IV-2b.

performance of the revised networks tended to increase as the window level of pyramid pooling increased.

We also investigated the effects of the window level in a temporal pyramid pooling operation on subject-level classification accuracy. For this analysis, we chose the network and TPP configurations which achieved performance improvements with a strong statistical significance ($p<.05$) as seen in Tables 3 and 4. Among the networks with a single TPP method, only ShallowNet++ achieved significant performance improvements on both datasets IV-2a and IV-2b. DeepNet++ and EEGNet++ could achieve significant performance improvements on dataset IV-2b only when the multilayer TPP approach was used.

Tables 7 and 8 show the effects of window level in a single (or multilayer) TPP on the performance for each subject. The numbers in each column indicate the performance improvement compared to the performance from the original networks (i.e., without the TPP approach). The numbers that are bold in Tables 7 and 8 indicate the winning methods for each subject.

First, how the window level of a single TPP layer used in ShallowNet++ affects the performance for each subject on both datasets is summarized in Table 7. As described in Table 7, it cannot be said that a specific window level was always effective for all the subjects in both the datasets. For example, the results for dataset IV-2a show that the 2-level window configuration was effective for subjects No.2, No.3, and No.7, while subjects No.1, No.5, No.8, and No.9 did benefit from the 3-level window configuration. A similar pattern is also observed for dataset IV-2b. In particular, subject No.4 in dataset IV-2a and subjects No.4 and No.8 in dataset IV-2b did not benefit from the proposed method, consistent with the results presented in Tables 5 and 6.

The effect of the window level in the multilayered TPP used in DeepNet++ and EEGNet++ architectures on subject-level performance is summarized in Table 8. Similar to the result in Table 7, we cannot conclude that a specific window level was always effective for all subjects. In the case of DeepNet++, the 2-level window configuration was effective for subjects No.5 and No.8, while subjects No.1,

TABLE 7. Effect of the window level in a single TPP approach (ShallowNet++).

	Dataset IV-2a		Dataset IV-2b	
Subject	*2-level*	*3-level*	*2-level*	*3-level*
#1	+7.99	**+8.68**	+6.25	+6.25
#2	**+3.47**	+1.39	**+3.21**	+2.14
#3	**+8.68**	+5.56	+2.50	+2.50
#4	-4.17	-2.08	-0.31	-0.63
#5	+9.03	**+17.36**	+1.25	**+3.12**
#6	+7.29	+7.29	**+5.63**	+3.75
#7	**+1.39**	+1.04	+0.62	+0.62
#8	-0.69	**+1.74**	-5.00	-3.13
#9	+1.04	**+4.51**	+2.50	**+3.13**

TABLE 8. Effect of the window level in a multilayer TPP approach on dataset IV-2b.

	DeepNet++		EEGNet++	
Subject	*2-level*	*3-level*	*2-level*	*3-level*
#1	+3.13	**+6.56**	**+1.56**	+0.31
#2	+1.79	**+3.21**	**+3.22**	+0.36
#3	+0.62	**+1.25**	+1.25	**+3.44**
#4	+0.00	-0.31	-0.31	-0.94
#5	**+6.88**	+6.56	-0.94	**+2.50**
#6	+0.94	**+3.13**	-0.94	**+1.25**
#7	+0.31	+0.31	+2.50	+2.50
#8	**+3.75**	+2.50	**+2.81**	+2.19
#9	-2.81	-1.56	**+0.63**	-2.19

TABLE 9. Number of parameters based on the window level of pyramid pooling on dataset IV-2a.

	Shallow++	*Deep++*	*EEGNet++*
original	47,364	284,479	3,700
2	38,404	286,079	3,956
3	38,884	324,479	4,276

TABLE 10. Number of parameters based on the window level of pyramid pooling on dataset IV-2b.

	Shallow++	*Deep++*	*EEGNet++*
original	8,082	265,802	1,634
2	6,482	270,202	1,442
3	7,042	279,002	1,506

TABLE 11. Training and testing times of CNN architectures for each dataset (unit: s).

	Dataset IV-2a		Dataset IV-2b	
	Training	*Testing*	*Training*	*Testing*
Shallow	54.86	0.08	8.52	0.01
Shallow++	53.66	0.08	8.73	0.01
Deep	161.2	0.13	89.15	0.18
Deep++	161.2	0.13	474.15	0.35
EEGNet	14.57	0.02	11.41	0.01
EEGNet++	16.54	0.02	12.98	0.01

No.2, No.3, and No.6 did benefit from the 3-level window configuration. In the case of EEGNet++, subjects No.1, No.2, No.8, and No.9 did benefit from the 2-level window configuration, while the performances for subjects No.3, No.5, and No.6 were improved when using the 3-level window configuration. Similar to the case of ShallowNet++, we could also find that certain subjects failed to benefit from the proposed multilayered TPP approach. The results from this investigation imply that a higher window level does not necessarily yield better performance in terms of subject-level motor imagery classification. Also, it was observed that network architecture and a subject's characteristics must be carefully considered to achieve a robust performance.

Next, we compared the number of trainable parameters based on the window level of the TPP. Table 9 and 10 summarize the results of datasets IV-2a and IV-2b, respectively. These results show that the number of parameters tends to increase as the window level of pyramid pooling increases. However, this does not apply to all cases, as was also reported in [20]. For example, the number of parameters of the ShallowNet architecture for the IV-2a dataset decreased from 47,364 (original) to 38,404 (2-level) and 38,884 (3-level). This is potentially due to the difference in the pooling sizes for each window, compared to that of the original network. This also implies that the performance of ShallowNet can be improved even with a lower number of trainable parameters. Based on these results, we configured our revised networks with a 3-level window. However, we could not observe any specific pattern in the effects of pooling size for each window on the overall performance. Therefore, we chose the pooling sizes used in each network that showed the best performance during the experiments.

Finally, we compared the training and testing times of each network for each dataset to validate the feasibility of the proposed approach for online BCI applications. According to the analysis presented in [16], the FBCSP method was substantially faster to train than the CNN-based approaches; however, the online application of the trained neural networks did not suffer from the speed disadvantage compared to the FBCSP method. The authors of [16] revealed that the high prediction speed of ShallowNet and DeepNet make them well suited for decoding in real-time BCI applications. As the processing time of the networks with the proposed method are highly dependent on the original networks, therefore, we compared the training and testing times of the revised networks that adopt the proposed TPP layers with those of the original networks. Table 11 summarizes the comparison of training and testing times required for processing all the trials of a single subject between the original and the revised networks. As described in this table, we could not find a significant difference between the original networks and the revised networks for all the cases except for DeepNet on dataset IV-2b. Therefore, it is believed that the prediction speeds of the revised networks using the proposed method were fast enough to be used for real-time BCI applications. However, it was also inferred that more sophisticated optimization techniques need to be considered to reduce the training time of the networks.

VI. CONCLUSION AND FUTURE WORK

In this paper, we discussed the concept of pyramid pooling designed to improve the performance of CNNs. We then implemented two types of pyramid pooling for decoding motor-imagery EEG signals. First, a basic TPP approach was applied by replacing the last standard pooling layer of the network with the TPP layer. This was similar to a SPP approach commonly used in the field of computer vision. Subsequently, we extended the concept of single-layer TPP to a multilayered approach by replacing every standard pooling layer of the network with a TPP layer. Furthermore, extensive experiments were conducted on the BCI competition IV-2a and 2b datasets to assess the effect of the proposed method. The experimental results demonstrated that the proposed method could successfully improve the performance of the original CNN architectures for decoding MI EEG signals. Specifically, we observed that a single TPP layer led to a significant improvement in the performance of ShallowNet. On the other hand, the results indicated that multilayer TPP is more useful for deeper networks such as DeepNet and EEGNet. However, the negative effects of the proposed method were also observed in some cases.

Even though we proposed a novel mechanism and validated its effectiveness through various experiments, there remains a lot of room to improve the quality of feature representation for decoding motor imagery EEG signals. Recently, various techniques for feature extraction and representation have been proposed and evaluated in the computer vision and deep learning fields. Examples of these include an atrous (dilated) convolution operation [46]–[48] and an encoder-decoder architecture with a deconvolution or up-sampling operation [49], [50]. Despite the success of the aforementioned techniques for visual feature representation, they have not been extensively studied in the MI-EEG BCI domain. Atrous convolution is known to enlarge the field of view of filters to incorporate a larger visual context even with the same number of parameters [46]. In the EEG domain, we expect that the proper adaptation of atrous convolutions can facilitate the extraction and representation of spatio-temporal features. However, the effects of atrous convolution in recent CNN architectures for motor imagery classification are still not clear. Hence, more studies on the optimized network architecture, such as dilation factors, are required. On the other hand, an upsampling mechanism was introduced and applied to various semantic segmentation tasks [49]. The authors of [49] applied an upsampling mechanism which can reconstruct dense feature maps from coarse representation of images, thereby improving the quality of semantic segmentation outputs. Despite the success of an encoder-decoder architecture in the computer vision field, obtaining a fine grained representation of EEG feature maps from the coarse ones is still challenging. However, there will be a lot of potential to address various issues of MI-EEG-based systems if an upsampling mechanism can be successfully realized and integrated with atrous convolution layers.

TABLE 12. Architecture of DeepNet. In case of DeepNet with a single TPP layer, only the pooling layer in the 4th block is replaced with the TPP layer. In case of DeepNet with MTPP layers, every max-pooling layer in the network is replaced with a TPP layer.

Blocks	Layers	Configuration
1	Convolution Convolution Max-pooling	25 kernels, 1×10 25 kernels, E×1 1×3
2	Convolution Max-pooling	50 kernels, 1×10 1×3
3	Convolution Max-pooling	100 kernels, 1×10 1×3
4	Convolution Max-pooling	200 kernels, 1×10 1×3
5	Fully connected	Flatten Softmax

TABLE 13. Architecture of ShallowNet. In case of ShallowNet, only a single pooling layer exists. Therefore, a pooling layer in the 2nd block is replaced with a TPP layer to implement ShallowNet with a TPP layer.

Blocks	Layers	Configuration
1	Convolution Convolution	40 kernels, 1×25 40 kernels, E×1
2	Average pooling	1×75
3	Fully connected	Flatten Softmax

In addition to the atrous convolution and upsampling mechanisms, self-supervised learning [51], [52] is another promising approach to overcome current limitations of MI-EEG-based applications. In a self-supervised learning pipeline, a network model first tries to learn the feature representations from the data themselves without relying on predefined annotations. This is done through a pretext (proxy) task which is a form of unsupervised learning where the data provide supervision. Once the model training through the pretext task is completed, intermediate layers (feature layers) of the trained network are used for fine tuning on a specific task (downstream task) of interest. For example, [53] proposed a self-supervised learning framework to learn image features from the object rotation prediction task (pretext task), and exploit the learned features for image classification, object detection, and segmentation tasks (downstream tasks). The authors of [53] showed that a model trained with a self-supervised learning mechanism can achieve a performance comparable to those of the models learned in a supervised manner. This approach is particularly useful for a domain where it is difficult to collect a large amount of training data and their corresponding labels. We believe that it will be possible to learn the informative feature representation of the EEG signals based on the self-supervised learning framework, if an appropriate design of a pretext task for the EEG domain is proposed.

As stated above, various advanced techniques for learning feature representation of data have been proposed in

TABLE 14. Architecture of EEGNet. In case of EEGNet with a single TPP layer, only the pooling layer in the 2nd block is replaced with the TPP layer. In case of EEGNet with MTPP layers, every average-pooling layer in the network is replaced with a TPP layer.

Blocks	Layers	Configuration
1	Convolution	8 kernels, 1×64
	Depth-wise convolution	16 kernels, E×1
	Average pooling	1×4
2	Separable convolution	16 kernels, 1×16
	Average pooling	1×8
3	Fully connected	Flatten
		Softmax

recent years. However, the application of these techniques to the MI-EEG field still requires significant research. For future studies, we plan to optimize the TPP module and integrate the proposed MTPP approach with other advanced deep learning techniques, such as atrous convolution, upsampling layers, and self-supervised learning, to ensure more robust and reliable performance. Finally, we will study how the improved version of the proposed approach can be applied to other domains.

APPENDIX
CNN ARCHITECTURES

See Tables. 12, 13, 14.

REFERENCES

[1] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of EEG-based brain–computer interface paradigms," *J. Neural Eng.*, vol. 16, no. 1, Jan. 2019, Art. no. 011001.

[2] N. Birbaumer, "Breaking the silence: Brain-computer interfaces (BCI) for communication and motor control," *Psychophysiology*, vol. 43, no. 6, pp. 517–532, Nov. 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.2006.00456.x

[3] R. Alazrai, H. Alwanni, Y. Baslan, N. Alnuman, and M. Daoud, "EEG-based brain-computer interface for decoding motor imagery tasks within the same hand using choi-williams time-frequency distribution," *Sensors*, vol. 17, no. 9, p. 1937, Aug. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/9/1937

[4] K.-T. Kim, T. Carlson, and S.-W. Lee, "Design of a robotic wheelchair with a motor imagery based brain-computer interface," in *Proc. Int. Winter Workshop Brain-Computer Interface (BCI)*, Feb. 2013, pp. 46–48.

[5] L. Jiang, E. Tham, M. Yeo, Z. Wang, and B. Jiang, "Motor imagery controlled wheelchair system," in *Proc. 9th IEEE Conf. Ind. Electron. Appl.*, Jun. 2014, pp. 532–535.

[6] B. Xu, W. Li, X. He, Z. Wei, D. Zhang, C. Wu, and A. Song, "Motor imagery based continuous teleoperation robot control with tactile feedback," *Electronics*, vol. 9, no. 1, p. 174, Jan. 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/1/174

[7] N. Padfield, J. Zabalza, H. Zhao, V. Masero, and J. Ren, "EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges," *Sensors*, vol. 19, no. 6, p. 1423, Mar. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/6/1423

[8] G. Rodríguez-Bermádez and P. J. García-Laencina, "Automatic and adaptive classification of electroencephalographic signals for brain computer interfaces," *J. Med. Syst.*, vol. 36, no. S1, pp. 51–63, Nov. 2012, doi: 10.1007/s10916-012-9893-4.

[9] Q. Novi, C. Guan, T. Huy Dat, and P. Xue, "Sub-band common spatial pattern (SBCSP) for brain-computer interface," in *Proc. 3rd Int. IEEE/EMBS Conf. Neural Eng.*, May 2007, pp. 204–207.

[10] H. Zhiping, C. Guangming, C. Cheng, X. He, and Z. Jiacai, "A new EEG feature selection method for self-paced brain-computer interface," in *Proc. 10th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2010, pp. 845–849.

[11] K. K. Ang, Y. Chin, H. Zhang, and G. Uan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jun. 2008, pp. 2390–2397.

[12] M. Z. Ilyas, P. Saad, M. I. Ahmad, and A. R. I. Ghani, "Classification of EEG signals for brain-computer interface applications: Performance comparison," in *Proc. Int. Conf. Robot., Autom. Sci. (ICORAS)*, Nov. 2016, pp. 1–4.

[13] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, 2008.

[14] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Frontiers Neurosci.*, vol. 6, p. 39, Mar. 2012.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[16] R. T. Schirrmeister, J. T. Springenberg, L. D. J. 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," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Aug. 2017, doi: 10.1002/hbm.23730.

[17] 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 brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, Oct. 2018, Art. no. 056013.

[18] G. Zhang and A. Etemad, "RFNet: Riemannian fusion network for EEG-based brain-computer interfaces," 2020, *arXiv:2008.08633*. [Online]. Available: https://arxiv.org/abs/2008.08633

[19] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 346–361.

[21] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, "Review of the BCI competition IV," *Frontiers Neurosci.*, vol. 6, p. 55, Dec. 2012. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2012.00055

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*. [Online]. Available: http://arxiv.org/abs/1511.07289

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[25] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 3856–3866. [Online]. Available: http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[27] K.-W. Ha and J.-W. Jeong, "Motor imagery EEG classification using capsule networks," *Sensors*, vol. 19, no. 13, p. 2854, Jun. 2019.

[28] N. Akhtar and U. Ragavendran, "Interpretation of intelligence in CNN-pooling processes: A methodological survey," *Neural Comput. Appl.*, vol. 32, no. 3, pp. 879–898, Feb. 2020, doi: 10.1007/s00521-019-04296-5.

[29] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2, Dec. 2005, pp. 1458–1465.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Dec. 2006, pp. 2169–2178.
[31] J. Yue, S. Mao, and M. Li, "A deep learning framework for hyperspectral image classification using spatial pyramid pooling," *Remote Sens. Lett.*, vol. 7, no. 9, pp. 875–884, Sep. 2016, doi: 10.1080/2150704X.2016.1193793.
[32] T. Qu, Q. Zhang, and S. Sun, "Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21651–21663, Oct. 2017, doi: 10.1007/s11042-016-4043-5.
[33] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sens.*, vol. 9, no. 8, p. 848, Aug. 2017.
[34] J. I. Toledo, S. Sudholt, A. Fornés, J. Cucurull, G. A. Fink, and J. Lladós, "Handwritten word image categorization with convolutional neural networks and spatial pyramid pooling," in *Structural, Syntactic, and Statistical Pattern Recognition*, A. Robles-Kelly, M. Loog, B. Biggio, F. Escolano, and R. Wilson, Eds. Cham, Switzerland: Springer, 2016, pp. 543–552.
[35] S. Guo, T. Yang, W. Gao, C. Zhang, and Y. Zhang, "An intelligent fault diagnosis method for bearings with variable rotating speed based on pythagorean spatial pyramid pooling CNN," *Sensors*, vol. 18, no. 11, p. 3857, Nov. 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3857
[36] J. Zhang, Y. Dai, F. Porikli, and M. He, "Multi-scale salient object detection with pyramid spatial pooling," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2017, pp. 1286–1291.
[37] J. Li, Y. Si, L. Lang, L. Liu, and T. Xu, "A spatial pyramid pooling-based deep convolutional neural network for the classification of electrocardiogram beats," *Appl. Sci.*, vol. 8, no. 9, p. 1590, Sep. 2018.
[38] P. Wang, Y. Cao, C. Shen, L. Liu, and H. T. Shen, "Temporal pyramid pooling-based convolutional neural network for action recognition," *IEEE Trans. Cir. Sys. Video Technol.*, vol. 27, no. 12, p. 2613–2622, Dec. 2017, doi: 10.1109/TCSVT.2016.2576761.
[39] C. Cheng, P. Lv, and B. Su, "Spatiotemporal pyramid pooling in 3D convolutional neural networks for action recognition," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3468–3472.
[40] Z. Zheng, G. An, and Q. Ruan, "Temporal pyramid pooling based relation network for action recognition," in *Proc. 14th IEEE Int. Conf. Signal Process. (ICSP)*, Dec. 2018, pp. 644–647.
[41] J. Zhu, Z. Zhu, and W. Zou, "End-to-end video-level representation learning for action recognition," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 645–650.
[42] Z. Yu, X. Xu, X. Chen, and D. Yang, "Temporal pyramid pooling convolutional neural network for cover song identification," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4846–4852, doi: 10.24963/ijcai.2019/673.
[43] S. Soleymani, A. Dabouei, H. Kazemi, J. Dawson, and N. M. Nasrabadi, "Multi-level feature abstraction from convolutional neural networks for multimodal biometric identification," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3469–3476.
[44] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: https://arxiv.org/abs/2004.10934
[45] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "Brain computer communication: Motivation, aim, and impact of exploring a virtual apartment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 4, pp. 473–482, Dec. 2007.
[46] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122
[47] R. Xi, M. Li, M. Hou, M. Fu, H. Qu, D. Liu, and C. R. Haruna, "Deep dilation on multimodality time series for human activity recognition," *IEEE Access*, vol. 6, pp. 53381–53396, 2018.
[48] R. Wang, J. Fan, and Y. Li, "Deep multi-scale fusion neural network for multi-class arrhythmia detection," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2461–2472, Sep. 2020.
[49] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[50] Y. Zhang, Z. Zhang, Y. Zhang, J. Bao, Y. Zhang, and H. Deng, "Human activity recognition based on motion sensor using u-net," *IEEE Access*, vol. 7, pp. 75213–75226, 2019.
[51] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6706–6716.
[52] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2070–2079.
[53] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*. [Online]. Available: http://arxiv.org/abs/1803.07728

KWON-WOO HA received the B.S. and M.S. degrees in computer engineering from the Kumoh National Institute of Technology, South Korea, in 2017 and 2020, respectively. His research interests include deep learning, machine learning, and brain–computer interfaces.

JIN-WOO JEONG received the Ph.D. degree in computer science and engineering from Hanyang University, South Korea, in 2013.

From 2013 to 2016, he was a Senior Research Engineer with the Software Research and Development Center, Samsung Electronics. Since 2016, he has been an Assistant Professor with the Department of Computer Engineering, Kumoh National Institute of Technology, Gumi, South Korea. His research interests include deep learning, machine learning, human–computer interaction, and multimedia information retrieval.