

Research Article

Automatic Sleep Stage Classification Based on Convolutional Neural Network and Fine-Grained Segments

Zhihong Cui,^{1,2} Xiangwei Zheng^{1,2}, Xuexiao Shao,^{1,2} and Lizhen Cui³

¹School of Information Science and Engineering, Shandong Normal University, Ji'nan 250014, China

²Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Ji'nan 250014, China

³School of Software, Shandong University, Ji'nan 250101, China

Correspondence should be addressed to Xiangwei Zheng; xwzhengcn@163.com

Received 13 March 2018; Revised 29 June 2018; Accepted 29 August 2018; Published 8 October 2018

Academic Editor: Matilde Santos

Copyright © 2018 Zhihong Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sleep stage classification plays an important role in the diagnosis of sleep-related diseases. However, traditional automatic sleep stage classification is quite challenging because of the complexity associated with the establishment of mathematical models and the extraction of handcrafted features. In addition, the rapid fluctuations between sleep stages often result in blurry feature extraction, which might lead to an inaccurate assessment of electroencephalography (EEG) sleep stages. Hence, we propose an automatic sleep stage classification method based on a convolutional neural network (CNN) combined with the fine-grained segment in multiscale entropy. First, we define every 30 seconds of the multichannel EEG signal as a segment. Then, we construct an input time series based on the fine-grained segments, which means that the posterior and current segments are reorganized as an input containing several segments and the size of the time series is decided based on the scale chosen depending on the fine-grained segments. Next, each segment in this series is individually put into the designed CNN and feature maps are obtained after two blocks of convolution and max-pooling as well as a full-connected operation. Finally, the results from the full-connected layer of each segment in the input time sequence are put into the *softmax* classifier together to get a single most likely sleep stage. On a public dataset called ISRUC-Sleep, the average accuracy of our proposed method is 92.2%. Moreover, it yields an accuracy of 90%, 86%, 93%, 97%, and 90% for stage W, stage N1, stage N2, stage N3, and stage REM, respectively. Comparative analysis of performance suggests that the proposed method is better, as opposed to that of several state-of-the-art ones. The sleep stage classification methods based on CNN and the fine-grained segments really improve a key step in the study of sleep disorders and expedite sleep research.

1. Introduction

Sleep plays an important role in physical health and quality of life. Sleep diseases, such as insomnia and obstructive sleep apnea, may cause daytime sleepiness, depression, or even death [1]. Therefore, there is an urgent demand for an effective way to diagnose and cure sleep-related diseases. Sleep-related disease research, defined as sleep medicine, is already an important branch of medicine and has been involved in several clinical problems.

Sleep stage classification is the first step in the diagnosis of sleep-related diseases [2, 3]. The crucial step in sleep study is to collect polysomnographic (PSG) data from subjects during

the hours of sleep. The PSG data include EEG, electromyography (EMG), and electrocardiography (ECG), as well as respiratory effort and other biophysiological signals. Human experts study the different time series records and assign each time segment to a sleep stage according to reference nomenclature, such as the guidelines of the American Academy of Sleep Medicine (AASM) or those of Rechtschaffen and Kales (R&K). In this study, we use the AASM criterion, which categorizes sleep data into five stages, which are wake (W), stages 1–3 (N1, N2, and N3), and rapid eye movement (REM) [4]. These stages are defined by the electrical activities recorded by sensors placed at different parts of the body, as per their differences in proportions over a night.

TABLE 1: The AASM staging criteria.

Sleep stage	Classification criteria
Wake (W)	Most epochs in this stage consist of alpha (8–13 Hz) activity or low voltage, mixed (2–7 Hz) with frequency activity.
N1	Most epochs in this stage have low-voltage, mixed (2–7 Hz) activity; small epoch contains alpha (8–13 Hz) activity. Slow rolling movement usually lasts several seconds often seen in early N1.
N2	Typical features of this stage are the appearance of sleep spindles, <i>K</i> complex, and some high voltage activities (<2 Hz).
N3	Most epochs in this stage begin to show high-voltage (>75 μ V) and low-frequency (<2 Hz) activity.
REM	This stage has a pretty active performance. It has relatively low-voltage mixed-frequency (2–7 Hz) EEG with episodic rapid eye movements and absent or reduced chin EMG activity.

In Table 1, the main proportions of those sleep stages are briefly introduced.

Sleep stage classification has been investigated for decades and many state-of-the-art methods and clinical applications have been developed. The following literature introduces several methods on sleep stage classification for clinical disorder diagnosis, including *K*-means clustering [5], artificial neural network [6], dual tree [7], empirical mode decomposition (EMD) [8], and support vector machine (SVM) [9]. However, these traditional methods are mostly based on biological signal recognition and substantial manual features extracted from the preprocessed signals, which are prone to local optimization. Furthermore, the pattern of human brain signals is more profound than the existing understanding of human beings, which may cause information loss in manual feature extraction. In addition, feature extraction is a tedious and time-consuming task. It needs hours of hard work by professional experts, which also means it may be subjective. Above all, the accuracy and convenience of sleep stage classification methods are critical factors in the diagnosis of sleep-related diseases.

Recently, researchers proposed a deep learning model named CNN, which was inspired by the biological simulation results on the visual cortex of mammals. Compared with the traditional methods, it reduces the complexity of the network and number of weights because of its shared-weight network structure, which is similar to a biological neural network. Furthermore, it simplifies the computation process due to its ability to classify the EEG data without handcrafted feature extraction. The CNN is widely used in the field of object recognition [10] and image segmentation [11]. Although, using CNN for EEG classification is currently quite popular [12, 13], the method can be further improved. For a better sleep stage classification performance, we presented a new method, in which CNN architecture is combined with fine-grained segments.

This paper is organized as follows. In Section 2, we briefly introduce several state-of-the-art-related works. In Section 3, we describe in detail the proposed CNN architecture with fine-grained segments for sleep stage classification from multichannel EEG signals. In Section 4, we test it on a public dataset and compare its performance with several state-of-the-art methods as well as our previous work [14]. The final section concludes the paper and points out the future work.

2. Related Work

Sleep stage classification has been of great interest in the past few decades [15]. There have already been several studies on it. For example, Acharya et al. [16] creatively proposed a new sleep stage classification method based on high-order spectra (HOS), in which the authors extracted features based on unique bispectrum and bicoherence plots of various sleep stages and then used the Gaussian mixture model (GMM) classifier for automatic identification. It finally achieved an accuracy of 88.7% in a situation with five sleep stages. Sharma et al. [17] presented a new technique for sleep stage classification based on iterative filtering. They first got the amplitude envelope and instantaneous frequency (AM-FM) after mode extraction using an iterative and discrete energy separation algorithm (DESA), and then used the AM-FM to compute the Poincare plot descriptors and statistical measures, which are the input factors to the final classifiers which include naive Bayes, *k*-nearest neighbor, multilayer perceptron, C4.5 decision tree, and random forest. It achieved an average accuracy of 86.2% in the case of five sleep stages. Liang [18] originally proposed an automatic sleep-scoring method combining multiscale entropy (MSE) and autoregressive (AR) models. Its overall sensitivity and kappa coefficient can reach a level of 88.1% and 81% in a situation with five sleep stages, respectively. Acharya et al. [19] presented a comprehensive comparative review and analysis of 29 non-linear dynamic methods for sleep stage classification. They not only gave a bispectrum and cumulate plot of every sleep stage but also gave a feature ranking based on *F* values, which represents the discriminative performance of 29 features. Thus, it has a significant influence on guiding the process of sleep stage classification. To develop a robust and accurate portable system for a huge dataset, Sharma [20] developed a new sleep stage identification system based on a novel set of wavelet-based features extracted from a large EEG dataset. The authors creatively used log-energy (LE) and signal-fractal-dimension (SFD) as well as signal-sample-entropy (SSE) to extract features from seven subbands (SBs) which are decomposed by a novel three-band time-frequency localized (TBTFL) wavelet filter bank (FB). It achieved an accuracy of 91.5% in the case with five sleep stages using SVM as a classifier. For a better sleep stage classification performance, Zhu et al. [21] proposed a method based on graph domain features from single-channel EEG. The authors obtained a different visibility graph (DVG) by

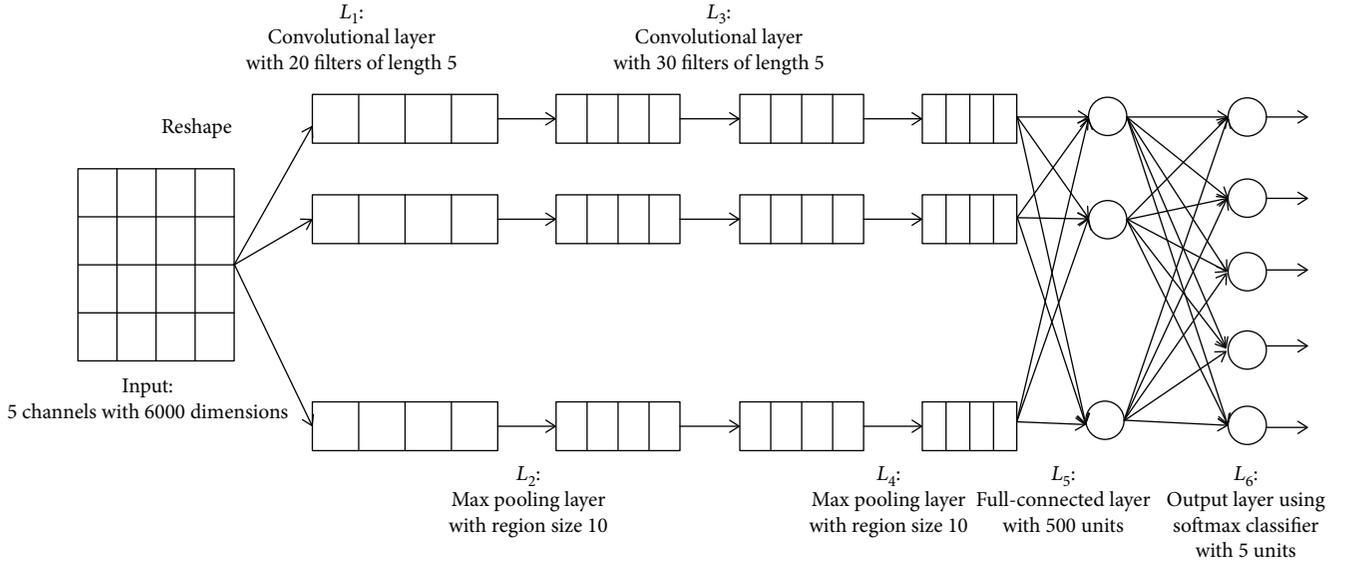


FIGURE 1: CNN architecture.

subtracting the edge set of the horizontal VG (HVG) from the edge set of the VG, and then putting nine features including the mean degrees (MDs) of each DVG and HVG as well as seven degree distributions (DDs) into the SVM to classify the 87% of sleep stages correctly. Tsinalis et al. [22] used CNN for automatic sleep stage scoring. It was designed to learn task-specific filters for classification based on single-channel EEG without using prior domain knowledge. Although, the CNN architecture has shown a very good performance in the field of classification, some researchers further optimized the CNN architecture for assigned tasks. For a better sleep stage classification, Chambon et al. [23] proposed the first end-to-end deep learning approach that performed automatic temporal sleep stage classification using multivariate and multimodel PSG signals. They constructed a general deep architecture which could extract information from EEG, EOG, and EMG channels and put the learned representations into a final *softmax* classifier. The temporal sleep stage classification means that the architecture learns from the temporal context of each sample. It can finally correctly classify 91% of the sleep stages.

In summary, there have been several papers on sleep stage classification, but none of these has combined CNN with fine-grained segments for sleep stage classification from multichannel time series. Thus, our method is original and novel.

3. Methods

In this section, we detail a CNN architecture combined with fine-grained segments for sleep stage classification from multichannel EEG time series.

In our CNN architecture, we first denote a segment of EEG signals by $X \in R_{N \times T}$, with its label $y \in Y$. Here, X is a 30-second-long EEG sample and $Y = \{W, N1, N2, N3, REM\}$. N is the number of channels and T denotes the time sampling points. Our classification task is then

redefined in this way: we firstly give an input sequence of $S_t^k = \{X_{t+0}, X_{t+1}, \dots, X_{t+k-1}\}$ which refers to an ordered sequence of k posterior segments of signals; then, we use CNN to learn the model $\hat{f}: (X_{N \times T})^k \rightarrow y$; finally, we get the sleep stage classification results [24].

Firstly, we present our CNN network architecture without fine-grained ($k=0$) segments. Next, we describe the optimized CNN with fine-grained segments ($k>0$). If $k=0$, the task is a standard sleep stage classification problem; otherwise, the fine-grained sleep stage classification is employed on the task.

3.1. CNN Architecture. According to the domain characteristics of sleep stage classification, CNN architecture usually contains five layers: an input layer for data inputting, a convolutional layer for automatic feature extraction, pooling layers for reducing network complexity, a full-connected layer, and an output layer for sleep stage classification [25, 26].

To overcome the difficulties in extracting features and transforming biological models into statistical models, we proposed a CNN architecture with seven layers. The CNN architecture is composed of one input layer (L_0), two convolutional layers (L_1 and L_2), two pooling layers (L_3 and L_4), one full-connected layer (L_5), and one output layer (L_6). Details are shown as Figure 1.

We now detail our proposed CNN architecture ($k=0$). As shown in Figure 1 and Table 2, the architecture starts with an input matrix followed by a reshape operation which is used to convert the data into the specified format. Then, the dimensions are permuted, as shown in layer 2 in Table 2. Next, two blocks of convolution and max-pooling are consecutively applied. Each block convolves its input signal with a certain number of estimated kernels of length 5 with stride 1, which is designed to extract a specified number of feature maps from the data. The outputs are then reduced

TABLE 2: Details of CNN architectural layers.

Layer	Layer type	Units	Units type	Kernel size	Stride	Output dimension
L_0 : Input						(N,T)
Reshape						(N,T,1)
L_1	Convolutional	20	Relu	(1,5)	(1,1)	(N,T,20)
L_2	Max-pooling			(1,10)	(1,10)	(N,T,20)
L_3	Convolutional	30	Relu	(1,5)	(1,1)	(N,T,600)
L_4	Max-pooling		Relu	(1,10)	(1,10)	(N,T,600)
L_5	Full-connected	500	Relu			500
L_6	<i>softmax</i>	5				5

with a max pooling layer (pool size of 10) with stride 10 at the same time. This layer is designed to reduce the size of feature maps while keeping the numbers in the maps unchanged. Then, the output of the two convolution blocks is put into a full-connected layer designed to aggregate the output features from the last layer and form global features for sleep stage classification. Finally, those features will be fed into a final layer with 5 neurons and a *softmax* is used to obtain a probability vector.

As mentioned above, the propagation and classification processes are the core of the method. For a better comprehension of our proposed CNN architecture, we will describe these two processes in Section 3.2 and Section 3.3, respectively.

3.2. Propagation. The propagation process contains forward propagation and back propagation. The forward-propagation process goes from L_1 to L_4 . Here, L_1 and L_3 are convolutional processes consisting of different numbers of output maps. The process starts from a ‘‘convolutional’’ operation between the input matrix from the previous layer and the convolution kernel matrix, and then it travels with the nonlinear conversion to get feature mappings. Thus, the output of the mapping m at level l can be described by the following:

$$x_m^l = f \left(\sum_{k=0}^{k=n} \sum_{i=0}^{i=SF*TS} x_{SF*TS*i,J+n}^{l-1} w_{m,i,j}^l + b_m^l \right), \quad (1)$$

where $x_{SF*TS*i}^{l-1}$ is the input of our CNN model, $0 \leq i < N$, and $0 \leq j < SF * TS$. N , SF , and TS stand for the number of EEG channels, signal frequency acquisition, and the time segments, respectively. In addition, n stands for the neurons in M filters, b_m^l is the set of weights at level l , and f is a *tanh* function.

L_2 and L_4 are the pooling layers, which can significantly reduce the amount of network computing while keeping the number of feature maps unchanged. Its mathematical theory is described as follows:

$$x_m^l = f \left(\beta_m^l D_s \left(x_m^{l-1} \right) + b_m^l \right), \quad (2)$$

where $D_s(\cdot)$ is the pooling function, β_m^l and b_m^l are the multiplicative bias and additive bias, respectively, and f is a mathematical function we choose. The back-propagation process going from L_4 to L_1 is achieved by using a gradient descent by minimizing the least mean square error, which can reduce the network’s propagation errors effectively.

3.3. Classification. The full-connected layer, as the name suggests, is fully connected with the front layer and the later output layer. It maps the feature maps produced by the convolutional layer to a fixed-length eigenvector, that is to say it gives a probability on each sleep stage. Its function is described as follows:

$$o_m^l = f \left(\sum_{i=1}^n x_m^{l-1} w_m^l + b_m^l \right), \quad (3)$$

where n is the neuron number of the previous layer and w_m^l is the connection weight with the previous layer.

Finally, the output layer uses a *softmax* classifier with 5 units to classify the sleep stages based on the probability given by the full-connected layer. The number of units stands for the sleep stage number that can be classified.

3.4. CNN with Fine-Grained Segments. In this subsection, we describe the CNN architecture with fine-grained segments. Although CNN already has the ability to classify the sleep stage data on its own, it also has some disadvantages. As mentioned above, sleep stage classification is based on the features extracted in each segment, which means the size of the segment is the key to it. However, as we can see from Figure 2, there are rapid changes in every second. Thus, we can automatically ratiocinate that the butte sharp variation in the neighboring two sleep stage segments may affect the classification results to some extent. More specifically, if we set a short time segment, CNN may not be able to extract sufficient features, but if we set a relatively longer time segment, there might be some bully feature maps misleading the classification results. Therefore, for a better sleep stage performance, we imported the fine-grained segments into our CNN architecture in order to find the best time segment. Its basic principle is shown in Figure 3.

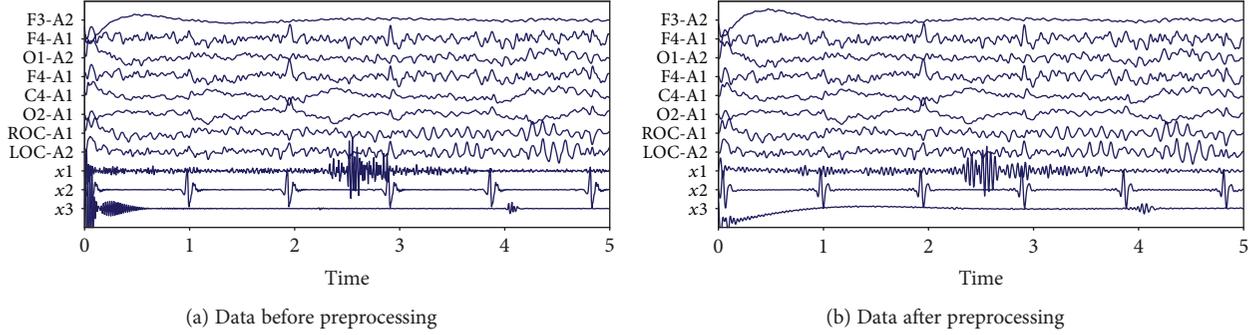


FIGURE 2: Data before and after preprocessing.

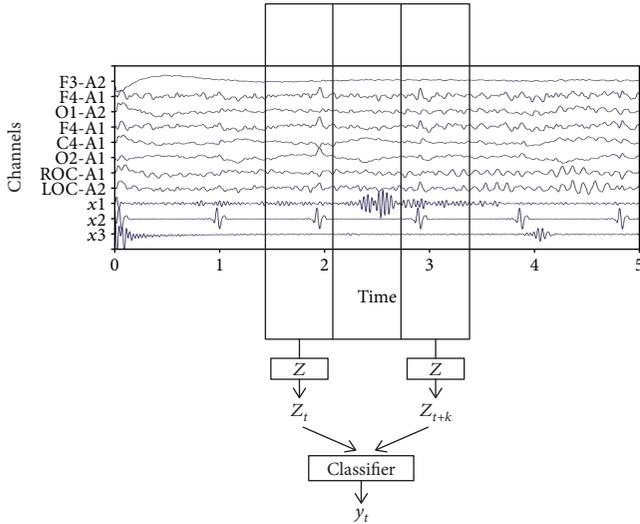


FIGURE 3: Time distribution of CNN with fine-grained segments.

We first describe the theory of the fine-grained method in multiscale entropy. As is shown in Figure 4, suppose there is a given time series x_1, x_2, \dots, x_N , we can construct a time series with fine-grained scale s : $\{Z^{(s)}\}$ which is a sequence of k posterior samples. We first choose a time window of length s , taking $s=3$ for an example, and then the first time series will be $\{x_1, x_2, x_3\}$. Next, the window of length s will be moved one by one until it traverses all the EEG data. Finally, we will get a new time series at scale s .

Based on the above theory, we proposed our CNN with fine-grained segments to reconstruct the time series at each scale, which is designed to resolve the problem of inaccurate analysis owing to the drastic reduction of the time series length.

In our proposed method, we firstly choose a sequence of $S_t^k = \{X_{t+0}, X_{t+1}, \dots, X_{t+k-1}\}$ as a predefined time segment using fine-grained methods mentioned above. Here, X_t stands for the multivariate data over 30 s, k is the total amount of X_t , and t is the serial number in EEG samples. Next, we use feature extractor Z on each sample in time series S_t^k to account for the statistical properties of the signals after fine-grained segments, which will get features expressed by Z_{t+k-1} . Then, we aggregate the features and get a feature series described by $[Z_t, Z_{t+k-1}]$, which is a vector of size $D(k)$. Finally, the obtained vector is fed into

a classifier to predict the label y_t associated with the sample to classify X_t . The method is illustrated in Figure 3.

3.5. Training. After designing all the parameters in the network, the next task is training our proposed CNN neural network. Our training is performed by minimizing the loss with a stochastic gradient using minibatches of data [22].

For better classification performance, we set several training parameters for the learning rate: μ is the momentum update for faster coverage, ϵ stands for the initialized learning rate, and wd is the decaying rate to avoid overshooting at the layer stage of GDS. In our training work, we set $\mu=0.9$, $\epsilon=0.01$, and $wd=0.005$. Weights were initialized with a normal distribution with mean $\mu=0$ and standard deviation $\sigma=0$.

The training of the time distribution was done in two steps. We firstly trained the multivariate network, especially its feature extractor, without employing the fine-grained method ($k=0$). The trained model was then used to set the weights of the feature extractor distributed in time. Then, we froze the weights of the feature extractor distributed in time and trained the final *softmax* classifier with aggregated features.

4. Experiments and Discussions

4.1. Datasets and Preprocessing. To show the robustness of our proposed CNN architecture, we tested our proposed CNN architecture on a public sleep dataset called ISRUC-Sleep.

The datasets consist of full overnight recordings of 116 subjects with three health statuses (healthy, sick, and under treatment). Each recording contains six EEG channels (i.e., C3_A2, C4_A1, F3_A2, O1-A2, O2-A1, and F4-A1), two EOG channels (i.e., LOC_A2 and ROC-A1), and three EMG channels (i.e., X1, X2, and X3) as well as an annotation file with detailed events. The recording rate is 200 Hz. Each 30 s epoch is divided into one of the five sleep stages mentioned above. All night recordings are labelled by two experts according to AASM rules. More details can be found in [4]. In each experiment, we obtain results using 10-fold cross-validation. Specifically, in each fold we use the recordings of 10 subjects for testing and the recordings of the remaining 106 subjects for training and validation. For each fold, we use the recordings from 10 randomly selected subjects for

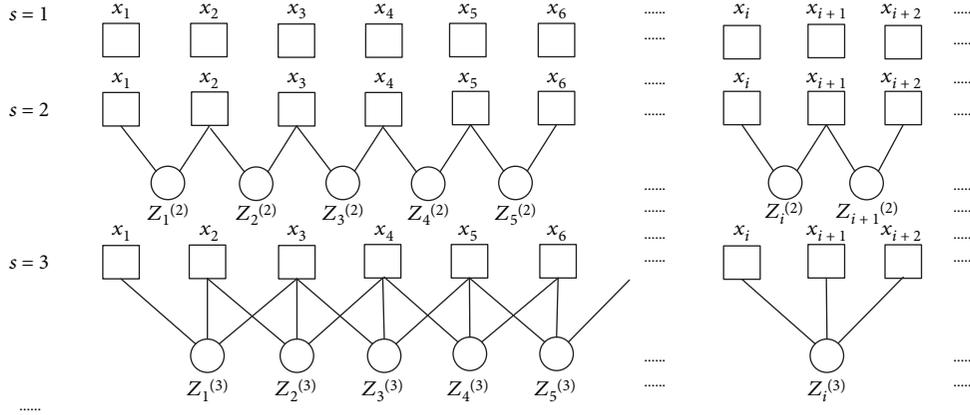


FIGURE 4: The principle of the fine-grained method.

validation and the recordings from the remaining 86 subjects for training. Each experiment was carried out 5 times, while the records for training, validation, and testing were shuffled to reduce the variance in metric evaluation. To learn discrimination between underrepresented classes (such as W and N1 stages), we proposed to use the balanced accuracy of a subject to train our CNN architecture in this paper. As we have 5 classes, it means that each batch has approximately 20% belonging to each class.

As we all know, the EEG recordings often have interference from a variety of factors, such as baseline drift, ocular motions, and white noise. Those artifacts do have a negative impact on the final sleep stage classification, so we need to do some preprocessing work. A 10th-order II_R Butterworth band filter is applied to the EEG signals in order to remove the noise and artefacts from EEG signals. In addition, we also use a 12th-order stop-band Butterworth Notch filter to decrease the interfaces. The preprocessing results are shown in Figure 2. We both plot a sleep data of 5 seconds with 11 channels. It can be clearly seen that our preprocessing work really removes the artefacts while keeping the basic characteristics of the signal unchanged at the same time, which will have a good effect on the sleep stage classification task.

4.2. Performance Metrics. To evaluate the performance of our CNN architecture, we adopt accuracy, specificity, precision, and sensitivity (SE) as defined in (5), (6), (7), and (8) to evaluate our models. We also use confusion matrix and loss to prove the good performance of our proposed CNN architecture.

The accuracy is obtained by dividing the correctly classified samples by all samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \quad (5)$$

Specificity is the correctly predicted proportion of all negative samples, which measures the ability of the classifier to recognize negative samples.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (6)$$

Sensitivity reflects the correctly predicted proportion of all positive samples, which measures the ability of the classifier to recognize positive samples.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (7)$$

Precision reflects the correctly predicted proportions of all position samples.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (8)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

$$\text{Loss} = -\frac{1}{M} * \sum_{i=1}^M y_i \log f(x_i), \quad (9)$$

where M is the number of training samples and x_i and y_i are the input and the expected output, respectively. $f(x_i)$ denotes the real output.

4.3. Experiments and Comparisons. In this section, we explicate the details and the outcomes of our experiments and expound some of the significance of the results. The performance of the proposed method is compared to several state-of-the-art papers for sleep stage classification published recently. To evaluate the performance of the proposed CNN architecture, we design the following three experiments in this section:

- (1) Sleep stage classification with our CNN architecture ($k=0$)
- (2) Adding channels gradually to evaluate the performance of our CNN architecture on sleep stage classification
- (3) Sleep stage classification using CNN combined with fine-grained segments

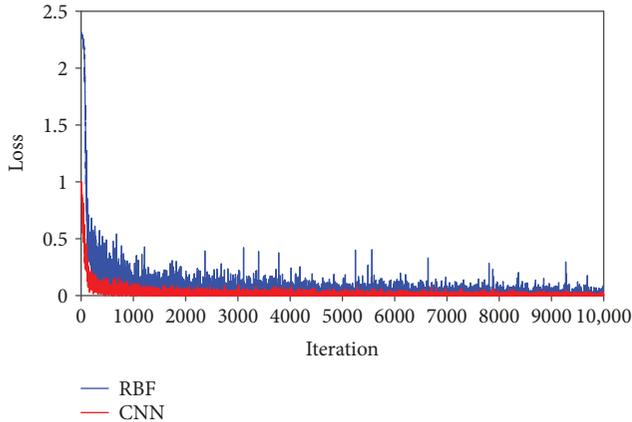


FIGURE 5: The loss comparison between CNN and RBF.

4.3.1. Experiment 1: The Performance of Our CNN Architecture. In this experiment, we aim to verify the performance of the CNN architecture ($k = 0$). To prove the performance, we compare it with our previous work [14], which used RBF for sleep stage classification. As introduced in Section 3, the input channels in our experiments must be exactly the same as the compared work [14], so $N = 5$. Meanwhile, we also use the same kind of channels (C3-A2, C4-A1, O1-A2, LOC-A2, and X1). In addition, all EEG recording signal rates are 200 Hz and each 30 s signal is regarded as a sleep stage, so the time sampling points is $T = 200 \times 30 = 6000$.

Finally, concerning the proposed approach, the specificity, precision, sensitivity, accuracy, and loss were used as performance metrics for each predictive model. The detailed results are shown in Figure 5 and in Tables 3 and 4.

Figure 5 shows the loss comparison between our proposed CNN architecture and RBF [14]. We can clearly see that our proposed CNN architecture has a lower and more stable loss than RBF. More specifically, our proposed CNN is basically stable in regard to the 5000th iteration while the RBF still has dramatic fluctuations until the 10,000th iteration. That is, our proposed CNN architecture has a faster convergence rate than RBF, which means it is more efficient than RBF. Meanwhile, our proposed CNN architecture has a loss which is smaller than 1 while RBF has a much bigger loss at the beginning. Above all, our proposed CNN architecture is much more efficient than RBF.

Tables 3 and 4 list the “best and worst values of every sleep stage” of CNN and RBF using four measurement standards including precision, sensitivity, specificity, and accuracy, respectively. Comparing these two tables, we can see that the best values of CNN in the first three measurements including precision, sensitivity, and specificity are almost same as the ones of RBF, but in regard to the worst value, things has changed. CNN has a better performance in situations having low sensitivity as compared to RBF, which is to say our proposed RBF performs better in recognizing sleep stages. When it comes to specificity, CNN and RBF have a pretty good performance. More specifically, both CNN and RBF can perfectly recognize all S2 and SS samples. Our proposed CNN architecture has an accuracy of 83.25%,

which is marginally higher than the RBF. Overall, our proposed CNN has better performance in terms of precision, sensitivity, specificity, and accuracy.

Above all, we can say that CNN has a good performance in sleep stage classification without handcrafted features.

4.3.2. Experiment 2: The Influence of Channel Number. As shown in the first experiment, although our proposed CNN architecture performs well in the sleep stage classification without the handcrafted features, it only achieves an average accuracy of 83.25%, which is impressive but needs to be improved. There is a way to improve its performance inspired by [27], which used 28 channels to compete the classification of sport imagination and achieved an accuracy of 89%. Thus, we can reasonably postulate that by adding the number of channels gradually, we can also find the best channel’s number for sleep stage classification of our proposed CNN architecture.

In this experiment, we will add two channels each time until the number of channels is 11. The channel number used at first is 5, so it will be 5, 7, 9, and 11 in this experiment. Concerning the proposed approach, the confusion matrices (C.M.) and these four performance metrics including precision, specificity, sensitivity, and accuracy are all used as performance metrics. Finally, we compared the experiment results with [22] and [23] to prove the excellent performance of our proposed method. The results are shown in Figures 6 and 7 and Table 5.

Figure 6 shows the confusion matrix comparison between our proposed CNN architecture under the four kinds of channel number mentioned above and those of the two papers that use multichannels [22, 23]. Comparing these six confusion matrices, we find that our proposed method achieves equal or higher diagonal coefficients in its confusion matrix under the four kinds of channel number compared to the other two methods. More specifically, our CNN using five channels can classify 78% of the epochs of N1, which is much higher than the two compared methods. The more channels we use, the higher accuracy of stage N1. In regard to stage N3 and REM, the comparison results are the same as stage N1. The fourth confusion matrix of our CNN using the same channels as [23] has higher diagonal coefficients in its confusion matrix, which means it has the best performance in classifying every sleep stage. In addition, our CNN under these four kinds of channel number perform better in every sleep stage compared to the results in [22], and it also performs better in recognizing S1, S2, and REM than [23]. Moreover, our proposed CNN architecture under 11 channels yields the highest diagonal coefficient in its confusion matrix. Figure 7 shows the accuracy comparison between another two methods with our CNN under four kinds of channel number. It shows that our CNN under 5 channels has a better accuracy than [22] and [23]. What’s more, our method under channel numbers of 7, 9, and 11 has a much higher performance in sleep stage classification than any of the other two methods. For a better comparison, we further research the CNN with 11 channels and find its best and worst performance in terms of precision, sensitivity, specificity, and accuracy. When compared with Tables 3 and 4,

TABLE 3: Evaluation results of the original CNN with 5 channels.

Networks	Stages	Precision		Sensitivity		Specificity		Accuracy	
		Best	Worst	Best	Worst	Best	Worst	Best	Worst
CNN	W	0.9510	0.7501	0.9616	0.5	0.9964	0.7997	0.9009	0.8078
	N1	0.8168	0.4284	0.8219	0.71	0.9807	0.7518	0.7928	0.6917
	N2	0.9110	0.6098	0.9033	0.3182	1	0.6974	0.9034	0.7560
	N3	0.9636	0.6028	0.9635	0.5657	1	0.8504	0.9724	0.7588
	REM	0.9229	0.593	0.8351	0.5258	0.9972	0.8067	0.9329	0.8067
	Average		0.8937		0.7669		0.9192		0.8325

TABLE 4: Evaluation results of our previous work using RBF.

Networks	Stages	Precision		Sensitivity		Specificity		Accuracy	
		Best	Worst	Best	Worst	Best	Worst	Best	Worst
RBF	W	0.9901	0.7621	0.9516	0.100	0.9486	0.8842	0.8867	0.7588
	N1	0.8943	0.7712	0.7988	0.5100	0.7845	0.5892	0.7226	0.5429
	N2	0.9924	0.8478	0.8578	0.2210	1	0.7356	0.9021	0.7788
	N3	0.9788	0.8034	0.90 88	0.5443	1	0.8932	0.9460	0.6821
	REM	0.9455	0.8732	0.8088	0.5087	1	0.7921	0.9045	0.7813
	Average		0.8793		0.6056		0.8592		0.817

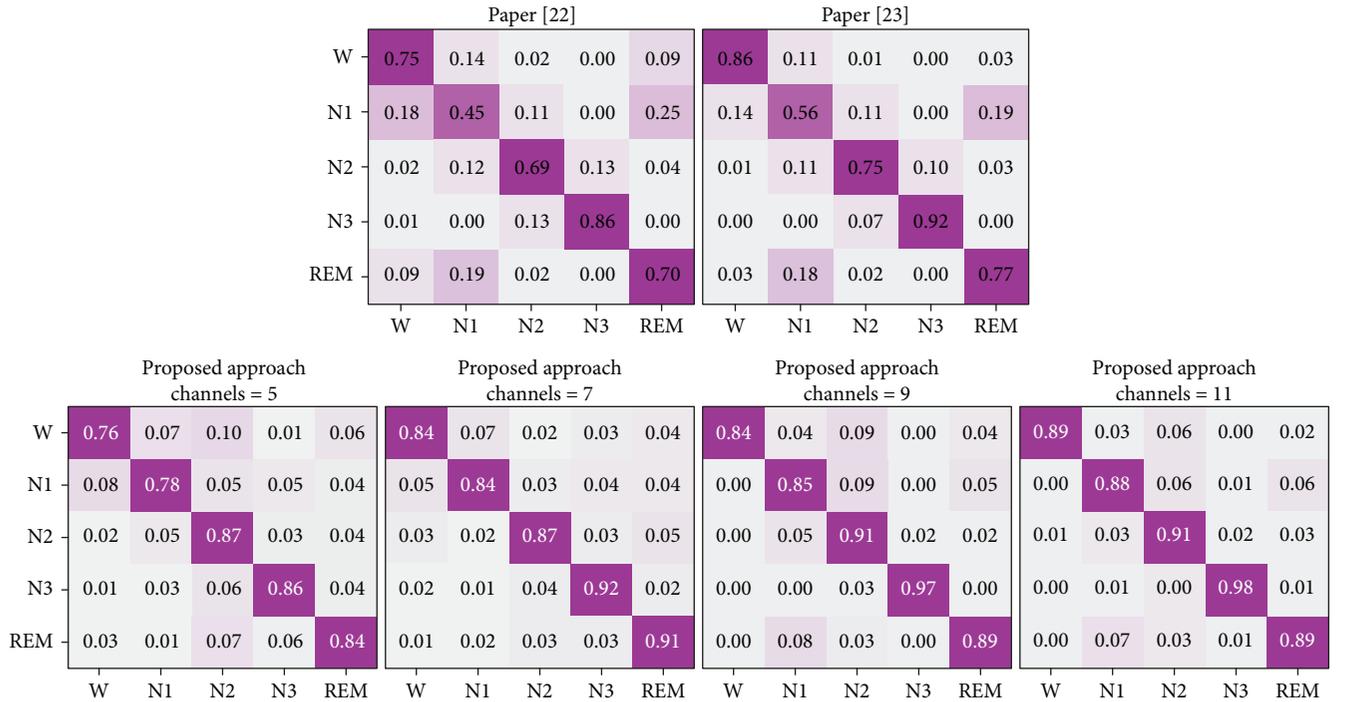


FIGURE 6: Confusion matrix comparison between the other two methods and the original CNN under four kinds of channel number.

Table 5 shows that both the worst precision performance and best precision performance of our CNN with 11 channels are much better than the original CNN and RBF. When it comes to sensitivity and accuracy, the situations are the same with the precision. It finally achieves the highest accuracy of 0.90, 0.80, 0.93, 0.95, and 0.92 of each sleep stage, respectively. Thus, in general, our proposed CNN model has a better performance in sleep stage classification.

We now compare the confusion matrices and the accuracy of our CNN model. The last four confusion matrices in Figure 6 show that the more channels we use, the higher the diagonal coefficient will be in its confusion matrix. According to the last four columns in Figure 7, we also find that the accuracy of the five sleep stages has also been improved by different degrees when increasing channels. More specifically, all sleep stages are improved by 7% on

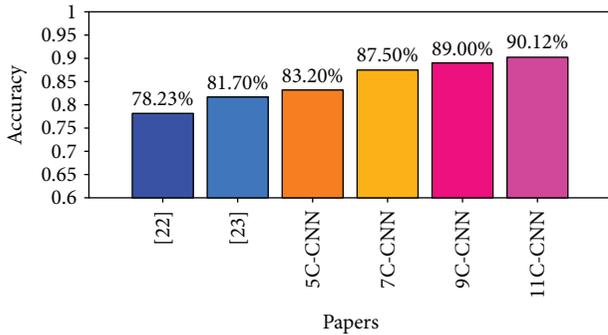


FIGURE 7: Accuracy comparison between two other methods and the original CNN under four kinds of channel number.

average. This consequence is much more impressive than in [22] and [23] which also added the recording sensors. Besides, the two channels added first significantly improved the accuracy of sleep stages W, N1, N3, and REM. The next two channels added yield equal or higher diagonal coefficients in the confusion matrix than our proposed method with 7 channels on N1, N2, and N3. Meanwhile, the improvement in stage S3 is the most obvious, which is 12%. When using 5 channels, our CNN architecture can classify approximately 83% of the sleep data perfectly. Then, we add 2 channels, and its accuracy increases by 4%. When the next two channels are added, our CNN architecture achieves an accuracy of 89.0% or so, and the performance of our proposed CNN architecture achieves an accuracy validated at a confidence level of 90.11% when we use 11 channels. That is, the increment in performance is greater than 0.1 when the number of channels goes from 5 to 11. However, we also find that the increase in accuracy each time is uneven and it decreased progressively. According to our research, the unevenness is because when we add two channels each time, the species is not the same every time, which means that the channels carry different quantities and influential features having varying impacts on the final sleep stage classification performance.

Here are some things we need to explain. Firstly, Figure 6 shows that stage S1 and stage W have the worst performance, with average sensitivities of 83.25% and 83.75%, while other stages are 88.75%–91.75%. According to our study, the reason stage W has a poor classification performance is that there are some dramatic fluctuations during the process of falling asleep as we can see from Figure 2(a). Those fluctuations are affected by external influences and cannot be completely removed as shown in Figure 2(b), and thus they directly affect the final classification results. The poor classification performance of S1 is also due to the transition between wake and sleep, which means the feature signals may not last longer than 30 s, so the experts cannot make a decision easily. Additionally, the sensitivity of S2 to SS is larger than 90%. When sleepers fall into stage S2, it becomes more difficult for them to gradually awaken and be more slightly responsive to the environment. Therefore, the sensitivity of S2 to SS has a better performance. When a sleeper goes into the REM stage, they tend to awake and are

accompanied by some other waves, so some of those stages are classified into the wake stage. Secondly, we can see from Figure 7 that the increment gets smaller as the number of channels increases. The first two added channels increase the accuracy by 3%, the next one improves it by 2.17%, and the final one raises it by 1.7%. Therefore, we can guess that if we continue to increase the number of channels, there will be a cutoff value beyond which the accuracy of the classification no longer increases. Thus, the increasing number of channels does optimize the classification stage, but it also has a limit.

Above all, our proposed methods will exhibit a better sleep stage classification performance when using more channels, but there will be a limit to the improvement as the number of channels keeps increasing.

4.3.3. Experiment 3: The Influence of Size k on CNN Model.

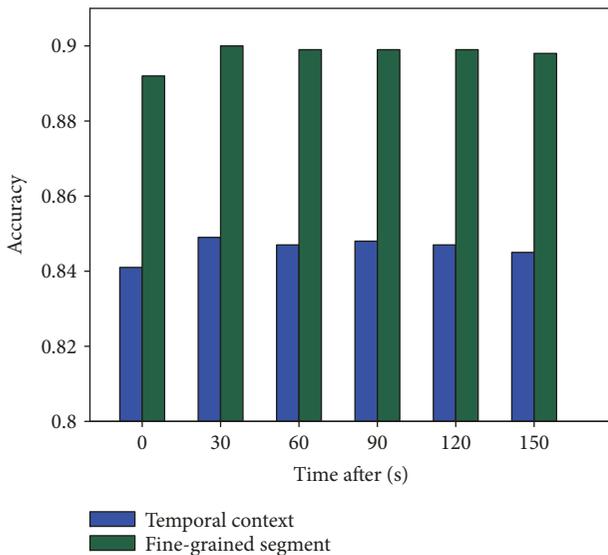
In this experiment, we investigate how the size of fine-grained segments influences the performance of our CNN architecture. First, we define the size of k . Next, we use the constructed series including the current sample and $k-1$ posterior segments to find the best k for sleep stage classification, which is inspired by [23]. Finally, we compare the highest accuracy and average sensitivity between the best k of our proposed method with that of several state-of-the-art methods. We demonstrate that by considering the data from the fine-grained segment samples, classification performance is increased, especially if the fine-grained segments are limited.

As shown in experiment 2, we achieved the best sleep stage classification performance when using 11 channels which is exactly the same as [23], which accomplishes its best temporal sleep stage classification when using 6 EEG channels, 2 EOG channels, and 3 EMG channels. For a better comparison, we also considered 11 channels in this experiment. In this experiment, we varied the size of the temporal input sequence S^k from $k=0$ to $k=5$, which is the same as [23], that is, we use a data of 150 s following each sample to classify. Finally, concerning the proposed approach, we use accuracy and confusion matrix as performance metrics for each predictive model. The results are reported in Figures 8 and 9 and Table 6.

Figure 8 shows the accuracy comparison between our proposed method which uses data including the current and succeeding segments and the method in [23] which uses the data containing the preceding and succeeding samples. Overall, our proposed method achieves a similar influence on sleep stage classification compared with [23] while using much less data. More specifically, our proposed CNN trained on 11 channels with succeeding 30 s of signals achieves the best performance in sleep stage classification. So did the method of [23], which was trained on 11 channels with -30 s/ $+30$ s of context. Our proposed method achieves an average accuracy of 92.20%. Figure 8 also shows that considering a few successive samples does enhance the classification performance, while considering too many succeeding samples decreases the classification performance. Figure 9 gives a confusion matrix comparison between our proposed method and the method in [23] both

TABLE 5: Evaluation results of CNN with 11 channels.

Networks	Stages	Precision		Sensitivity		Specificity		Accuracy	
		Best	Worst	Best	Worst	Best	Worst	Best	Worst
Our CNN with 11 channels	W	0.9000	0.8051	0.9800	0.7354	0.9800	0.8700	0.9220	0.8678
	N1	0.8429	0.6234	0.80	0.65	0.9612	0.7212	0.8133	0.7022
	N2	0.9422	0.6087	0.934	0.5079	1	0.8055	0.9300	0.8321
	N3	0.9213	0.7000	0.9650	0.6454	0.9870	0.8366	0.9512	0.8965
	REM	0.9534	0.7300	0.8878	0.7699	0.9877	0.8311	0.9243	0.8612
	Average		0.9200		0.9053		0.9221		0.9012

FIGURE 8: Accuracy comparison between Chambon et al. [23] and our CNN under different sizes of k .

under their highest accuracy. It shows that our proposed method yields a higher diagonal coefficient in its confusion matrix than in [23].

To further prove the efficiency of our method, we do a deeper research on the CNN with posterior 30 s and compare it with other methods. Table 6 gives the best and worst situations of this comparison in terms of precision, sensitivity, specificity, and accuracy. We find that the proposed method gives a wonderful performance in every sleep stage in terms of those four evaluation indices. In regard to accuracy, it achieves the best accuracies of 0.97, 0.94, 0.96, 0.98, and 0.93 for each sleep stage, respectively, which is much more impressive compared with Tables 4 and 5 [14]. More specifically, it increases the average specificity of the original CNN by 2%, which means it has a better performance in classifying negative samples. The situation of sensitivity is the same with the specificity. It also achieves a comparable performance in precision compared with the original CNN. When compared with Table 4 [14], our proposed method achieves much higher performance in both the best and worst situations in terms of these four performance metrics.

Above all, our proposed method yields superior performance in sleep stage classification.

4.4. Discussions. In this section, we will discuss the architectural characteristics of our proposed method and compare it in perspective with some state-of-the-art methods.

It has several advantages. Firstly, the computation performance of our CNN architecture is quite small thanks to specific architectural choices. It can be evaluated by the computation number of the parameters and convergence speed as well as the dimension of the convolution filters and pooling regions. Compared with [23] and [28], the whole network of our proposed method does not exhibit more than 10^4 parameters when considering the CNN architecture without the following sleep stage data, and not more than 10^5 parameters when using the fine-grained segments. This significant simplicity is mainly because of our decision to use small convolution filters and large pooling regions. This is quite simple and compact compared to the recent approach in [23] which has up to $\sim 14 \cdot 10^7$ parameters and in [28] which has up to $\sim 6 \cdot 10^5$ parameters for the feature extractor and $2 \cdot 10^7$ parameters for the sequence learning part using BiLSTM. Besides, the loss in the training process is basically stabilized at the 1500th time and its average loss is less than 1, which is pretty small. Thus, the CNN architecture we proposed for multichannel EEG signals has a fast convergence and high efficiency. We also need to notice the size of filters. Some studies use smaller convolutional filters, such as 2, 3, 5, or 7 [29], but they must use a larger number of feature maps from 64 to 512 [29], which increases the computation. We find the best filter size for our CNN after several tests.

Our CNN method turns out to be agnostic. It can deal with different kinds of signals including EEG, EOG, and EMG.

Our CNN architecture also has pretty strong compatibility and flexibility. It can learn naturally from the fine-grained segments because it only relies on the aggregation of temporal features and a *softmax* classifier. Our proposed architecture is designed to only extract features from 30 s, so when we add the posterior samples, we do not need to change our CNN architecture. Thanks to this architecture, we can easily evaluate the influence of our proposed methods (fine-grained segments). However, the methods described in [30] and [31] are designed to address an input matrix of 150 s, so it is complicated when we need to dispose a signal of 120 s.

Finally, the proposed method has a great potential on real applications. For real-world applications, our proposed CNN

	Chambon et al., 2017 temporal context 30 s					Fine-grained segments succeeding 30 s				
W	0.83	0.11	0.01	0.00	0.05	0.90	0.04	0.04	0.01	0.01
N1	0.12	0.55	0.10	0.00	0.23	0.01	0.86	0.05	0.00	0.07
N2	0.01	0.09	0.76	0.10	0.04	0.01	0.03	0.93	0.02	0.02
N3	0.02	0.00	0.07	0.90	0.01	0.00	0.01	0.01	0.97	0.00
REM	0.02	0.11	0.01	0.00	0.85	0.00	0.07	0.03	0.00	0.90
	W	N1	N2	N3	REM	W	N1	N2	N3	REM

FIGURE 9: Confusion matrix comparison between Chambon et. al. [23] and fine-grained segments succeeding 30 s.

TABLE 6: Evaluation results of our proposed method succeeding 30 s.

Networks	Stages	Precision		Sensitivity		Specificity		Accuracy	
		Best	Worst	Best	Worst	Best	Worst	Best	Worst
Proposed method	W	0.9200	0.8351	0.9716	0.7612	0.9876	0.8700	0.9742	0.8078
	N1	0.8790	0.6384	0.8674	0.69	0.9744	0.7812	0.9433	0.8422
	N2	0.9417	0.6054	0.9328	0.5079	1	0.8054	0.9645	0.8023
	N3	0.9512	0.7028	0.9766	0.6954	1	0.8912	0.9812	0.9065
	REM	0.9347	0.7700	0.8906	0.7822	0.9972	0.8433	0.9365	0.7612
	Average		0.9274		0.9253		0.9477		0.9220

architecture can be adopted for other physiological signal classification performance. Future applications can include attempting to combine a wireless and wearable EEG device with the proposed computerized sleep stage classification. In this way, our proposed method can contribute to the feasibility of a sleep quality evaluation device, long-term sleep monitoring, and home-based daily care. In conclusion, we can anticipate that the proposed automatic sleep stage classification method based on CNN and fine-grained segments will enable personal sleep monitoring, assist clinicians in analyzing sleep data, and help diagnose sleep disorders.

But it also has disadvantages. It does not consider the impact of the signal species. On our second experiment, the increment is different when adding two different kinds of channels each time. So it is hard to evaluate the influence of signal species on sleep stage classification.

5. Conclusions and Future Work

In this paper, we present a CNN architecture combined with fine-grained segments for automatic sleep stage classification from multichannel EEG time series. This architecture has two pairs of convolutional layers and max-pooling layers with their own kernel size and stride, which are designed based on the unique features of our multichannel EEG signals. It also has one full-connected layer and one output layer for classification. To evaluate the performance of our proposed method, we conducted experiments to check three aspects. First, we verified the superior performance our CNN architecture by comparing it with our previous work described in [14]. It finally has an accuracy validated at a

confidence of 83.25%, which means it can successfully perform sleep stage classification, but it is marginally worse in terms of accuracy. Secondly, to achieve a better classification performance, we gradually increased the number of channels to find its influence on our proposed CNN architecture. The results demonstrated that the larger the number of channels we used, the better is the performance exhibited by our CNN model. Finally, we verified the performance of the CNN architecture with fine-grained segments. The proposed method showed excellent performance with an average accuracy of 92.20%, which is superior to the other state-of-the-art approaches. In short, the proposed method is fast, robust, and fully automatic. Above all, our proposed CNN architecture is very successful in sleep stage classification.

When adding different kinds of channels to our CNN architecture, the accuracy is raised each time by a different degree. Thus, in the future, we will elaborate on the influence of channel species on our proposed CNN architecture. We continue to find the perfect channel types for our CNN architecture.

Data Availability

Data used in the preparation of this article were obtained from the ISRUC-SLEEP dataset (http://sleeptight.isr.uc.pt/ISRUC_Sleep/). The investigators within the ISRUC-SLEEP contributed to the design and implementation of ISRUC-SLEEP and/or provided data, but did not participate in the analysis or in writing this report. The dataset is described in the following article: Khalighi Sirvan, Teresa Sousa, Jose Moutinho Santos, and Urbano Nunes.

ISRUC-Sleep: a comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*, 124 (2016):180–192. It can be downloaded at https://www.researchgate.net/publication/283734463_ISRUCSleep_A_comprehensive_public_dataset_for_sleep_researchers.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

We are grateful for the support of the National Natural Science Foundation of China (61373149, 61672329, 61572295), the National Key R&D Program (2017YFB1400102, 2016YFB1000602), and SDNSFC (no. ZR2017ZB0420).

References

- [1] D. Y. Kang, P. N. DeYoung, A. Malhotra, R. L. Owens, and T. P. Coleman, “A state space and density estimation framework for sleep staging in obstructive sleep apnea,” *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 6, pp. 1201–1212, 2018.
- [2] F. Mendonça, S. S. Mostafa, F. Morgado-Dias, J. L. Navarro-Mesa, G. Juliá-Serdá, and A. G. Ravelo-García, “A portable wireless device based on oximetry for sleep apnea detection,” *Computing*, pp. 1–17, 2018.
- [3] Y.-Y. Cheung, B.-C. Tai, G. Loo et al., “Screening for obstructive sleep apnea in the assessment of coronary risk,” *The American Journal of Cardiology*, vol. 119, no. 7, pp. 996–1002, 2017.
- [4] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes, “ISRUC-sleep: a comprehensive public dataset for sleep researchers,” *Computer Methods and Programs in Biomedicine*, vol. 124, pp. 180–192, 2016.
- [5] Z. R. Zamir, N. Sukhorukova, H. Amiel et al., “Optimization-based features extraction for K-complex detection,” *Anziam Journal*, vol. 55, pp. C384–C398, 2013.
- [6] L. Wei, Y. Lin, J. Wang, and Y. Ma, “Time-frequency convolutional neural network for automatic sleep stage classification based on single-channel EEG,” in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 88–95, Boston, MA, USA, 2018.
- [7] D. Jha, J.-I. Kim, and G.-R. Kwon, “Diagnosis of Alzheimer’s disease using dual-tree complex wavelet transform, PCA, and feed-forward neural network,” *Journal of Healthcare Engineering*, vol. 2017, Article ID 9060124, 13 pages, 2017.
- [8] D. Wang, D. Ren, K. Li et al., “Epileptic seizure detection in long-term EEG recordings by using wavelet-based directed transfer function,” *IEEE Transactions on Biomedical Engineering*, p. 1, 2018.
- [9] T. Lajnef, S. Chaibi, P. Ruby et al., “Learning machines and sleeping brains: automatic sleep stage classification using decision-tree multi-class support vector machines,” *Journal of Neuroscience Methods*, vol. 250, pp. 94–105, 2015.
- [10] R. Zhao, W. Ouyang, H. Li, and X. Wang, “Saliency detection by multi-context deep learning,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1265–1274, Boston, MA, USA, 2015.
- [11] E. Y. Huan, G. H. Wen, S. J. Zhang et al., “Deep convolutional neural networks for classifying body constitution based on face image,” *Computational and Mathematical Methods in Medicine*, vol. 2017, Article ID 9846707, 9 pages, 2017.
- [12] H. Spitzer, K. Amunts, S. Harmeling, and T. Dickscheid, “Parcellation of visual cortex on high-resolution histological brain sections using convolutional neural networks,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. 920–923, Melbourne, VIC, Australia, 2017.
- [13] S. Sakhavi, C. Guan, and S. Yan, “Parallel convolutional-linear neural network for motor imagery classification,” in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2736–2740, Nice, France, 2015.
- [14] Z. Cui and X. Zheng, “A sleep stage classification algorithm based on radial basis function networks,” in *Proceedings of the 2nd International Conference on Crowd Science and Engineering - ICCSE’17*, pp. 56–60, Beijing, China, 2017.
- [15] R. Boostani, F. Karimzadeh, and M. Nami, “A comparative review on sleep stage classification methods in patients and healthy individuals,” *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 77–91, 2017.
- [16] U. R. Acharya, E. C.-P. Chua, K. C. Chua, L. C. Min, and T. Tamura, “Analysis and automatic identification of sleep stages using higher order spectra,” *International Journal of Neural Systems*, vol. 20, no. 6, pp. 509–521, 2010.
- [17] R. Sharma, R. B. Pachori, and A. Upadhyay, “Automatic sleep stages classification based on iterative filtering of electroencephalogram signals,” *Neural Computing and Applications*, vol. 28, no. 10, pp. 2959–2978, 2017.
- [18] S. F. Liang, C. E. Kuo, Y. H. Hu, Y. H. Pan, and Y. H. Wang, “Automatic stage scoring of single-channel sleep EEG by using multiscale entropy and autoregressive models,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1649–1657, 2012.
- [19] U. R. Acharya, S. Bhat, O. Faust et al., “Nonlinear dynamics measures for automated EEG-based sleep stage detection,” *European Neurology*, vol. 74, no. 5-6, pp. 268–287, 2015.
- [20] M. Sharma, D. Goyal, P. V. Achuth, and U. R. Acharya, “An accurate sleep stages classification system using a new class of optimally time-frequency localized three-band wavelet filter bank,” *Computers in Biology and Medicine*, vol. 98, pp. 58–75, 2018.
- [21] G. Zhu, Y. Li, and P. P. Wen, “Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1813–1821, 2014.
- [22] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, “Automatic sleep stage scoring with single-channel EEG using convolutional neural networks,” 2016, <http://arxiv.org/abs/1610.01683>.
- [23] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, “A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 758–769, 2018.

- [24] A. Drouin-Picaro and T. H. Falk, "Using deep neural networks for natural saccade classification from electroencephalograms," in *2016 IEEE EMBS International Student Conference (ISC)*, pp. 1–4, Ottawa, ON, Canada, 2016.
- [25] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2620–2623, Milan, Italy, 2015.
- [26] F. C. Morabito, M. Campolo, C. Ieracitano et al., "Deep convolutional neural networks for classification of mild cognitive impaired and Alzheimer's disease patients from scalp EEG recordings," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pp. 1–6, Bologna, Italy, 2016.
- [27] Z. Tang, K. Zhang, C. Li, and S. Zhang, "Motor imagery classification based on deep convolutional neural network and Its application in exoskeleton controlled by EEG," *Chinese Journal of Computers*, vol. 2016, no. 254–4164, pp. 1–15, 2016.
- [28] Z. Tang, K. Zhang, C. Li, and S. Zhang, "Motion classification based on deep convolutional neural network and its application in brain control exoskeleton," *Chinese Journal of Computers*, vol. 40, no. 6, pp. 1367–1378, 2017.
- [29] X. Shuyuan, W. Bei, Z. Jian, Z. Qunfeng, Z. Junzhong, and M. Nakamura, "An improved K-means clustering algorithm for sleep stages classification," in *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1222–1227, Hangzhou, China, 2015.
- [30] S. Biswal, J. Kulas, H. Sun et al., "SLEEPNET: automated sleep staging system via deep learning," 2017, <http://arxiv.org/abs/1707.08262>.
- [31] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J. F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel EEG," *Biomedical Signal Processing and Control*, vol. 42, pp. 107–114, 2018.

