EEG-Based Sleep Stage Classification via Neural Architecture Search

Gangwei Kong, Chang Li, *Member, IEEE*, Hu Peng, Zhihui Han, and Heyuan Qiao

***Abstract*— With the improvement of quality of life, people are more and more concerned about the quality of sleep. The electroencephalogram (EEG)-based sleep stage classification is a good guide for sleep quality and sleep disorders. At this stage, most automatic staging neural networks are designed by human experts, and this process is time-consuming and laborious. In this paper, we propose a novel neural architecture search (NAS) framework based on bilevel optimization approximation for EEG-based sleep stage classification. The proposed NAS architecture mainly performs the architectural search through a bilevel optimization approximation, and the model is optimized by search space approximation and search space regularization with parameters shared among cells. Finally, we evaluated the performance of the model searched by NAS on the Sleep-EDF-20, Sleep-EDF-78 and SHHS datasets with an average accuracy of 82.7%, 80.0% and 81.9%, respectively. The experimental results show that the proposed NAS algorithm provides some reference for the subsequent automatic design of networks for sleep classification.**

***Index Terms*— Electroencephalogram (EEG), sleep stage classification, neural architecture search (NAS), bilevel optimization approximation.**

I. Introduction

SLEEP is very important for a person. Sleep takes up about one third of a person's life. Sleep disorders can affect the quality of sleep and lead to a number of other diseases, such as cardiovascular and metabolic diseases. Sleep deprivation can also lead to severe impairment of cognitive and motor skills, making reactions slower and increasing the risk of accidents [1]. Therefore, the diagnosis of sleep disorders becomes particularly important.

Polysomnography (PSG) is the most important test to diagnose sleep disorders by continuous monitoring to understand the patient's condition. This record usually includes readings from four channels, namely electroencephalography (EEG), electrocardiogram (ECG), electrooculogram (EOG) and electromyogram (EMG) [2]. EEG has attracted the attention of a wide range of researchers because of its low cost, convenience and non-invasive nature [3]. In sleep classification studies, EEG is used to create different sleep stages (30-second recording segments), which are then divided into different sleep stages by experts. The classification process should follow the guidelines of the American Academy of sleep medicine (AASM) [4]. In the AASM manual the state of sleep is split into five sleep stages: wakefulness (W), non-rapid eye movement (NREM) sleep stage 1 (N1), NREM sleep stage 2 (N2), NREM sleep stage 3 (N3) and rapid eye movement (REM) sleep stage. Sleep stage classification is commonly used to aid in the diagnosis of sleep disorders [5], but this manual process is very exhaustive, tedious, and time-consuming and the results are also influenced by the subjective consciousness of the raters. Reference [6]. As such, automatic sleep stage classification systems are required to assist sleep specialists.

The traditional sleep EEG classification method is mainly divided into two steps: 1) extract features from preprocessed EEG signals; 2) construct sleep stage classifiers. In traditional methods, many features are designed based on prior knowledge of sleep. As shown in Table I and Table II [7], we see the spectral range of each wave and the type of waves included in each stage of sleep. Generally speaking, in the traditional machine learning algorithms, they will first design and extract various features from the time domain and frequency domain, and then use the feature selection algorithm to further eliminate redundancy and select the most discriminative features. Finally, the selected features are input into the traditional machine learning model for classification, such as fuzzy c-means algorithm (FCM) [8], support vector machines (SVM) [9], random forest (RF) [7], [10], Naive Bayes [11].

In recent years, with the rapid development of deep learning (DL), researchers have a strong interest in this algorithm which can automatically learn features [12], [13]. DL algorithms can achieve end-to-end classification and are able to extract the most representative features from large amounts of data without the need for rich prior knowledge, making them superior

Manuscript received 19 August 2022; revised 30 November 2022 and 28 December 2022; accepted 16 January 2023. Date of publication 23 January 2023; date of current version 6 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071165, Grant 62171177, and Grant 41901350; in part by the Fundamental Research Funds for the Central Universities under Grant JZ2021HGTB0078; and in part by the Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, under Grant HBIR202206. *(Corresponding author: Hu Peng.)*

Gangwei Kong, Chang Li, Zhihui Han, and Heyuan Qiao are with the Department of Biomedical Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: gwkong@mail.hfut.edu.cn; changli@hfut.edu.cn; hanzh@hfut.edu.cn; qiao_heyuan@hfut.edu.cn).

Hu Peng is with the Department of Biomedical Engineering and the School of Instrument Science and Opto-Electronics Engineering, Hefei University of Technology, Hefei 230009, China, and also with the Anhui Province Key Laboratory of Measuring Theory and Precision Instrument, Hefei University of Technology, Hefei 230009, China (e-mail: hpeng@hfut.edu.cn).

Digital Object Identifier 10.1109/TNSRE.2023.3238764

TABLE I
FREQUENCY BANDS OF EEG RHYTHMS

Rhythm	Frequency Band (Hz)
Delta (δ)	0-4
Theta (θ)	4-8
Alpha (α)	8-12
Sigma (σ)	12-15
Beta 1 (β_1)	15-22
Beta 2 (β_2)	22-30
Gamma 1 (γ_1)	30-40
Gamma 2 (γ_1)	40-49.5

TABLE II
RELATIONSHIP BETWEEN CHARACTERISTIC WAVES OF EEG AND DIFFERENT STAGES OF SLEEP

Sleep Stage	Included Characteristic Waves
W	α , β
N1	α , θ
N2	k complex , spindle wave , δ , θ
N3	δ , spindle wave
REM	α , β , θ

to traditional algorithms to some extent. As one of the deep learning algorithms, convolutional neural networks (CNNs) is widely used in computer vision, such as object detection, image segmentation, image fusion and face recognition [14], [15], [16], [17]. In the field of biomedical engineering, CNN is also widely used in medical imaging and classification of one-dimensional biological signals, such as EEG and ECG. In recent years, many sleep EEG classification methods based on CNN are also developing rapidly [18]. In [19], the authors used successive convolution and pooling layers with fully-connected layers and the overall accuracy was 74%. In [20], the authors designed a deeper CNN network to verify that network depth could improve network performance. In [21], the authors transformed the raw EEG into a logarithmic power spectrum and then performed sleep EEG classification by a joint classification and prediction framework. In addition, the researchers found that there were certain transition rules for sleep stages, and the next possible stage could be determined by the previous stages [4]. In the field of DL, recurrent neural networks (RNNs) can extract time-dependent features of sleep EEG very well. For example, in [22], the authors implemented CNN and bidirectional-long short term memory (LSTM) for automatic EEG classification. In [23], the authors classified single-channel EEG signals by means of 4-class LSTM RNN and 2-class LSTM RNN cascades. In addition, other researchers have used attentional mechanisms [24], self-supervision [25], and other methods to classify sleep EEG.

Although DL approaches have shown outstanding advantages in EEG-based sleep classification, most of the existing architectures are designed by human experts, which requires certain prior knowledge and experience, and it is also a time-consuming and error-prone process [26]. In addition, it is difficult for experts to design ideal models due to people's inherent mindset. Therefore, people hope to search network architectures automatically by algorithms, which can greatly liberate the creativity of researchers and reduce the heavy network design cost. On the same clinical problem, the authors in [27] and [28] implemented the classification of emotional EEG based on reinforcement learning and transformer, respectively.

To solve the above problems, we introduce the neural architecture search (NAS) algorithm and propose a fast and low-cost method for automatic design of sleep EEG classification task models. At the same time, in order to simplify the model, we only design the network through the simple CNN architecture stack, and give up those common network structures, such as LSTM and attention mechanism. The experimental results show that the model has good performance.

The main contributions of this paper can be summarized as follows:

1. Through the NAS framework based on bilevel optimization approximation, the automatic classification network structure based on sleep EEG is automatically designed. The algorithm aims to find the optimal CNN structure in the discrete search space, which can extract the features from the original EEG signals, and realize sleep EEG classification. NAS not only saves man-made design time, but also ensures the accuracy of network structure.
2. We implement search model by a bilevel optimization approximation. The model is further optimized by searching deeper networks through search space approximation and search space regularization.
3. We have conducted extensive experiments on public data sets. The experimental results show that in the method based on CNN, the network performance we designed is equivalent to the latest scoring system, which proves the effectiveness of the algorithm.

The rest of this paper is organized as follows: Section II presents related work. Section III introduces the proposed method, and Section IV shows the experiments and results on three datasets sleep-EDF-20, sleep-EDF-78 and SHHS. Finally, we discuss and conclude our research in Section V and Section VI, respectively.

II. RELATED WORK

In this section, we briefly introduce the development of CNN and the background of NAS.

A. CNN

In recent years, CNN has made great achievements in image classification, target detection and other fields due to its powerful performance. CNN has promoted feature extraction from the manual design stage to the self-learning stage, and the network model is also constantly updated and iterated from the original models such as LeNet [29] to the current models such as AlexNet, VGG, ResNet [30], [31], [32] and other models. In the field of brain-computer interface, CNN is able to extract and generalize temporal information better, and is widely used in the classification of EEG data. For example, EEG-based emotion recognition and sleep stage classification [19], [27].

Numerous studies have shown that feature representation and final performance depend heavily on the network structure. Researchers have designed various complex structures to

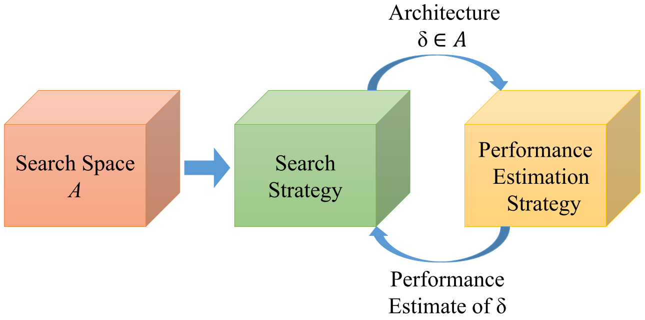


Fig. 1. NAS basic framework. A suitable structure δ is selected in the search space A by different search strategies, and then a suitable evaluation index is selected to evaluate the selected structure δ and feed it back to the front-end network. In this cycle, the structure δ with the best evaluation index is finally selected.

achieve better data feature representation. As the complexity of the network increases, the performance of the network continues to improve, but the parameters also keep increasing, which in turn increases the difficulty of parameter tuning. Therefore, people hope that the machine can adjust parameters and design a reasonable network structure by itself on the premise of ensuring the network performance.

B. NAS

The NAS has emerged to meet people's hopes and has quickly become a new research hotspot. The significance of NAS is to solve the tuning problem of deep learning models, which is a new approach to automate the design of network structures. As described above, with the complexity of the model, the design of the neural network architecture has shifted from manual design to automatic machine design [33]. As shown in Fig. 1, NAS mainly includes search space, search strategy and evaluation strategy. One of the main constraints on the development of NAS is it needs huge GPU resources. In recent years, with the continuous optimization and improvement of search strategies, the demand for hardware resources has been greatly reduced. Common search strategies include stochastic search, Bayesian optimization, evolutionary algorithms, reinforcement learning, and gradient-based algorithms [26]. In [34], the authors used reinforcement learning for neural network architecture search and outperformed previously hand-designed networks on image classification and language modeling tasks. In [35], the authors introduced evolutionary algorithms to NAS. In [36], the authors proposed the differentiable structure search method, which continuous sizes the search space. Currently, NAS has been widely used in object detection, semantic segmentation, image classification, etc [37], [38], [39].

III. Methodology

In this section, we first introduce the basic framework of the proposed sleep EEG classification, and then introduce the structure of the cell used in detail. Finally, the search principle is briefly described.

A. Basic Framework of NAS

In this experiment we utilize differentiable architecture searcharts (DARTS) [36] as our baseline framework. The DARTS algorithm is much simpler than many existing algorithms, it does not involve any controllers, hypnetworks, and the architecture uses a differentiable structure search approach, which greatly reduces memory consumption. Its goal is to find a cell with optimal performance as the building block of the final architecture, the details of the cell will be described later. Then the deeper network is retrained by stacking optimal cells. Based on this, this paper achieves end-to-end sleep EEG classification by extracting the most effective and realistic information from raw EEG through a data-driven approach. The schematic flow of the overall framework is shown in Fig. 2. In general, there are two steps: 1) search for the optimal cell; 2) combine the optimal cell into a model and retrain the model. As shown in Fig. 2, first, we divide the raw EEG data into training samples and validation samples. The training is roughly divided into two steps: 1) fix the architectural parameters and train the model parameters with the training samples; 2) fix the model parameters and train the architectural parameters with the validation samples. The two steps are crossed and both use gradient descent to minimize the cross-entropy loss of the model on the data. Finally, we select the cell with the best performance for overlaying and retrain from zero. In order to achieve accurate sleep stage classification, we use 20-fold cross-validation and take the average classification accuracy of the test samples as the final classification result. The specific details will be shown in the following sections.

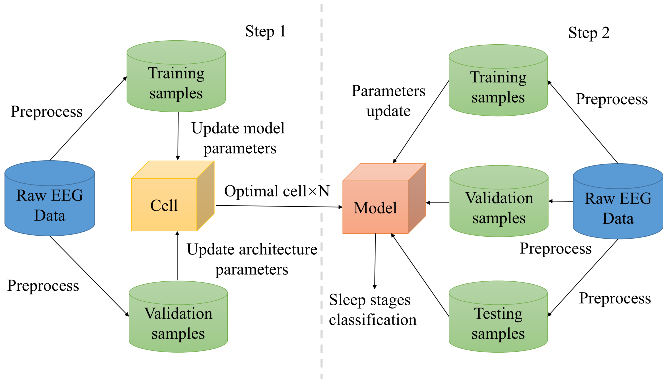


Fig. 2. Flowchart of the proposed method. The left side is the search cell, the right side is the training of the model composed of the optimal cell and the classification of sleep EEG.

B. The Construction of Cell

Cell is a directed acyclic graph consisting of an ordered sequence of N nodes. Taking the four nodes in Fig. 3 as an example. Each gray matrix in the graph, called node, represents a feature map, and we need to connect these nodes by some operations (e.g., convolution, pooling). The colored connecting lines between each feature map represent the operations. Assuming that there are only three optional operations in the figure, to make the search space continuous, we assign a weight α to all operations between two nodes and relax the categorical choice of a particular operation to a softmax over all possible operations to achieve a continuum of discrete operations. Then the weights α are optimized by

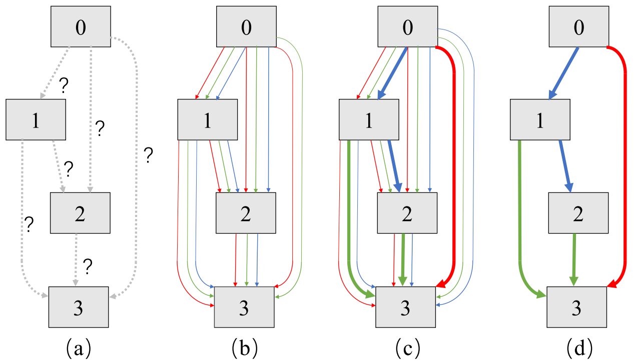


Fig. 3. (a) The search phase, the connection operation between two nodes is unknown; (b) the set of all operations; (c) the weight of each operation after training, the thicker the line means the larger the weight; (d) the operation with the largest weight is selected to form the final cell.

gradient descent, and finally the operation with the largest weight α is taken for each node to form the final cell structure graph. Taking Fig. 3 as an example, the colored connecting line in the figure is the structure parameter α in the search experiment, node denotes the feature map x, and o denotes the operation, then any node in the middle can be expressed by the following formula:

$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)}), \tag{1}$$

$$\bar{o}^{(i,j)}(x) = \sum_{o \in O} \frac{\exp(\alpha_o{}^{(i,j)})}{\sum_{o' \in O} \exp(\alpha_{o'}{}^{(i,j)})} o(x), \tag{2}$$

where i and j denote the ordinal number of the node, O denotes the set of candidate operations, and $o(\cdot)$ denotes some function applied to $x^{(i)}$, and each intermediate node is computed based on all the nodes before it. The operation mixing weights for a pair of nodes (i, j) are parameterized by a vector $\alpha^{(i,j)}$ of dimension $|O|$. The task of architecture search then reduces to learning a set of continuous variables $\alpha = \{a^{(i,j)}\}$. At the end of search, a discrete architecture can be obtained by replacing each mixed operation $\bar{o}^{(i,j)}$ with the most likely operation, i.e., $o^{(i,j)} = \arg\max_{o \in O} \alpha_o{}^{(i,j)}$.

In this paper, the cell consists of 7 nodes, which are 2 input nodes, 4 intermediate nodes and 1 output node, and the output node is deeply connected between all intermediate nodes, as shown in Fig. 4, k denotes the ordinal number of the cell. Meanwhile, in 1/3 and 2/3 of the network is the reduction cell, and the rest is the normal cell, the reduction cell shares the weight $\alpha_{reduction}$, and the normal cell shares the weight $\alpha_{reduction}$.

C. Designing Convolution Network

As described above, this is a bilevel optimization problem. Our goal is to search for a suitable cell. $\mathcal{L}_{train}$, $\mathcal{L}_{val}$ are used to represent the training loss and validation loss, respectively. The above losses are defined not only by the structure α, but also by the weight matrix ω in the network. The goal of network search is to find some structure α^* that minimizes the validation loss $\mathcal{L}_{val}(\omega^*, \alpha^*)$, i.e, $\alpha^* = \arg\min_{\alpha} \mathcal{L}_{val}(\omega^*, \alpha)$.

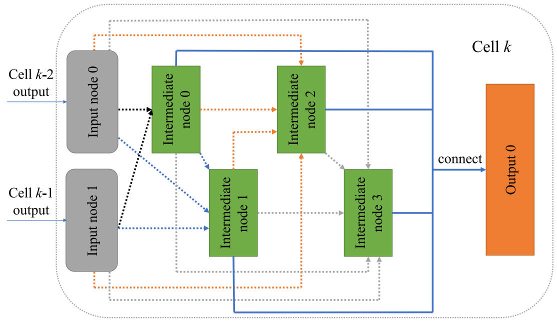


Fig. 4. The dashed line indicates that all intermediate nodes are connected to the forward node unknown, and different colors indicate that different intermediate nodes are connected to all forward nodes.

The weight ω^* associated with the structure is obtained by minimizing the training loss: $\omega^* = \arg\min_{\omega} \mathcal{L}_{train}(\omega, \alpha^*)$. Therefore, it can be summarized as follows:

$$\min_{\alpha} \mathcal{L}_{val}(\omega^*(\alpha), \alpha), \tag{3}$$

$$s.t.\ \omega^*(\alpha) = \arg\min_{\omega} \mathcal{L}_{train}(\omega, \alpha), \tag{4}$$

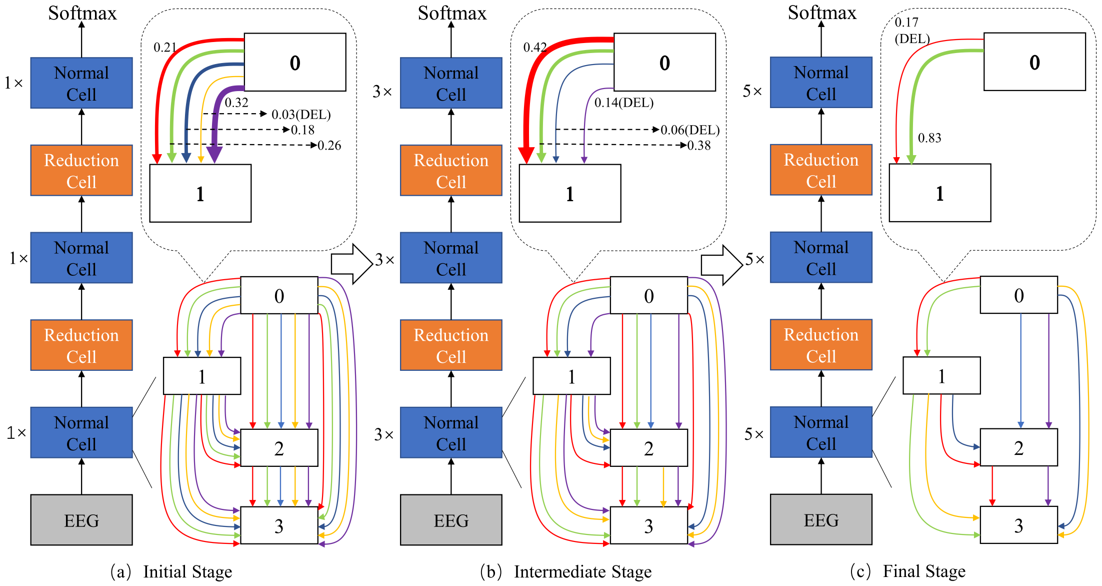
where α is the upper level variable and ω is the lower level variable. Due to the limitation of hardware facilities and training time, it seems difficult to accurately evaluate the architectural gradient by bilevel optimization, so we borrow the approximate gradient estimation from DARTS, called bilevel optimization approximation, as follows:

$$\nabla_{\alpha} \mathcal{L}_{val}(\omega^*(\alpha), \alpha) \approx \nabla_{\alpha} \mathcal{L}_{val}(\omega - \varepsilon \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha), \tag{5}$$

where ω represents the current weight used by the algorithm and ε represents the learning rate of the internal optimization. We approximate $\omega^*(\alpha)$ with $\omega - \varepsilon \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha)$ and use single training step only once for ω throughout the process, achieving a bilevel optimization approximation solution.

At the same time, in the experiment, we found that the cell of DARTS search, *skip-connect* will dominate. Since *skip-connect* is parameterless, the network does not learn features well when there are too many *skip-connects*. In addition, DARTS is validated by searching the optimal cell through a network of 8-layer cell structure and then stacked 20-layer optimal cells to form a deeper network structure to verify performance. We found that the optimal cells searched in the shallow network structure were not well classified when stacked into a deeper network structure. In [40], the authors likewise identified this problem and referred to this phenomenon as depth gap. Based on this, the author of [40] proposed a progressive differentiable architecture search (P-DARTS), which was solved by search space approximation and search space regularization. The basic flow of its algorithm is shown in Fig. 5.

For simplicity, only an intermediate stage and normal cell are shown in the figure, and the reduction cell is similar to this. The search process is divided into three main stages: initial stage, intermediate stage, and final stage. There can be multiple intermediate stages depending on the actual need. For each stage K, the search network includes N_K cells and the



(a) Initial Stage (b) Intermediate Stage (c) Final Stage

Fig. 5. The overall pipeline of P-DARTS.

operation space is O_K. In the initial stage K, N_K is the smallest and O_K is the largest. After training the network, we rank the candidate operations for each connection according to the learned parameter α_K, and then discard the operations with smaller α_K. Meanwhile, we gradually deepen the search network by stacking the cells. At the intermediate stage $K+1$, the number of cells in the new network is N_{K+1}, i.e., $N_{K+1} > N_K$, and the new operation space is O_{K+1}, i.e., $O_{K+1} < O_K$. And so on, the operation with the largest weight is selected in the final stage. The cell in the Fig. 5 increases from 5 in the initial stage to 11 and 17 in the middle and final stages, while the number of candidate operations decreases accordingly from 5 to 3 and 2. By the above method, we achieve the matching of layers instead of searching in the shallow network and then verifying in the deep network. However, at the same time, as the number of layers increases, the requirement for GPU memory gradually increases, so we gradually reduce the candidate operation by judging the number size of α_K, i.e., the search space approximation.

Search space regularization consists of two main parts. The first part is to insert operation-level dropout after each *skip-connect* operation in order to partially cut off the *skip-connect*. But we cannot cut off *skip-connect* endlessly, which would cause *skip-connect* to disappear and be detrimental to the search of the network, so we set a dropout rate for each stage in order to better search the network. We gradually decay the dropout rate during the training process in each search stage, so that the *skip-connect* path is cut off at the beginning and treats equally afterward when parameters of other operations are well learned. The second part is the architectural optimization, i.e., controlling the number of *skip-connect* in the final search stage to M. If the number of *skip-connect* is not M, the cell is searched for the M *skip-connects* with the largest weight in the cell, and the other weights are set to 0, and the cell is reconstructed by iteration. Through search space regularization, the problem of the number of *skip-connect* is well solved.

TABLE III
DATA INTRODUCTION

Dataset	W	N1	N2	N3	REM	Total
Sleep-EDF-20	8285 19.6%	2804 6.6%	17799 42.1%	5703 13.5%	7717 18.2%	42308
Sleep-EDF-78	65951 33.7%	21522 11.0%	69132 35.4%	13039 6.7%	25835 13.2%	195479
SHHS	46319 14.3%	10304 3.2%	142125 43.7%	60153 18.5%	65953 20.3%	324854

The bilevel optimization approximation, search space approximation and search space regularization allow us to search for networks with better performance under hardware device constraints. The long training time of bilevel optimization and depth gap problems are largely solved.

IV. Experiment and Analysis

In this section, we first introduce the public data sets used in our experiment. Then we describe the experimental setup in detail. Finally, we present the experimental results and compare the results of our method with other methods on Sleep-EDF-20, Sleep-EDF-78 [41], [42] and SHHS [43], [44] databases.

A. Data Materials

In our experiments, we use three publicly available datasets, namely, Sleep-EDF-20, Sleep-EDF-78, and SHHS as shown in Table III.

1) Sleep-EDF-20: It contains 39 PSG records for 20 subjects (one night for subject 13 and two full nights for the rest). It contains two different studies on healthy

TABLE IV
DATA FORMAT

Dataset	Array	Array Shape	Array Content
Sleep-EDF	Data Lables	$n \times 1 \times 3000$ $n \times 1$	numbers × channel × data numbers × channel
SHHS	Data Lables	$n \times 1 \times 3750$ $n \times 1$	numbers × channel × data numbers × channel

subjects: Sleep Cassette (SC), which studies the effect of age on sleep, and Sleep Telemetry (ST), which studies the effect of temazepam on sleep. Each PSG record contains two EEG channels (Fpz-Cz and Pz-Oz), one EOG channel, one chin EMG channel, and event markers. Both EEG and EOG use 100 Hz sampling. Sleep scoring was performed by the sleep specialist in 30-second epochs according to the Rechtschaffen & Kales manual [45], and the corresponding sleep maps were scored manually. Each epoch was labeled into one stage of W, N1, N2, N3, N4, REM, Movement (M) and UNKNOWN.

2) Sleep-EDF-78: It is an extended dataset of Sleep-EDF-20 containing 153 PSG records for 78 subjects (two full nights of PSG records for all subjects except for the first night for subjects 36 and 52, and the second night for subject 13). The rest of the records were identical to Sleep-EDF-20.

3) SHHS: It is a multi-center cohort study of the cardiovascular and other consequences of sleep-disordered breathing. The subjects suffer from various diseases including lung diseases, cardiovascular diseases and coronary diseases. To minimize the impact of these diseases, we followed the study in [46] to select subjects, who are considered to have a regular sleep (e.g., Apnea Hypopnea Index or AHI less than 5). Eventually, 329 out of 6,441 subjects were selected for our experiments. Notably, we selected the C4-A1 channel with a sampling rate of 125 Hz.

The format of the data set is shown in the Table IV. For Sleep-EDF, n indicates the total number of sleep EEG markers for each subject by the expert, 1 indicates that the selected channel is a single channel named Fpz-Cz, and 3000 indicates the sampling point of the data, using a time of 30 seconds and a sampling frequency of 100 Hz. For SHHS, n indicates the total number of sleep EEG markers for each subject by the expert, 1 indicates that the selected channel is a single channel named C4-A1, and 3750 indicates the sampling point of the data, using a time of 30 seconds and a sampling frequency of 125 Hz.

B. Implementation Details

We built our model using PyTorch 1.4 and trained it on a NVIDIA GTX3060 GPU. In order to explain the details of the experiment more clearly, we introduce it from the following aspects:

1) Evaluation Index: We adopt the accuracy (ACC) rate on the test set as an indicator to verify the performance of our method, in a 20-fold cross-validation way. We get the final accuracy by the final confusion matrix after 20-fold cross-validation. In addition to this, we use three other metrics to evaluate the performance of the model, namely macro-averaged F1-score ($MF1$), Cohen Kappa (k), and the macro-averaged G-mean (MGm) [24]. Both $MF1$ and MGm are commonly used to evaluate the performance of the models on imbalanced datasets. Given the true positives(TP), false positives (FP), true negatives (TN), false negatives (FN), precision (PR), recall (RE) and specificity (SP). They are calculated as in binary classification by considering one class as the positive class and the other four classes as the negative class. The overall accuracy ACC, $MF1$, k and MGm are defined as follows.

$$ACC = \frac{\sum_{i=1}^{N} TP_i}{S}, \tag{6}$$

$$MF1 = \frac{1}{N}\sum_{i=1}^{N} \frac{2 \times PR_i \times RE_i}{PR_i + RE_i}, \tag{7}$$

$$k = \frac{p_o - p_e}{1 - p_e}, \tag{8}$$

$$MGm = \frac{1}{N}\sum_{i=1}^{N} \sqrt{SP_i \times RE_i}, \tag{9}$$

where i stands for stage, S is the total number of samples and N is the number of sleep stages. Each parameter in the formula is defined as follows.

$$PR_i = \frac{TP_i}{TP_i + FP_i}, \tag{10}$$

$$RE_i = \frac{TP_i}{TP_i + FN_i}, \tag{11}$$

$$p_o = ACC, \tag{12}$$

$$p_e = \frac{\sum_{i=1}^{N} (TP_i + FP_i) \times (TP_i + FN_i)}{S^2}, \tag{13}$$

$$SP_i = \frac{TN_i}{TN_i + FP_i}. \tag{14}$$

2) Search Space: We include the following operations in O: 1×3 and 1×5 separable convolutions, 1×3 and 1×5 dilated separable convolutions, 1×3 max pooling, 1×3 average pooling, identity, and $zero$. All operations are of stride one (if applicable) and the convolved feature maps are padded to preserve their spatial resolution. We use the ReLU-Conv-BN order for convolutional operations, and each separable convolution is always applied twice.

3) Settings of NAS: In the search phase, we perform random initialization and the learning rate follows the cosine scheme. The SGD optimizer with lr = 0.001, weight decay 0.001 is used for the network parameters ω. The Adam optimizer with learning rate lr = 0.0006, weight decay 0.001, and momentum = (0.5, 0.999) is used for the structural parameters α. For each stage, we run 25 epochs, where the first 10 epochs only tune the network parameters, while learning the network parameters

TABLE V
THE DIVISION DETAILS OF 20-FOLD CROSS-VALIDATION

Dataset	Size	Fold	Training	Validation	Testing	Array Shape
Sleep-EDF-20	20	20-fold	17 subjects	2 subjects	1 subject	64 × 1 ×3000
Sleep-EDF-78	78	1st to 19th fold 20th fold	70 subjects 72 subjects	4 subjects 4 subjects	4 subjects 2 subjects	64 × 1 ×3000 64 × 1 ×3000
SHHS	329	1st to 19th fold 20th fold	300 subjects 310 subjects	12 subjects 12 subjects	17 subjects 7 subjects	64 × 1 ×3750 64 × 1 ×3750

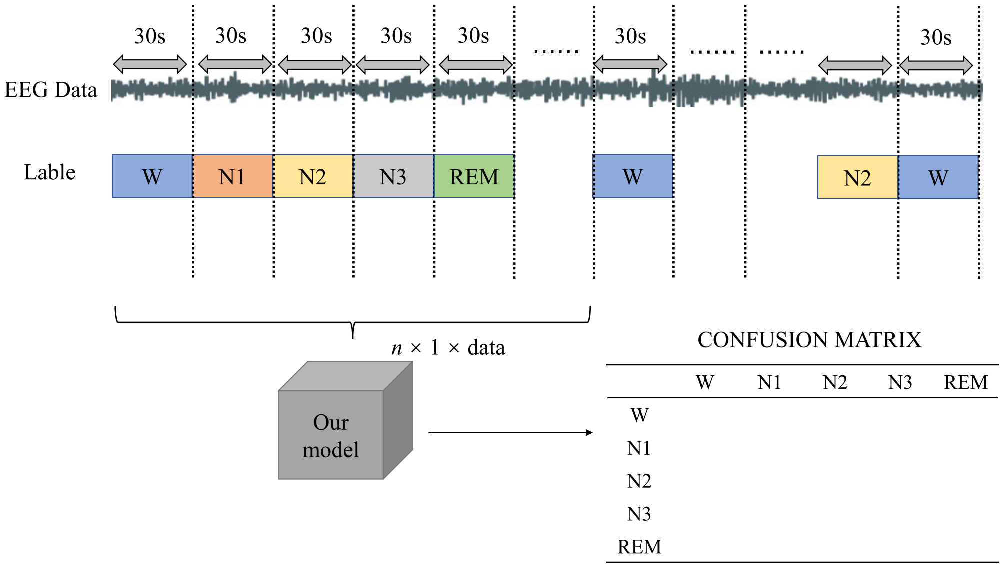


Fig. 6. Data segmentation and management.

and architectural parameters in the remaining 15 epochs. When training the final model, we use a similar setup as above, using the SGD optimizer, and run a total of 50 epochs. Finally, through experimental validation, we finally find the cell retaining 2 $skip - connect$ operations. It is worth mentioning that our network uses a progressive search of 5, 8, 11, and the dropout rate is set to 0.0, 0.4, 0.7.

4) Data Processing: In this experiment, we used three major publicly available datasets. In order to use the AASM criteria, we combine stages N3 and N4 into a single stage N3 and exclude Movement and UNKNOWN classes. To increase the number of sleep stages studied, we include only 30 minutes of waking time before and after sleep [24]. The data division also differs in the network search phase and the network training phase. As shown in the Fig. 2, in the step 1, we divide the data into training set and validation set, where the training set and validation set each account for 50%. In the step 2, we divide the dataset into training set, validation set and test set, and the division details are shown in the Table V, and the training model is validated by 20-fold cross-validation. We use the raw EEG signal as the input to the network. Depending on their different sampling rates for each data set and the number of subjects, the specific administration is shown in Table V. As shown in the Table V, we used a 20-fold cross-validation approach to evaluate the model, avoiding some problems caused by poorly divided data sets. We selected different subjects in each fold as the test set as well as randomly selected certain subjects as the validation set. We can see from the division of the table that each subject made a test set. We selected the final confusion matrix to evaluate our model. In addition, 64 in the table indicates the number of categories per input, whose different values can be selected depending on the hardware conditions. We give an example in Fig. 6. Fig. 6 represents the entire PSG record of a particular object. We divide it into different epochs with a time interval of 30 seconds. According to the sampling rate, the data of sleep-EDF is 3000 and the data of SHHS is 3750. we input our model with 64 epochs as a set of data samples and the last set if less than 64 is the final remaining data as input. Finally, as shown in Table V, we performed 20 cross-validations to obtain the final confusion matrix.

5) Experimental Details: In this experiment, due to hardware conditions and time constraints, we adopt a bilevel optimization approximation for network search as shown in Equation 5. In addition, through experiments and comparisons, we finally choose $M = 2$, i.e., containing two $skip-connects$. In addition, it is worth mentioning that we use the P-DARTS search framework, and the specific parameter settings are shown in **Settings of NAS**. The specific process is as follows: first we randomly divide the smaller dataset Sleep-EDF-20 into a training set and a validation set, each with 50%. We perform four searches, and then select the cell with the highest accuracy

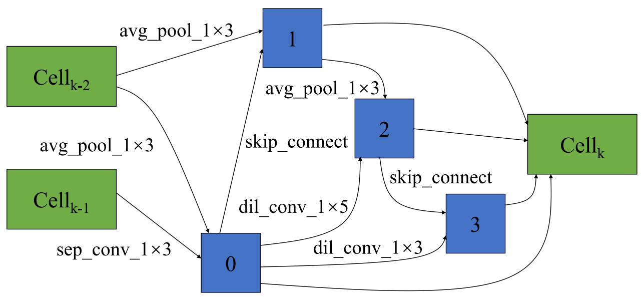


Fig. 7. Normal cell.

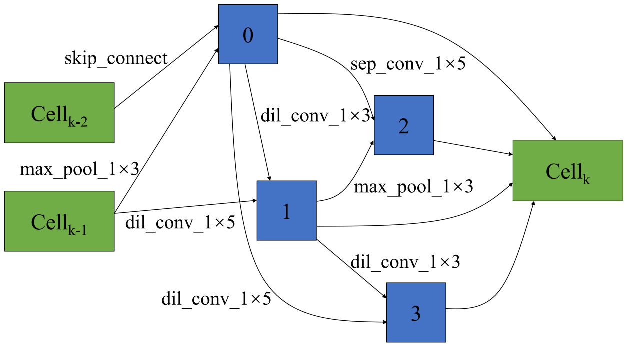


Fig. 8. Reduction cell.

in the validation set as the final cell. Finally, we build the final cell into the final network and train it on Sleep-EDF and SHHS for validation.

6) Cell Details: In our model, there are two types of cells: Normal and Reduction. Normal cell is used when the resolution of input features and output features are the same, and Reduction cell is used when the resolution of output features is half of input features. The design of the Reduction cell is basically the same as that of the Normal cell, except that a convolution operation with stride=2 is added to the input features to reduce the resolution. In the overall network architecture, Normal cell and Reduction cell are designed based on the principle of inserting one Reduction cell in every N Normal cells. For reduction cell, we can extract more information of the bottom layer by down sampling. At the same time, we increase the number of channels, in order to prevent the loss of feature information.

C. Result and Analysis

The proposed NAS discovers the optimal convolution cell and reduction cell are shown in Fig. 7 and Fig. 8, respectively, then we connected these cells in a certain order to form the final network architecture shown in Fig. 9. It is worth mentioning that after extensive experimental validation, we find that it is more ideal to stack 11 cells under the two major premises of ensuring network performance and training time, so the overall framework of the model is shown in Fig. 9. There are 11 layers, of which layer 4 and layer 8 are reduction cell and the rest are convolution cell. We use the original signal as the initial input to the network, and the input of layer N is the output of layer $N-1$ and layer $N-2$. Each layer contains four intermediate nodes, and the output of each layer is the connection of the four intermediate nodes, as detailed in Fig. 4. The operation between each convolution cell and reduction cell is detailed in Fig. 7 and Fig. 8. At the same time, we expanded the channels when we input the first cell. Besides, for the purpose of reducing the number of parameters in the dense connection passed to the softmax layer, we add a global average pool layer to average all the activations of each channel after the last cell layer while avoiding overfitting and making parameters ω updated stably.

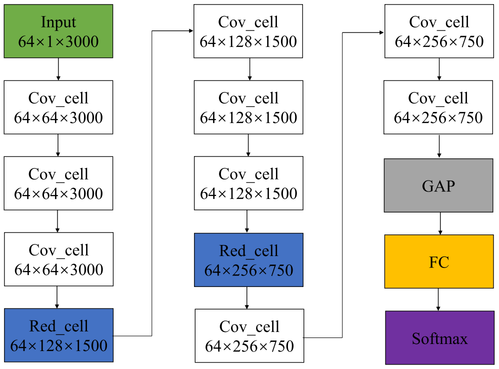


Fig. 9. The final network.

TABLE VI
CONFUSION MATRIX FOR THE FPZ-CZ CHANNEL OF SLEEP-EDF-20

	Predicted					Per-class metrics			
	W	N1	N2	N3	REM	*PR*	*RE*	*F1*	*GM*
W	7200	574	122	28	361	86.2	86.9	86.6	91.6
N1	384	1013	604	4	799	42.0	36.1	38.8	59.0
N2	430	322	15584	543	920	88.1	87.6	87.8	89.5
N3	50	1	667	4983	2	89.6	87.4	88.5	92.7
REM	284	503	716	4	6210	74.9	80.5	77.6	87.0

Table VI, Table VII, and Table VIII show the confusion matrices of the proposed model on three datasets with a single channel and a single epoch (i.e., 30-second EEG signal) as input. The confusion matrices are obtained by summing the confusion matrices for each fold after 20-fold cross-validation. Each row represents the number of samples classified by experts, while each column represents the number of epochs predicted by our model. The tables also show the per-class precision, recall, F1 score and G-mean value for each class.

From the Table VI, Table VII and Table VIII, it can be concluded that the various metrics of stage N1 are lower, and the remaining four stages W, N2, N3 and REM stages outperform stage N1. This may be due to the fact that the number of N1 stage in the dataset is lower than that of the other stages. For each stage, the W stage is mostly mistaken for the N1, N2 and REM stages, the N1 stage is mistaken for the W, N2 and REM stages, and the REM stage is mostly mistaken for the N1, N2 stages. Another fact derived from the tables is that stage N3 is largely confused with stage N2 only.

D. Comparison With Other Methods

In order to better validate the reliability of the proposed method, we conduct extensive experiments on Sleep-EDF-20,

TABLE VII
CONFUSION MATRIX FOR THE FPZ-CZ CHANNEL OF SLEEP-EDF-78

	Predicted					Per-class metrics			
	W	N1	N2	N3	REM	*PR*	*RE*	*F1*	*GM*
W	61002	2869	1029	92	959	89.7	92.5	91.1	93.5
N1	4431	7329	7035	101	2626	46.1	34.1	39.2	56.9
N2	812	2715	61055	1934	2616	80.1	88.3	84.0	88.2
N3	9	7	2610	10387	26	82.3	79.7	81.0	88.7
REM	1737	2985	4468	100	16545	72.7	64.0	68.1	78.5

TABLE VIII
CONFUSION MATRIX FOR THE C4-A1 CHANNEL OF SHHS

	Predicted					Per-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	38597	1031	3283	326	3082	86.3	83.3	84.8	90.3
N1	1326	3022	2710	50	3196	49.8	29.3	36.9	53.9
N2	2548	1004	124768	2928	10877	81.3	87.8	84.4	86.0
N3	424	5	16106	43287	331	92.3	72.0	80.9	84.3
REM	1828	1005	6518	295	56307	76.3	85.4	80.6	89.2

Sleep-EDF-78 and SHHS dataset. In addition, we compared our method with other methods: Cont-CNN [19], Joint-PreCNN [21], Multi-DeepCNN [20], DeepSleepNet [22], SleepEEGNet [47], ResnetLSTM [48]. Brief descriptions for each model are as follows.

- **Cont-CNN** [19] used convolutional neural networks (CNN) for automatic single-channel EEG-based sleep scoring.
- **Joint-PreCNN** [21] introduced a joint classification and prediction formulation. Based on the dependency of sleep periods, the input periods were jointly classified and their adjacent labels were predicted in the contextual output.
- **Multi-DeepCNN** [20] designed a deep CNN and verified that the network performance gain depended mainly on the network depth rather than using multiple channel information.
- **DeepSleepNet** [22] implemented a two-step training algorithm to classify sleep stages through CNN and LSTM structures.
- **SleepEEGNet** [47] used the same CNN architecture as DeepSleepNet and incorporates an encoder-decoder with an attention mechanism.
- **ResnetLSTM** [48] extracted features by ResNet and then used LSTM for sleep EEG classification.

Table IX, and Table X show the comparison between the six models and our model. We can notice that our model achieves a better classification performance than the other six models. The six models can be divided into two categories, the first three are CNN models and the last three are CNNs combined with other model structures, such as LSTM, attention mechanisms. All the models in Table IX are CNN architectures and have not other structures such as LSTM. The difference is that Joint-PreCNN converts the single-channel raw signal into a logarithmic power spectrum, and Multi-DeepCNN uses 3 channels and 5 epochs as inputs. We can observe that our model achieves a significant improvement in all three metrics, ACC, $MF1$ and k. Of which, ACC is 82.7%, $MF1$ is 75.9 and k is 0.76. All three performance metrics are higher than other models. In Table X, we compare with models of other structures. DeepSleepNet utilized CNN to extract time-invariant features, and bidirectional-LSTM to learn transition rules among sleep stages automatically from EEG epochs. Similarly, ResnetLSTM used the structure of the LSTM. SleepEEGNet used the same CNN architecture as DeepSleepNet to extract features and incorporated an encoder-decoder with an attention mechanism to classify. By comparing, we can observe that the performance of our model is basically the same as that of the other three models, and even has a slight improvement in accuracy. In summary, NAS can automatically extract features and design better CNN architectures than those designed manually. At the same time, by comparing with other network structures, we have reasons to believe that NAS can also extract some features that are neglected by manually designed CNN architectures as well as its ability to extract features on unbalanced datasets.

TABLE IX
COMPARISON WITH THE FIRST THREE MODELS

Dataset	Method	*ACC*	*MF1*	*k*
Sleep-EDF-20	Cont-CNN	74.0	69.8	0.65
	Joint-PreCNN(one-to-one)	79.8	72.0	0.72
	Multi-DeepCNN	81.0	72.0	0.73
	Ours	**82.7**	**75.9**	**0.76**

TABLE X
COMPARISON WITH THE LAST THREE MODELS

Dataset	Method	*ACC*	*MF1*	*k*	*MGm*
Sleep-EDF-20	DeepSleepNet	81.9	**76.6**	**0.76**	**86.9**
	SleepEEGNet	81.5	**76.6**	0.75	85.3
	ResnetLSTM	82.5	73.7	**0.76**	81.9
	Ours	**82.7**	75.9	**0.76**	84.0
Sleep-EDF-78	DeepSleepNet	77.8	71.8	0.70	81.6
	SleepEEGNet	74.2	69.6	0.66	**82.3**
	ResnetLSTM	78.9	71.4	0.71	80.8
	Ours	**80.0**	**72.7**	**0.72**	81.2
SHHS	DeepSleepNet	81.0	**73.9**	0.73	**82.6**
	SleepEEGNet	73.9	68.1	0.65	82.7
	Joint-PreCNN	81.1	71.2	**0.74**	80.4
	Ours	**81.9**	73.5	**0.74**	80.7

V. DISCUSSIONS

In recent years, there has been a growing interest in sleep quality. Traditional classification-based methods and deep learning-based classification methods have shown an explosive growth. Each method has its advantages and disadvantages. The traditional classification method can extract some important neurophysiological features in sleep EEG from the original signal by frequency domain analysis, time domain analysis [49], [50], which is not available in artificial intelligence methods. Then we select the most representative features for classification. In [51], the authors have proposed a new method of time-frequency representation (TFR) which is based on the Fourier-Bessel decomposition method (FBDM). Since recorded EEG signals are non-stationary which lead to time-varying amplitude and spectrum. Hence, they proposed

FBDM method is well suited for the classification of these sleep EEG signals. They convert EEG signals to TFR images, which are then classified by CNN networks. In [52], a new technique for automated classification of sleep stages based on iterative filtering of EEG signals is presented. In order to perform sleep stages classification, the EEG signals are decomposed using iterative filtering method. The discrete energy separation algorithm (DESA) is applied to the modes to determine amplitude envelope and instantaneous frequency functions. The extracted amplitude envelope and instantaneous frequency functions have been used to compute Poincarè plot descriptors and statistical measures, which are applied as input features for different classifiers in order to classify sleep stages. The classifiers namely, naïve Bayes, k-nearest neighbor, multilayer perceptron, C4.5 decision tree, and random forest are applied in order to classify the EEG epochs corresponding to various sleep stages. However, these methods require a large amount of a priori knowledge, while the advantage of AI methods is that the raw signal is used as input to achieve end-to-end classification. We learn the features through neural networks and no longer need a large amount of prior knowledge. Many researchers have demonstrated that end-to-end automatic classification of raw signals using individual EEG channels and convolutional neural networks is possible without any a priori knowledge. However, most of the networks at this stage are based on manual design, which makes it challenging to explore more suitable architectures. This study aims to automatically search convolutional neural network architectures for EEG-based end-to-end sleep stage recognition. In this paper, we propose an automatic search method NAS for classification tasks. Unlike traditional application of reinforcement learning or genetic evolution algorithms on discrete and non-minimizable search spaces, our approach is based on relaxation of the architectural representation, allowing the use of gradient descent efficient search architecture, which greatly saves search time and hardware memory footprint and enables training models on a single GPU.

NAS is mainly used in the field of image classification, and there are still very few applications in the field of EEG. In this paper, we demonstrate through extensive experiments that the one-dimensional EEG signal based on NAS still has good performance. As shown in Table IX and Table X, the proposed NAS is competitive compared with CNN and other methods based on CNN, indicating that the model obtained by automatic search can replace manual design to a certain extent, freeing up a lot of manpower and reducing the cost of designing networks. Separate convolution, dilated separable convolution, max pooling, average pooling and skip-connect are used in our model. We use the neural architecture search (NAS) method to combine the above operations and finally form an optimal cell. Compared with other structures, we are searching for an optimal cell by NAS algorithm instead of designing by hand, which saves a lot of labor and time in designing the network. Besides, the combination of our structures is more flexible. We implement different combinations by our algorithm and then select an optimal cell, while the manually designed convolutional networks are often fixed. Of course, in addition to the above advantages, we also use separate convolution to speed up the computation, dilated convolution to improve the perceptual field and extract more features, and use $skip - connect$ to speed up the convergence. These methods are rarely used in convolutional neural networks that deal with the same problem. We also used the structure of Relu-Convolution-Batch Normalization for convolutional operations to improve the generalization ability of the model.

Although we have achieved good performance with our network, the network we found is not necessarily optimal. During the network search, the network optimization is very sensitive to the initialization value. Under the same conditions each time, we have performed more than four searches with random initialization, each time deciding the final architecture based on the optimal results obtained on the validation set. Due to time cost constraints, we were unable to perform multiple searches on the network to obtain all model architectures.

VI. Conclusion

In this paper, we propose an end-to-end deep learning method named NAS for sleep EEG recognition. The main contribution of this method is the implementation of automatic design of neural networks, which provides a new way of thinking for the subsequent study of sleep EEG classification. The experimental results on three public datasets demonstrated that the performance of our model reaches certain criteria under various evaluation matrices, proving the effectiveness of the method. In the future, we will further improve our method to overcome the problem of low accuracy in the N1 stage. In addition, based on the huge search space, it is worthwhile to investigate how to design a more concise and efficient model.

VII. Acknowledgment

The authors would like to thank Prof. Min Wu and Dr. Emadeldeen Eldele of Nanyang Technological University for providing them with the SHHS dataset and for their answers and help.

References

[1] F. S. Luyster, P. J. Strollo, P. C. Zee, and J. K. Walsh, "Sleep: A health imperative," *Sleep*, vol. 35, no. 6, pp. 727–734, Jun. 2012.

[2] N. Goshtasbi, R. Boostani, and S. Sanei, "SleepFCN: A fully convolutional deep learning framework for sleep stage classification using single-channel electroencephalograms," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2088–2096, 2022.

[3] M. Xu, X. Xiao, Y. Wang, H. Qi, T.-P. Jung, and D. Ming, "A brain–computer interface based on miniature-event-related potentials induced by very small lateral visual stimuli," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 5, pp. 1166–1175, Jan. 2018.

[4] R. B. Berry et al., "The AASM manual for the scoring of sleep and associated events," *Rules, Terminol. Tech. Specifications, Darien, Illinois, Amer. Acad. Sleep Med.*, vol. 176, p. 2012, Oct. 2012.

[5] T. Sudhakar, G. Hari Krishnan, N. R. Krishnamoorthy, B. Janney, J. M. Pradeepa, and J. P. Raghavi, "Sleep disorder diagnosis using EEG based deep learning techniques," in *Proc. 7th Int. Conf. Bio Signals, Images, Instrum. (ICBSII)*, Mar. 2021, pp. 1–4.

[6] A. Malhotra et al., "Performance of an automated polysomnography scoring system versus computer-assisted manual scoring," *Sleep*, vol. 36, no. 4, pp. 573–582, 2013.

[7] P. Memar and F. Faradji, "A novel multi-class EEG-based sleep stage classification system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 1, pp. 84–95, Jan. 2017.

[8] M. Obayya and F. E. Z. Abou-Chadi, “Automatic classification of sleep stages using EEG records based on fuzzy C-means (FCM) algorithm,” in *Proc. 31st Nat. Radio Sci. Conf. (NRSC)*, Apr. 2014, pp. 265–272.

[9] B. Koley and D. Dey, “An ensemble system for automatic sleep stage classification using single channel EEG signal,” *Comput. Biol. Med.*, vol. 42, no. 12, pp. 1186–1195, 2012.

[10] A. Malafeev et al., “Automatic human sleep stage scoring using deep neural networks,” *Frontiers Neurosci.*, vol. 12, p. 781, Nov. 2018.

[11] S. I. Dimitriadis, C. Salis, and D. Linden, “A novel, fast and efficient single-sensor automatic sleep-stage classification based on complementary cross-frequency coupling estimates,” *Clin. Neurophysiol.*, vol. 129, no. 4, pp. 815–828, 2018.

[12] X. Chen, C. Li, A. Liu, M. J. McKeown, R. Qian, and Z. J. Wang, “Toward open-world electroencephalogram decoding via deep learning: A comprehensive survey,” *IEEE Signal Process. Mag.*, vol. 39, no. 2, pp. 117–134, Mar. 2022.

[13] C. Li et al., “Emotion recognition from EEG based on multi-task learning with capsule network and attention mechanism,” *Comput. Biol. Med.*, vol. 143, Apr. 2022, Art. no. 105303.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[15] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[16] Y. Zhang, Y. Liu, P. Sun, H. Yan, X. Zhao, and L. Zhang, “IFCNN: A general image fusion framework based on convolutional neural network,” *Inf. Fusion*, vol. 54, pp. 99–118, Feb. 2020.

[17] S. Wen, M. Dong, Y. Yang, P. Zhou, T. Huang, and Y. Chen, “End-to-end detection-segmentation system for face labeling,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 3, pp. 457–467, Jun. 2021.

[18] N. Parekh, B. Dave, R. Shah, and K. Srivastava, “Automatic sleep stage scoring on raw single-channel EEG: A comparative analysis of CNN architectures,” in *Proc. 4th Int. Conf. Elect., Comput. Commun. Technol. (ICECCT)*, Sep. 2021, pp. 1–8.

[19] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, “Automatic sleep stage scoring with single-channel EEG using convolutional neural networks,” 2016, *arXiv:1610.01683*.

[20] M. Sokolovsky, F. Guerrero, S. Paisarnsrisomsuk, C. Ruiz, and S. A. Alvarez, “Deep learning for automated feature discovery and classification of sleep stages,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 6, pp. 1835–1845, Nov. 2020.

[21] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. De Vos, “Joint classification and prediction CNN framework for automatic sleep stage classification,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, May 2019.

[22] A. Supratak, H. Dong, C. Wu, and Y. Guo, “DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 11, pp. 1998–2008, Nov. 2017.

[23] N. Michielli, U. R. Acharya, and F. Molinari, “Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals,” *Comput. Biol. Med.*, vol. 106, pp. 71–81, Mar. 2019.

[24] E. Eldele et al., “An attention-based deep learning approach for sleep stage classification with single-channel EEG,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 809–818, 2021.

[25] C. Yang, D. Xiao, M. Brandon Westover, and J. Sun, “Self-supervised EEG representation learning for automatic sleep staging,” 2021, *arXiv:2110.15278*.

[26] M. Wistuba, A. Rawat, and T. Pedapati, “A survey on neural architecture search,” 2019, *arXiv:1905.01392*.

[27] C. Li, Z. Zhang, R. Song, J. Cheng, Y. Liu, and X. Chen, “EEG-based emotion recognition via neural architecture search,” *IEEE Trans. Affect. Comput.*, early access, Nov. 24, 2021, doi: 10.1109/TAFFC.2021.3130387.

[28] C. Li, Z. Zhang, X. Zhang, G. Huang, Y. Liu, and X. Chen, “EEG-based emotion recognition via transformer neural architecture search,” *IEEE Trans. Ind. Informat.*, early access, Apr. 26, 2022, doi: 10.1109/TII.2022.3170422.

[29] 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, Nov. 1998.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[33] D. Baymurzina, E. Golikov, and M. Burtsev, “A review of neural architecture search,” *Neurocomputing*, vol. 474, pp. 82–93, Feb. 2022.

[34] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” 2016, *arXiv:1611.01578*.

[35] E. Real et al., “Large-scale evolution of image classifiers,” Mar. 2017, *arXiv:1703.01041*.

[36] H. Liu, K. Simonyan, and Y. Yang, “DARTS: Differentiable architecture search,” 2018, *arXiv:1806.09055*.

[37] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, “DetNAS: Backbone search for object detection,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[38] C. Liu et al., “Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 82–92.

[39] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[40] X. Chen, L. Xie, J. Wu, and Q. Tian, “Progressive differentiable architecture search: Bridging the depth gap between search and evaluation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1294–1303.

[41] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye, “Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG,” *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000.

[42] A. L. Goldberger et al., “Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[43] G.-Q. Zhang et al., “The national sleep research resource: Towards a sleep data commons,” *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 10, pp. 1351–1358, 2018.

[44] S. F. Quan et al., “The sleep heart health study: Design, rationale, and methods,” *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.

[45] E. A. Wolpert, “A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects,” *Arch. Gen. Psychiatry*, vol. 20, no. 2, pp. 246–247, 1969.

[46] P. Fonseca, N. den Teuling, X. Long, and R. M. Aarts, “Cardiorespiratory sleep stage detection using conditional random fields,” *IEEE J. Biomed. Health Informat.*, vol. 21, no. 4, pp. 956–966, Jul. 2017.

[47] S. Mousavi, F. Afghah, and U. R. Acharya, “SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach,” *PLoS ONE*, vol. 14, no. 5, May 2019, Art. no. e0216456.

[48] Y. Sun, B. Wang, J. Jin, and X. Wang, “Deep convolutional network method for automatic sleep stage classification based on neurophysiological signals,” in *Proc. 11th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2018, pp. 1–5.

[49] 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,” *Int. J. Neural Syst.*, vol. 20, no. 6, pp. 509–521, Dec. 2010.

[50] L. Fraiwan, K. Lweesy, N. Khasawneh, M. Fraiwan, H. Wenz, and H. Dickhaus, “Time frequency analysis for automated sleep stage identification in fullterm and preterm neonates,” *J. Med. Syst.*, vol. 35, no. 4, pp. 693–702, Aug. 2011.

[51] V. Gupta and R. B. Pachori, “FBDM based time-frequency representation for sleep stages classification using EEG signals,” *Biomed. Signal Process. Control*, vol. 64, Feb. 2021, Art. no. 102265.

[52] R. Sharma, R. B. Pachori, and A. Upadhyay, “Automatic sleep stages classification based on iterative filtering of electroencephalogram signals,” *Neural Comput. Appl.*, vol. 28, no. 10, pp. 2959–2978, Oct. 2017.