Learning Long-range Temporal Structure for Sleep Stage Classification

Jinzhuo Wang*, Navin Cooray, Christine Lo, Michele Hu, Huy Phan, Maarten De Vos

***Abstract*—Recent progress in automatic sleep stage classification suggests there are advantages in mining temporal dependencies in consecutive epochs. However, existing solutions are still limited in short-range sequential learning. This paper instead presents a framework called SegNet that can map long-range epochs to stage labels. We escape from conventional ways of treating each epoch equally by considering consecutive epochs with the same label as a whole, and consequently converting sleep stage classification to temporal detection scenario with the aims (1) to locate temporal regions that we call segments in which epochs share the same labels and (2) to predict the stage label for each segment. We show how to incorporate these two phases in a unified framework and efficiently train it. We demonstrate promising classification accuracy on two public datasets and visualize reasonable stage structures generated by our model.**

***Index Terms*—Sleep stage classification, SegNet, long-range dependencies, segment detection, deep neural network**

I. Introduction

SLEEP plays an important role in physiological homeostasis. Nowadays, a large number of people are suffering from sleep-related disorders, such as sleep apnea syndrome and insomnia [1]. In order to diagnose sleep-related diseases, sleep quality is usually evaluated using polysomnography (PSG) devices which record multiple physiological signals. One important factor to evaluate sleep quality is the distribution of different sleep stages [2]. Traditionally, PSG recordings are visually examined and scored by experts according to standardised rules. [3]. However, clinical sleep scoring is time consuming and prone to human error [4]. Thus, automatic sleep stage classification, which has shown to be capable of outperforming manual scoring [5], can be an appropriate solution to produce reliable and repeatable sleep stage classification results.

Researchers often crop the collected overnight PSG signals for a into 30-second segments named epochs and perform automatic stage classification on these consecutive epochs. Solutions on this task start from early attempts incorporating hand-crafted features with statistical models, and are recently dominated by end-to-end deep learning architectures [6]. A recent work [5] summarizes four types of methods according to the length of input and output, including one-to-one, one-to-many, many-to-one and its proposed many-to-many paradigms. The reported results demonstrate the advantages of mining temporal dependencies, by encouraging the deep neural network to learn mapping from raw signals to stage labels in a sequence to sequence manner. However, existing work often learn temporal dependencies for relative short-range consecutive epochs [5] [7], while in many cases tens of consecutive epochs share the same label. According to the statistics on a public large-scale dataset [8], there are more than 85% epochs that have the same label as the previous ones, which encourages us to model long-range dependencies. To this end, this paper aims to construct deep neural architecture that can capture long-range stage structure.

In terms of long-range structure modeling, a key observation is that there are only around 15% epochs that have a stage change termed transition point. Motivated by this observation, we encourage our network to first produce a distribution of transition points over the long-range input sequences, and use this distribution to produce structured segments and then predict the label for each segment. Another important issue when sending long-range epochs to a network is the high dimension of the feature maps and the over-fitting risk. We tackle this issue with a carefully designed segment pooling structure, which is able to aggregate information over the learned segments. In this way, we can obtain a compact feature summarization for each segment, which is prepared for classifier layers to produce final stage prediction. This segment pooling strategy preserves relevant information with dramatically lower cost, thus enabling the network training over long epoch sequences under a reasonable budget in both time and computing resources. We construct a model named SegNet to achieve the above functions. We show SegNet can generate reasonable sleep stage structures and achieve promising classification accuracy on two public datasets.

The rest of this paper is structured as follows. Section II reviews relevant work on sleep stage classification and discusses the relations to our method, followed by our proposed framework and implementation details in Section III. Then, we provide experimental results and comprehensive analysis in Section IV. Finally, we conclude our paper in Section V.

II. Related Work

We first review recent work on the task of automatic sleep stage classification from two perspectives: Traditional statistical models based on hand-crafted features and pure end-to-end deep learning architectures based on raw PSG signals. Then, we cover a line of recent work relevant to our network design and training techniques. For more comprehensive surveys, we refer to latest review articles [9] [10].

Jinzhuo Wang, Navin Cooray, Christine Lo and Michele Hu are with the Nuffield Department of Clinical Neurosciences, University of Oxford, UK. Huy Phan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London. Maarten De Vos is with the Department of Electrical Engineering, Catholic University of Leuven, Belgium. *Corresponding author: jinzhuo.wang@ndcn.ox.ac.uk

Early methods often follow a two-step pipeline. Raw PSG signals are first sent to a carefully designed feature extractor with signal processing approaches to obtain a high-level compact feature summary, which is then sent to a statistical classifier to produce stage prediction. In particular, the feature extraction procedure starts from the measured data and derives values intended to be informative and non-redundant. An example can be the time-domain signal power over the entire epoch [11]. Feature extraction techniques can be linear and non-linear and grouped into three major categories: temporal domain methods, frequency domain methods and hybrid of temporal and frequency domain methods [12] [13] [14]. These techniques allow representing data in a acceptable dimensional space while resulting in increased performance of the classifier [15]. As for the classifier part, artificial neural networks (ANN) and random forest are two common choices. An ANN-based scoring system with varied performance in a broad range of accuracy was reported [16] depending on the recognized stages. The authors in [13] carried out a comparative study to identify the most effective features and the most efficient algorithm to classify the sleep stages. Another study [17] also tried to identify optimal signal processing and classifier methods, focusing on online sleep staging using a single EEG channel. In the comprehensive survey of [14] several sleep stage classification techniques using EEG signals were reported and compared, with accuracy ranging from 70 to 94% on various datasets.

Recently, with the impressive success of deep learning, researchers tend to build large-scale deep network based on raw PSG signals [14] [18] [19] [20] [21] [22] [23] [24] [25]. Specifically, a deep belief net was built in [26] to learn probabilistic representations from raw PSG signals. Convolutional neural networks was also applied to extract time-invariant features from raw single EEG channel [27]. Until 2016, the results from the literature indicate that applying deep learning on hand-engineered features outperformed that on raw signals [27]. One reason might be lacking the exploration of temporal information among epochs that sleep experts often use when they determine the sleep stages. Since 2017, a line of work showed that deep learning based on raw PSG signals are capable of obtaining state-of-the-art performance, including a recent study [5] that trained a sequence to sequence network with RNN block and attention block, taking up to 30 epochs as input. A novel cascaded RNN model was designed in [28] also using single-channel EEG, achieving 86.7% accuracy over five stages on sleep-EDF dataset. Instead of using RNN to learn temporal dependencies, our proposed model first learns to locate temporal regions of input sequences and then predicts semantic stage labels for segments. Authors in [29] showed that a purely U-net [30] like convolutional network can process a whole night PSG signals and reported promising results using input sequence of 35 epochs.

However, most work in both methods use a single epoch as training data while lacking the exploration of temporal dependencies among consecutive epochs, which is one of the key evidences that assist sleep experts in identifying the sleep stages. We therefore encourage our model to consider long-range epochs by taking as input up to 128 epochs and learn temporal stage structure. There are existing efforts in the literature that also leverage long-term input such as 10/20/30 epochs [5], 35 epochs [29] and 50 epochs [21]. However, their approaches only trained the networks to learn the stage labels for the consecutive input epochs all at once while did not incorporate explicit transition rule learning. In our proposed SegNet, we encourage the network to learn the stage transition distribution as well as the stage label within a unified framework.

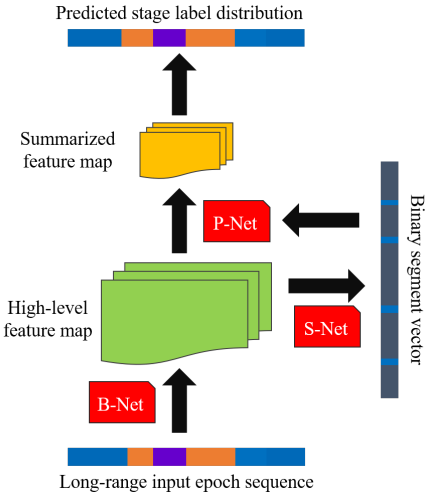

Figure 1: The proposed SegNet consists of mainly three sub-networks, including a backbone network, segment network and prediction network, short for B-Net, S-Net and P-Net in three red rectangles in the figure. Workflow and training details can be found in Section III-A and III-C.

Our network design and training schedule share similar properties with popular object detection systems [31]. The proposed segment pooling layer to obtain a compact feature summary over the large size input is inspired by the successful ROI pooling implementation in Fast RCNN [32]. The iterative training procedure in SegNet training is similar to that of Faster CNN [33]. In our case, the backbone network and segment network are updated in the first learning step while fixed in the second step. This two-step alternating training can be run for more iterations until negligible improvements are observed.

III. Method

Instead of conventional ways of treating each epoch as individual training information, we view sleep staging as a detection problem in temporal domain with discussed prior knowledge on the transition point distribution to assist model design. In particular, we encourage our model to firstly generate the structured segment and then predict stage label for all epochs in each segment at once.

A. *Framework overview*

Fig. 1 illustrates the proposed SegNet framework. SegNet consists of three sub-networks including a backbone network,

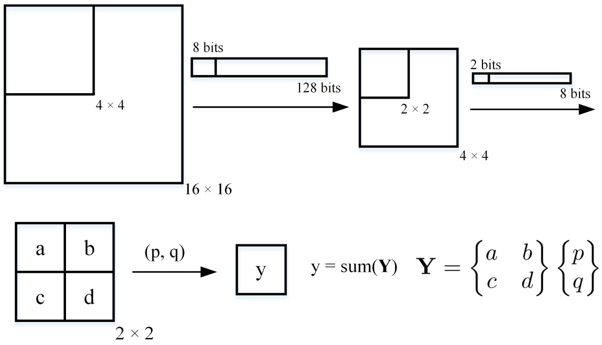
8 bits
128 bits
4 × 4
16 × 16
2 bits
8 bits
2 × 2
4 × 4
a
b
c
d
(p, q)
y
y = sum(Y)
$\mathbf{Y} = \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \begin{Bmatrix} p \\ q \end{Bmatrix}$
2 × 2

Figure 2: Illustration of segment pooling layer. The size of each channel in high-level feature maps is 16×16 and the segment vector is of size 1×128. These two parts are combined and calculated to produce an output summarized feature map of size 4×4, leading to a downsample of $1/16$ times.

a segment network and a prediction network, short for B-Net, S-Net and P-Net, respectively. B-Net takes as input consecutive long-term epochs and produces a core feature map via several convolutional and max pooling layers. This feature map can be regarded as a basic high-level feature and branches into B-Net and P-Net. As seen in the right branch in Figure 1, S-Net is a also sequence of convolutional layers followed by fully connected layers that produce a binary segment vector of the same size as the input epoch length. This segment vector can be viewed as the estimation of the stage distribution over the input epoch sequence. Note that this part has a ground truth that can be used to supervise the training of B-Net and S-Net. The second branch is P-Net which takes inputs from B-Net and S-Net. P-Net combines these two inputs with a segment pooling layer to produce a compact summarization of the core feature map which is then sent to a few fully connected layers that link to the ground truth to produce final prediction.

B. *Segment pooling layer*

One key component in SegNet is the segment pooling layer in P-Net which links to S-Net. It receives both the output of B-Net and the output of S-Net, and conducts a channel-wise pooling operation to summarize the feature of input sequence according to the learned stage distribution. In particular, it works by dividing each channel of the core feature map to 4×4 sub-windows and then applying the segment vector to each sub-window, obtaining the corresponding output grid cells. In our case, each channel of the output of B-Net is of size $256 = 16 \times 16$ and the segment vector is 1×128. For each bin in the output of B-Net, the value comes by applying each part of the segment vector of $128/16 = 8$ size to each bin of the core feature map of $16/4 \times 16/4$ size. The operation details are illustrated in Figure 2. This process results in a compact B-Net output feature map of $1/16$ size, which approximately corresponds to previously discussed epoch distribution, i.e. 15% epochs have the different labels with previous ones. This pooling operation is applied independently to each feature map channel. On the other hand, since the proposed model takes as input a long-range sequential epochs, the core feature map is expected to be of high dimension (e.g. 256×256 in our implementation), the segment pooling layer is also responsible for dimension reduction, reducing the risk of over-fitting for P-Net training.

C. *Two-step training strategy*

In the training stage, long-range epoch sequences are generated with a moderate size of stride to obtain sufficient training data. We then apply a two-step training strategy similar to that of Faster RCNN [33]. Specifically, in the first step, we train the backbone network and the segment network according to the ground truth of stage distribution. In the second step, we train the P-Net using the generated segment vector, while keeping the B-Net and S-Net parameters fixed. Note that we can perform this two-step alternating training for more iterations. In practice, we conduct twice two-step training and observed negligible improvements in further iterations. At test time, we use the trained SegNet on the test epoch sequences without stride to produce sleep stage prediction. Optimization for both steps are performed by minimizing a generalized dice loss [34] between predicted sequential vectors and the ground truth. This cost function is suggested given data imbalance problems such as sleep staging [29].

D. *Model specification*

Our model receives a 128 epoch sequence of single EEG channel as input leading to a size of $1 \times (128 \times 30 \times \mathrm{fs})$. EEG signals on different datasets were re-sampled at $\mathrm{fs} = 100\mathrm{Hz}$ using poly-phase filtering with automatically derived FIR filters [29]. We regard 1 as input channel number and $128 \times 30 \times \mathrm{fs}$ as the feature dimension for each channel. The B-Net consists of sequential layers as follows: $\mathrm{C}(128, 256, \lfloor \mathrm{fs}/8 \rfloor, \lfloor \mathrm{fs}/16 \rfloor) \rightarrow \mathrm{P}(4,4) \rightarrow \mathrm{C}(256, 512, 4) \rightarrow \mathrm{C}(256, 512, 4) \rightarrow \mathrm{C}(512, 512, 4) \rightarrow C(512, 256, 4) \rightarrow \mathrm{DownSample}(256)$, where $\mathrm{C}(n, m, k, s)$ stands for a 1D convolutional layer with n input channel, m output channel, kernel size of k and stride s (stride is set to 1 if not specified), $\mathrm{P}(k, s)$ denotes a 1D max-pooling layer with window size of k and stride s, and $\mathrm{Flatten}$ is the reshape operation along with feature dimension. The S-Net consists of sequential layers as follows: $\mathrm{C}(256, 64, 16, 4) \rightarrow \mathrm{C}(64, 16, 8) \rightarrow \mathrm{Flatten} \rightarrow \mathrm{FC}(864, 128) \rightarrow 128{*}\mathrm{FC}(128, 2)$, where $\mathrm{FC}(n, m)$ is a fully connected layer with n input units and m output units, and $k * \mathrm{FC}(n, m)$ are k $\mathrm{FC}(n, m)$ layers. The P-Net consists of sequential layers as follows: $\mathrm{C}(256, 128, 4) \rightarrow \mathrm{Flatten} \rightarrow \mathrm{FC}(1664, 512) \rightarrow \mathrm{FC}(512, 128) \rightarrow 128 * \mathrm{FC}(128, 5)$. All Conv1D layers are followed by a BatchNorm1D layer and a ReLU layer. SegNet architecture details are summarized in Table I. The total parameter number of SegNet is $3,157,504$, while the number of two popular end-to-end deep learning models DSN [18] and Utime [29] are $26,440,965$ and $1,220,317$, respectively. The code is implemented in pytorch [35] and available at https://github.com/wangjinzhuo/wearables/tree/master/segnet.

Table I: SegNet architecture details. The computation flow follows the training strategy described in Section III-C.

ID	Layer type	Input (ch×dim)	Output (ch×dim)	Filter num	Filter size	Filter stride	Activation	Parameter num
B-Net								
1	Input	$1 \times (3000 \times 128)$	$1 \times (3000 \times 128)$	-	-	-	-	-
2	Reshape	1×384000	1×384000	-	-	-	-	-
3	Conv1D-BN	1×384000	32×47999	32	16	8	ReLU	$1 \times 32 \times 16$
4	Max-Pool	32×47999	32×5998	-	16	8	-	-
5	Conv1D-BN	32×5998	64×1498	64	8	4	ReLU	$32 \times 64 \times 8$
6	Conv1D-BN	64×1498	128×748	128	4	2	ReLU	$64 \times 128 \times 8$
7	Conv1D-BN	128×748	256×373	256	4	2	ReLU	$128 \times 256 \times 4$
8	Down-Sample	256×373	256×256	-	-	-	-	-
B-Sum	-	-	-	-	-	-	-	$344,576$
S-Net								
9	Conv1D-BN	256×256	64×61	64	16	4	ReLU	$256 \times 64 \times 16$
10	Reshape	64×61	3904	-	-	-	-	-
11	Linear	3904	512	-	-	-	-	3094×512
12	Linear	512	128	-	-	-	-	512×128
13	Multi-Linear	128	128×2	-	-	-	-	$128 \times (128 \times 2)$
S-Sum	-	-	-	-	-	-	-	$1,682,432$
P-Net								
14	Seg-Pool	256×256	256×16	-	-	-	-	-
15	Conv1D-BN	256×16	128×13	128	4	1	-	$256 \times 128 \times 4$
16	Reshape	128×13	1664	-	-	-	-	-
17	Linear	1664	512	-	-	-	-	1664×512
18	Linear	512	128	-	-	-	-	512×128
19	Multi-Linear	128	128×5	-	-	-	-	$128 \times (128 \times 5)$
P-Sum	-	-	-	-	-	-	-	$1,130,496$
Sum	-	-	-	-	-	-	-	$3,157,504$

IV. Experiment

A. Dataset and setup

We evaluated our model using a single EEG channel from two public datasets: Montreal Archive of Sleep Studies (MASS) [8] and Sleep-EDF [36] [37]. In the available MASS cohort 1, there were 200 PSG recordings from 200 healthy participants. Each recording contained 20 scalp-EEG, 2 EOG (left and right), 3 EMG and 1 ECG channels. These recordings were manually classified into one of the five sleep stages (W, N1, N2, N3 and REM) by a sleep expert according to the AASM standard [3]. We evaluated our model using F4-Cz channel and resampled it to 100 Hz using polyphase filtering with automatically derived FIR filters [29]. There were movement artifacts at the beginning and the end of each participant's recordings that were labeled as UNKNOWN and excluded. Sleep-EDF dataset included the Sleep Cassette set and Sleep Telemetry set. We used the first part which contained 20 participants aged 25-34 aiming at studying the age effects on sleep in healthy participants with a total of 39 nights PSG signals. Each PSG recording contained 2 scalp-EEG signals from Fpz-Cz and Pz-Oz channels, 1 EOG (horizontal), 1 EMG, and 1 oro-nasal respiration signal. All EEG and EOG had the same sampling rate of 100 Hz. These recordings were manually scored by well-trained technicians according to the 1968 Rechtschaffen and Kales manual [38]. We evaluated our model using the EEG-Fpz-Cz channel without any further preprocessing. We merged the S3 and S4 stages into a single stage N3 and excluded MOVEMENT and UNKNOWN stage to keep the same AASM standard as in the MASS dataset. Following the settings in [18], we included 30 minutes before and after the sleep periods.

Table II: Model investigation results on MASS SS1 set

Model choice	Accuracy	Model choice	Accuracy
Baseline	0.80	B2-SP2-AT1	0.81
B1-SP1-AT2	0.81	B2-SP2-AT2	0.85
B1-SP1-AT3	0.83	B2-SP2-AT3	0.86
B1-SP1-AT4	0.83	B2-SP2-AT4	0.88
B1-SP2-AT1	0.78	B3-SP1-AT1	0.76
B1-SP2-AT2	0.80	B3-SP1-AT2	0.77
B1-SP2-AT3	0.81	B3-SP1-AT3	0.80
B1-SP2-AT4	0.82	B3-SP1-AT4	0.81
B2-SP1-AT1	0.85	B3-SP2-AT1	0.79
B2-SP1-AT2	0.88	B3-SP2-AT2	0.81
B2-SP1-AT3	0.89	B3-SP2-AT3	0.81
B2-SP1-AT4	0.91	B3-SP2-AT4	0.83

We followed the common train-val-test settings used in [7]. For MASS, We performed 20-fold cross validation. In addition, we use the participant-independent schemes to randomly divide the training set and test set. At each iteration, 200 participants were split into training, validation, and test set with 180, 10, and 10 participants. For Sleep-EDF, we conducted leave-one-participant-out cross validation for all 20 participants. At each iteration, 19 training participants were divided into 15 participants for training and 4 participants for validation. We evaluated the performance of our SegNet using overall accuracy, macro-averaging F1-score (F1), Cohen's Kappa coefficient [39], sensitivity, selectivity, and also class-wise sensitivity and selectivity.

We used the following details to train SegNet: In the first step, we employed SGD optimization algorithm with $1e^{-4}$ learning rate and other default settings in pytorch. $(50, 0.5)$ StepLR was utilized as learning rate scheduler. In the second step, we used Adam optimization algorithm with $1e^{-5}$ learning rate and other default settings in pytorch. Learning rate

Table III: Model comparison of state-of-the-art networks for automated sleep staging

Model	#Input epoch	Network input shape	#total parameters
DeepSleepNet [18]	1	ch$\times fs(200) \times 30$	46,371,589
SeqSleepNet [5]	{10,20,30}	ch$\times fs(100) \times 30 \times \{10, 20, 30\} \rightarrow (29 \times 129) \times \{10, 20, 30\}$	137,476 (20 epochs)
Utime [29]	35	ch$\times fs(100) \times 30 \times 35$	1,220,317
SegNet	128	ch$\times fs(100) \times 30 \times 128$	3,674,464

scheduler was equipped with $(100, 0.1)$ StepLR. Both steps are stopped when no improvement occurs on validation set. In practice, we run $(200, 500)$ epochs for two steps on MASS dataset, and $(100, 300)$ on Sleep-EDF dataset. The total run time for training a SegNet on MASS dataset is around 11 hours on a GeForce GTX 1080Ti with 11GB GPU memory, where the first step takes around 5 hours and the second step takes 13 hours.

B. Model exploration

We evaluated a few common settings and determined the best ones for our model according to overall accuracy. Then we applied these settings to report the best results of our model and compared with other methods. These experiments were conducted on MASS SS1 subset which contains 53 participants. Keeping the main framework fixed, we experimented on a few system implementations as follows:

- **Baseline**. This setting is the same as described in Section III-D. The update of three sub-networks follows the descriptions in Section III-C. Two-step training is performed for only once.
- **B-Net optimization**. To investigate the behavior of three sub-networks, we mainly consider the update of B-Net since it acts as feature extractor of the sequential epochs which contributes to stage distribution prediction of S-Net as well as final label prediction of P-Net. We performed three settings including B1 which encourages SegNet to update B-Net with S-Net and P-Net in two-step training iterations, B2 which updates B-Net only in the first step training for all iterations, and B3 which updates B-Net only in the first training step in the first iteration.
- **Segment pooling layer.** We tested two manners for segment pooling layer. The first one named SP1 is implemented in Figure 2. The second choice is namedSP2 that applies two duplicated segment vectors from S-Net output on the output of B-Net feature maps, and performs max pooling in a standard way. Note that both manners are channel-wise, which means each feature map in B-Net output uses the same pooling manner.
- **Alternative training Iteration**. Conducting two-step alternative training described in Section III-C can possibly increase SegNet ability in both segmentation and prediction. In our experiments, we tested iterating two-step training for more times. We denote n times of alternative training as ATn.

Table II summarizes the result comparison of the different settings of SegNet described above. We can observe B2-SP1-AT4 achieves the best overall accuracy. In most cases, using the update manner of B2 and the pooling manner of SP1 perform better than the other choices. Also, performing more iterations of two-step training can increase the accuracy. In practice, negligible improvements were obtained after 4 times. Given the results in Table II, we used as the best settings of B2-SP1-AT4 for SegNet and report the performance on two datasets and comparison with other state-of-the-art methods in the followings.

Table IV: Confusion matrix of SegNet on MASS test set

	Predicted					Per-class Metrics		
	W	N1	N2	N3	REM	Sens.	Sel.	F1
W	5638	502	93	45	83	0.89	0.89	0.89
N1	220	2973	921	79	258	0.69	0.67	0.68
N2	129	519	23719	712	291	0.90	0.94	0.92
N3	102	82	827	5389	120	0.85	0.83	0.84
REM	181	419	523	59	7912	0.91	0.87	0.89

Table V: Confusion matrix of SegNet on Sleep-EDF test set

	Predicted					Per-class Metrics		
	W	N1	N2	N3	REM	Sens.	Sel.	F1
W	6238	502	124	172	138	0.87	0.87	0.87
N1	211	3373	629	257	358	0.70	0.69	0.71
N2	369	504	20719	512	591	0.90	0.91	0.90
N3	102	187	626	5889	46	0.85	0.86	0.85
REM	124	219	827	59	6912	0.85	0.84	0.85

C. Results and comparison

We mainly compare with three state-of-the-art deep learning based methods, i.e. DeepSleepNet [18], SeqSleepNet [5] and Utime [29]. We also compare other competitive results including a simple shallow CNN [27], a multi-task learning method [7], a time distributed multivariate network [20], a RNN-based architecture [42], a random forest model [40], a two-stage sequential learning approach [41], and a very recent graph-based convolutional network [21]. Since the original papers did not provide complete results for all the metrics, we wrote a pytorch-based model zoo for DeepSleepNet, SeqSleeqNet and Utime following the original implementations and is available at https://github.com/wangjinzhuo/wearables. The network shape and parameters of four main architectures are summarized in Table III.

Table IV and V show confusion matrices obtained from test sets sourced from two datasets. Each row and column represent the number of 30s epochs of each sleep stage provided by the sleep expert and predicted by our SegNet, respectively. The diagonals indicate the number of epochs that are correctly classified by our model. The last three columns in each row is the per-class performance metrics computed from the confusion matrix. We can see a quite balanced performance over F1 score for each class, which often suffers in many other methods due to data imbalance, especially for the N1 sleep stage. This advantage might be due to our long-range stage structure modeling that does not treat each epoch equally and consequently evades the severe data imbalance problem.

Table VI: The performance comparison of the state-of-the-art approaches on the MASS dataset

MASS	Overall metrics					Class-wise sensitivity					Class-wise selectivity				
	Acc.	Kappa	F1	Sens.	Sel.	W	N1	N2	N3	REM	W	N1	N2	N3	REM
DeepSleepNet (2017) [18]	0.81	0.73	0.76	0.75	0.95	0.80	0.52	0.86	0.69	0.91	0.88	0.46	0.85	0.85	0.79
SeqSleepNet (2019) [5]	0.87	0.81	0.83	0.82	0.96	0.89	0.60	0.91	0.80	0.94	0.91	0.65	0.89	0.84	0.91
Utime (2019) [29]	0.85	0.75	0.77	0.78	0.92	0.83	0.55	0.87	0.74	0.93	0.89	0.53	0.86	0.84	0.83
Chambon et al. (2019) [20]	0.73	0.64	0.67	-	-	-	-	-	-	-	-	-	-	-	-
Jiang et al. (2019) [40]	0.85	0.75	0.77	-	-	-	-	-	-	-	-	-	-	-	-
Sun et al. (2019) [41]	0.88	0.82	0.82	-	-	-	-	-	-	-	-	-	-	-	-
Jia et al. (2020) [21]	**0.89**	**0.83**	**0.84**	-	-	-	-	-	-	-	-	-	-	-	-
SegNet (ours)	0.87	0.82	0.85	**0.83**	**0.95**	0.90	0.62	0.89	0.82	0.93	0.91	0.66	0.89	0.85	0.91

Table VII: The performance comparison of the state-of-the-art approaches on Sleep-EDF dataset

Sleep-EDF	Overall metrics					Class-wise sensitivity					Class-wise selectivity				
	Acc.	Kappa	F1	Sens.	Sel.	W	N1	N2	N3	REM	W	N1	N2	N3	REM
O. Tsinalis. (2016) [27]	0.79	0.69	0.74	0.75	0.89	0.79	0.52	0.84	0.69	0.89	0.86	0.45	0.86	0.84	0.78
DeepSleepNet (2017) [18]	0.82	0.76	0.77	0.76	0.93	0.81	0.53	0.87	0.71	0.92	0.88	0.48	0.87	0.86	0.80
SeqSleepNet (2019) [5]	0.85	0.77	0.78	0.77	0.94	0.86	0.51	0.84	0.85	0.81	0.85	0.54	0.81	0.82	0.80
Utime (2019) [29]	0.86	0.79	0.78	0.79	0.92	0.87	0.52	0.86	0.87	0.85	0.85	0.55	0.85	0.85	0.81
SegNet (ours)	**0.88**	**0.80**	**0.81**	**0.83**	**0.94**	0.89	0.60	0.87	0.82	0.90	0.91	0.61	0.89	0.85	0.91

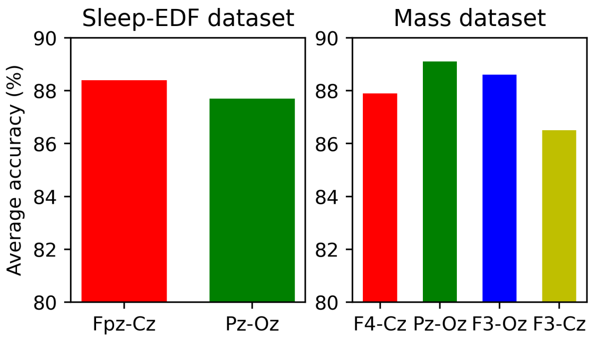

Figure 3: Performance comparison of using other EEG channels as input on MASS and Sleep-EDF datasets.

The average F1 scores for all the class on MASS dataset and Sleep-EDF dataset are 84.6 and 84.1, respectively.

Table VI and VII demonstrate a thorough comparison of our method and other automatic sleep scoring methods across different metrics on two datasets. The results show that our SegNet outperforms most of other methods. In particular, SegNet achieved 0.87 and 0.88 overall accuracy on MASS and Sleep-EDF dataset, respectively. Notably, a very recent published paper [21] constructed a novel deep graph neural network called GraphSleepNet and obtained the best overall accuracy. However, GraphSleepNet employed all the signals including EEG, EOG, EMG and ECG, while our SegNet and other methods used only one channel EEG signal.

D. Generalization to other input channels

We also examined the generalization ability of SegNet by applying three other EEG channels (Pz-Oz, F3-Oz, F3-Cz) from the MASS dataset and Pz-Oz from the Sleep-EDF dataset, in comparison with the previously used channel. The results are shown in Figure 3. From the figure, we can observe SegNet is generalized well on other EEG channels. In particular, on MASS dataset, the overall accuracy is is 0.5% higher than that of the original one used in previous experiments. On the Sleep-EDF dataset, the performance of Pz-Oz is very close to that of using Fpz-Cz in terms of most metrics. We also test multiple EEG channel inputs and observe no clear evidence on stage scoring performance improvement. However, adding EOG and EMG signals improved accuracy, echoing similar results to GraphSleepNet [21].

In Figure 4, we demonstrate the quality of binary stage distribution generated by S-Net and the ground truth on MASS dataset. This figure is designed to to examine and validate how SegNet can predict state transitions from segments with consecutive sleep stages. From three chosen sequences of 100 epochs, we can see S-Net is able to segment the sequential epochs and predict transition points quite well compared with the ground truth. In test case of two datasets, we collected the output of S-Net and compared them with binary ground truth, and found the accuracy is similar to that of overall accuracy. This indicates SegNet can learn reasonable transition points.

E. Long-range hypnogram analysis

To show the effectiveness of SegNet in long-range stage structure modeling, we output the predicted stage from SegNet and compare it with ground truth. Figure 5 and 6 show the output hypnogram and the posterior probability distribution per stage of SegNet on both the MASS and Sleep-EDF datasets. It can be seen that the output hypnogram aligns very well with the corresponding ground truth. In many periods, SegNet makes nearly perfect prediction. This is because SegNet approaches automated sleep staging as a state-transistion detection problem, by first performing segmentation on sequential epochs with similar characteristics (estimated similar sleep stage), followed by the final sleep stage prediction.

V. Conclusion

We propose a novel framework SegNet to perform automatic sleep stage classification based on raw single-channel EEG. Instead of the conventional way in most popular methods of treating each sleep epoch as an individual training data, we consider long-range sleep structure and learn temporal dependencies for sequences as long as 128 epochs. We incorporate

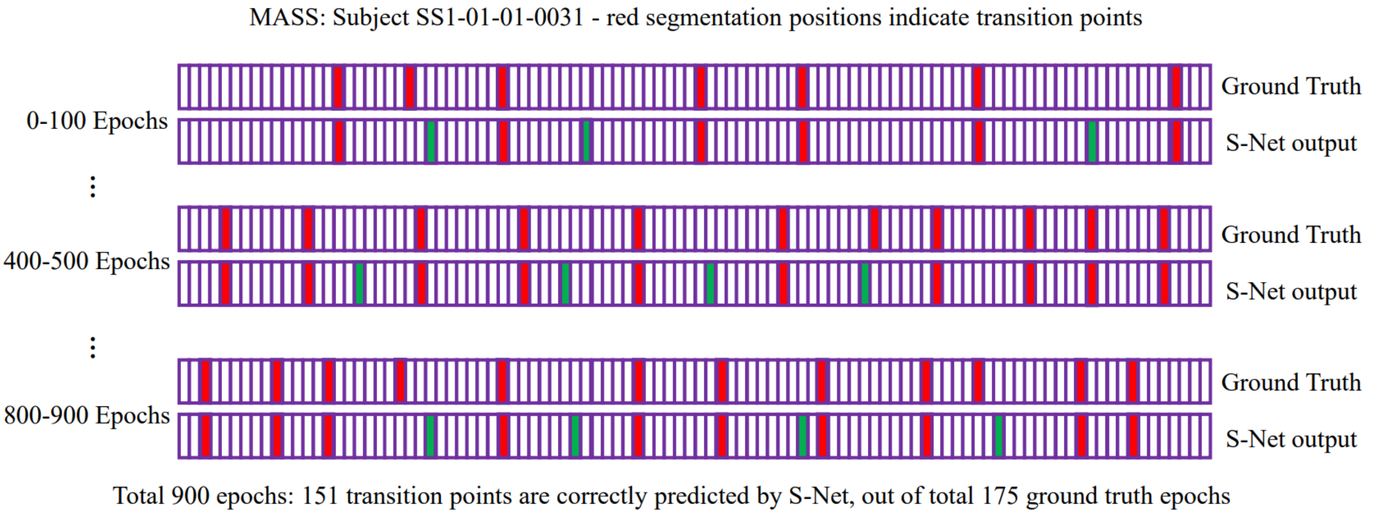
MASS: Subject SS1-01-01-0031 - red segmentation positions indicate transition points
0-100 Epochs
400-500 Epochs
800-900 Epochs
Ground Truth
S-Net output
Ground Truth
S-Net output
Ground Truth
S-Net output
Total 900 epochs: 151 transition points are correctly predicted by S-Net, out of total 175 ground truth epochs

Figure 4: Comparison of S-Net output and the segmentation ground truth on one participant of MASS dataset.

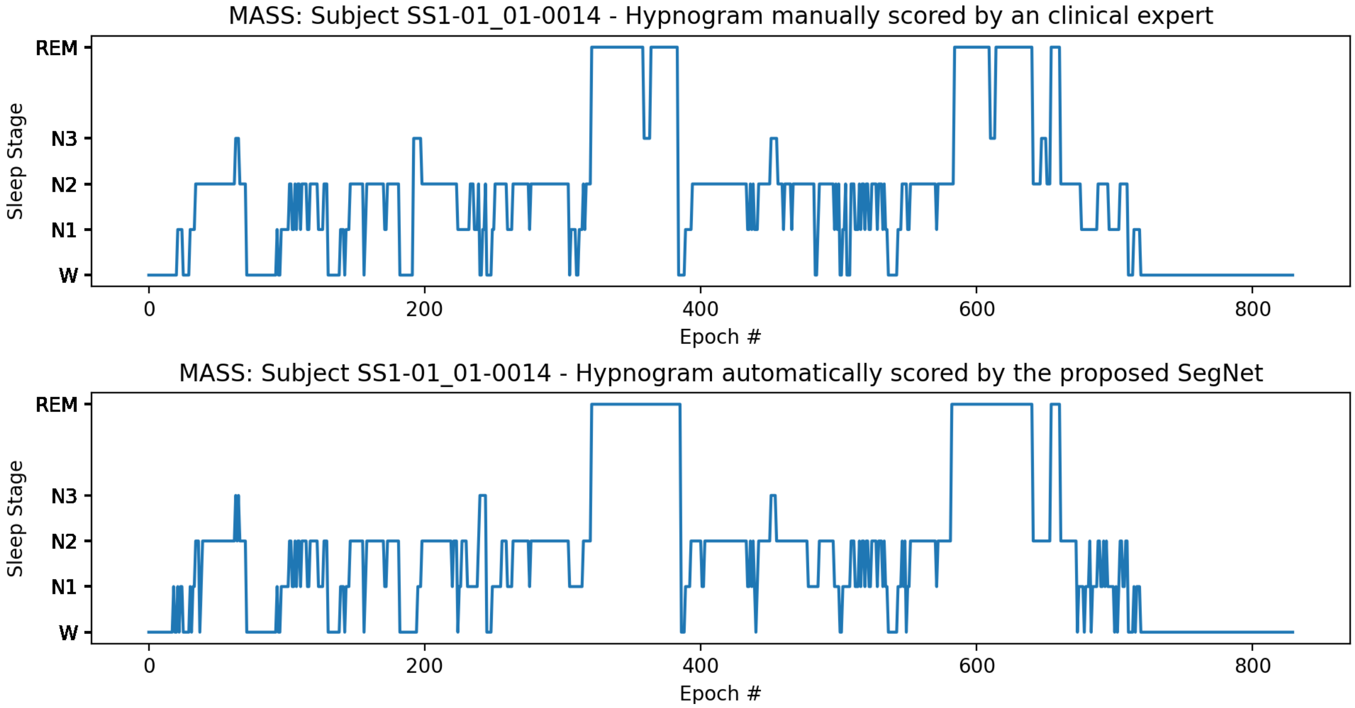

Figure 5: Hypnogram comparison of the proposed SegNet and the ground truth on one participant of MASS dataset.

the learning of sleep stage structure and label prediction in a unified framework with a carefully designed segment pooling layer, enabling training over long epoch sequences under a reasonable budget in both time and computing resources. The proposed SegNet demonstrates the ability to produce promising classification accuracy and high-quality long-range sleep structure on public datasets. We believe SegNet can be a positive contribution in automated sleep scoring through the learning and utilisation of sleep stage transition rules.

References

[1] D. Riemann, C. Baglioni, C. Bassetti, B. Bjorvatn, L. Dolenc Groselj, J. G. Ellis, C. A. Espie, D. Garcia-Borreguero, M. Gjerstad, M. Gonçalves *et al.*, “European guideline for the diagnosis and treatment of insomnia,” *Journal of sleep research*, vol. 26, no. 6, pp. 675–700, 2017.

[2] R. S. Rosenberg and S. Van Hout, “The american academy of sleep medicine inter-scorer reliability program: sleep stage scoring,” *Journal of clinical sleep medicine*, vol. 9, no. 01, pp. 81–87, 2013.

[3] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, B. V. Vaughn *et al.*, “The aasm manual for the scoring of sleep and associated events,” *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine*, vol. 176, p. 2012, 2012.

[4] B. Koley and D. Dey, “An ensemble system for automatic sleep stage classification using single channel eeg signal,” *Computers in biology and medicine*, vol. 42, no. 12, pp. 1186–1195, 2012.

[5] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, “Seqsleepnet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging,” *IEEE Transactions on Neural*

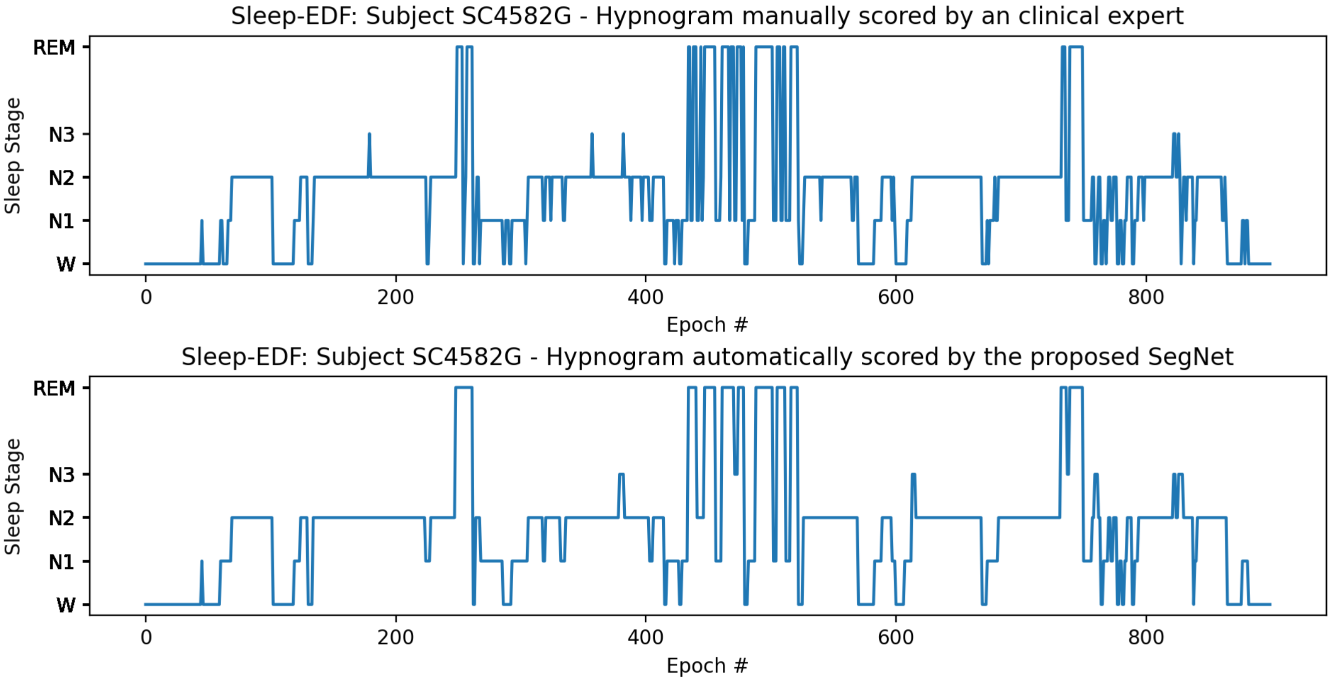

Figure 6: Hypnogram comparison of the proposed SegNet and the ground truth on one participant of Sleep-EDF dataset.

Systems and Rehabilitation Engineering, vol. 27, no. 3, pp. 400–410, 2019.

[6] O. Faust, H. Razaghi, R. Barika, E. J. Ciaccio, and U. R. Acharya, "A review of automated sleep stage scoring based on physiological signals for the new millennia," *Computer methods and programs in biomedicine*, 2019.

[7] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction cnn framework for automatic sleep stage classification," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1285–1296, 2018.

[8] C. O'reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research," *Journal of sleep research*, vol. 23, no. 6, pp. 628–635, 2014.

[9] L. Fiorillo, A. Puiatti, M. Papandrea, P. L. Ratti, P. Favaro, C. Roth, P. Bargiotas, C. Bassetti, and F. D. Faraci, "Automated sleep scoring: A review of the latest approaches," *Sleep medicine reviews*, 2019.

[10] C. Berthomier, V. Muto, C. Schmidt, G. Vandewalle, M. Jaspar, J. Devillers, G. Gaggioni, S. L. Chellappa, C. Meyer, C. Phillips *et al.*, "Exploring scoring methods for research studies: Accuracy and variability of visual and automated sleep scoring," *Journal of Sleep Research*, p. e12994, 2020.

[11] I. J. Rampil, "A primer for eeg signal processing in anesthesia," *The Journal of the American Society of Anesthesiologists*, vol. 89, no. 4, pp. 980–1002, 1998.

[12] S. Motamedi-Fakhr, M. Moshrefi-Torbati, M. Hill, C. M. Hill, and P. R. White, "Signal processing techniques applied to human sleep eeg signals—a review," *Biomedical Signal Processing and Control*, vol. 10, pp. 21–33, 2014.

[13] B. Şen, M. Peker, A. Çavuşoğlu, and F. V. Çelebi, "A comparative study on classification of sleep stage based on eeg signals using feature selection and classification algorithms," *Journal of medical systems*, vol. 38, no. 3, p. 18, 2014.

[14] K. A. I. Aboalayon, M. Faezipour, W. S. Almuhammadi, and S. Moslehpour, "Sleep stage classification using eeg signal analysis: a comprehensive survey and new investigation," *Entropy*, vol. 18, no. 9, p. 272, 2016.

[15] T. Lan, "Feature extraction feature selection and dimensionality reduction techniques for brain computer interface," *Doctor of Philosophy in Electrical Engineering examined and approved thesis. Oregon Health & Science University, OHSU Digital Commons, Scholar Archive, Paper*, vol. 706, 2011.

[16] M. Ronzhina, O. Janoušek, J. Kolářová, M. Nováková, P. Honzík, and I. Provazník, "Sleep scoring using artificial neural networks," *Sleep medicine reviews*, vol. 16, no. 3, pp. 251–263, 2012.

[17] M. Radha, G. Garcia-Molina, M. Poel, and G. Tononi, "Comparison of feature and classifier algorithms for online automatic sleep staging based on a single eeg signal," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 1876–1880.

[18] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[19] A. Koushik, J. Amores, and P. Maes, "Real-time sleep staging using deep learning on a smartphone for a wearable eeg," *arXiv preprint arXiv:1811.10111*, 2018.

[20] S. Chambon, V. Thorey, P. J. Arnal, E. Mignot, and A. Gramfort, "Dosed: a deep learning approach to detect multiple sleep micro-events in eeg signal," *Journal of Neuroscience Methods*, vol. 321, pp. 64–78, 2019.

[21] Z. Jia, Y. Lin, J. Wang, R. Zhou, X. Ning, Y. He, and Y. Zhao, "Graphsleepnet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, 2020, pp. 1324–1330.

[22] A. H. Ansari, O. De Wel, K. Pillay, A. Dereymaeker, K. Jansen, S. Van Huffel, G. Naulaers, and M. De Vos, "A convolutional neural network outperforming state-of-the-art sleep staging algorithms for both preterm and term infants," *Journal of Neural Engineering*, vol. 17, no. 1, p. 016028, 2020.

[23] A. N. Olesen, P. Jennum, E. Mignot, and H. B. Sorensen, "Msed: a multi-modal sleep event detection model for clinical sleep analysis," *arXiv preprint arXiv:2101.02530*, 2021.

[24] F. Li, R. Yan, R. Mahini, L. Wei, Z. Wang, K. Mathiak, R. Liu, and F. Cong, "End-to-end sleep staging using convolutional neural network in raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 63, p. 102203, 2021.

[25] A. Guillot and V. Thorey, "Robustsleepnet: Transfer learning for automated sleep staging at scale," *arXiv preprint arXiv:2101.02452*, 2021.

[26] M. Längkvist, L. Karlsson, and A. Loutfi, "Sleep stage classification using unsupervised feature learning," *Advances in Artificial Neural Systems*, vol. 2012, 2012.

[27] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel eeg using convolutional neural networks," *arXiv preprint arXiv:1610.01683*, 2016.

[28] N. Michielli, U. R. Acharya, and F. Molinari, "Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals," *Computers in biology and medicine*, vol. 106, pp. 71–81, 2019.

[29] M. Perslev, M. Jensen, S. Darkner, P. J. r. Jennum, and C. Igel, "U-time: A fully convolutional network for time series segmentation applied to sleep staging," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 4415–4426.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[31] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[32] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[34] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[36] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[37] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[38] J. A. Hobson, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *Electroencephalography and Clinical Neurophysiology*, vol. 26, no. 6, p. 644, 1969.

[39] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[40] D. Jiang, Y.-n. Lu, M. Yu, and W. Yuanyuan, "Robust sleep stage classification with single-channel eeg signals using multimodal decomposition and hmm-based refinement," *Expert Systems with Applications*, vol. 121, pp. 188–203, 2019.

[41] C. Sun, C. Chen, W. Li, J. Fan, and W. Chen, "A hierarchical neural network for sleep stage classification based on comprehensive feature learning and multi-flow sequence learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 5, pp. 1351–1366, 2019.

[42] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 324–333, 2017.