Sleep stage classification using long-range temporal structure

1st Someshwara Raju D
200580175
Huy Phan
MSc in AI

***Abstract*—Recent developments in the automatic classification of sleep stages point to potential benefits of mining temporal dependencies over subsequent epochs. Short-range sequential learning is still a limitation of current solutions. In its place, this research introduces a system called SegNet that can convert long range epochs into stage labels. SegNet uses segment pooling to combine the temporal region data to the long-range epoch sequence to generate core feature map. This feature map is further used to predict stage labels using 2-step training process. The model's ability to perform was tested using the Sleep-EDF public database.**

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

I. Introduction

Ever wondered why you feel fresh and relaxed after a good night's sleep? When you sleep, your body goes through a series of changes that allow you to get the rest you need for your general health. Sleep allows the brain and body to settle down and engage in restorative processes, enabling greater physical and mental performance the next day and over time. What happens during sleep, including how different stages of sleep unfold, highlights the complexities of sleep and its importance to our well-being (Ramar et al. 2021).

Sleep disorders like insomnia, restless legs syndrome, narcolepsy, and sleep apnea, which affect many people today, can have a negative impact on every aspect of your life, including your safety, relationships, academic and professional performance, thinking, mental health, weight, and the onset of diabetes and heart disease. Typically, polysomnography (PSG) equipment, which records several physiological signals, is used to measure sleep quality to diagnose these sleep disorders. The distribution of various sleep stages is a crucial feature to consider when assessing the quality of your sleep. Clinical sleep scoring takes time and is prone to human mistakes (Rosenberg & Van Hout 2013). The ability of automatic sleep stage classification to exceed manual scoring (Berthomier et al. 2020) makes it a viable option for generating accurate and consistent sleep stage classification findings.

Before understanding automatic sleep stage classification, let's understand what sleep stages are. Sleep stages progress cyclically from stage N1 through stage R, then start over at stage N1. An entire sleep cycle lasts between 90 and 110 minutes on average, with each stage lasting between 5 and 15 minutes. The early sleep cycles feature lengthy deep sleep intervals and very brief REM naps. The following list provides a summary of the sleep phases' characteristics based on AASM's(American Academy of Sleep Medicine) (Berry et al. 2012) scoring guidelines.

1) Stage W: EEG recording: beta waves have the highest frequency and the lowest amplitude (alpha waves appear when a person is calm and comfortable when awake).
   The wake stage, also known as stage W, is the initial phase and is dependent on whether the eyes are open or closed. Beta waves dominate when the eyes are open. Alpha waves become the dominating pattern as people get sleepy and close their eyes.
2) Stage N1: EEG recording: low voltage theta waves
   The beginning of this stage of sleep, which is the lightest one, occurs when more than half of the alpha waves are replaced by low-amplitude mixed-frequency (LAMF) activity. Skeletal muscles have tone, and breathing usually happens at a regular rate. This phase of sleep lasts for about one to five minutes and accounts for 5% of overall sleep time.
3) Stage N2: EEG recording: sleep spindles and K complexes
   This stage represents deeper sleep as your heart rate and body temperate drop. It is distinguished by the presence of K-complexes, sleep spindles, or both. Sleep spindles and K complexes appear to be key players in memory consolidation, according to numerous studies (Antony et al. 2019) (Gandhi MH 2022). Initially lasting around 25 minutes, this stage of sleep eventually accounts for about 4 per cent of all sleep. It gets longer with each subsequent cycle. The period of sleep during which bruxism (tooth grinding) takes place.
4) Stage N3: EEG recording: delta waves - lowest frequency, highest amplitude
   Another name for N3 is slow-wave sleep (SWS). The signals in this stage of sleep, known as delta waves, have substantially lower frequencies and higher amplitudes and are indicative of the deepest period of sleep. The hardest stage to awaken from is this one, and for some people, even really loud noises (over 100 dB) won't wake them up. During this phase, the body heals and regenerates tissues, develops bone and muscle, and fortifies the immune system.
5) Stage REM: EEG recording: beta waves, which resemble

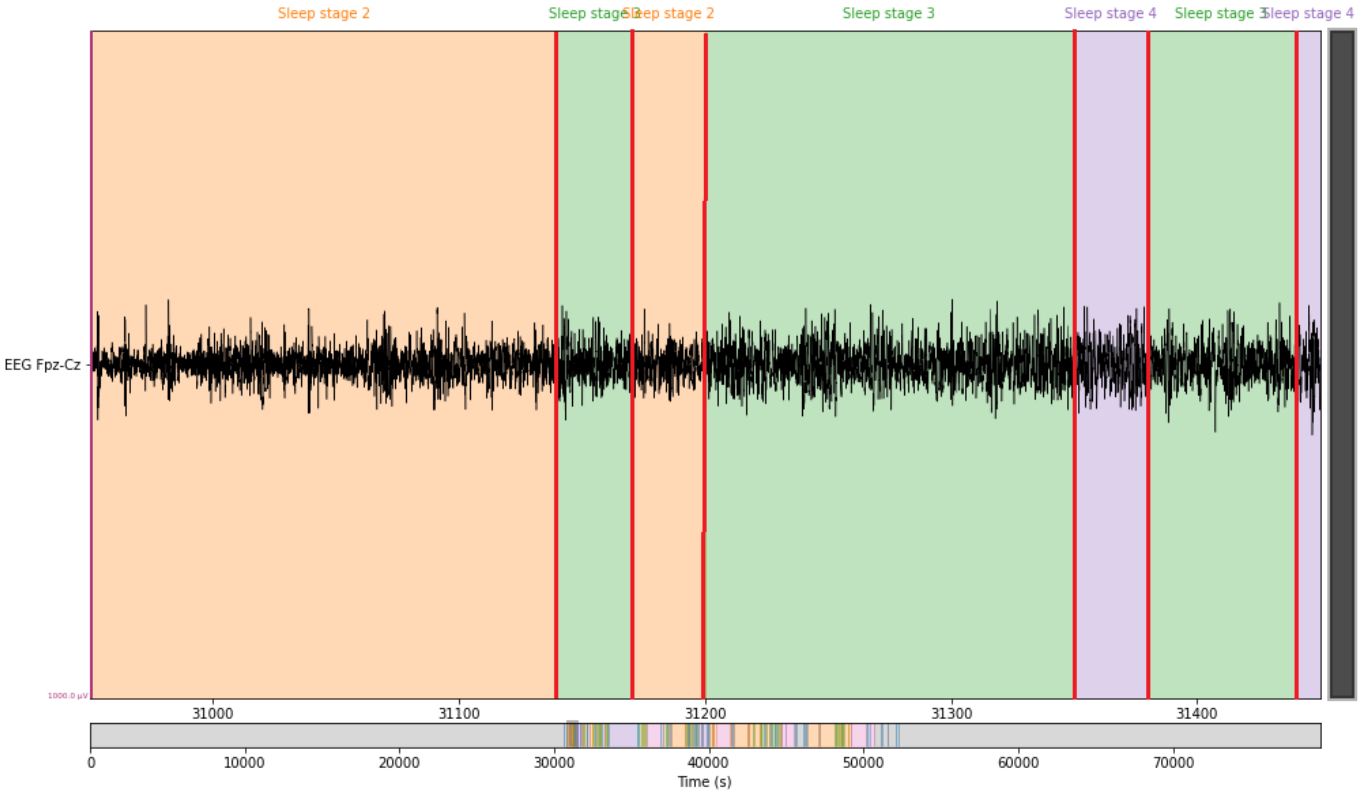

Fig. 1. A 500sec part of an EEG(Fpz-Cz) signal with sleep stage labels. Red lines denote the transition points.

brain waves during waking
REM is not regarded as a restful sleep stage because it is connected to dreaming. The skeletal muscles are atonic and immobile, with the exception of the eyes and diaphragmatic breathing muscles, which continue to be active, even if the EEG resembles that of an awake person. The breathing rate, however, starts to fluctuate more. This stage typically begins 90 minutes after you fall asleep, and it lasts all night long, with each REM cycle becoming longer. The first period usually lasts for ten minutes, and the last one can last for an hour. Dreaming, nightmares, and penile/clitoral tumescence all happen during REM.

In order to execute automatic stage classification tasks, researchers frequently crop the nightly PSG signals recorded by researchers into 30-second chunks known as epochs. End-to-end deep learning architectures have lately taken the lead in providing solutions to this task, building on early approaches that combined hand-crafted features with statistical models (Faust et al. 2019). According to the length of the input and output, recent work (Phan et al. 2019) describes four different technique types, including the many-to-one, many-to-one, and its proposed many-to-many paradigms. The published results highlight the benefits of mining temporal dependencies by pushing the deep network to learn the mapping from raw signals to stage labels in a sequence-to-sequence fashion. Existing research, however, frequently identifies temporal dependencies between very short-range epochs (Phan et al. 2019) (Phan et al. 2018), even though tens of such epochs frequently have the same label. A public large-scale dataset's statistics (O'reilly et al. 2014) show that more than 85% of epochs have the same label as their predecessors, which motivates us to characterise long-range dependencies. This paper aims to build a deep neural architecture that can capture long-range stage structure in order to achieve this.

A crucial finding in the context of long-range structural modelling is that only around 15% of epochs include a stage change known as a transition point. Due to this finding, I trained the network to generate a distribution of transition points over the long-range input sequences, use this distribution to generate structured segments, and finally predict the label for each segment using the structured segment distribution. Figure 1 shows one of the input EEG signals used to train the model. It depicts a 500sec signal showing the stage labels. The red line shows the transition points between the stages. The bar in the lower part of the figure shows the stages for the whole signal. We can see the Wake(grey) stage covers the major part and N1(blue) stage covers the least. Due to this, the high dimension of the feature maps and the over-fitting danger are significant issues when feeding long-range epochs to a network. In order to solve this problem, I implemented a segment pooling structure(inspired by ROI pooling in (Girshick 2015)) that is able to aggregate over the learned segments. By doing this, we can construct a concise

feature summarization for each segment that is ready for classifier layers to produce the prediction for the final stage. This segment pooling technique saves key information at a significantly lower cost, making it possible to train the network over lengthy epoch sequences within a realistic time and computer resource budget. To accomplish the aforementioned duties, a model was built called SegNet. The model was tested on the Sleep-EDF public dataset.

II. Related Work

This section reviews recent research on the task of automatically classifying sleep stages from the perspectives of two different deep learning architectures: pure end-to-end deep learning architectures based on raw PSG signals and traditional statistical models based on hand-crafted features. It will follow with the discussion of some new research pertinent to our network architecture and training methods. The most recent review publications (Sarkar et al. 2022) (Fiorillo et al. 2019) (Berthomier et al. 2020) are cited for surveys that are more in-depth.

Machine learning workflow follows steps such as data preprocessing, feature extraction, feature selection/dimensionality reduction, and classification. In the pre-processing stage, bias, noise, or artefacts can be found in the PSG raw signals. Finding the most relevant information is made possible by feature extraction, feature selection, and dimensionality reduction procedures. All the data are then sent to the classifier to identify the various stages of sleep in the final classification step. Feature extraction can be linear and non-linear, which can be divided into three primary categories: temporal domain methods, frequency domain methods, and hybrid temporal and frequency domain methods (Motamedi-Fakhr et al. 2014) (Aboalayon et al. 2016). These methods enable data representation in a suitable dimensional space while enhancing classifier performance. The most frequently used techniques in recent works are the wavelet transform in the time-frequency domain, non-parametric analysis in the frequency domain, and standard statistics in the time domain (Aboalayon et al. 2016). As for the classifier part, artificial neural networks (ANN) and random forest are two common choices (Aboalayon et al. 2016). Study (Radha et al. 2014) focused on online sleep staging utilising a single EEG channel in an effort to determine the best signal processing and classifier techniques. Several EEG-based sleep stage classification algorithms were published and compared in the thorough study of (Aboalayon et al. 2016), with accuracy ranging from 70 to 94% on different datasets.

Now we will see how deep learning has improved sleep stage detection. Recent research has tended to develop extensive deep networks based on raw PSG signals due of deep learning's excellent results (Supratak et al. 2017) (Jia et al. 2020) (Olesen et al. 2021) (Li et al. 2021) (Phan et al. 2020). A deep belief net was specifically developed in (Längkvist et al. 2012) to learn probabilistic representations from unprocessed PSG signals. In order to extract time-invariant information from a single raw EEG channel, convolutional neural networks were also used (Tsinalis et al. 2016). According to the literature's findings up till 2016, using deep learning on manually created features performed better than using it on unprocessed signals(Tsinalis et al. 2016). One explanation could be a lack of the analysis of temporal information between epochs that sleep experts frequently employ to identify the various stages of sleep. The number of epochs in a single dataset for sleep scoring is enormous, and the dataset contains a wide range of data. The variety of the patients and the collection of recorded signals may not be adequately described by the feature-based method. For this reason, a number of studies over the past few years have directly applied deep learning algorithms to unprocessed PSG signals. Recent research (Phan et al. 2019) that trained a sequence to sequence network with RNN block and attention block using up to 30 epochs as input demonstrated that deep learning based on raw PSG signals is capable of achieving state-of-the-art performance. Using single-channel EEG, an unique cascaded RNN model was created in (Michielli et al. 2019), attaining 86.7% accuracy over five stages on the sleep-EDF dataset. My suggested model first learns to identify temporal sections of input sequences rather than utilising RNN to understand temporal connections and then predicts semantic stage labels for segments. This will also help improve the model's performance by reducing the learning parameters.

One of the major pieces of evidence that helps sleep specialists identify the different stages of sleep is the examination of temporal connections among successive epochs; however, most work in the previous methods only explores one epoch as training data. As a result, we encourage our model to take into account long-range epochs by learning the temporal stage structure and using up to 128 epochs as input. We urge the network in the proposed SegNet to learn the stage label and the distribution of stage transitions within a coherent framework.

The network architecture and training schedule have characteristics in common with well-liked object detection systems (Zhao et al. 2019). The successful ROI pooling implementation in Fast RCNN (Girshick 2015) served as inspiration for the proposed segment pooling layer, which aims to achieve a compact feature summary over the vast size input. Similar to Faster RCNN, SegNet training uses an iterative 2-step training process (Ren et al. 2015). In our example, the segment and backbone networks are updated during the first learning stage and fixed during the second. The second stage is focused on learning the stages by updating the prediction network. More iterations of this two-step alternating training can be performed up until minimal gains are seen.

III. Method

The problem was approached as an object detection problem to solve the task at hand. The Computer Vision field has various novel network architectures that tackle this problem. This paper tackles the sleep staging task by considering it as a detection problem in the temporal domain. The knowledge of transition points assists the model to learn this temporal distribution. The model first tries to generate the structured

segment using transition points knowledge and then predicts the stage labels for all epochs at once. The following subsections describe the model in more detail.

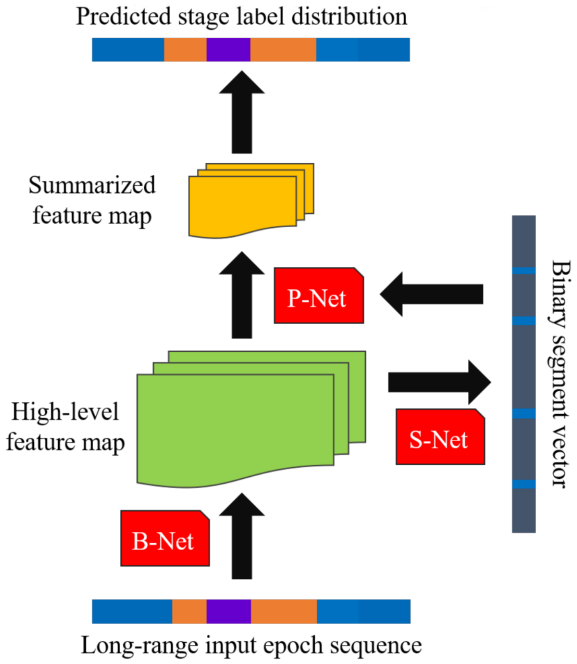

Fig. 2. Proposed network consisting of 3 subnetworks: B-Net, S-Net, and P-Net.

A. Framework Overview

A backbone network, a segment network, and a prediction network, also known as BNet, S-Net, and P-Net, respectively, are the three sub-networks that makeup SegNet. Figure 2 shows the proposed model architecture. Consecutive long-term epochs are sent into the B-Net, which then uses many convolutional and max pooling layers to create a core feature map. This feature map, which branches into the S-Net and P-Net, can be considered a fundamental high-level feature. S-Net, which produces a binary segment vector with the same size as the input epoch length, is also a series of convolutional layers followed by fully connected layers, as seen in the right branch of Figure 2. This binary segment vector has the estimation of the stage distribution. In simple terms, the segment vector will hold the information of transition points over the input sequence. The S-Net and B-Net hold the ground truth of transition points which helps in the training process. Next, the second branch is P-Net which takes inputs from B-Net and S-Net. In order to provide a concise summary of the main feature map, P-Net combines these two inputs with a segment pooling layer. This compact summary is then delivered to a few fully connected layers that connect to the ground truth to produce the final prediction.

B. Segment Pooling Layer

The major component in the SegNet is the segment pooling layer. This layer is present in the P-Net, which is linked to S-Net. The pooling layer receives both the output of B-Net (high-level feature map) and S-Net (transition points estimation). The aim of this layer is to combine these two inputs and summarize the input features according to the learned stage distribution. In turn, it also reduces the size of the feature maps.

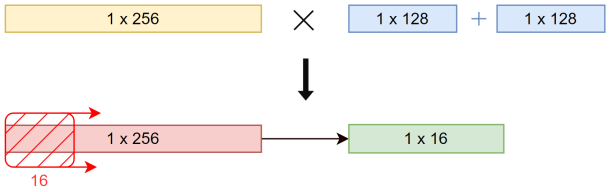

Fig. 3. Segment pooling layer. Each feature map channel (yellow box) is combined with two segment vectors concatenated with each other (blue boxes), creating a 1x256 vector(red box). Max pool (size 16) operation is performed on it, to get a 1 x 16 vector (green box).

It works by dividing each channel of the feature map into 1/16 sub-window and then applying the segment vector to it. In our case, B-Net outputs a feature map of size 256 x 256. That is, each channel size is 256, and the segment vector is of 1 x 128 size. The segment vector is concatenated with itself to make it of size 1 x 256. This vector is then combined with each channel. The output is a 256 x 256 feature map. Then the max pool operation is applied on this feature map with kernel size 16. This process divides the feature map into 256/16 bins and applies max pool operation in each bin. The output of this operation gives a vector of size 256 x 16. Each channel size got reduced by 1/16, representing the distribution of 15% epochs having different labels than previous ones. However, because the basic feature map for the proposed model is planned to have a high dimension (e.g., 256 x 256 in our implementation), the pooling operation lowers the risk of over-fitting for P-Net training because of the dimension reduction.

C. Two-Step training

The training was performed in a 2-step manner, similar to the 2-step training strategy used in Faster-RCNN (Ren et al. 2015). Input to the model in the training stage is a long-range epoch sequence. It was generated with a moderate size stride(64) to obtain sufficient training data. Figure 4 shows this process. A 128 sequence length of epochs was generated with a stride of 64. Let's take an example to explain this. Assume the input signal array as-10x1x3000, and we have to generate sequences of 5 epochs with a stride of 2. Like pooling operation, here we will capture 5 epoch sequence, move 2 strides, and then capture the next 5 epoch sequence. We follow this until the end of the sequence. For our example, this generates 3 sequences of length 5x1x3000.

The first step in the 2-step training is used to learn the transition point distribution. It is done by training the backbone and segment network according to the ground truth transition point distribution. In the second step, P-Net is trained while keeping the backbone(B-Net) and segment (S-Net) networks

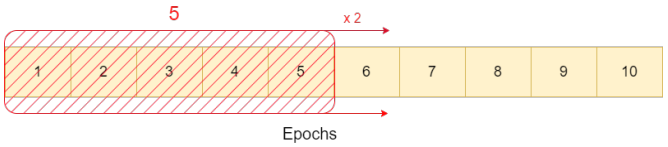

Fig. 4. Long-range epoch generation

fixed. P-Net is trained according to the true labelled data with sleep stages. This 2-step training can be performed multiple times until the results become stagnant. Generalised dice loss (Sudre et al. 2017) was used in both the steps as a loss function. Optimization was carried out by minimising this loss function between predicted sequential vectors and the actual data. Given issues with data imbalance like sleep staging, this cost function is recommended (Perslev et al. 2019). In multiple sleep staging research, it was observed that cross-entropy and Dice loss was majorly used. So, the combination of both, DiceBCE loss is also a better option for a segmentation task, explored in this paper (Rajput n.d.). It was also explored as a loss function in the training process.

D. Model specification

The model receives a 128 30s-epoch sequence of a single EEG channel as input leading to a size of 1 × (128 × 30 × fs). The sequence length 128 was chosen after multiple trials and tests. EEG signal in the sleep-edf dataset was sampled at fs=100Hz. The input shape can be read as:

- 1: number of input channels
- 128 x 30 x fs: feature dimension, where 128 is the sequence size and 30 is seconds of input per epoch.

This input is taken by B-Net, which consists of sequential convolution1D, batch norm and relu layers. Multiple convolutional layers help in extracting the best features from the input. These features are then passed to S-Net, which consists of sequential Conv1D, batch norm, relu and linear layers. This is followed by a custom multi-linear layer (MLL). MLL is responsible for taking a flattened array (output from linear layer) and producing an output of size (sequence length, channel number, number of classes). The output of B-Net and S-Net is combined by the segment pooling layer and passed to P-Net. P-Net consists of two convolutional layer followed by multiple linear layers. Each conv1d layer was followed by batchnormilazation and ReLu activation layer, similar to S-Net. The output of linear layer is passed to the MLL layer, which outputs an array of size (1, 128, 5), where 5 represents five stages of sleep. Table I summarizes the model's architecture in detail.

IV. Dataset

The model was evaluated using a single EEG channel data from the Sleep-EDF public dataset (Goldberger et al. 2000) (Kemp et al. 2000). The sleep-edf database has 197 PolySomnoGraphic(PSG) whole-night sleep recordings with chin EMG, EEG, and EOG and event markers. They were manually scored by well-trained technicians according to the Rechtschaffen and Kales manual (Rechtschaffen 1968). The dataset contains sleep cassette(SC) and telemetry data. This paper uses the cassette part. SC data were acquired from a 1987–1991 research on the effects of age on sleep in 20 healthy Caucasians aged 25–101, without any sleep-related medication. Two PSGs lasting almost 20 hours each were captured during two successive day-night cycles at the subjects' residences. The subjects went about their daily business while sporting a customised cassette recorder in the shape of a Walkman. The EOG and EEG signals were each sampled at 100 Hz. This paper focuses on EEG data. Each EEG signal consists of two channels: Fpz-Cz and Pz-Oz. The model was trained and evaluated on the Fpz-oz channel. PSG files were annotated with seven sleep stages (Wake, N1, N2, N3, N4, REM, and Unknown). I merged the N3 and N4 stages into a single N3 stage and removed the Unknown stage to keep it consistent with the American Academy of Sleep Medicine (AASM) standard (Berry et al. 2012). This also helped reduce the number of classes for the model to predict which reduced model training parameters. Like other sleep staging datasets, the Sleep-EDF dataset also suffers from data imbalance. In particular, stage N2 makes up between 45% and 55% of the entire sleep time and contributes to most of the class. Stage N1 only accounts for 2 to 5% (Altevogt et al. 2006) of the total. Also, most data contain the Wake stage, and the subject was wearing the headset all day which also can be seen in the bottom bar in figure 2. This inspired me to only keep the wake periods of 30mins before and after sleep, as done in (Supratak et al. 2017).

V. Experiment

The 20 participants in the dataset were divided into 15, 4, and 1 representing train, validation, and test respectively. Several parameters were tried for the convolutional and linear layers for the model architecture. The model was evaluated using overall accuracy and macro-average F1 score. Below experiments were performed in Steps 1 and 2 of the training process. All the experiments were stopped once the validation accuracy did not increase after 10 epochs and was run for 2 iterations. Step 1 and Step 2 were run for 100 and 200 epochs, respectively.

- Step 1:
  Input sequence lengths ranging from 35 to 128 were tested. Learning rate was scheduled using StepLR with (step_size=50, gamma=0.1) parameters.
  - Learning rate: 1e-2 to 1e-7 was tested.
  - Batch Size: 1, 64, 128 batch sizes were tested. Each batch size was tested with all the learning rates, and the best performing parameters were kept.
  - Optimizers: Stochastic gradient descent (SGD) and Adam.
  - Loss: Generalized Dice Loss(GDL) and DiceBCE loss functions were used, both helps to tackle imbalance in the data.
- Step 2:
  Learning rate was scheduled using StepLR with

TABLE I
PROPOSED MODEL ARCHITECTURE

B-Net	Layer Type	Input (ch X dim)	Output (ch X dim)	Filter number	Filter Size	Stride	Activation
1	Input	1 x (3000 x 128)	1 x (3000 x 128)	-	-	-	-
2	Reshape	1 x 384000	1 x 384000	-	-	-	-
3	Conv1D-BN	1 x 384000	32 x 47999	32	16	8	ReLU
4	Max-Pool	32 x 47999	32 x 5998	-	16	8	-
5	Conv1D-BN	32 x 5998	64 x 1498	64	8	4	ReLU
6	Conv1D-BN	64 x 1498	128 × 748	128	4	2	ReLU
7	Conv1D-BN	128 × 748	256 x 373	256	4	2	ReLU
8	Down-sample	256 x 373	256 x 256	-	-	-	-
S-Net							
9	Conv1D-BN	256 x 256	64 x 61	64	16	4	ReLU
10	Reshape	64 x 61	3904	-	-	-	-
11	Linear	3904	512	-	-	-	-
12	Linear	512	128	-	-	-	-
13	Multi-Linear	128	128 x 2	-	-	-	-
P-Net							
14	Seg-Pool	256 x 256	256 x 16	-	-	-	-
15	Conv1D-BN	256 x 16	128 x 13	128	4	1	ReLU
16	Conv1D-BN	128 x 13	102 x 10	102	4	1	ReLU
17	Reshape	102 x 10	1020	-	-	-	-
18	Linear	1020	512	-	-	-	-
19	Dropout	512	512	-	-	-	-
20	Linear	512	312	-	-	-	-
21	Dropout	312	312	-	-	-	-
22	Linear	312	128	-	-	-	-
23	Multi-Linear	128	128 x 5	-	-	-	-

(step_size=100, gamma=0.1) parameters. All the hyper-parameters testing was performed similarly to Step 1.

The architecture of the segment pooling layer was also changed to check the update in accuracy. The change was done in the B-Net feature map output and segment vector. Feature map size was changed to output 128 x 128, and a single segment vector was used instead of concatenating as explained before. The results of the hyperparameter tuning and pooling layer changes are discussed in more detail in the results section.

VI. RESULTS

TABLE II
ACCURACY ACHIEVED FOR EACH BATCH SIZE AND LEARNING RATE IN STEP 1 USING SGD AS OPTIMISER. FORMAT:- COLUMNS: BATCH_SIZE, ROWS: LEARNING RATE, VALUES: ACCURACY (ACHIEVED AT WHICH EPOCH IN BRACKETS)

Step 1 (SGD)	1	64	128
1.00E-02	-	90.52 (100)	88.06 (80)
1.00E-03	-	72.88 (99)	-
1.00E-04	88.40 (100)	52.87 (54)	-
1.00E-05	-	51.41 (97)	-
1.00E-06	-	51.37 (24)	-
1.00E-07	-	46.08 (8)	-

TABLE III
ACCURACY ACHIEVED FOR EACH BATCH SIZE AND LEARNING RATE IN STEP 2 USING SGD AS OPTIMISER. FORMAT:- COLUMNS: BATCH_SIZE, ROWS: LEARNING RATE, VALUES: ACCURACY (ACHIEVED AT WHICH EPOCH IN BRACKETS)

Step 2 (SGD)	1	64	128
1.00E-02	-	21.08 (31)	-
1.00E-03	-	21.67 (90)	-
1.00E-04	22.68 (109)	21.74 (10)	-
1.00E-05	-	20.38 (5)	-
1.00E-06	-	23.10 (2)	-
1.00E-07	-	19.01 (3)	-

TABLE IV
ACCURACY ACHIEVED FOR EACH BATCH SIZE AND LEARNING RATE IN STEP 2 USING ADAM AS OPTIMISER. FORMAT:- COLUMNS: BATCH_SIZE, ROWS: LEARNING RATE, VALUES: ACCURACY (ACHIEVED AT WHICH EPOCH IN BRACKETS)

Step 2 (Adam)	1	64	128
1.00E-02	43.80 (8)	46 (45)	45.95 (10) (1)
1.00E-03	44.20 (9)	52 (87)	60.68 (70)
1.00E-04	45.06 (55)	35.23 (65)	39.07 (42)
1.00E-05	36.241% (92)	38.16 (100)	36 (99) (2nd)
1.00E-06	32.56 (59)	26.76 (116)	20.87 (114)
1.00E-07	-	21.28 (10)	-

To start with a positive note, the novel model is good at predicting transition points (Step 1) but performs poorly in predicting stage labels (Step 2), which was the actual task. The tables II, III, and IV show the best validation accuracies achieved at which epoch (in brackets) for each batch size and learning rate combination.

We can see that the model had no problem in Step 1. It gave very good results in predicting temporal structures. Adam was not tested for Step 1 as SGD was giving good results. Best accuracies for each batch size (1, 64, and 128) found were : 88.40%, 90.52%, 88.06% respectively. These accuracies were achieved in the first iteration of the training. In the following

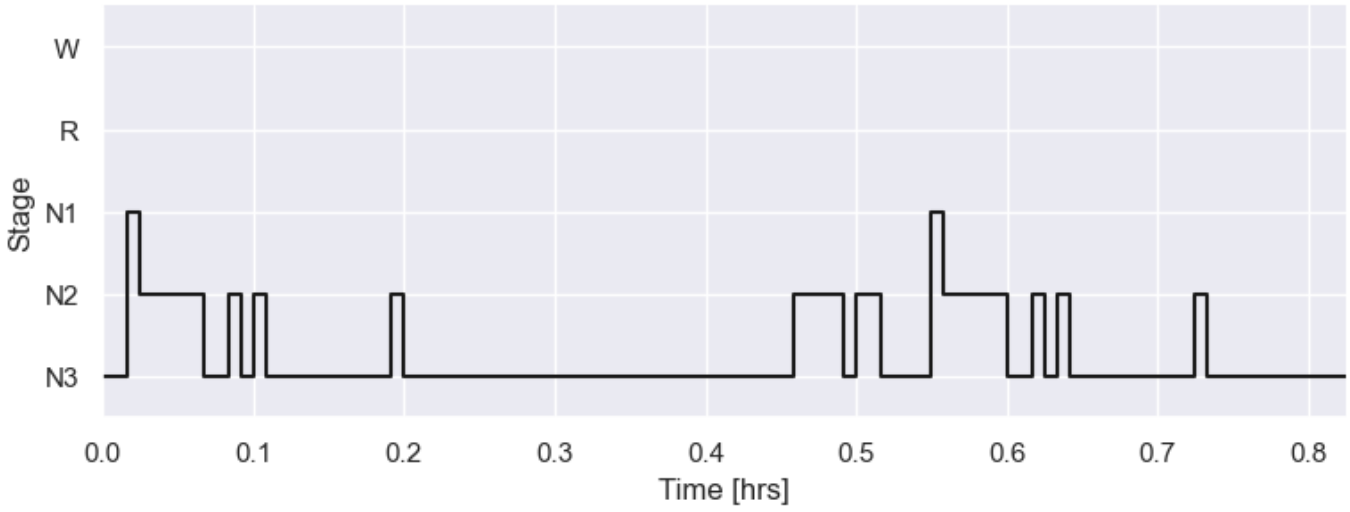

Fig. 5. Hypnogram of random sample of ground truth

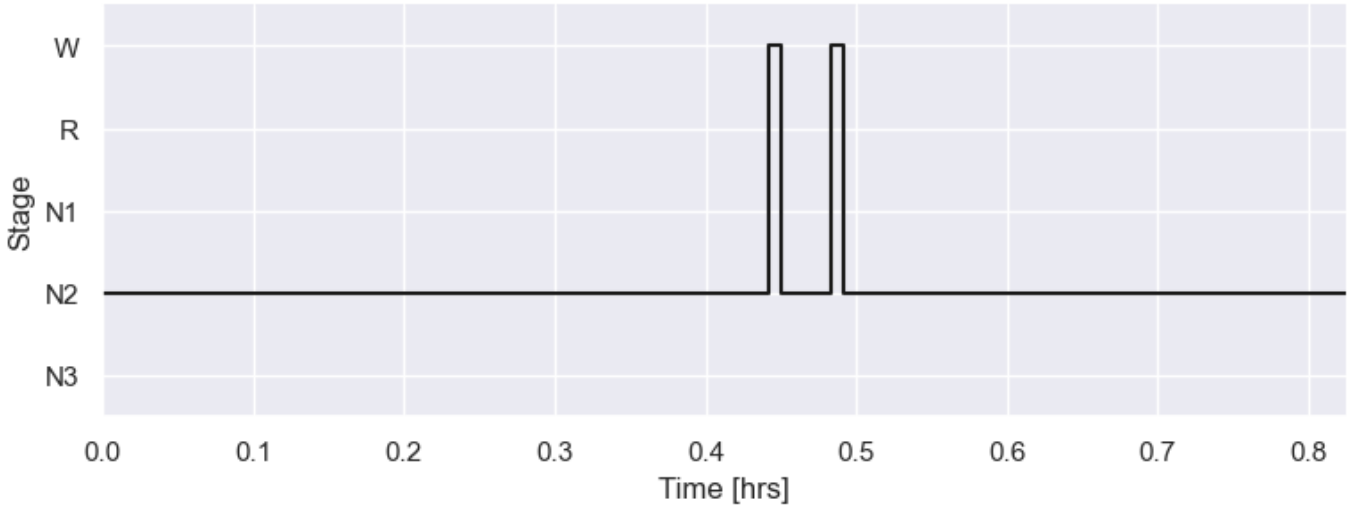

Fig. 6. Hypnogram of model prediction on the ground truth's random sample

iterations, accuracies didn't improve much (0 to 5%).

The challenge came in finding the best parameters for training Step 2 (P-Net). The best accuracy and F1 score found was 60.68% and 0.37, respectively, using these parameters: Batch size: 128, Learning rate: 1e-3. After each linear layer in P-Net, dropout layers were added and L2 regularization was used to tackle overfitting issues. By observing the experiment results (tables III & IV), we can see that Adam was performing better than SGD. Batch size 1 and 128 for Step2(SGD) was not explored more as batch size of 64 was giving poor results. A high learning rate was able to produce good accuracies in fewer epochs, but after reaching a point, the accuracies start to decrease. StepLR(100, 0.1) helped reduce this issue but still wasn't enough to achieve good results. On the other hand, a low learning rate produced slow accuracy increments. For example, batch size 128 and learning rate 1e-6 gave less than 1% increase after 100 epochs. P-Net wasn't giving good accuracies even after adding more Convolution and Linear layers. Increasing batch size more than 128 was going out of GPU memory.

The best parameters found for Step 2 were again tested with the increased number of iterations (4). An increase in accuracy was found. The third iteration gave the best accuracy which didn't increase in the 4th.

After all these observations, the below parameters were found to be the best ones:

Batch Size: 128

- Step 1
  - Optimizer: SGD
  - Learning Rate: 1e-2
  - Step LR: 50, 0.1
- Step 2

- Optimizer: Adam
- Learning Rate: 1e-3
- Step LR: 100, 0.1

All the accuracies were obtained using DiceBCE loss as it gave better results than Generalized Dice Loss. It was observed that the accuracy was reaching a point and then started reducing, showcasing a sign of overfitting. But, adding more dropout layers and increasing regularization didn't solve the issue. Due to the early stopping criteria, the model training was stopped after 10 epochs if no improvement was seen in the validation accuracy.

After getting the best model parameters, the model was tested on the held-out test data. Also, "EEG Pz-Oz'' channel was used to test for generalization of the model. Held-out test data got accuracy and F1 score of 42% and 0.3, respectively. Testing on the 'Pz-Oz' channel got 31% accuracy and 0.17 F1 score. This confirms the overfitting issue faced by the model. Figure 7 shows the classification matrix of the predictions made on the test data using the best parameters. We can see that the N1 stage is getting wrongly classified a lot. It was expected as N1 data points is very less compared to others.

Figures 5 & 6 shows the ability of the model to predict sleep stages. This further states the issue with the model. It predicts the majority of classes (Wake and N2) most of the time, which is the main reason for the bad performance.

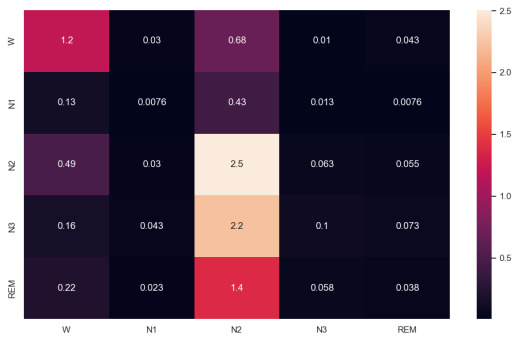

Fig. 7. Confusion matrix achieved at Step 2.

VII. Conclusion and Future Work

A novel architecture was proposed to perform automatic sleep stage classification. It was specially built to learn from the temporal structures observed in EEG signals. The proposed model's input consisted of a long-range sleep structure of 128 epochs which was different from the conventional single epoch training done today. The model consisted of three sub-networks: B-Net, S-Net, and P-Net. B-Net and S-Net learn temporal structures combined by the B-Net's features map by segment pooling layer producing a compact feature map. This is taken by P-Net to predict stage labels. The model produces good results in predicting the temporal structures but fails to give promising results for stage labels. The idea of 2-step training and segment pooling can be further researched to obtain good results in sleep scoring. As we know, manual sleep scoring focuses on temporal structures found in the sleep EEG signal. This concept is very useful for the task of automatic sleep scoring. Also, this method can be further tested on multi-channel inputs and other signals like EOG, ECG etc. Further experimentations on P-Net can be done by increasing the batch size and training data which wasn't possible on the current hardware.

References

Aboalayon, K. A. I., Faezipour, M., Almuhammadi, W. S. & Moslehpour, S. (2016), 'Sleep stage classification using eeg signal analysis: a comprehensive survey and new investigation', *Entropy* **18**(9), 272.

Altevogt, B. M., Colten, H. R. et al. (2006), 'Sleep disorders and sleep deprivation: an unmet public health problem'.

Antony, J. W., Schönauer, M., Staresina, B. P. & Cairney, S. A. (2019), 'Sleep spindles and memory reprocessing', *Trends in neurosciences* **42**(1), 1–3.

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

Berthomier, C., Muto, V., Schmidt, C., Vandewalle, G., Jaspar, M., Devillers, J., Gaggioni, G., Chellappa, S. L., Meyer, C., Phillips, C. et al. (2020), 'Exploring scoring methods for research studies: Accuracy and variability of visual and automated sleep scoring', *Journal of sleep research* **29**(5), e12994.

Faust, O., Razaghi, H., Barika, R., Ciaccio, E. J. & Acharya, U. R. (2019), 'A review of automated sleep stage scoring based on physiological signals for the new millennia', *Computer methods and programs in biomedicine* **176**, 81–91.

Fiorillo, L., Puiatti, A., Papandrea, M., Ratti, P.-L., Favaro, P., Roth, C., Bargiotas, P., Bassetti, C. L. & Faraci, F. D. (2019), 'Automated sleep scoring: A review of the latest approaches', *Sleep medicine reviews* **48**, 101204.

Gandhi MH, E. P. (2022), 'Physiology, k complex', *National Center for Biotechnology Information* .
URL: *https://pubmed.ncbi.nlm.nih.gov/32491401/*

Girshick, R. (2015), Fast r-cnn, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 1440–1448.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K. & Stanley, H. E. (2000), 'Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals', *circulation* **101**(23), e215–e220.

Jia, Z., Lin, Y., Wang, J., Zhou, R., Ning, X., He, Y. & Zhao, Y. (2020), Graphsleepnet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification., *in* 'IJCAI', pp. 1324–1330.

Kemp, B., Zwinderman, A. H., Tuk, B., Kamphuisen, H. A. & Oberye, J. J. (2000), 'Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg', *IEEE Transactions on Biomedical Engineering* **47**(9), 1185–1194.

Längkvist, M., Karlsson, L. & Loutfi, A. (2012), 'Sleep stage classification using unsupervised feature learning', *Advances in Artificial Neural Systems* **2012**.

Li, F., Yan, R., Mahini, R., Wei, L., Wang, Z., Mathiak, K., Liu, R. & Cong, F. (2021), 'End-to-end sleep staging using convolutional neural network in raw single-channel eeg', *Biomedical Signal Processing and Control* **63**, 102203.

Michielli, N., Acharya, U. R. & Molinari, F. (2019), 'Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals', *Computers in biology and medicine* **106**, 71–81.

Motamedi-Fakhr, S., Moshrefi-Torbati, M., Hill, M., Hill, C. M. & White, P. R. (2014), 'Signal processing techniques applied to human sleep eeg signals—a review', *Biomedical Signal Processing and Control* **10**, 21–33.

Olesen, A. N., Jennum, P., Mignot, E. & Sorensen, H. B. (2021), 'Msed: a multi-modal sleep event detection model for clinical sleep analysis', *arXiv preprint arXiv:2101.02530* .

O'reilly, C., Gosselin, N., Carrier, J. & Nielsen, T. (2014), 'Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research', *Journal of sleep research* **23**(6), 628–635.

Perslev, M., Jensen, M., Darkner, S., Jennum, P. J. & Igel, C. (2019), 'U-time: A fully convolutional network for time series segmentation applied to sleep staging', *Advances in Neural Information Processing Systems* **32**.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y. & De Vos, M. (2018), 'Joint classification and prediction cnn framework for automatic sleep stage classification', *IEEE Transactions on Biomedical Engineering* **66**(5), 1285–1296.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y. & De Vos, M. (2019), 'Seqsleepnet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **27**(3), 400–410.

Phan, H., Mikkelsen, K., Chén, O. Y., Koch, P., Mertins, A., Kidmose, P. & De Vos, M. (2020), 'Personalized automatic sleep staging with single-night data: a pilot study with kullback–leibler divergence regularization', *Physiological measurement* **41**(6), 064004.

Radha, M., Garcia-Molina, G., Poel, M. & Tononi, G. (2014), 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, pp. 1876–1880.

Rajput, V. (n.d.), 'Robustness of different loss functions and their impact on network's learning'.

Ramar, K., Malhotra, R. K., Carden, K. A., Martin, J. L., Abbasi-Feinberg, F., Aurora, R. N., Kapur, V. K., Olson, E. J., Rosen, C. L., Rowley, J. A. et al. (2021), 'Sleep is essential to health: an american academy of sleep medicine position statement', *Journal of Clinical Sleep Medicine* **17**(10), 2115–2119.

Rechtschaffen, A. (1968), 'A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects', *Brain information service* .

Ren, S., He, K., Girshick, R. & Sun, J. (2015), 'Faster r-cnn: Towards real-time object detection with region proposal networks', *Advances in neural information processing systems* **28**.

Rosenberg, R. S. & Van Hout, S. (2013), 'The american academy of sleep medicine inter-scorer reliability program: sleep stage scoring', *Journal of clinical sleep medicine* **9**(1), 81–87.

Sarkar, D., Guha, D., Tarafdar, P., Sarkar, S., Ghosh, A. & Dey, D. (2022), 'A comprehensive evaluation of contemporary methods used for automatic sleep staging', *Biomedical Signal Processing and Control* **77**, 103819.
URL: *https://www.sciencedirect.com/science/article/pii/S174680942200*

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Jorge Cardoso, M. (2017), 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, pp. 240–248.

Supratak, A., Dong, H., Wu, C. & Guo, Y. (2017), 'Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **25**(11), 1998–2008.

Tsinalis, O., Matthews, P. M., Guo, Y. & Zafeiriou, S. (2016), 'Automatic sleep stage scoring with single-channel eeg using convolutional neural networks', *arXiv preprint arXiv:1610.01683* .

Zhao, Z.-Q., Zheng, P., Xu, S.-t. & Wu, X. (2019), 'Object detection with deep learning: A review', *IEEE transactions on neural networks and learning systems* **30**(11), 3212–3232.