EEG DATA SLEEP STAGING WITH DEEP NEURAL NETWORKS

Tam Phi, Drexel University, College of Engineering, thp29@drexel.edu

Anup Das, Drexel University, College of Engineering, ad3639@drexel.edu

Abstract: Sleep pattern is an important indicator of human mental and physical health performance. Accurate identification of sleep stages assists doctors in correct diagnosis of sleep disorders in patients. The traditional manual method of sleep scoring is time-consuming and reliant on clinical experience of experts. As a result, new methods of automated sleep staging based on machine learning techniques have been studied in last years to solve the problem that traditional method carries. In this paper, we take spectrogram images of EEG signals, recorded throughout 8 hours of patients' sleep, as input in a deep neural network. These features are extracted in the time domain.

I. Introduction

Sleep takes up a third of human lives span, bearing a direct relationship between sleep quality and human's mental and physical performance. Due to stress and machinery, disturbances of sleep have surged in modern times. Additionally, studies have shown that neurological and psychological disorders can disrupts normal sleep patterns. Not only do sleep disorders affect human's physical performance but they also have long-lasting negative impacts on one's cognitive ability such as attention, memory, and learning. As a result, accurate sleep scoring is crucial in giving patients the correct diagnosis and treatment. Visual method of scoring sleep stages are common practices in real world but it requires significant expertise and it is time-consuming. Previous literatures have suggested methods of automated sleep staging to help with this problem **[1].**

For the last three decades, the Rechtschaffen and Kales (R&K) manual is widely used as the foundation for sleep scoring. The manual breaks down sleep stages into: Non-Rapid Eye Movement (NREM), REM, and Movement Time (MT). Within NREM there are stage 1, stage 2, stage 3, and stage 4, and within REM there is stage R. MT does not classify as either sleep or wakefulness. While R&K has remained the standard for sleep staging, there are limitations. Therefore, the new standard by Academy of Sleep Medicine (AASM) with modified definition sleep stages turn popular. In the new standard, stage R and W is kept, stage 1-4 is changed to stage 1-3, merging stage 3 and 4 into 1 stage, and MT is disregarded **[6]**.

This paper presents a method of sleep staging using spectrogram and deep neural networks classifier.

II. Proposed Method

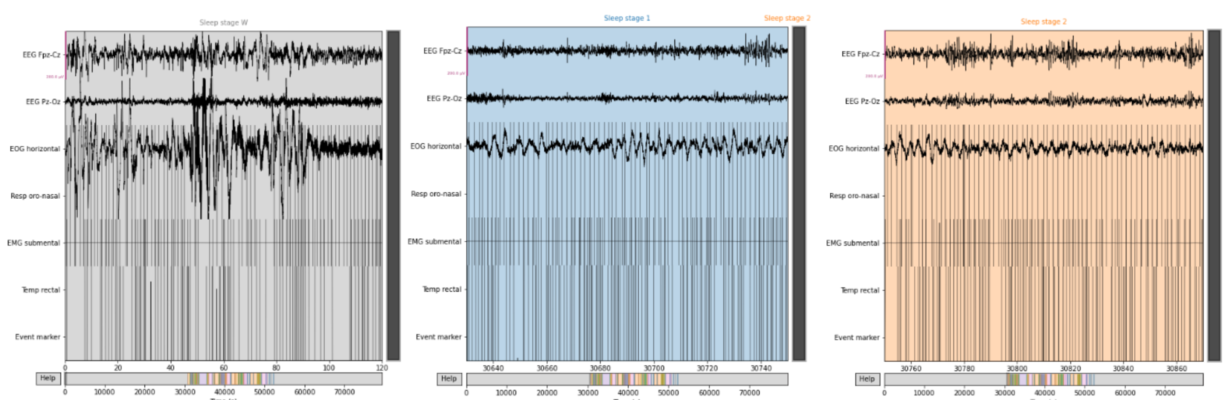
There are many approaches to sleep staging or sleep classification problem in adults. What are the input data (HOS-based, spectrogram wavelet, or time-series data, etc.)? Which learning model is used to classify them? In "A comparative review on sleep stage classification methods in patients and healthy individuals" by Boostani R., Karimzadeh F., and Nami M., compares 5 different methods and their accuracy achieved. According to the paper, the highest performance method uses Random Forrest classifier with input data feature from wavelet coefficients **[1]**. In this project, another method of features and classifier is proposed: Spectrogram images with CNN classifier. The project builds on the approach in "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring". Extraction of features is from the PhysioNet database for sleep signals, using 1 channel of EEG (Fpz-Cz) sampled at 100Hz to generate spectrogram images (time-domain extraction). Then these images will be fed into CNN model **[2]**. While the second paper score sleep stages according to the America Academy of Sleep Medicine (AASM) and employs a large scale, powerful model (VGG), the proposed method will use a simple, lightweight, and fast-to-compute one (MobileNet by Google).

1. Image creations:

The proposed method used the Sleep-EDF Database provided by PhysioNet [3]. A subset of EEG data from patients with affect sleep pattern is sampled at 100 Hz over a whole night sleep, approximately 20 hours. The signal is recorded in 30 seconds temporal window, or commonly referred to as epochs. These epochs are manually scored accordingly to the R&K standards by well-trained clinician. Class labels were created **[2]**.

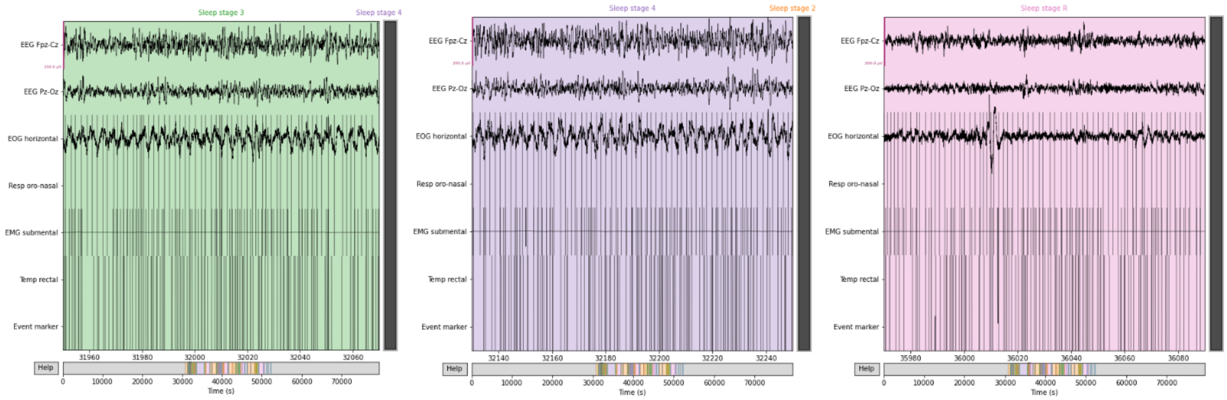
EEG data is presented as images in time-frequency domain. Fourier analysis is employed to perform spectral estimation on the raw signals, assuming a set of properties of the dataset: periodicity, stationarity, and infinite signals. These assumptions do not fully capture the characteristics of EEG signals, resulting in bias and high variance in slight change of frequencies. A common practice to reduce bias and variances is to use a processing method called *multi-taper spectral estimation* **[4]**. After applying the estimation on the dataset, values are converted to RGB values to be plotted as images in the size of 64x64 pixels. Images were created using one of two EEG channels: Fpz-Cz.

The raw data, spectrogram images, as well as the PSD plots can be observed below. The visualization of raw data and PSD plots are made possible with PhysioNet toolkit and library **[7].**



Stage W Stage 1 Stage 2

Figure 1.1. Raw EEG signal with Fpz-Cz and Pz-Oz channels. Each graph is visualized in three 30 seconds epochs to show 6 stages: Wake, REM, 1, 2, 3, 4 and color coded. The y-axis is in Hz, while the x-axis is in seconds (s). Because the signal is sampled at 100 Hz, for every 30 seconds, there are 3000 data points. X-axis also contains a bar that corresponds time to data points. These plots are acquired over sleeping period of the first patient in the database.



Stage 3 Stage 4 Stage R

Figure 1.2 Raw EEG signal with Fpz-Cz and Pz-Oz channels. Each graph is visualized in three 30 seconds epochs to show 6 stages: Wake, REM, 1, 2, 3, 4 and color coded. The y-axis is in Hz, while the x-axis is in seconds (s). Because the signal is sampled at 100 Hz, for every 30 seconds, there are 3000 data points. X-axis also contains a bar that corresponds time to data points. These plots are acquired over sleeping period of the first patient in the database.

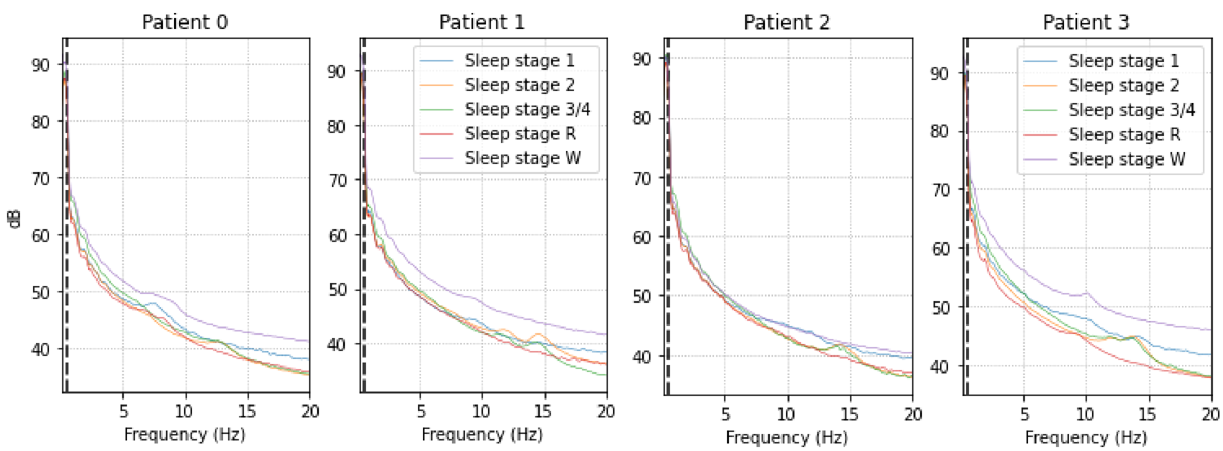


Figure 2. PSD plot of each sleep stage epochs that consists of the signature for each stage. These signatures possess a similar trend: as the Frequency (Hz) increases, the intensity (dB) decreases in a non-linear fashion. These plots are acquired from data of 4 patients.

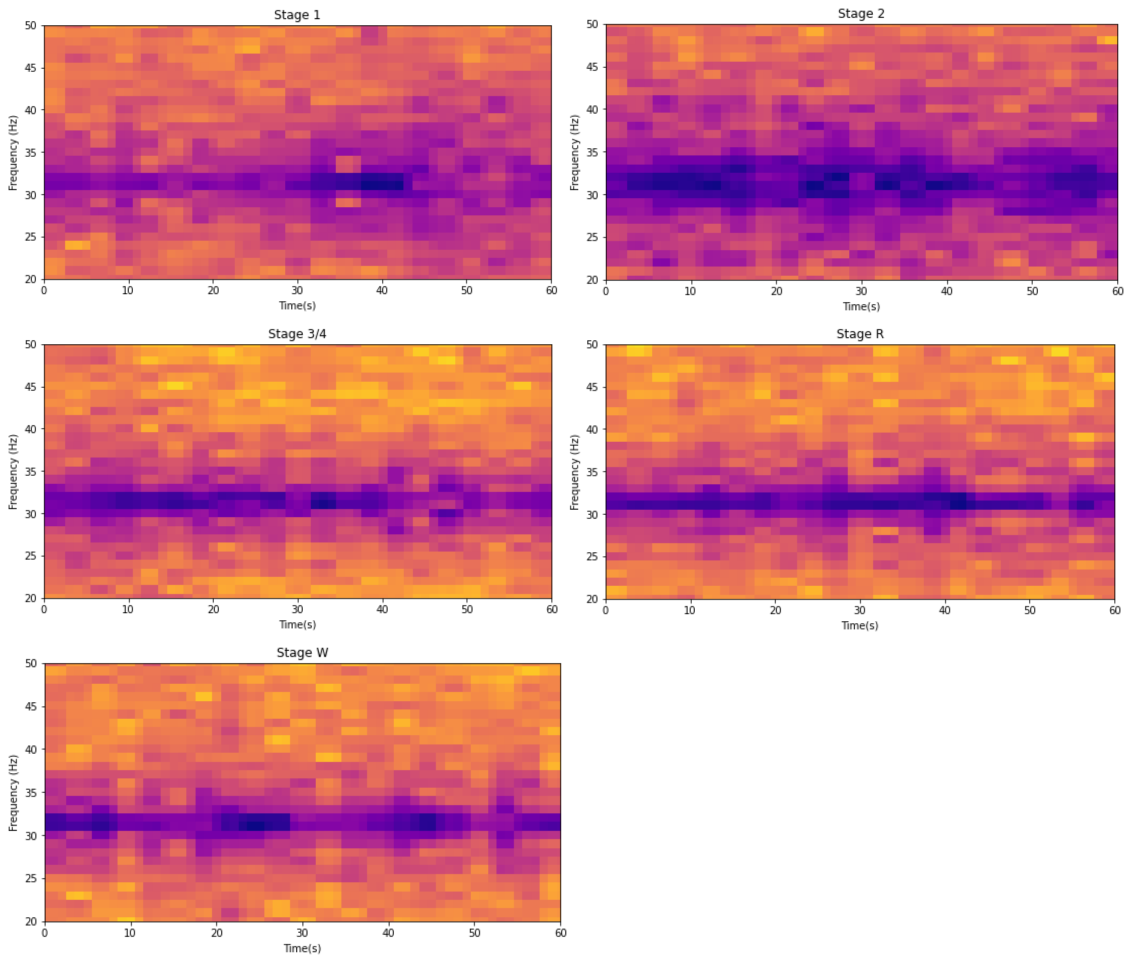


Figure 3. Spectrogram of sleep stages: W, 1, 2, 3/4, R in two 30 seconds epochs. The y-axis is in Hz, while the x-axis is in seconds (s). Due to similarities in characteristics of Stage 3 and Stage 4 **[6]**, they have been grouped into 1 class, leaving the total number of classes to be 5, aligning with the AASM standard. These plots are acquired from multiple patients.

2. Data pre-processing:

The spectrogram images are converted into RGB values between 0 and 1 **[8]** to create pixel images with a dimension of 64x64. Because these pixel images only have 1 channel, while the model in use (MobileNet) requires a minimum dimension of (32,32,3) for input, according to Keras application library [5], the information of one channel of each feature set is pasted into a new axis 3 times using NumPy library, emulating a data dimension of 3 channels. Total number of classes is 5 (W,1,2,3,R).

	Feature	Label
Train set	(110060,64,64,3)	(110060)
Test set	(5241,64,64,3)	(5241)
Validation set	(5241,64,64,3)	(5241)

Table 2. Size and dimension of training, test, and validation set to feed to the neural network model.

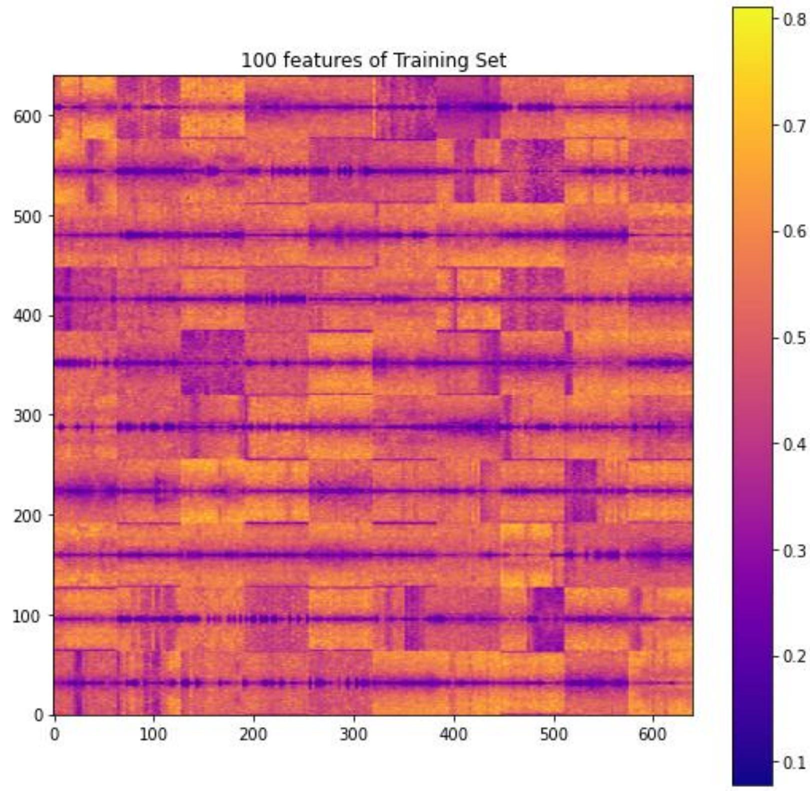


Figure 4. Visualization of training set. 100 features of size (64,64,3) are plotted in plasma gradient. Color bar indicates RGB value of each pixel.

3. Model Selection:

In this proposed approach, the state-of-the-art MobileNet is chosen for its lightweight-ness and fast training time. MobileNet architecture, at its core, is based on depthwise separable convolution apart from the first layer, which presents a full convolution. The architecture consists of a layer of convolution followed by ReLU nonlinearity except for the final fully connected layer. Unlike the standard convolution, MobileNet utilizes 3x3 depthwise separable convolution, resulting in 8 to 9 times less computation, at a cost of minor decrease in accuracy **[5]**.

Type / Stride	Filter Shape
Conv / s2	$3 \times 3 \times 3 \times 32$
Conv dw / s1	$3 \times 3 \times 32$ dw
Conv / s1	$1 \times 1 \times 32 \times 64$
Conv dw / s2	$3 \times 3 \times 64$ dw
Conv / s1	$1 \times 1 \times 64 \times 128$
Conv dw / s1	$3 \times 3 \times 128$ dw
Conv / s1	$1 \times 1 \times 128 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw
Conv / s1	$1 \times 1 \times 128 \times 256$
Conv dw / s1	$3 \times 3 \times 256$ dw
Conv / s1	$1 \times 1 \times 256 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw

Conv / s1	1 × 1 × 256 × 512
5x Conv dw / s1	3 × 3 × 512 dw
5× Conv / s1	1 × 1 × 512 × 512
Conv dw / s2	3 × 3 × 512 dw
Conv / s1	1 × 1 × 512 × 1024
Conv dw / s2	3 × 3 × 1024 dw
Conv / s1	1 × 1 × 1024 × 1024
Avg Pool / s1	Pool 7 × 7

Table 3. MobileNet architecture **[5]**

For this project, the dropout rate of MobileNet has been modified from default value of 0.001 in Keras function to 0.4 to reduce overfitting problem. The optimizer of choice is Adaptive Movement Estimation (Adam), which is a variance of Stochastic Gradient Descent (SGD) that is computational and memory efficient **[8]**. While the loss function is *sparse_categorical_crossentropy* because the model will predict images of spectrogram, which has floating points values (i.e. [0.5, 0, 0.2 …]). These hyper-parameters and model instantiate result in the model's total parameters of 5,023,605, trainable parameters of 4,996,24, and the non-trainable parameters of 27,360, which is small comparing to larger ones such as VGG or ResNet with total parameters ranging from 25,636,712 to 143,667,240 **[9]**.

III. Experimental Results

The proposed method performs extensively to score 5 stages (Wake, REM, 1, 2, 3/4). The model train on 110060 features with 100 epochs and a batch size of 220 on NVDIA GeForce GTX 1060.

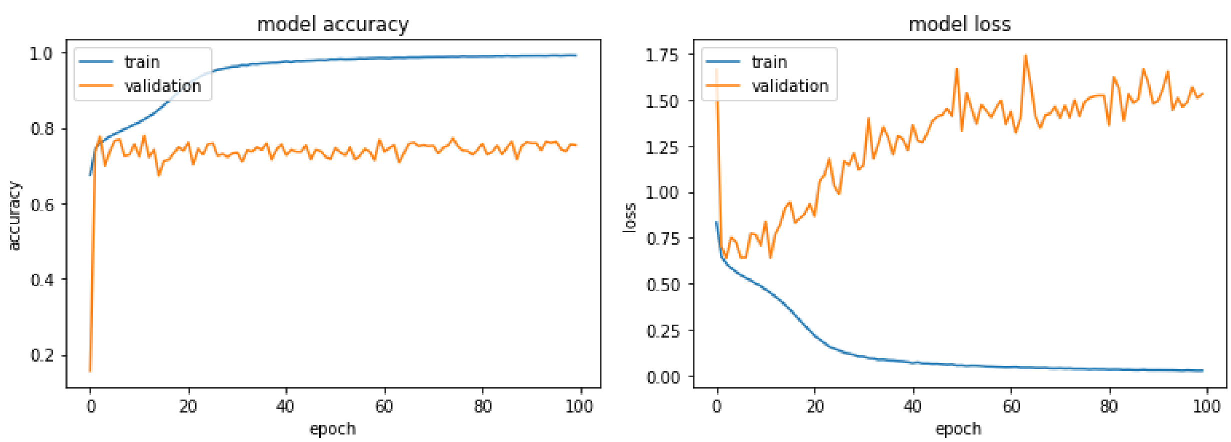


Figure 5. Accuracy and loss of training and validation set over 100 epochs. As the epochs increase, the accuracy for training increases and achieved a high of 99%, while it plateaus around 70 – 80% for validation. However, the trend for loss is opposite for train and validation set, while the loss decreases for training, it increases for validation set.

After the MobileNet model successfully performed on the training and validation set. A new test set will be used to evaluate the performance of the pre-trained model on features it has never seen before. Figure 6 and Table 4 summarize the performance.

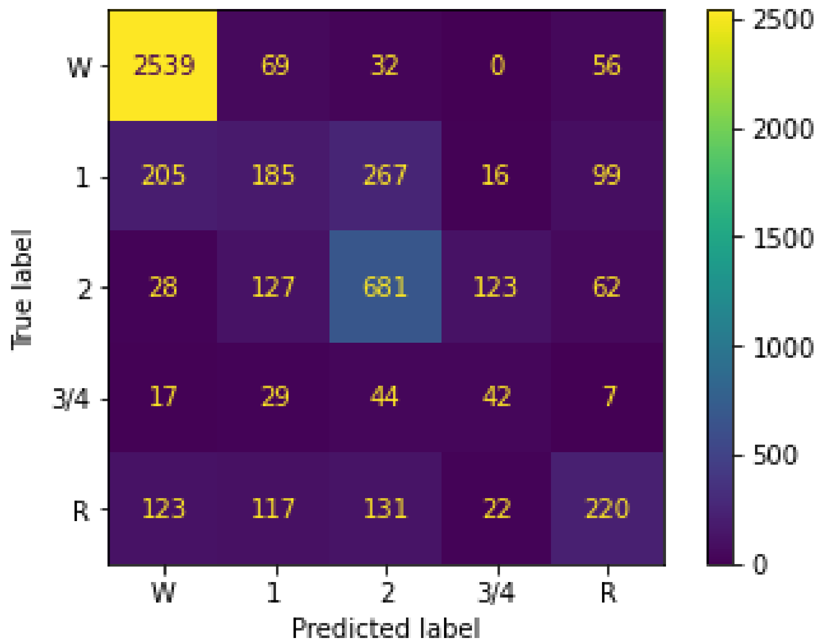


Figure 6. Confusion matrix of model prediction on test set. The model does best on stage W classification, second best on stage 2, third on stage R, fourth on stage 1, and worst on stage 3/4.

	precision	**recall**	**f1-score**	**support**
Stage W	0.87	0.94	0.91	2696
Stage 1	0.35	0.24	0.28	772
Stage 2	0.59	0.67	0.63	1021
Stage 3/4	0.21	0.30	0.25	139
Stage R	0.50	0.36	0.42	613

Table 4. Classification report: precision, recall, f-1 call, and support (total number of occurrences in the set) for the test set.

IV. Conclusion

In this study, the method of automated sleep staging using spectrogram, extracted from raw EEG data, is presented. This technique use 30 seconds epoch of sleep data to map raw features into images and feed them to deep neural networks models. The training set yields high accuracy during the training process; however, the validation and test set perform at a lower accuracy. Looking at the loss trend of validation set, it is safe to conclude that this model has problems of overfitting despite applying a large dropout rate of 0.4. Fixes for improvement of the model can be:

1. Increase and diversify the training set: use more features, and introduce pictures that are rotated, or zoomed in as 'new' data. A problem with this is resource exhaust as not every computers is equipped to handle a large amount of data

2. Simplify the training model

Overall, the accuracy is high when comparing to previous literature that employed Random Forest Classifier [1]

References

[1] Reza Boostani, Foroozan Karimzadeh, Mohammad Nami, "A comparative review on sleep stage classification methods in patients and healthy individuals", Computer Methods and Programs in Biomedicine, 2017, vol. 140, pp. 77-91, doi: 10.1016/j.cmpb.2016.12.004.

[2] A. Vilamala, K. H. Madsen and L. K. Hansen, "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring," 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), 2017, pp. 1-6, doi: 10.1109/MLSP.2017.8168133.

[3] 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," in IEEE Transactions on Biomedical Engineering, vol. 47, no. 9, pp. 1185-1194, Sept. 2000, doi: 10.1109/10.867928.

[4] D. J. Thomson, "Spectrum estimation and harmonic analysis," in Proceedings of the IEEE, vol. 70, no. 9, pp. 1055-1096, Sept. 1982, doi: 10.1109/PROC.1982.12433.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for Mobile Vision Applications," arXiv.org, 17-Apr-2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[6] 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, doi: 10.1016/j.smrv.2011.06.003.

[7] 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," Circulation, vol. 101, no. 23, 2000, doi: 10.1161/01.CIR.101.23.e215.

[8] S. Kanwal, M. Uzair, H. Ullah, S. D. Khan, M. Ullah and F. A. Cheikh, "An Image Based Prediction Model for Sleep Stage Identification," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 1366-1370, doi: 10.1109/ICIP.2019.8803026.

[9] K. Team, "Keras Documentation: Keras applications," Keras. [Online]. Available: https://keras.io/api/applications/.