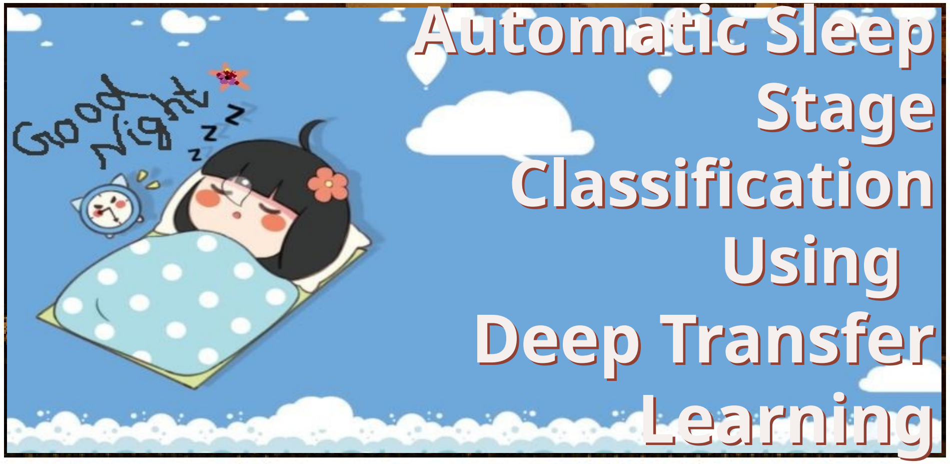


Automatic Sleep Stage Classification Using Deep Transfer Learning

Presented By: Rumana Sultana

Table of Content

Problem Statement and Background

Why Sleep Stage Classification?

- Depression

Sleep disorder

consequence to heart attack

even death

- Pandemic situation increased this situation.
- Need to improve human lifestyle and healthiness.
- Significant importance to diagnose the symptoms and diseases.

Problem Statement and Background

Why does data science solution need?

- Time consuming and expensive sleep stages labeling.
- Difficulty of ML and DL training for big data set.
- Data insufficiency for home based sleep monitoring devices.
- More dependency on cloud server.
- Latency and failure of service is an overhead.
- Need to develop a data-efficient model from raw sleep data.
- Edge devices can get accuracy and macro F1 score above 87%-83.



Previous Research works

- DeepSleepNet [3] – A sleep stage scoring model based on single channel EEG.
  - Sleep data set: MASS and Sleep-EDF.
  - Technique: Representation learning and sequence residual learning.
  - Accuracy: macro F1 score achieved: MASS: 86.2%-81.7, Sleep-EDF: 82.0%-76.9.
- The Survey paper [4]- described about edge intelligence and intelligent edge:
  - Technique: practical implementation methods.
  - Enabling technologies.
  - Future challenges.
- The research paper [5]- initiated a model for automatic sleep staging:
  - Sleep data set: MASS and Sleep-EDF.
  - Technique: deep transfer learning.
  - Accuracy: 84:3%(source domain) and 85:2% (target domain database).

Analysis and Exploration of Data set

- **Sleep-EDF Expanded Database**- 197 whole-night Polysomnographic sleep recordings,
- Including EEG, EOG, Chin EMG, and event markers of around 20 subjects.
- **Downloaded from-**https://www.physionet.org/static/published-projects/sleep-edfx/sleep-edf-database-expanded-1.0.0.zip.
- **Data and Annotation Files:**
  - *PSG.edf files
  - SC*PSG.edf
  - *Hypnogram.edf

Table 1: Sleep stage

Stage Label	Sleep Stage
W	wakefulness
R	rapid eye movement
N1	light sleep
N2	Deeper sleep
N3&4	deep sleep

Analysis and Exploration of Data set

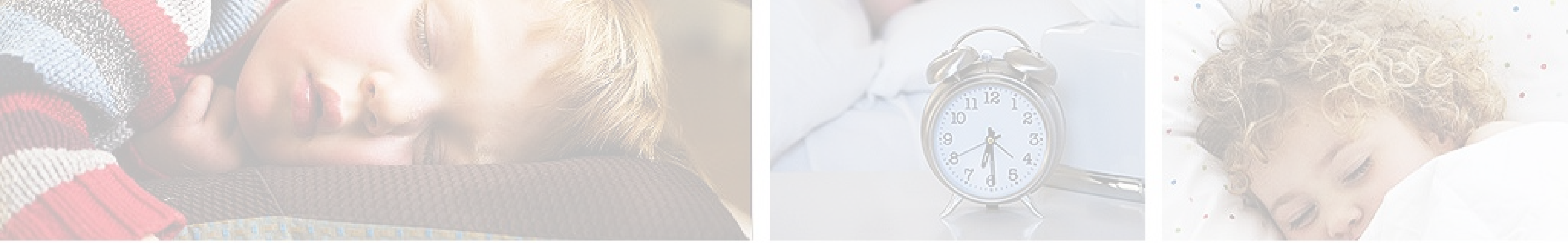
- **Sleep Cassette Study and Data:**
  - 153 SC*files of healthy Caucasians aged 25-101, with no medication.
  - **Variables**: Filename- SC4ssNE0-PSG.edf.

    Where, ss=subject number, N=night.
  - **Missing Values**: The first night of subjects 36 and 52, and the second night of subject 13.

- **Sleep Telemetry Study and Data:**
  - 44 ST*files of temazepam effects on sleep in 22 Caucasians males and females, difficulty with falling asleep.
  - **Variables**: Filename- ST7ssNJ0-PSG.edf

    where, ss=subject number, N=night

- EOG, EMG, and EEG signals sampled at 100Hz.



Methodology and Implementation

- Train a DeepSleepNet as source domain
- Method: Deep neural network.
- Fine tune:

  source domain → target domain.

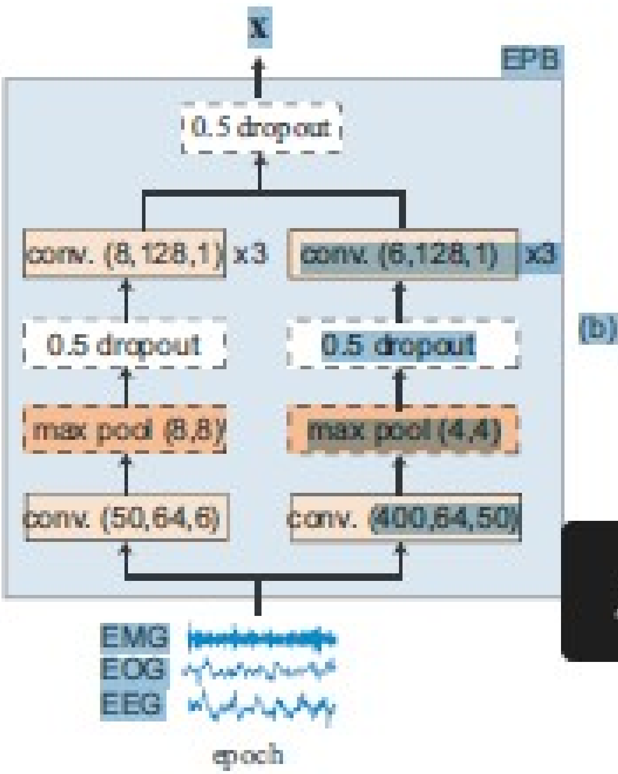


Figure 1: DeepSleepNet model

Methodology and Implementation (Cont.)

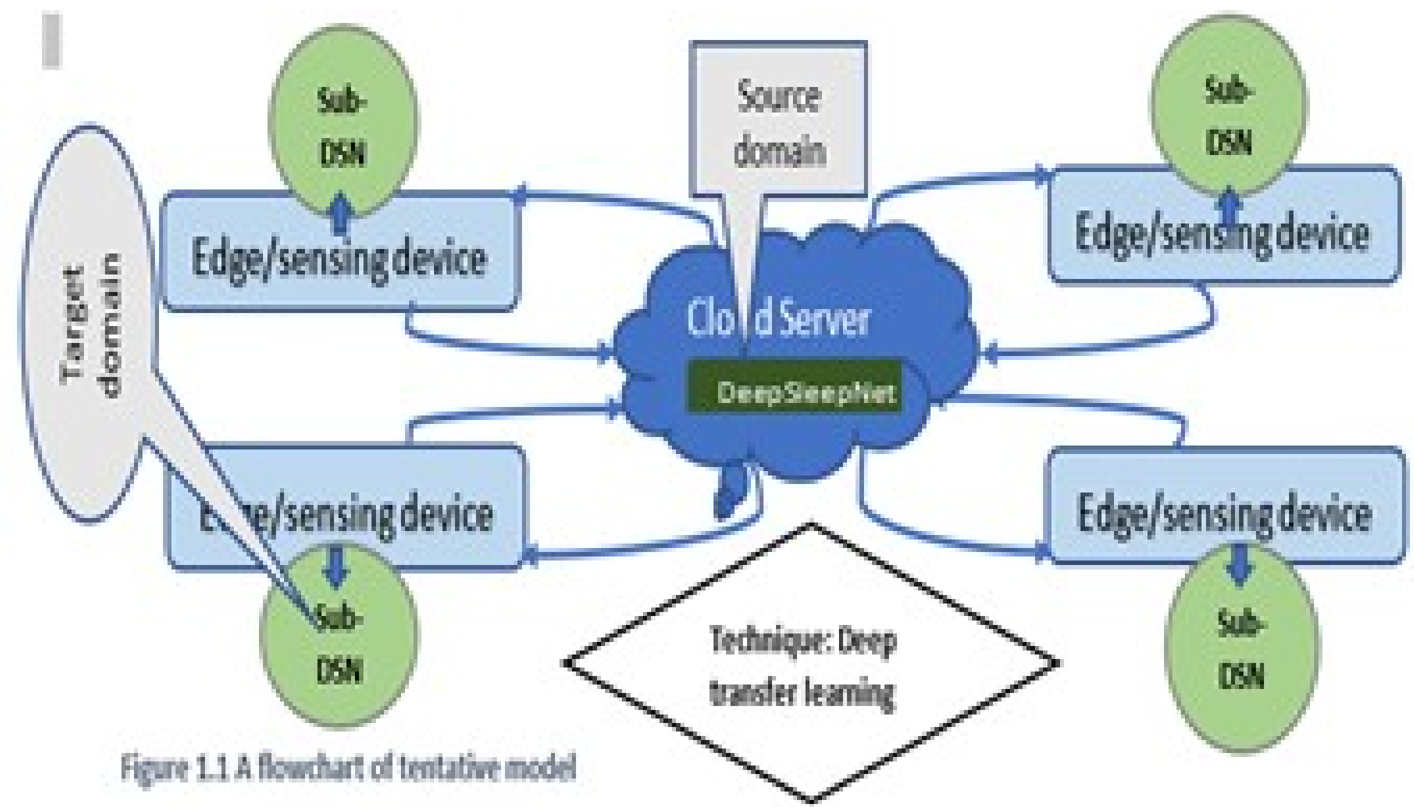
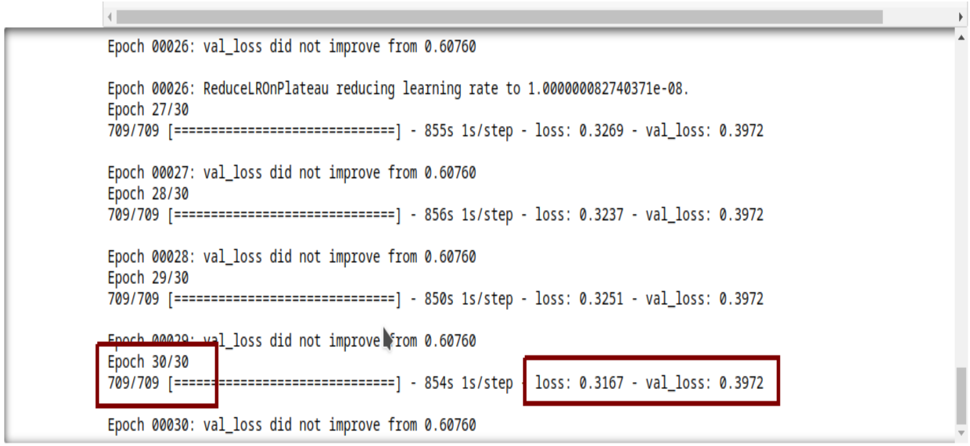


Figure 2: Flowchart



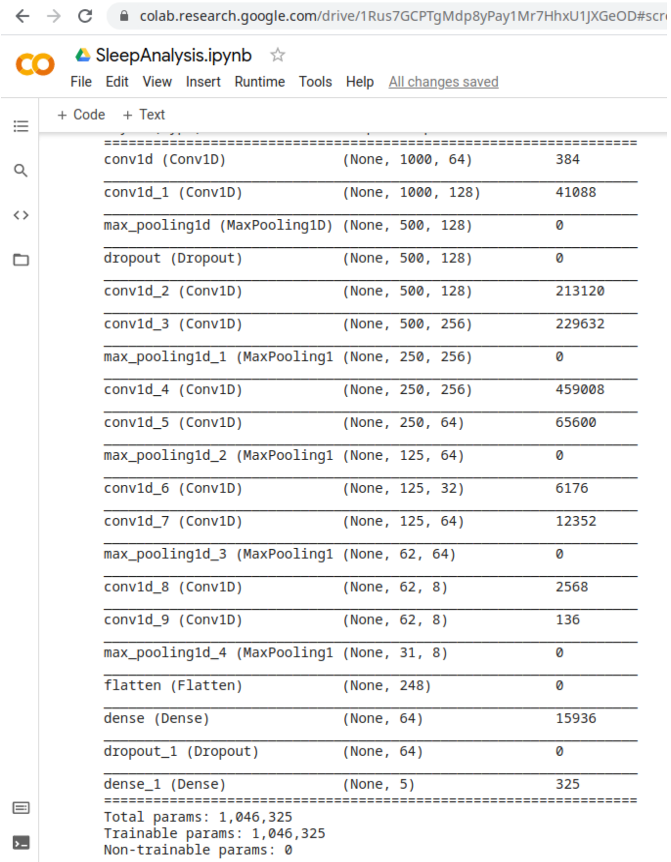
Technology

- Convolutional Neural Network
- Deep Transfer Learning
- Butter worth Filter



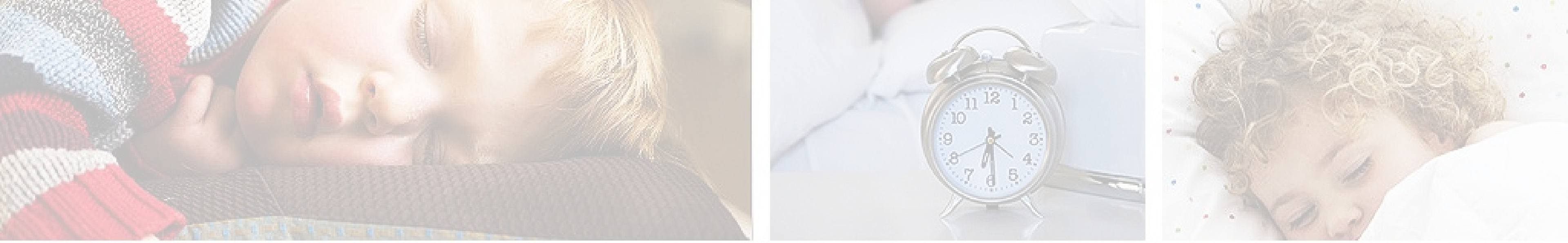
```
Epoch 00026: val_loss did not improve from 0.60760

Epoch 00026: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-08.
Epoch 27/30
709/709 [============================] - 855s 1s/step - loss: 0.3269 - val_loss: 0.3972

Epoch 00027: val_loss did not improve from 0.60760
Epoch 28/30
709/709 [============================] - 856s 1s/step - loss: 0.3237 - val_loss: 0.3972

Epoch 00028: val_loss did not improve from 0.60760
Epoch 29/30
709/709 [============================] - 850s 1s/step - loss: 0.3251 - val_loss: 0.3972

Epoch 00029: val_loss did not improve from 0.60760
Epoch 30/30
709/709 [============================] - 854s 1s/step - loss: 0.3167 - val_loss: 0.3972

Epoch 00030: val_loss did not improve from 0.60760
```

Figure 4: New CNN Model Training




```
=================================================================
conv1d (Conv1D)              (None, 1000, 64)          384
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 1000, 128)         41088
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 500, 128)          0
_________________________________________________________________
dropout (Dropout)            (None, 500, 128)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 500, 128)          213120
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 500, 256)          229632
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 250, 256)          0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 250, 256)          459008
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 250, 64)           65600
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 125, 64)           0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 125, 32)           6176
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 125, 64)           12352
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 62, 64)            0
_________________________________________________________________
conv1d_8 (Conv1D)            (None, 62, 8)             2568
_________________________________________________________________
conv1d_9 (Conv1D)            (None, 62, 8)             136
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 31, 8)             0
_________________________________________________________________
flatten (Flatten)            (None, 248)               0
_________________________________________________________________
dense (Dense)                (None, 64)                15936
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 325
=================================================================
Total params: 1,046,325
Trainable params: 1,046,325
Non-trainable params: 0
```

Figure 3: New Model developed



Methodology and Implementation(cont.)

PLATFORMS

- MATLAB (for data preparation)
- Python3,
- TensorFlow 2,
- Pytorch (for network training and evaluation)
- NumPy,
- SciPy,
- Sklearn,
- H5py.

Data Pre-processing

- Used Butterworth filtering to extract 30s window
- Eliminate high frequency noise
- Cutoff frequency of 30Hz.
- Wrap them into PyTorch data set

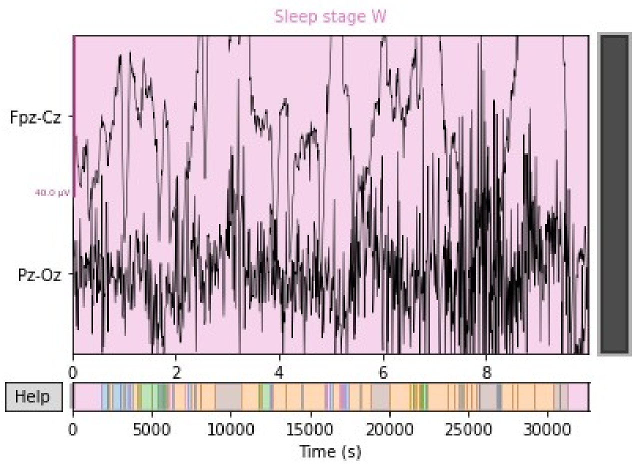


Figure 5: Plot of raw data

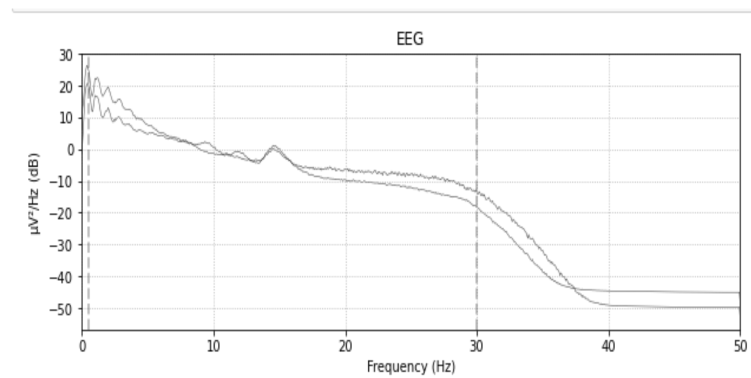


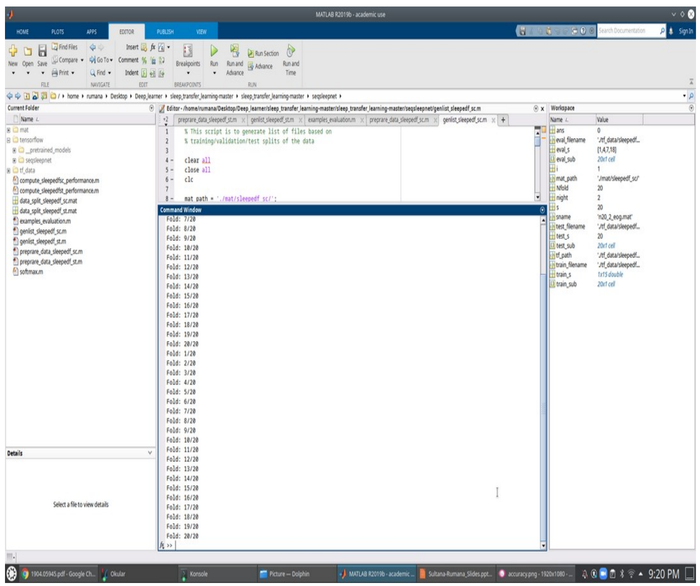
Figure 6: Plot of filtered data

Data Pre-processing

(Cont…)

Matlab data preprocessing [6]

- **Sleep cassette data (first 20 subjects)**
- **Sleep telemetry data (placebo only)**



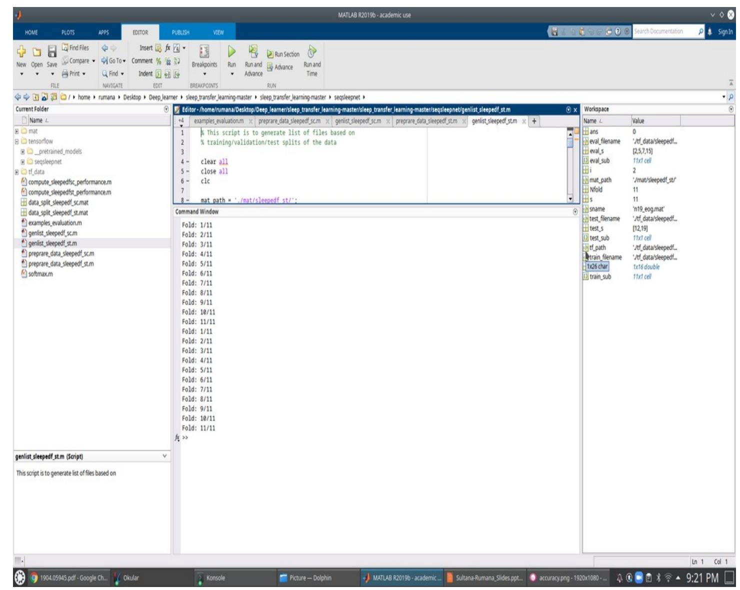


Figure 7: Matlab Data pre-processing

Testing and Result Analysis

Preliminary training with the pre-processed data in first way[8].

With number of examples in each set:

- **Training: 18845**
- **Validation: 4094**
- **Test: 9850**

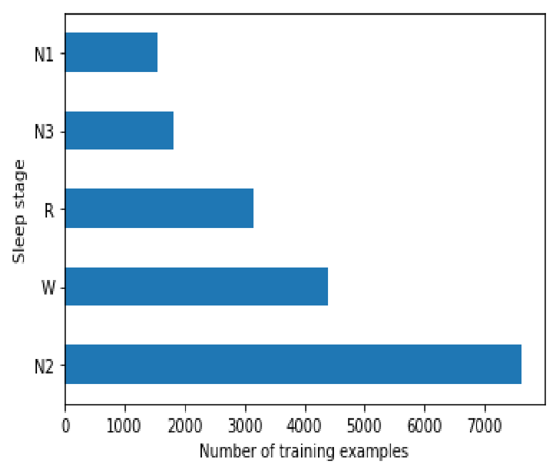
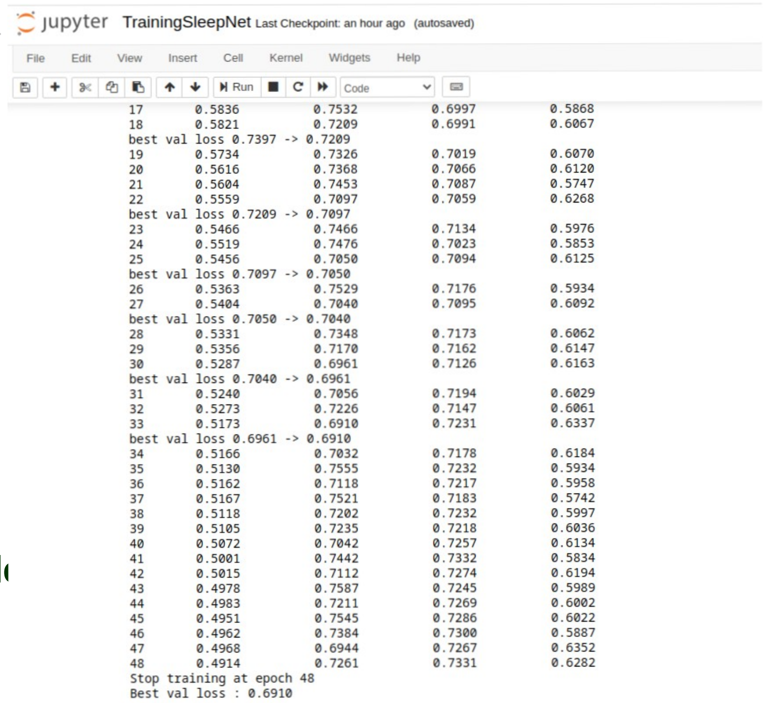


Figure 8: Number of training exampl




```
17        0.5836          0.7532          0.6997          0.5868
18        0.5821          0.7209          0.6991          0.6067
best val loss 0.7397 -> 0.7209
19        0.5734          0.7326          0.7019          0.6070
20        0.5616          0.7368          0.7066          0.6120
21        0.5604          0.7453          0.7087          0.5747
22        0.5559          0.7097          0.7059          0.6268
best val loss 0.7209 -> 0.7097
23        0.5466          0.7466          0.7134          0.5976
24        0.5519          0.7476          0.7023          0.5853
25        0.5456          0.7050          0.7094          0.6125
best val loss 0.7097 -> 0.7050
26        0.5363          0.7529          0.7176          0.5934
27        0.5404          0.7040          0.7095          0.6092
best val loss 0.7050 -> 0.7040
28        0.5331          0.7348          0.7173          0.6062
29        0.5356          0.7170          0.7162          0.6147
30        0.5287          0.6961          0.7126          0.6163
best val loss 0.7040 -> 0.6961
31        0.5240          0.7056          0.7194          0.6029
32        0.5273          0.7226          0.7147          0.6061
33        0.5173          0.6910          0.7231          0.6337
best val loss 0.6961 -> 0.6910
34        0.5166          0.7032          0.7178          0.6184
35        0.5130          0.7555          0.7232          0.5934
36        0.5162          0.7118          0.7217          0.5958
37        0.5167          0.7521          0.7183          0.5742
38        0.5118          0.7202          0.7232          0.5997
39        0.5105          0.7235          0.7218          0.6036
40        0.5072          0.7042          0.7257          0.6134
41        0.5001          0.7442          0.7332          0.5834
42        0.5015          0.7112          0.7274          0.6194
43        0.4978          0.7587          0.7245          0.5989
44        0.4983          0.7211          0.7269          0.6002
45        0.4951          0.7545          0.7286          0.6022
46        0.4962          0.7384          0.7300          0.5887
47        0.4968          0.6944          0.7267          0.6352
48        0.4914          0.7261          0.7331          0.6282
Stop training at epoch 48
Best val loss : 0.6910
```

Figure 9: Primary training

Testing and Result Analysis (Cont…)

Preliminary training result shows:

- ✔ **Test Balance**
- ✔ **Accuracy: 0.757**
- ✔ **Test Cohen"s**
- ✔ **Kappa: 0.714**

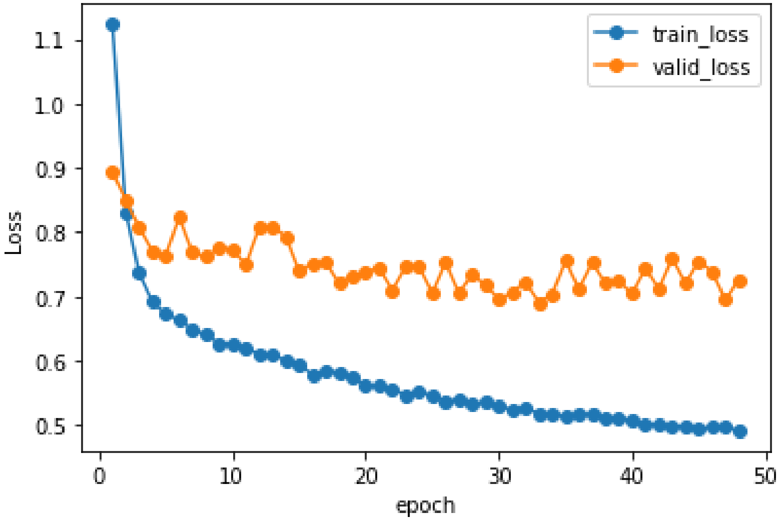


Figure 10: Visualization of loss

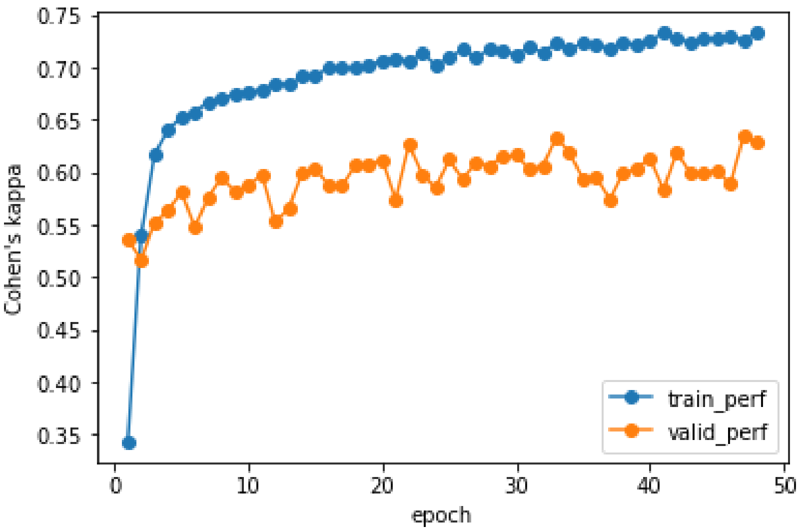


Figure 11: Visualization of performance

Testing and Result Analysis

(Cont...)

Final training with examples:

- **Training: 45351**
- **Validation: 5039**
- **Test: 5599**

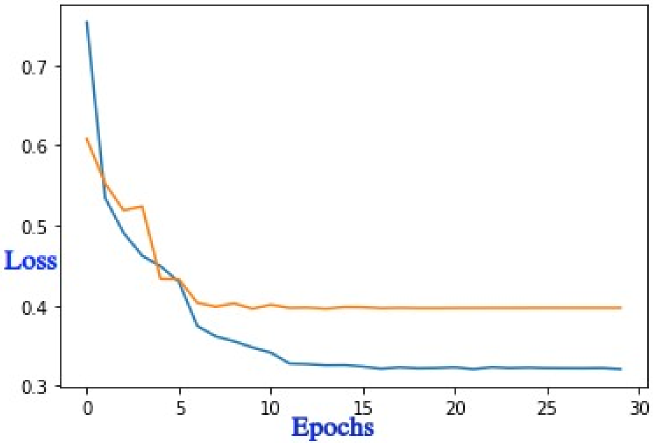


Figure 12: Loss and validation loss Graph with epochs

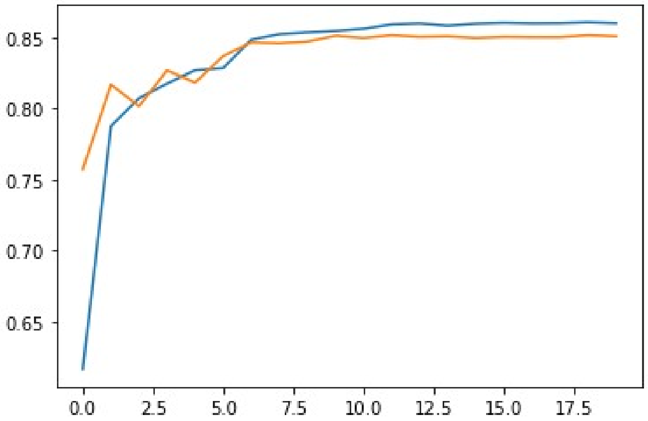


Figure 13: Accuracy and validation accuracy graph with epochs

Testing and Result Analysis

(Cont...)

F1 score: 0.7799128954339096

	precision	recall	f1-score	support
0	0.93	0.93	0.93	1231
1	0.52	0.35	0.42	435
2	0.87	0.90	0.89	2341
3	0.88	0.85	0.87	613
4	0.77	0.82	0.79	979
accuracy			0.85	5599
macro avg	0.79	0.77	0.78	5599
weighted avg	0.84	0.85	0.84	5599

Figure 14: Performance after training

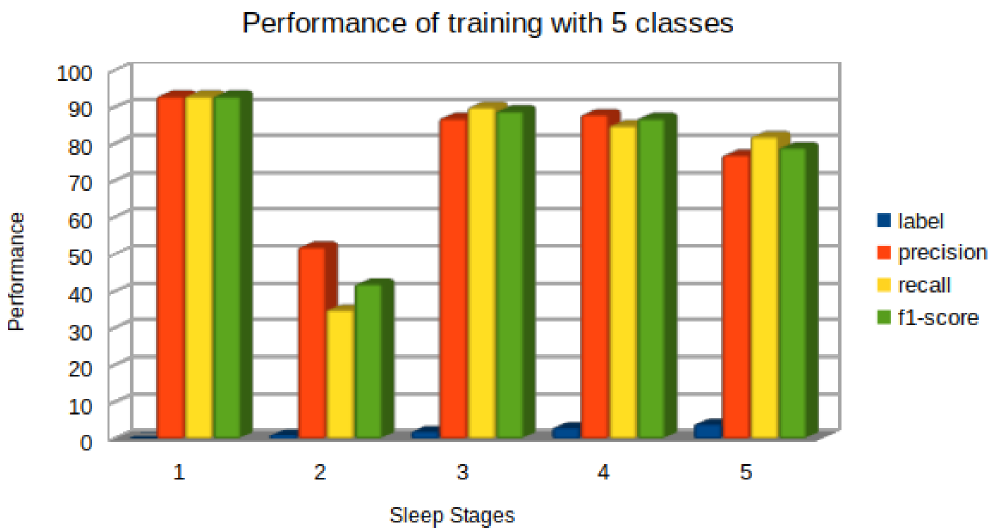
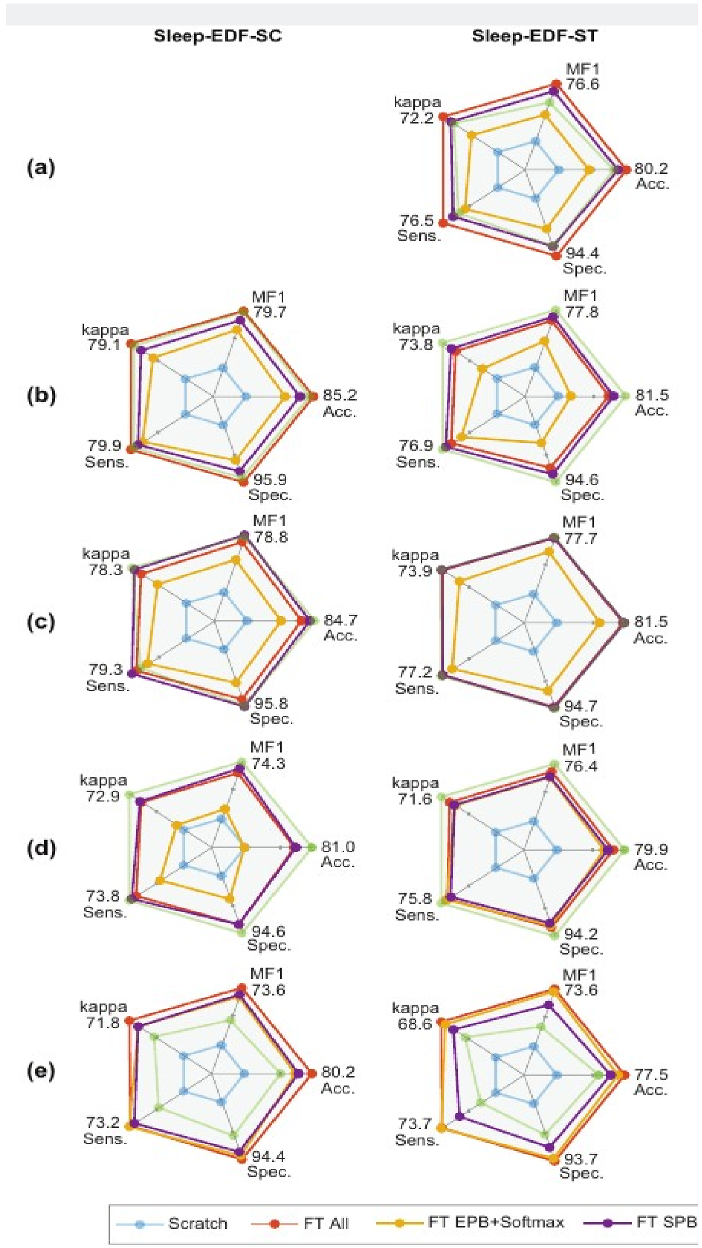


Figure 15: Performance visualization

Testing and Result Analysis

(Cont...)



- ✔ **Expected result**
- ✔ **To Increase**
- ✔ **About 3%**

Figure 16: Performance pattern after fine tuning[6]



Challenges and Future plan (Cont...)

- Big challenge is to train a model at home computer without GPU.
- Fine tune the new trained CNN model to target domain.
- Increasing performance for both as possible.

Conclusion

- Ensure sound sleep, ensure healthy life.
- Enrich sleep analysis with mobile computing.
- Increase reliable home-based sleep monitoring.

Thank you

Questions

???

References

[1] https://poweredtemplate.com/05036/0/index.html

[2] https://pngtree.com/free-backgrounds-photos/sleeping

[3] 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.

[4] Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), 869-904.

[5] Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2017). A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282.

[6] Phan, H., Chén, O. Y., Koch, P., Lu, Z., McLoughlin, I., Mertins, A., & De Vos, M. (2020). Towards more accurate automatic sleep staging via deep transfer learning. IEEE Transactions on Biomedical Engineering.

[7] The sleep-EDF database download link, https://www.physionet.org/static/published-projects/sleep-edfx/sleep-edf-database-expanded-1.0.0.zip.Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301,1982].

[8] https://colab.research.google.com/github/hubertjb/dl-eeg-tutorial/blob/main/sleep_staging_physionet.ipynb#0.-Setting-up-the-environment.

[9] Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., & Gramfort, A. (2018). A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(4), 758-769