Subject and Posture Classification with Convolutional Neural Network

Ciuche Bianca Andreea
biancaandreea.ciuche@studenti.unipd.it

Mukaj Bleona
bleona.mukaj@studenti.unipd.it

Abstract

Sleeping is one of the most important activities in our daily lives which affects our health. However, very few people could really understand their sleeping habits, which is important to avoid potential sleep-related diseases. A benefit of sleep monitoring is that it can lead to positive changes. Humans are more likely to change habits when they track them. Pressure sensing mats consist of gridded and flexible force sensors that are now commercially available for continuously measuring pressure distribution under body parts in different in-bed postures. In this study, we propose a convolutional neural network (CNN) model for three different classification tasks; the first task is capable of accurately detecting subjects,the second task is capable of detecting three standard sleeping postures(supine, right and left) and the last one is a multitask classification of subject identification and posture recognition at the same time. We evaluate the performance of our models applying it on two different data-sets. Our model showed really promising results for both experiments used in each classification tasks.

1 Introduction

Sleep plays a vital role in good health and well-being throughout your life. Getting enough quality sleep at the right times can help protect your mental health, physical health, quality of life, and safety. The pattern of in-bed postures during sleep is one of the key indicators of sleep quality and can be beneficial for several medical diagnosis.

Lately, sleep posture has been shown to be important in monitoring health conditions such as congestive heart failure (CHF), pressure ulcers, and even blood pressure abnormalities [1]. As such, recent studies have shown in-bed postures have a high influence on occurrence of sleep diseases such as apnea [2]. Conventional system for sleep study using neurophysiological signals, polysomnography (PSG), is extremely constrictive and expensive [3]. Furthermore, many PSGs are performed exclusively in hospital environments requiring patients to stay overnight in hospitals.

In this paper, to easily detect subjects and sleep posture from a data-set we present three different classification tasks using convolutional neural network. By taking into consideration two different data-sets of pressure sensor, which includes different subjects and various sleeping postures, the classification tasks implemented are the following ones: subject identification, posture recognition and joint subject identification and posture recognition, where for each one of these we compare the results on the two data-sets presented. Researchers have also found that sleep quality is related to sleeping position and frequent sleep postural changes, like snoring or extensive body movement may result in a shorter sleep duration [4]. Commercial pressure sensing mats have recently been available to continuously measure pressure distribution under the body while being in bed. These pressure mats report very valuable data which can be used directly by machine learning algorithms to track postures and assess the experienced stress by each body parts [5].Our method can provide valuable information in a faster

way, which can be used in smart home and clinical settings.The informations achieved can be relevant to the relationships of particular diseases.
This report is structured as follows. In Section II we describe the state of the art, the pre-processing techniques, learning frame and results are respectively presented in Sections III,IV and V. The proposed signal processing technique is detailed in Section IV and its performances evaluation is carried out in Section VI. Concluding remarks are provided in Section VII.

2 Related Work

Over the years, many different sleep assessment methods have appeared, especially in the last years with new methods and models.Monitoring and relieving pressure experienced at different postures has therefore become a critical step in reducing pressure ulcer incidences in hospitals and nursing homes.
There has been an increasing interest in sleep quality evaluation, not only for clinical research and for treating sleep disorders, but also in the fields related to the healthcare and health promotion[6], but however, unlike the data-set used in our study, the number of posture classes in previous studies were limited. There has also been increased attention on studies regarding the effects of sleeping postures on sleep apnea [7] and ulcer prevention using advanced monitoring technologies [8] for applications in smart beds and these papers helped us a lot on understanding how really our study can be applied in the real world scenario, but the methods suffer from occlusions, lightening conditions, as well as viewpoint angle and calibration effects.
Another study [9] showed that assessment of in-bed postures can be considered of high importance in evaluation of sleep quality and bed-bound patient's health status. They used one of the data-set we have taken into consideration in our work, but instead working with raw data as we did, they extract 18 statistical features from the pressure data.
Another interesting paper is [10], where is used a deep end-to-end CNN framework proposed for the joint classification of subject identification and posture recognition with in-bed pressure-sensors. Their work is an efficient and highly accurate model that performs both tasks simultaneously, subject and posture recognition together at the same time, where conventional classification algorithms fail to achieve a good accuracy for unseen data. They showed that their method significantly outperforms other works done related to this study.
Their proposed algorithm can ultimately be used in clinical and smart home environments as a complementary tool with other available automated patient monitoring systems.

3 Processing Pipeline

In our project we implemented a similar convolutional neural network for each of the classification tasks, the architecture of the models differs only in the output layer.
After trying various implementations for the architecture of our models we take inspiration from the paper [10]. Every classification is carried out by a feed-forward pass that consists of two blocks composed by the following layers: Convolution layer,BatchNormalization layer, MaxPool layer and LeakyRelu layer. Subsequently follows two others blocks with structure: Convolution layer, BatchNormalization layer and LeakyRelu layer.This four blocks are capturing the important properties of each single frame. The classification is then achieved by feeding the out come to one dense layer. In the case of the subject identification task and the posture recognition task the dense layer is followed by one SoftMax layer, instead for the joint subject identification and posture recognition task the dense layer is followed by two SoftMax layers used in parallel to classify subjects and posture separately.
We used the Dropout technique to prevent overfitting. Below, in figure 1 is illustrated the architecture of our model.

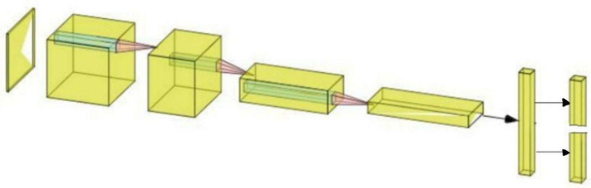


Figure 1: Our proposed framework is presented.

4 Signals and Features

In this study we used a public data-set, PmatData, which contains in-bed posture pressure data from multiple adult participants using two different types of pressure sensing mats. All 13 participants were healthy individuals with no history of sleeping disorders or pressure ulcers. Individuals 1-8 participated in both experiments while individuals 9-13 only participated in Experiment-I. PmatData contains pressure data from two separate experiments. These two experiments are:

- **Experiment-I:** The first data-set contains pressure data from 13 participants in 8 standard postures and additional states. Size of pressure mat is 32*64. This is the raw data collected reporting numbers in range of [0-10,000] for each sensor. Sampling rate is 1Hz. Each file includes the data frames around 2-mins (around 120 frames). For this data-set we created an input array of shape (20024, 2048), which contains all the pressure mats, and two labeled arrays of shape (20024,), one contains the subject labels and the other the posture labels for every pressure mat. The labels for subject recognition in this data-set are all subjects from 1st participant to 13d one and for posture recognition, we have three standard classes: Supine -Left -Right.

- **Experiment-II**:The second data-set contains pressure data from 8 participants in 29 different states of 3 standard postures. Furthermore, experiment II was collected separately for both regular and air-alternating pressure mattresses. Size of pressure mat is 27*64. This is the raw data collected reporting numbers in range of [0-500] for each sensor. Sampling rate is 1Hz. Each file contains the average of around 20 frames recordings. The data is collected for both sponge and air mattresses separately. For this data-set we create an input array of shape (462, 1728) and two labeled arrays of shape (462,), one contains the subject labels and the other the posture labels for every pressure mat. Labels here, for the subject recognition, are only for the first 8 participants, while the labels for posture recognition remain the same three standard classes of: Supine-Left-Right.

For experiment-i each file contains multiple frames of recordings, with the 2048 columns representing the 32x64 pressure sensor measurements, and each row giving a new set of samples. For experiment-ii each file contains one average of multiple frames of recordings. In contrast with the experiment-i text files, the 27 columns and 64 rows represent the 27x64 sensor matrix.

In the pre-processing step we attempted to reduce the noise caused by occasional malfunctioning pressure sensors. Such artifacts often take place when individual sensors are subjected to large pressure values outside of the allowed voltage range.

- To cope with this problem, we used for both data-sets the Median Filter of size 3x3x3 pixels, we tried first a filter size of 1x3x1 but the image was not as clear as the 3x3x3 size. The main idea of the median filter is to run through the signal entry by entry, sort them and replacing each entry with the median of neighboring entries. In the figure 2 we can see the difference of an image before and after applying the median filter.

- Second thing we did, only for the first experiment, was eliminating all the images that were not clear enough (most of the values of raw data were 0).

- And to conclude with the pre-processing we normalized the input data by dividing each value with the maximum range that a sensor could get, respectively for each data-set, we divided

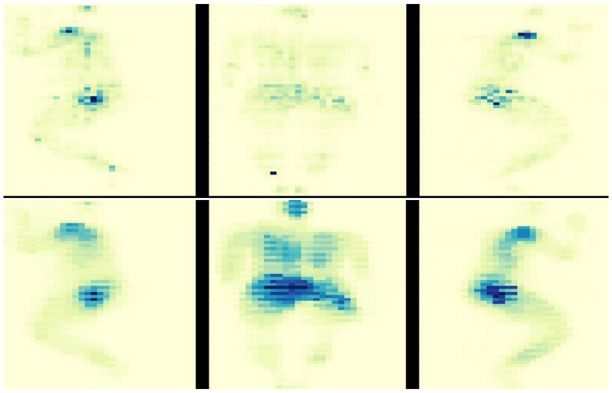


Figure 2: First three samples are presented in the original raw format, while other three are presented after applying the filter

by 10,000 (range for each sensor was from 0-10,000) and by 500 since the second data-set had its range for each sensor varying from 0-500.

- Before implementing the model, we divided both data-sets into three parts: train, test and validation sets; with a test size of 0,25. Also, for experiment-ii, since we have less data, to prevent overfitting we increased the number of the data-set.

5 Learning Framework

Our proposed models were implemented using TensorFlow platform. After applying different ones, we choosed the model which performed best and got the highest results.
As mentioned in the earlier section, as a final model, we used a convolutional neural network for three of the classification tasks presented. In the following points we described in deeper the architecture of our model.

- First Block: Conv2D layer of filter size 32, kernel size (3,3) with a stride of 1, a BatchNormalization layer, MaxPooling layer of kernel size (3,3), and a LeakyReLU layer with activation coefficient of 0,2. After the LeakyReLU layer,to prevent overfitting we added a Dropout layer with a rate of 10% . Now let us take a deeper look and see how these layers interacts with each-other as showed in figure 3

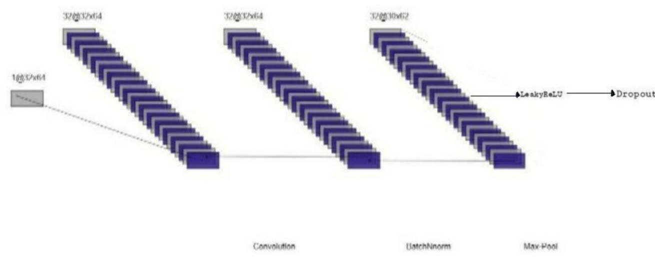
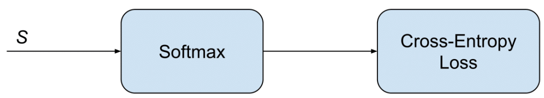


Figure 3: The core of the first block

- Second Block: It has again the exact same layers as the first block but here we change the filter size of Conv2D from 32 to 64 and the Dropout rate to 20% .
- Third Block: It consists of Conv2D layer, BatchNormalization layer and a LeakyRelu layer, so starting from this block we removed the max-pooling and also change the filter size of Conv2D from 64 to 128 and the Dropout rate to 30%.
- Fourth Block: This one has also the same layers used in the third block, but we apply the filter size of 264 of Conv2D from 128 that was in the previous one, and also increased the Dropout rate of 40%.
- Fifth Block: In the last block we have total different layers. We used Flatten layer which it flattens the output of the convolutional layers to create a single long feature vector, one Dropout of 50%, and Dense layers with a softmax activation, and size 256 and 14 or 4, depending on the classification task: subject or posture , or for the multitask problem where we use two parallel softmax Dense.

We trained our convolutional neural network using the Adam optimizer with a fixed learning rate

to 0.00002. Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks.
The loss function used in the model is the categorical cross-entropy, which is used for multi-class classifications,this loss function is also called Softmax Loss and it is composed by a Softmax activation plus a Cross-Entropy loss. With this loss function we will train a CNN to output a probability over the C classes for each frames. In the specific (and usual) case of Multi-Class classification the labels are one-hot, so only the positive class C_p keeps its term in the loss. In figure 4 is shown a diagram of loss function used.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

Figure 4: Categorical Cross-Entropy loss

There is only one element of the target vector t which is not zero $t_i = t_p$. So discarding the elements of the summation which are zero due to target labels, we can write:

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

where S_p is the CNN score for the positive class.

For the first two classification tasks, subject identification and posture recognition, it is used as in general, only one categorical cross-entropy loss function. Meanwhile when we implement the multitask classification of joint subject identification and posture recognition, we used two of categorical cross-entropy loss functions (since we implement the model for classifying at the same time subject and posture recognition).

6 Results

As explained in the earlier sections, we used a deep learning approach, i.e. a convolutional neural network model, for our three classification tasks: subject identification, posture recognition, and joint subject and posture recognition. We used three different metrics for the evaluation:

- ***Accuracy***: it is the fraction of predictions our model got right
- ***F1 score***: it is the harmonic mean of the precision and recall
- ***Confusion Matrix***: is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Let's initially consider the subject identification task, in which we need to identify each participant for both data-frames. The Convolutional Neural Network model on the first experiment achieved a high accuracy (99.4%) even though there is a bit overfitting as we can see in the Figure 5.

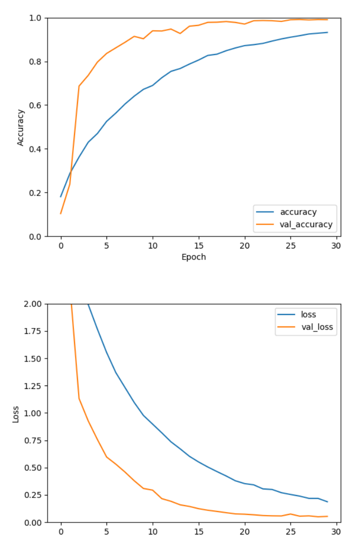


Figure 5: Accuracy and loss for subject identification

The performance of our model can be also observed in the following Confusion Matrix showed in Figure 6.

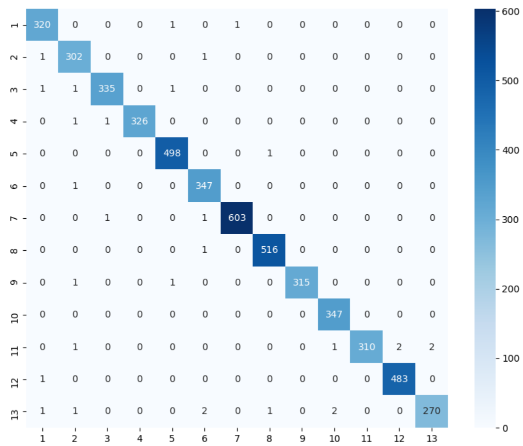


Figure 6: Confusion Matrix subject identification

After this analysis, we checked the performance of the model using the data-set of Experiment-ii and we obtained more or less the same behaviour for the overfitting, instead the values of the evaluation metrics are quite different as showed in the Table 1.

Metric	Experiment i	Experiment ii
Accuracy	99.4%	83.9%
F1 score	99.3%	82.5%

Table 1: Subject identification Results

From these results we can observe that the model has a better performance on the first experiment, in our opinion, this due to the fact that the second experiment has less participants and less frames recording for each participant.

Let's now focus our analysis on the posture recognition classification task. The performance obtained with the data-set of experiment-i for the posture recognition is better than the one obtained for subject identification, in fact in the following figure 7 we can observe that there is not any overfitting effect.

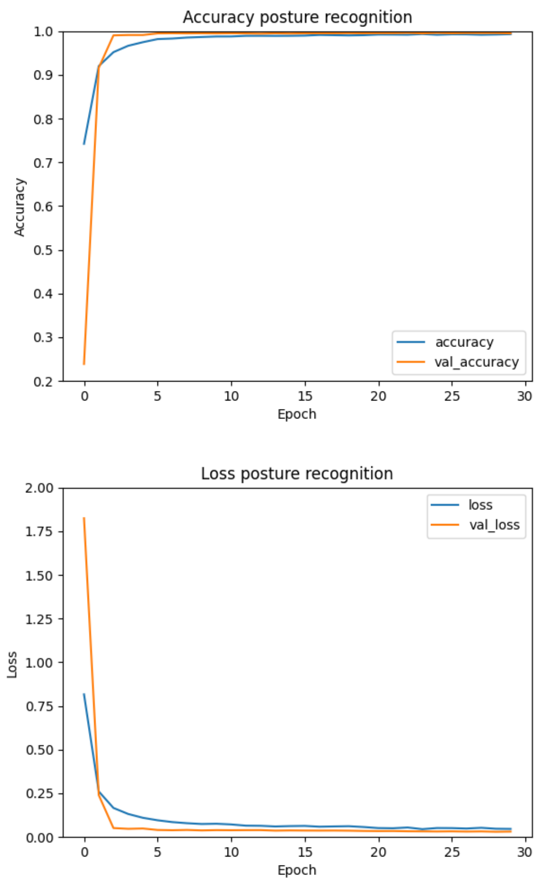


Figure 7: Accuracy and loss posture recognition

We obtained an accuracy of 99.3% and a f1-score of 99.3% and below, in figure 8, is showed the confusion Matrix performance.

Comparing the result of the first and the second experiments for this classification task we observed that the model has a decreasing performance for the second one as we can see in Table 2. So the model has the same behaviour as for the subjects identification.

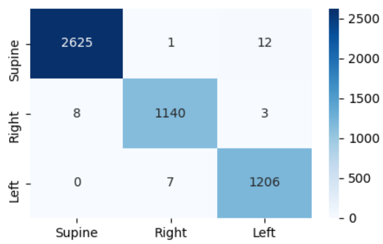


Figure 8: Confusion Matrix posture recognition

Metric	Experiment i	Experiment ii
Accuracy	99.3%	74.9%
F1 score	99.3%	75.4%

Table 2: Posture recognition results

As the final step we focus on the analysis of the last classification task i.e. joint subject identification and posture recognition. Since we implemented a multitasking learning for this classification, we now show in the tables (3,4) the performance for each task on each data-set.

Metric	Subject Id.	Posture Rec.
Accuracy	99.36%	99.32%
F1 score	99.36%	99.32%

Table 3: Experiment-i: Multitasking learning results

For experiment-i if we compare the results of the multitasking learning with results obtained in the first two classification task we observed similar performance. Instead, for the experiment-ii the results in the multitasking learning decrease. In our opinion this different behaviour between the two experiments is due to the fact that the two data-sets are very different.

Metric	Subject Id.	Posture Rec.
Accuracy	80.27%	79.06%
F1 score	79.78%	77.62%

Table 4: Experiment-ii: Multitasking learning results

7 Conclusions

In this project, the convolutional neural network model is implemented for the three different classification tasks mentioned in the earlier section. Our work is efficient and highly accurate. It can be very beneficial to clinical and smart home environments as a complementary tool. The future work can focus on pressure data-sets with higher temporal resolution in order to measure physiological signals such as heart-rate.
From this project we encountered difficulties on achieving the model that performed best, because before choosing our final one, we implemented various models and checked if these performed good with the data-sets of both experiments.

What we learned from our work are summarized in the following points.

- Importance of sleeping posture;
- Working with sensors data;
- Pre-processing techniques;
- Working with two different data-sets at same time;
- Implementing robust convolutional neural network models;
- Multitasking learning;
- How to handle a classification problem in real life.

References

[1] Moein Enayati, Marjorie Skubic, James M Keller and Mihail Popescu, "Sleep Posture Classification Using Bed Sensor Data and Neural Networks", in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* July 2018.

[2] C.H.Lee, D.K.Kim, S.Y.Kim, C.S.Rhee, and T.B.Won, Won, "Changes in site of obstruction in obstructive sleep apnea patients according to sleep position" in *The Laryngoscope*, vol. 125, no. 1, pp. 248–254, 2015.

[3] M. L. Blood, R. L. Sack, D. C. Percy, and J. C. Pen, "A comparison of sleep detection by wrist actigraphy, behavioral response, and polysomnography," in *Sleep,* vol. 20, no. 6, pp. 388–395, 1997.

[4] Hao T, Xing G, Zhou G, "Unobtrusive Sleep Quality Monitoring using Smartphones" in: *Proceedings of the 8th ACM conference on (SenSys)*, 2013.

[5] S. Ostadabbas, R. Yousefi, M. Nourani, M. Faezipour, L. Tamil, and M. Q. Pompeo, "A resource-efficient planning for pressure ulcer prevention," Information Technology in *Biomedicine, IEEE Transactions* on, vol. 16, no. 6, pp. 1265–1273, 2012.

[6] M. B. Pouyan, J. Birjandtalab, M. Nourani, and M. M. Pompeo, "Automatic limb identification and sleeping parameters assessment for pressure ulcer prevention," Computers in *biology and medicine*, vol. 75, pp. 98–108, 2016.

[7] R. D. Cartwright, "Effect of sleep position on sleep apnea severity," Sleep, vol. 7, no. 2, pp. 110–114, 1984.

[8] S. Ostadabbas, M. B. Pouyan, M. Nourani, and N. Kehtarnavaz, "In bed posture classification and limb identification," in *IEEE Biomedical circuits and systems conference (BioCAS)*, 2014, pp. 133–136.

[9] M. Baran Pouyan, J. Birjandtalab, M. Heydarzadeh, M. Nourani, "A Pressure Map Dataset for Posture and Subject Analytics " in: *Quality of Life Technology Laboratory*, 2017.

[10] Vandad Davoodnia and Ali Etemad, "Identity and Posture Recognition in Smart Beds with Deep Multitask Learning", in: *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.