Convolutional neural networks for subject identification and in-bed posture recognition

Marc Berret†

***Abstract*—Sleep posture analysis is a widely monitored sleep indicator in medical environment, and recent research have shown that in-bed posture plays a key role in sleep quality. Moreover, tracking sleep posture over time can help prevent certain diseases such as apnea and pressure ulcers. In this paper, I present an accurate and efficient deep convolutional neural network capable of classifying patients and their in-bed postures using commercial pressure sensing mattresses. The model significantly outperforms by making almost no errors on the three main postures (supine, right side, left side). In addition, it achieves an average accuracy of up to 99.8% for the prediction of subjects and 17 different postures, which has never been accomplished before. A 10-fold cross validation was applied to confirm the results and robustness of the model. Finally, this research paper can be use in medical and smart home environments as an additional tool to their existing systems.**

***Index Terms*—Identity and Sleep Posture Analysis, Convolutional Neural Networks**

I. Introduction

Sleep is an essential function of the human body and is important for our well-being. It has been proven that a lack of sleep disrupts daily life and is problematic for health. Indeed, prolonged sleep deprivation is associated with an increased risk of hypertension, diabetes, obesity, depression, heart attack, and stroke [1]. Furthermore, it has been shown that certain patterns of in-bed posture increase the risk of sleep apnea [2], [3] and pressure ulcers [4]. Therefore, it is beneficial to develop a system detecting a patient's posture in order to improve sleep quality and avoid the risk of illness.

The traditional system used to study sleep is visual inspection of neurophysiological signals named polysomnography (PSG) [5]. This method presents some disadvantages such as beeing expensive, time-consuming and can only be performed in a medical environment. Recently, pressure sensing mattresses have been commercialised and enable to continuously measure pressure distribution under the body while lying in bed. Pressure sensing mattresses would be the solution and the source data for machine learning algorithms to track in-bed postures.

The problem at stake is to determine and implement a model capable of accurately predicting the subjects and their sleeping postures simultaneously. The relevance of this paper is to further improve prediction results by taking into account available computing resources. Our method relies on deep convolutional neural networks, carefully selected to improve the performance of previous research papers. The proposed algorithm can ultimately be deployed in medical and smart home environments as a complementary tool with other available automated patient monitoring systems.

†Department of Mathematics, University of Padova, Italy, email: marcantoine.berret@studenti.unipd.it

Contributions

The main contributions of this report are :

- to define an efficient, automatic and sustainable pre-processing pipeline for pressure sensing mattresses data.
- to analyze different neural network architectures in order to find the best model structure able to identify the patient and recognize the posture at the same time. The best model will need to be simple, fast, accurate and reliable.
- to apply a 10-fold cross validation to fully evaluate this best model and compare it with other models from other research papers.

This report is structured as follows. In Section II we describe the state of the art; a high level description of our architecture and pre-processing techniques are respectively presented in Sections III and IV. A more detailed explanation of the proposed networks is provided in Section V and their performance evaluation is carried out in Section VI. The paper is then closed with some final thoughts in the concluding remark in Section VII.

II. Related Work

Several research papers aimed at detecting in-bed postures have been published. For instance, camera-based systems have been used to study posture patterns. Unfortunately, a number of problems have been observed with this method, including image noise due to the lighting conditions and the fact that these systems do not respect patient privacy [6]. Other techniques exist such as wearable sensors [7] or electrode-based systems [8] to monitor sleep patterns and estimate in-bed postures. However, these techniques can be uncomfortable for the subject.

In recent years, there has been great interest in posture recognition and limb locations using high density pressure sensors. In 2012, research paper [9] used 42 x 192 array of pressure sensors and later in 2016, research paper [10] used 48 x 128 array of pressure sensors both for classifying 4 sleeping positions: supine, prone, right and left. Although these two approaches report a very high performance of around 97.8%, two limitations remain: the size of the pressure sensor arrays is too large and the number of postures to be classified is limited.

In 2017, the research paper [11] made available "the first publicly-available dataset of pressure sensor data which includes various sleeping postures". This dataset contains 17 different sleeping postures from multiple adult participants. It allows research to be carried out on a wider range of postures using a 32 x 64 array of pressure sensors. In this paper, they implemented feature extraction to obtain 18 unique statistical features from the pressure distribution per frame/posture for each participant. Then, they trained a feed forward neural network per posture with the following architecture: an input layer, 3 hidden layers with 20, 20 and 10 nodes respectively, and an output layer to classify the subject and the subject's posture. They obtained interesting results with between 80% and 85% accuracy for supine, right and left postures on 10-fold cross validation, but these results can be improved by using deep convolutional neural networks.

A recent research paper [12] published in 2019 used the previous dataset to implement a deep convolutional neural network. They have proposed image pre-processing with a median filter to reduce noise and a loss function that can be modified with a λ parameter. They achieved very good results for the classification of supine, right and left postures with an accuracy of 99.2% on 10-fold cross validation using a data augmentation with 50 epochs. However, their performance was less impressive when it came to identifying subjects (89.7% of accuracy) and recognising one of the 17 postures (93.2% of accuracy).

With respect to all these studies, the aim of this paper is to implement a more powerful and fast model with a focus on its memory occupation to be able to detect subjects and the 17 different postures.

III. Processing Pipeline

The processing pipeline is composed of 4 main steps and is summarized by the diagram Fig. 1.

First of all, for each subject, we have one text file per posture, each line of which contains a pressure matrix of dimension 32 x 64. Thus, the first step was to transform the raw data into usable data with the correct labels (Data Loading). The second step was to normalize and filter the images to improve the accuracy and to accelerate the learning of the neural networks (Data Pre-processing). The third step was to train different models on the data and compare their performance using metrics, running time and complexity (Learning Framework). In parallel, a 10-fold cross validation was performed on the best model to obtain more reliable results. Finally, the fourth step is the multi-label classification to predict two classes with the same model and enable the identity and posture recognition.

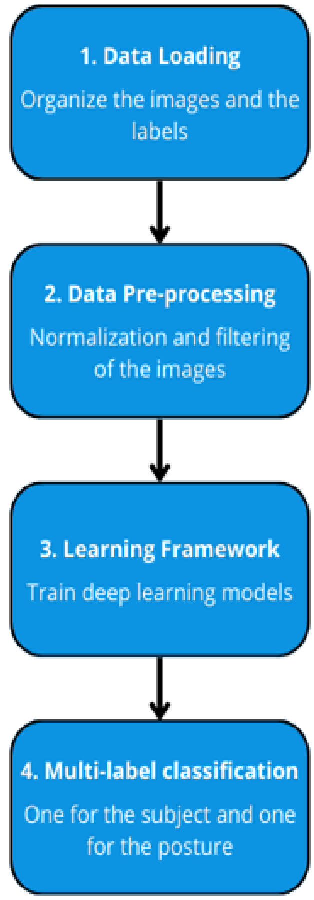

Fig. 1: Processing pipeline

IV. Signals and Features

I used the first experiment from the public dataset PmatData [11]. Data is collected using Vista Medical FSA SoftFlex 2048. The dataset includes pressure data from 13 participants in 8 standard postures and 9 additional uncommon postures. It contains 13 files corresponding to the 13 participants, with each file containing 17 documents corresponding to the 17 postures. Each document contains several pressure matrices collected at a sampling rate of 1 Hz. Each row is a new pressure matrix of size 32 x 64, and the sensors report numbers in the range [0-1000]. The age of participants was between 19 and 34 years, with a height between 170 and 186 cm and a weight between 63 and 100 Kg. In addition, there are three main postures: supine, right side and left side.

In the pre-processing step, I began by transforming the raw data into interpretable data for the deep learning models. I extracted all the 32 x 64 pressure matrices and created a unique document for each of them with an identifier name. In addition, I created a reference table that associates each identifier name with the label of the participant and the posture. I also removed each of the first and last three pressure matrices from the recordings, as the images were not usable. This may be due to the unstable positioning of individuals when the camera starts recording. Then, I applied a gaussian filter to the pressure matrices with a parameter $\sigma = 0.5$ in order to reduce the noise caused by occasional malfunctioning pressure sensor and to smooth the pressure sensor values. Finally, I performed a MinMax normalization on each of the pressure matrices to obtain pressure sensor values between 0 and 1. This helps the numerical stability of the models, accelerates gradient descent and avoids the vanishing gradient problem.

The Fig. 2 shows the effect of pre-processing on different example pressure matrices.

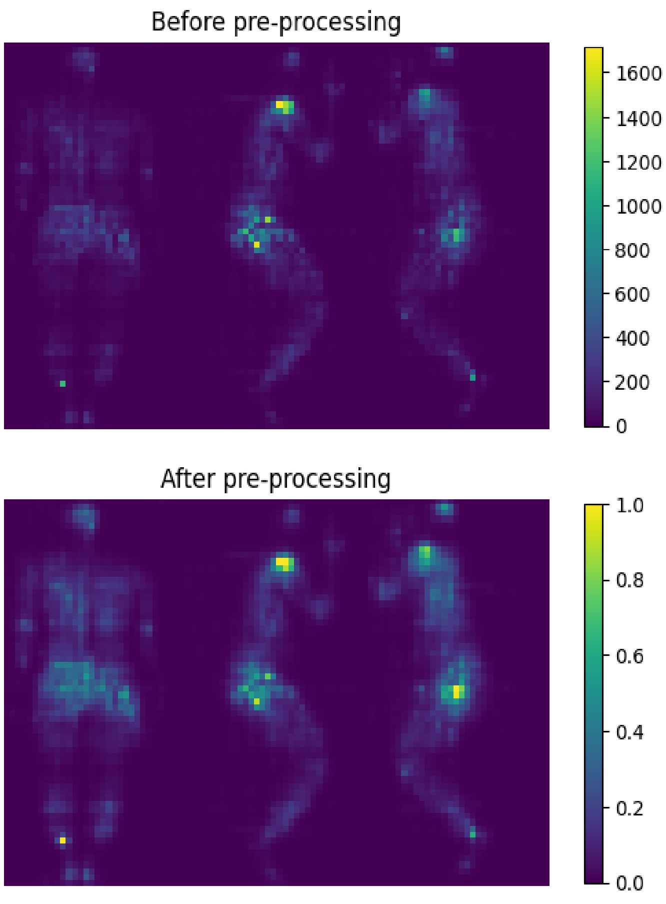

Fig. 2: Pre-processing

To train and evaluate the performance of the different models, I split the dataset into a training set (70%: 13243 examples), a validation set (20%: 3784 examples) and a test set (10%: 1892 examples).

V. Learning Framework

Different neural network architectures were trained using TensorFlow on a T4 GPU from Google Colaboratory free version. Then, they were evaluated according to various performance and complexity criteria. The diagram in Fig. 3 provides a visual reference of the different models proposed.

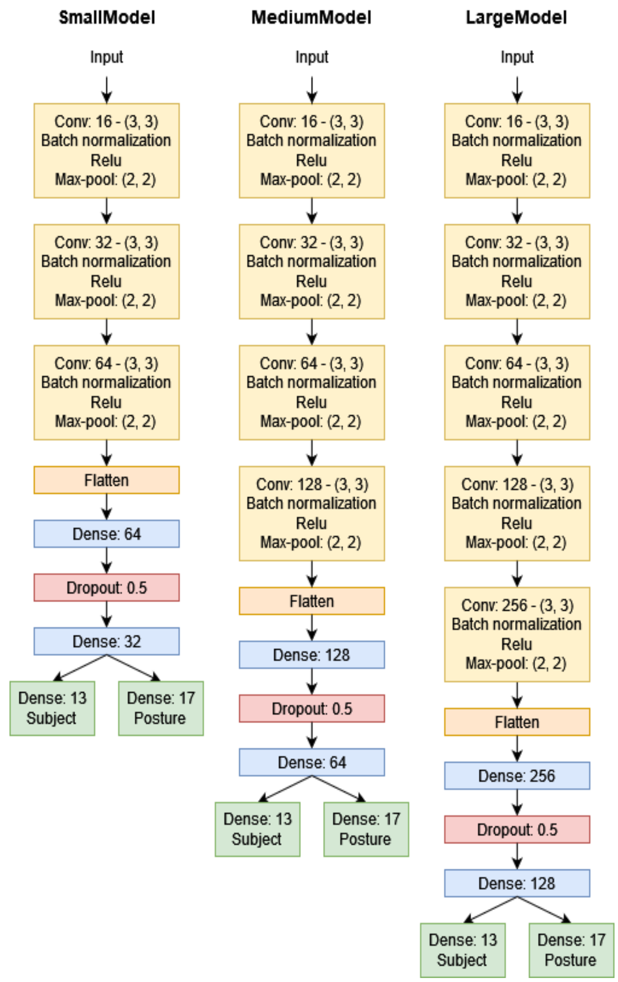

Fig. 3: Different models proposed

The models are composed of an input layer corresponding to the pressure matrix of size 32 x 64 x 1 and several blocks of convolution, batch normalization, activation relu and max pooling layers. The models are implemented with a base of three 2D convolutional layers, respectively with 16, 32 and 64 filters. The medium model adds one 2D convolutional layer with 128 filters and the large model adds two 2D convolutional layers with 128 filters and 256 filters. Every filter of the layers has a size of (3, 3) and it is applied with a stride of (1, 1). In addition, the padding parameter is set to *same* in each layer in order to keep the output size. These convolution layers are used to extract the presence of local or global features in the pressure matrix. Then, after each convolution layer, a batch normalization layer is applied, followed by a relu activation layer to obtain fast and robust results but also to avoid vanishing gradients. Finally, a 2D max pooling layer is applied with a pool size of (2, 2) to reduce image size while preserving important features. Then, after repeating these different blocks, a flattening layer is added to correctly format the input for the next layers. In addition, two fully connected layers are implemented with a relu activation function. The number of neurons in the first layer corresponds to the number of filters in the convolution layer of the last block, and the number of neurons in the second layer is the number of neurons in the first layer divided by two. Between these two layers, a dropout layer is inserted to deactivate 50% of the neurons in the first fully connected layer, helping the model to generalize and reduce overfitting. Finally, classification is achieved by two dense parallel layers of 13 and 17 neurons respectively, each with a softmax activation function to classify subjects and postures separately.

The different neural networks were trained over 8 epochs with a batch size of 32. A loss function sparse categorical cross entropy was used for the subject and another for the posture. For training the models, we consider a loss function that combines the two losses and minimizes both at the same time:

$$\mathcal{L} = \mathcal{L}_{subject} + \mathcal{L}_{posture}. \tag{1}$$

Training was performed on the training set and for each epoch, the loss function and accuracy for subject and posture recognition on the validation set were monitored and analyzed. In addition, two callbacks were used: the first was EarlyStopping to stop training if the validation loss did not decrease after 3 epochs, and the second was TimingCallback, a manually-created callback to accurately measure the training time of the different models for later comparison. Moreover, Adam optimizer was used with the default learning rate 0.001 and with a focus on model accuracy. In fact, I have chosen accuracy because the posture data are very well balanced: the right class is the one with the fewest examples (1064) and the supine star class is the one with the most examples (1189); the subject data are also globally balanced.

At the end of training, classification capability was evaluated on the validation set according to several scores: accuracy of subject and posture, recall, f1-score, training time. Moreover, model complexity and memory occupation are taken into account to determine the best model.

Finally, after selecting the best model according to the validation set, the best model was used to classify the test set data and acquire further performance measures. To confirm the performance of the best model, a 10-fold cross validation was also performed.

VI. RESULTS

A. Hyper-parameters validation

In order to determine the best model, I have chosen to train different neural network architectures (SmallModel, MediumModel, LargeModel). These models differ in the number of blocks convolution, batch normalization, activation relu and max pooling. In addition, for each of them, I activated or deactivated the dropout layer to evaluate the effect on the models. The Tab. 1 shows the accuracy results of the different models on the validation set. As highlighted previously, the data are fairly balanced between classes, which is why we can compare them with the accuracy metric.

Model	**Subject acc**	**Posture acc**
SmallModel	0.9992	0.9987
SmallModelWithDropout	0.9609	0.9110
MediumModel	0.9989	0.9992
MediumModelWithDropout	0.9987	0.9989
LargeModel	0.9992	0.9995
LargeModelWithDropout	0.9984	0.9987

TABLE 1: Performance of the different models

We can see that the models perform very well, especially the medium and large models. The dropout affects performance for the small model, perhaps because it has fewer neurons in the fully connected layers. On the other hand, performance is not affected for the medium and large models with the dropout, and I'm convinced of the importance of a dropout layer for preventing overfitting while training deep neural networks. So far, the medium and large models with dropout look promising.

The graph in Fig. 4 shows the training times of the different models.

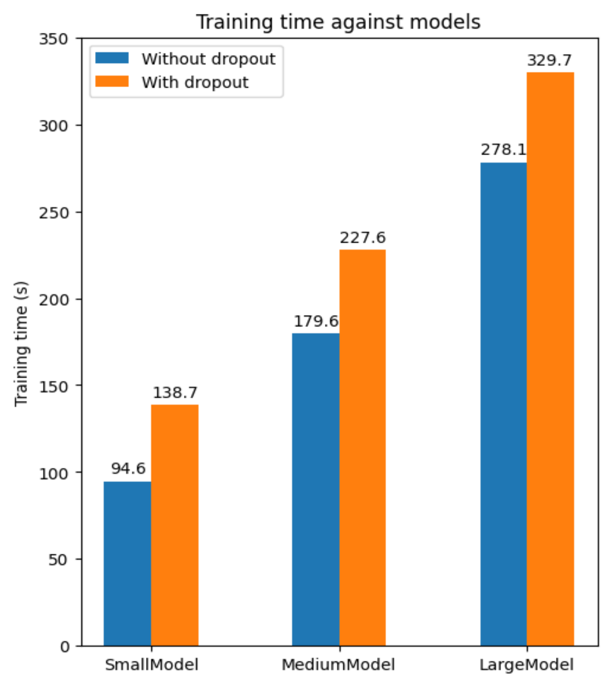

Fig. 4: Training times of the different models

The training time over 8 epochs with a T4 GPU is almost 4 minutes for the medium model with dropout and almost 6 minutes for the large model with dropout. In addition, we can see that applying a dropout layer to a neural network typically increases the training time by around 50 seconds, which is information worth taking into account.

The Tab. 2 shows the size and number of parameters of the different models.

Model	Model size (MB)	Parameters
SmallModel	1.89	157950
MediumModel	2.84	239518
LargeModel	6.54	562398

TABLE 2: Complexity of the different models

We can see that all the models have low memory requirements, which is a considerable advantage. Furthermore, the medium model is twice as light as the large model, and its architecture has half as many parameters.

Based on the results obtained so far, we choose the medium model with a dropout layer because it gives us excellent results on the validation set, it has relatively fast training time, it is a compact model in terms of memory occupation / number of parameters and finally because it prevents overfitting with the dropout layer.

B. Results of the medium model with dropout on the test set

Once our medium model with dropout has been trained on the training set (70%) and selected with the validation set (20%), we will now evaluate its performance on the test set (10%). The number of examples to be predicted is 1892, corresponding to 60 steps with a batch size of 32. The time to predict each step is 0.12 seconds and each image is 0.0038 seconds, which is also very fast. Analysis of the results of the medium model's prediction with dropout on the test set shows that it made no error in predicting the classification of the subject among the 13 possible classes, and also no error in predicting the classification of the posture among the 17 possible classes: we have 100% accuracy. The model significantly outperforms and I carried out a 10-fold cross validation to confirm these results.

C. 10-fold cross validation of the medium model with dropout

To confirm our results, I applied a 10-fold cross validation. We split the data into 10 subsets, where 90% was used for training and 10% was used for testing. The process was repeated 10 times. We compared the performance of our model on the 3 main postures (supine, right side, left side) with 10-fold cross validation, to [11] and [12]. These research papers also use 10-fold cross validation on the 3 main postures. The research paper [11] uses a separately trained model on each of the 3 main postures and the research paper [12] uses a deep convolutional neural network. The results are presented in Tab. 3 and we note that our model significantly outperforms [11] and is equivalent to [12] on performance for the 3 main postures. In fact, our proposed model makes only 6 errors on the 3 postures with 10-fold cross validation.

	Supine	Right	Left
Ref. [11]	85.5	80.4	82.3
Ref. [12]	99.9	100	100
Proposed method	99.9	99.9	99.9

TABLE 3: Accuracy over 3 postures with 10-fold cross validation (in %)

The Tab. 4 shows overall performance according to several metrics on subject identification and recognition of all postures (a total of 17) with 10-fold cross validation.

	Accuracy	Precision	Recall	F1
Subject	99.82	99.81	99.79	99.79
Posture	99.83	99.83	99.84	99.83

TABLE 4: Overall performance with 10-fold cross validation (in %)

Our proposed model achieved a subject identification accuracy of 99.82% and posture recognition of 99.83% among 17 postures which is highly significant. In comparison, the model presented in the recent paper [12] achieved a subject identification accuracy of 89.7% and posture recognition of 93.2% among 17 postures using data augmentation. Our proposed model is particularly impressive in its capability to perform both for the 3 main postures and for all postures.

The Fig. 5 shows the confusion matrix for posture recognition with the model proposed for this experiment.

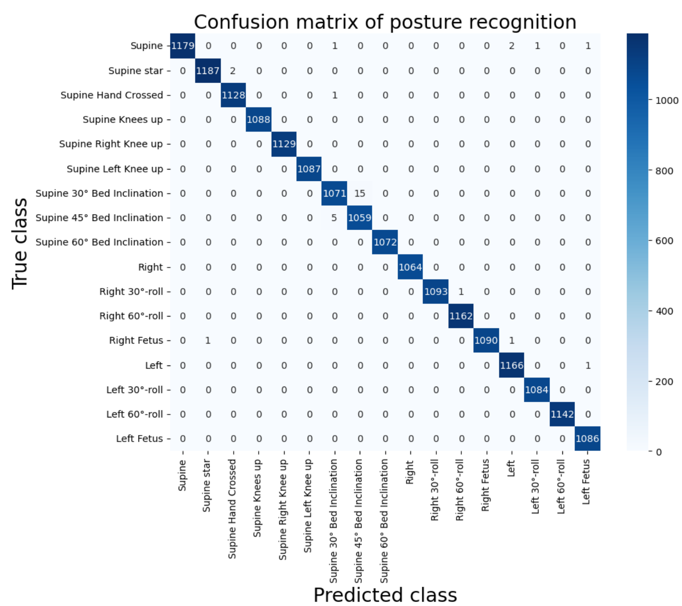

Fig. 5: Confusion matrix for posture recognition with 10-fold cross validation

The confusion matrix provides a closer look at our predictions. It reveals that most of the misclassified sub-posture fall within the correct main posture (i.e. supine, right side, and left side). For example, the model sometimes mispredicts the supine position inclined by 30° instead of 45°, and vice versa.

VII. Concluding Remarks

Sleep has been an important topic of research in recent years and tracking sleep posture over time can help prevent many diseases. This paper defined an antomatic and sustainable method of pre-processing for pressure sensing mattresses data. In addition, different neural network architectures were presented, and the medium model with the dropout layer was selected as the most promising. In fact it achieved impressive results with 99.8% accuracy for the identification of subjects and the recognition of 17 different possible in-bed postures. It also has the advantage of being robust and compact, with a training time of less than 4 minutes and a prediction time of 0.12 seconds per batch of 32 images. Medical and smart home environments can use the methods and algorithms presented in this paper as a complementary tool to their existing systems. Future work may involve using pressure sensing mattresses data with other tools such as respiration rate or heart rate to accurately target patient symptoms and reliably predict potential risks of associated diseases.

To conclude this paper, I wanted to present some final thoughts on what I have learned and the challenges encountered. I gained experience on programming and problem solving. For example, one big challenge I met was being able to predict two classes at the same time. I had to change all the data structures in the input model when creating the datasets, modify the loss functions and compile the model. Another important point was developing models from scratch and finding the best one by changing the hyperparameters. I realized about the importance of convolution neural network architecture. In addition, the 10-fold cross validation took time and I learnt how to separate the data, trained several models and recovered performance measurements. Finally, this project was an opportunity to develop my skills in time management and introduced me to the world of research.

References

[1] B. M. Altevogt and H. R. Colten, "Sleep disorders and sleep deprivation: an unmet public health problem," *National Academies Press*, 2006.

[2] R. D. Cartwright, "Effect of sleep position on sleep apnea severity," *Sleep: Journal of Sleep Research & Sleep Medicine*, 1984.

[3] C. H. Lee, D. K. Kim, S. Y. Kim, C.-S. Rhee, and T.-B. Won, "Changes in site of obstruction in obstructive sleep apnea patients according to sleep position: a dise study," *The Laryngoscope*, vol. 25, pp. 248–254, Jan. 2015.

[4] S. Bhattacharya1 and R. K. Mishra, "Pressure ulcers: Current understanding and newer modalities of treatment," *Indian Journal of Plastic Surgery*, Jan. 2015.

[5] 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," *Sleep*, vol. 20, no. 6, pp. 388–395, 1997.

[6] S. Ostadabbas, M. B. Pouyan, M. Nourani, and N. Kehtarnavaz, "In-bed posture classification and limb identification," *IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, pp. 133–136, Dec. 2014.

[7] A. Sadeh and C. Acebo, "The role of actigraphy in sleep medicine," *Sleep medicine reviews*, vol. 6, pp. 113–124, Apr. 2002.

[8] H. J. Lee, S. H. Hwang, S. M. Lee, Y. G. Lim, and K. S. Park, "Estimation of body postures on bed using unconstrained ecg measurements," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, pp. 985–993, Nov. 2013.

[9] S. Boughorbel, F. Bruekers, and K. de Groot, "Pressure-sensor system for sleeping-posture classification," *Proceedings of Measuring Behavior*, p. 258, Aug. 2012.

[10] Q. Sun, E. Gonzalez, and Y. Sun, "On bed posture recognition with pressure sensor array system," *IEEE SENSORS*, pp. 1–3, Oct. 2016.

[11] M. B. Pouyan, J. Birjandtalab, M. Heydarzadeh, M. Nourani, and S. Ostadabbas, "A pressure map dataset for posture and subject analytics," *IEEE-EMBS International Conference on Journal of Biomedical & Health Informatics (BHI)*, Sept. 2017.

[12] V. Davoodnia and A. Etemad, "Identity and Posture Recognition in Smart Beds with Deep Multitask Learning," *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct. 2019.