Classification of Sleep Videos Using Deep Learning

Jeehyun Choe* A. J. Schwichtenberg† Edward J. Delp*
* Video and Image Processing Laboratory, School of Electrical and Computer Engineering,
Purdue University, West Lafayette, Indiana USA
† Department of Human Development and Family Studies,
Purdue University, West Lafayette, Indiana, USA

Abstract

Videosomnography (VSG) is a group of video-based methods used to record and label sleep versus awake states in humans. Traditional behavioral-VSG (B-VSG) labeling requires visual inspection of the video by a trained technician to determine whether a subject is sleep or awake. B-VSG is not used to label sleep stages (e.g., slow wave or REM sleep), rather it solely labels whether a subject is asleep or awake at a particular time. In this paper we describe an automated VSG sleep detection system which uses deep learning approaches to label frames in a sleep video as "sleep" or "awake" in young children. We examine 3D Convolutional Networks (C3D) and Long Short-term Memory (LSTM) relative to motion information from selected Groups of Pictures of a sleep video and test temporal window sizes for back propagation. We compared our proposed VSG methods to traditional B-VSG sleep-awake labels. C3D had an accuracy of approximately 90% and the proposed LSTM method improved the accuracy to more than 95%. The analyses revealed that estimates generated from the proposed LSTM-based method with long-term temporal dependency are suitable for automated sleep or awake labeling.

1. Introduction

Pediatric sleep medicine is a field that focuses on typical and atypical sleep patterns in children. Within this area, a video recording of a sleeping child (a sleep video) is used to analyze sleep duration, fragmentation, and timing. One analysis method is videosomnography (VSG) which includes the labeling of sleep vs. awake intervals from the sleep video [13, 23, 17, 12, 25]. VSG is commonly used for infants/toddlers or children with sensory sensitivities because their compliance rates with other (more invasive) sleep analysis methods can be low [13, 17, 12]. Traditional behavioral videosomnography (B-VSG) includes manual labeling of the sleep video as "sleep" or "awake" by a trained technician [23]. B-VSG is not used to label sleep stages (e.g., slow wave or REM sleep), rather it solely labels whether a subject is asleep or awake at a particular time. B-VSG labeling is time consuming and expensive and because of this it has had limited use within the pediatric sleep medicine field. Actigraphy, the use of a wearable accelerometer, is another commonly used method to analyze sleep/awake. In many cases children will not tolerate wearing the accelerometer while sleeping. Previous studies document actigraphy's accuracy in detecting sleep but relatively low specificity for detecting night waking [22]. Polysomnography (PSG), which monitors many body functions including brain (EEG), eye movements (EOG), and heart rhythm (ECG) is the gold standard for sleep analysis but it does not capture typical sleep well [17, 26]. It is expensive and pediatric use can have low compliance. For this reason, PSG is not the most common sleep method used in homes or research.

In this paper we describe an automated VSG method, also known as auto-VSG, to replace or assist B-VSG , while maintaining high levels of accuracy. It is important to note that our goal is to label each frame of a sleep video with the label "sleep" or "awake." In this work we are not interested in labeling sleep stages, such as REM sleep.

Auto-VSG is a growing area in sleep analysis with preliminary studies using signal/image processing systems that use motion during sleep for sleep/awake labeling [13, 10, 19, 16, 18, 25]. In these studies, motion is estimated using frame differencing [19, 18] or motion vectors [13, 10, 16]. However, each of these studies were conducted in a controlled setting and do not account for the wide range of camera positions and lighting variations that are common among in-home VSG recordings. In [25] we describe preliminary motivation for the use of an auto-VSG to label sleep/awake but the system requires user input when the camera position changes.

In this paper, we propose a new approach to classify in home sleep videos as sleep vs. awake that adjust for

978-1-7281-1198-8/19/$31.00 ©2019 IEEE
DOI 10.1109/MIPR.2019.00028

these 'in the wild' factors using deep learning. The contributions in this paper are: (1) we describe the key factors in sleep video classification (i.e., movements over long period of time) that are not addressed in commonly used action classification problems (Section 2) (2) we propose a sleep/awake classification system with a recurrent neural network using simple motion features (Section 3) (3) we experimentally show our system successfully learns long-term dependencies in sleep videos and outperform one of the recent method that has been successful in public action dataset (Section 4).

2 Related Work

2.1 Motion and Long Term Dependencies in VSG

One common assumption that can be used to classify sleep vs. awake is that there is less motion of the subject during sleep than when awake [2]. However, sleep and awake patterns are not that simple. To tell whether the subject is sleep or awake, not only the current movements matter but it also depends on what the movement patterns are in long term time interval.

Typically in VSG, sleep onset is established based on information from more than 20 minutes of observed video and awakenings must include purposeful movements and be more than one minute in duration. Similarly, actigraphy methods use both a motion index (the amount of motion within a time segment [2]) and information about the duration before and after the target minute [24]. Both movement and temporal information are needed to accurately capture sleep and awake states.

2.2 Long Short- Term Memory Networks (LSTM)

A Recurrent Neural Network (RNN) is a deep learning network used for processing sequential data by forming a memory through recurrent connections from the previous inputs to the current output [21, 8]. Similar to Convolutional Neural Network (CNN) spatially sharing parameters, a RNN temporally shares parameters assuming that the same parameter can be used for different time increments (i.e., the conditional probability distribution over the variables at time t+1 given the variables at time t is stationary) [7].

For a standard RNN, the range of input sequence that can be accessed is quite limited in practice because of the "vanishing gradient" effect (VGE) [11, 8]. VGE is a problem that gradients propagated over many recurrent connections tend to vanish mainly due to the exponentially smaller weights given to long-term interactions (involving multiplication of many Jacobians) compared to short-term ones [7]. Long Short-Term Memory Networks (LSTM) [11, 9] is a special type of RNN which enables long-range learning by reducing VGE. LSTM uses a structure known as gates that can regulate the removal or addition of information. It is based on the idea of creating paths through time that have derivatives that neither vanish nor explode [7]. While the repeating module in a standard RNN contains a single layer, the one in LSTM contains four interacting layers–forget gate layer, input gate layer, update layer, and output layer. LSTM has been widely used for processing various sequential data and has been successful in language processing such as speech/text recognition, and machine translation.

2.3 Video Classification Using Deep Learning

The interest in image classification using deep learning began in 2012 with the ImageNet challenge [15], video classification using deep learning is still in the early stages with many recent studies focused on specific set of actions [14, 3, 4, 27]. These methods make use of the basic idea in Convolutional Neural Networks (CNN) classification approaches in still images to solve video classification problems. Karpathy *et al.* [14] presented slow fusion methods for large scale video classification using CNNs. This was one of the early works on deep learning video classification to extend the connectivity of the CNN in the time dimension to learn spatio-temporal features. Another approach incorporating temporal information is the Long-term Recurrent Convolutional Networks (LRCN) proposed by Donahue *et al.* [3, 4]. They first obtained visual features from each frame using a conventional CNN and then used the features as inputs to the recurrent models. The advantage of LRCN [3, 4] is that it can learn unique appearance in video while also learning temporal patterns of variable lengths. However in LRCN, the the spatial and temporal information is processed in two separate steps that there is a limit to learning spatial changes over time (i.e. motion). Another approach for human action recognition is spatio-temporal CNN filters (C3D) proposed by Tran *et al.* [27]. C3D extends the conventional CNN with an additional temporal dimension by using 3-dimensional CNN kernels in all convolutional layers. C3D [27] uses one network to learn both spatial and temporal information at the same time by using 3-dimensional convolution kernels that include the temporal dimension. This network can learn motion changes over time, but with limited temporal range (e.g. the length is fixed to 16 consecutive frames at a time). It was reported in [27] that C3D performed similar or better compared to other methods including deep networks [14] and LRCN [3] on an action recognition public dataset UCF101 [1]. While there has been improvements for specific action recognition datasets, whether these methods can be generalized for use in other types of video classification problems is an open question.

In our method, we designed the system to learn temporal motion changes over a longer period time by using simple motion feature rather than learning the appearance over short period of time like in CD3. By doing this, we could achieve better performance while removing the need for massive amount of training dataset. Our system is described in section 3.

3 Proposed Method

In this section we describe our proposed method for labeling frames of a sleep video as "sleep" or "awake" from RGB/infrared videos using motion information. Figure 1 shows our system. First, we define consecutive video

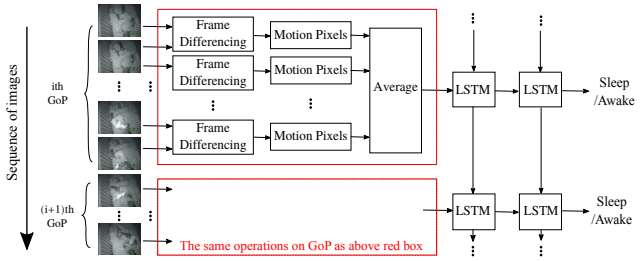

Figure 1. Proposed Sleep Detection System: Sleep/Awake Using a Motion Index and LSTM.

frames in small groups as Group of Pictures (GoP). The proposed system uses frame differencing within GoP to obtain motion information (described in detail in Section 3.1) and two-layer LSTM architecture to incorporate information from previous video GoPs.

3.1 Motion Detection/Motion Index

We shall assume that the child is the only source of motion in the sleep video. Background subtraction is widely used for detecting movements from static cameras [20]. One of the simple background subtraction methods is frame differencing which detects motion in frame by taking the difference (subtraction) between the current frame and the previous frame (the background model). For a sequence of gray scale images in GoP at constant frame rate and size, we take the frame difference in each consecutive pair. This difference indicates whether each pixel in the frame is classified as "moved" or "not moved." A pixel is classified as "moved" if Equation (1) is true

$$|I_{i-1}[x,y] - I_i[x,y]| > T \tag{1}$$

where $I_i[x,y]$ is a pixel in frame i, and T is a threshold for determining movement for one pixel. For our experiments the value of T is empirically determined and the value we use is described in Section 4.2. We quantify the amount of motion as the number of pixels classified as "moved." We define the motion index for a GoP as the average of the amount of motion for each frame pair in the GoP. The red box shown in Figure 1 is the motion detection block and the output of this block is the motion index for each GoP.

We minimize the use of the empirically driven parameters (only using one parameter T) by using deep learning methods that learn the sleep vs. awake patterns based on the motion index.

3.2 Loss Function

For an imbalanced dataset where one class has much larger number of samples than the other class, the trained model can be biased toward the class in the majority. A typical sleep video dataset is imbalanced where the number of sleep labels dominates awake labels. To compensate for this data imbalance, class-wise weights can be set in the loss function. We define the weight w_j for the class (sleep or awake) j as

$$w_j = 1 - \frac{n_j}{\sum_j n_j} \tag{2}$$

where n_j is the number of samples in class j. The idea is that when there is more data for class j, a smaller weight is assigned. Using these weights, the weighted softmax cross entropy loss function for sequence data $(x_0, y_0), \cdots, (x_i, y_i), \cdots, (x_n, y_n)$ is defined as

$$L = \sum_i w_{y_i} \left(-\log \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \right) \tag{3}$$

where y_i is the actual class index for the sample x_i, and f_j is the predicted probability of x_i belonging to class j. With uniform weights across classes ($w_0 = \cdots = w_j = \cdots$), the loss function L becomes the regular softmax cross entropy.

4 Experiments

4.1 Dataset

Our sleep dataset consists of in home sleep videos of 30 different nights for various children. The sleep videos are for children from 9 to 30 months of age. Each night is a different sleep video sequence of a different child. Our total data set consists of 30 children for 30 nights. The camera used for recording is Swann ADW-400 Digital Guardian Camera & Recorder. It records in color mode during the day and switches to black and white/infrared mode at night. This project was approved by the Purdue University Institutional Review Board. The sleep videos have spatial resolutions of 320×240 pixels at 13-16 frames/s (fps) or 640×480

pixels at 7-10 fps. The entire night is recorded as a sequence of videos with time stamps embedded in the video frame and the length of each video is 10 minutes and 14 seconds. Along with the sleep videos, B-VSG labels for sleep onset, offset, and awakenings were used in the analyses. This information was obtained as ground truth from trained observers. We did not use the audio due to too much noise in the signal.

For preprocessing, videos were sub-sampled at 4 fps. Then, the GoP (16 frames) were obtained. While the B-VSG labels are in units of minutes, a GoP in our settings corresponds to a 4-second duration. GoPs that do not fully belong to sleep or awake (i.e., partly Sleep and partly Awake GoPs), were not used in the experiment. How we divided the sleep dataset into training and testing sets is shown in Table 1. As we can see from Table 1, there is an imbalance between the two classes of "Sleep" and "Awake". For the training set of 20 children, the number of continuous sequences were 33 and the length ranged from 378 to 11,352 GoPs. In case where a child had some "out of bed" time, the corresponding GoPs were excluded from our training/test sets hence resulting in multiple sequences for one child.

Table 1. Training/Test Set Division of Sleep Dataset.

Sleep Dataset	# GoPs for Sleep	# GoPs for Awake	# GoPs in total
Train (20 children)	179,108	13,691	192,799
Test (10 children)	88,234	7,781	96,015
Total (30 children)	267,342	21,472	288,814

4.2 Implementation Details

For the motion index threshold T we used $T = 30$ (11.7% difference in gray scale intensity levels) and image size of 320×240 in gray scale for obtaining the motion index for all the GoPs. Our Long Short- Term Memory Network (LSTM) described in Section 2.2 was implemented using Python and TensorFlow. For the LSTM, we used a hidden unit size of 128 and 2 layers of cells with dropout layers with probability of 0.5. The softmax cross entropy loss was used as the cost function for training. The Adagrad method [5] was used for gradient descent optimization. To reduce computational complexity, we organized all the training set as a sequence of GoPs and put them in mini-batches of size 30. Since the number of training GoPs is 192,799 and is not dividable by our batch size 30, the remaining last 19 GoPs were discarded hence resulting in a $30\times$6,426 matrix. The GoPs in the first column (the first batch) were assumed to be the start of each sequence although it would not exactly match with the actual start of the sequence for each child. The size of each mini-batch was $30\times$(back propagation window size). The network is initialized with a vector of zeros and gets updated after reading each GoP. The number of epochs we used for training is 10.

For training and testing the spatio-temporal CNN filters (C3D), we used the implementation provided in Caffe [27].

4.3 Results

To assess the performance , we used five metrics, which are Accuracy, $ACC = \frac{TP+TN}{TP+TN+FP+FN}$, Precision, $PRE = \frac{TP}{TP+FP}$, Recall, $REC = \frac{TP}{TP+FN}$, Specificity, $SPEC = \frac{TN}{TN+FP}$ and Cohen's kappa (κ). The test dataset is shown in Table 1. Table 2 and Table 3 show the results for different classification models on the same test set. Table 2 is the result using uniform weights for the loss function. In this table, C3D-f is C3D pre-trained on Sports-1M dataset [14] and finetuned on our sleep dataset. C3D-t is C3D trained on our sleep dataset from scratch. Our proposed models are denoted as LSTM-k, where the number k refers to the size of the back propagation window in the unit of number of GoPs that is used during training the model. The duration of one GoP in unit of seconds is $16 \text{ frames}/4 \text{ fps} = 4$ seconds (i.e., 5 GoPs are 20 seconds, and 30 GoPs are 2 minutes). Table 2 shows the result for models trained with regular softmax cross entropy loss function. Compared to using one GoP at a time for classification (i.e., C3D-f and C3D-t), using multiple GoPs (i.e., LSTM-k) improved accuracy while maintaining high recall. LSTM-5 improved the performance across all five metrics. However, the specificity is low due to the data imbalance. Since the model is trained to minimize the overall loss including both sleep and awake GoPs, the specificity that involves only awake GoPs is not giving consistent results. Next, Table 3 shows the result for models trained using weighted loss as described in Section 3.2. We can see that the specificities are improved.

Table 2. Results [%] for the number of test GoPs $n = 96,015$. Models trained using loss with uniform weights.

Model	ACC	PRE	REC	$SPEC$	κ
C3D-f	84.25	93.52	89.03	30.07	0.15
C3D-t	89.54	93.29	95.49	22.05	0.20
LSTM-5	**95.60**	**96.43**	**98.87**	**58.55**	**0.66**
LSTM-15	92.89	92.87	99.93	12.98	0.21
LSTM-30	94.31	94.83	99.23	38.62	0.50
LSTM-50	92.77	92.71	99.99	10.90	0.18
LSTM-75	93.51	93.61	99.74	22.85	0.34
LSTM-85	92.53	93.56	98.66	22.95	0.30

Table 3. Results [%] for the number of test GoPs $n = 96,015$. Models trained using weighted loss.

Model	ACC	PRE	REC	$SPEC$	κ
LSTM-5	93.33	97.81	94.87	**75.88**	0.61
LSTM-15	93.46	97.75	95.06	**75.22**	0.62
LSTM-30	95.47	96.70	98.44	**61.88**	0.67
LSTM-50	95.58	95.79	99.57	50.31	0.63
LSTM-75	20.48	98.62	13.66	97.83	0.02
LSTM-85	92.69	93.85	98.51	26.74	0.34

There are good agreements between traditional B-VSG and our proposed methods (LSTM-5 on loss with uniform weights is $\kappa = 0.66$ in Table 2, and LSTM-5/15/30/50 on weighted loss are $\kappa > 0.6$ in Table 3). On the rest of the methods including C3D, there are fair to poor agreements ($\kappa < 0.4$).

Figure 2 and 3 are the ROCs [6]. Unlike accuracy and precision, ROCs are insensitive to changes in class distribution since it is based upon True Positive rate and False Positive rate. Note that in Table 2 and Table 3 recall and specificity are obtained based on the discrete outputs generated with a threshold of 0.5–we take the predicted class as the one with the higher probability. For the ROCs, we used the monotonicity of thresholded classifications [6]. Figure 2

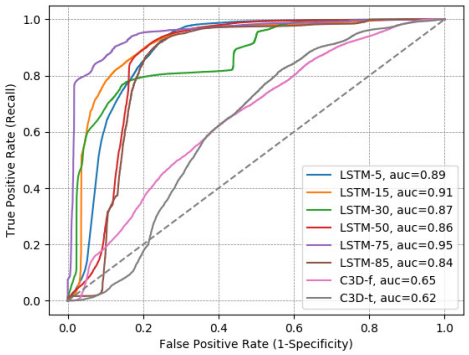

Figure 2. ROCs. Models trained using loss with uniform weights.

shows that all the proposed methods (LSTM-k) have higher Area Under the ROC (AUC) than the C3D models. Except for the AUC drop at $k = 85$ due to the long back propagation stages in the training, the rest of all the LSTM-k models have AUC higher than 0.85. C3D-t and C3D-f have AUC of 0.62 and 0.65 respectively both giving much lower performance compared to the proposed methods. LSTM-75 where 75 GoPs corresponds to 5-minute duration gave the highest performance (AUC=0.95). Figure 3 shows the result

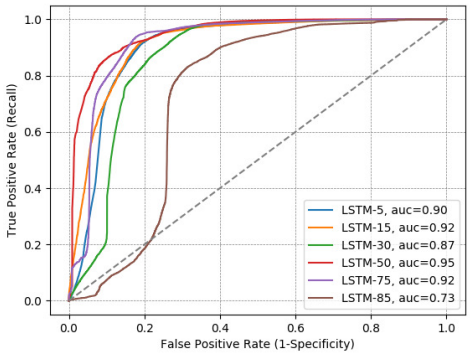

Figure 3. ROC curve. Models trained using weighted loss.

for the models trained using the weighted loss described in section 3.2. Even though LSTM-75 has low accuracy in Table 3 for specific threshold, the overall performance shown in Figure 3 is similar to other models (AUC=0.92). For the models trained using the weighted loss, LSTM-50 (duration of 3 minutes and 20 seconds) gave the highest performance (AUC=0.95). The overall results show the significance of using the long-term temporal motion information in sleep vs. awake classification. For sleep video classification, the proposed method outperforms the recent video classification methods even though it used minimal visual information of only one motion index for each GoP.

C3D is good for classifying unique appearance and short action in each class by learning spatio-temporal features in videos but due to the limited temporal range it takes, C3D did not work well for classifying sleep videos that have long temporal dependencies. Also, due to the slight changes in appearance between the sleep/awake states and few actions in sleep videos, learning appearance pattern in C3D did not contribute well on improving the overall performance. Our proposed method enabled capturing the temporal history of motion changes by using LSTM on sequence of GoPs and simple motion feature for each GoP.

5 Conclusions

In this paper, we described a system for sleep vs. wake classification that utilize long-term dependency and data imbalance. From the prior knowledge that motion is the key factor for determining sleep versus awake in B-VSG, we described a motion index to summarize the motion information for each GoP and then combined this with the recurrent model to label each GoP as asleep or wake. Our experiment demonstrated interesting results that using LSTM with simple motion feature for GoP outperformed one of the latest general video classification methods for sleep vs.

awake video classification. We also showed how weighting the loss function can affect various performance metrics for imbalanced sleep dataset. The design of our system is based on the prior knowledge in sleep medicine (i.e. the motion changes over long duration is the key factor in determining sleep vs. wake) and in signal processing (i.e. methods for simple motion feature in video). For future work, more general video classification methods that require less prior knowledge should be further investigated.

References

[1] UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CRCV-TR-12-01*, 2012. University of Central Florida, Orlando, FL.

[2] S. Ancoli-Israel, R. Cole, C. Alessi, M. Chambers, W. Moorcroft, and C. Pollak. The role of actigraphy in the study of sleep and circadian rhythms. *SLEEP*, 26(3):342–392, May 2003.

[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. 2014.

[4] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017.

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, July 2011.

[6] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[8] A. Graves. Supervised sequence labelling with recurrent neural networks. 385, 2012.

[9] A. Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850v5*, pages 1–43, June 2014.

[10] A. Heinrich, X. Aubert, and G. Haan. Body movement analysis during sleep based on video motion estimation. *Proceedings of the IEEE 15th International Conference on e-Health Networking, Applications and Services*, pages 539–543, October 2013. Lisbon, Portugal.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] D. Hodge, A. M. Parnell, C. D. Hoffman, and D. P. Sweeney. Methods for assessing sleep in children with autism spectrum disorders: A review. *Research in Autism Spectrum Disorders*, 6(4):1337–1344, October 2012.

[13] O. S. Ipsiroglu, Y. A. Hung, F. Chan, M. L. Ross, D. Veer, S. Soo, G. Ho, M. Berger, G. McAllister, H. Garn, G. Kloesch, A. V. Barbosa, S. Stockler, W. McKellin, and E. Vatikiotis-Bateson. "diagnosis by behavioral observation" home-videosomnography–a rigorous ethnographic approach to sleep of children with neurodevelopmental conditions. *Front Psychiatry*, 6(39):1–15, March 2015.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, June 2014. Columbus, OH.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Proceedings of Advances in Neural Information Processing Systems*, pages 1097–1105, December 2012.

[16] W. Liao and C. Yang. Video-based activity and movement pattern analysis in overnight sleep studies. *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 1–4, December 2008. Tampa, FL.

[17] M. Moore, V. Evans, G. Hanvey, and C. Johnson. Assessment of sleep in children with autism spectrum disorder. *Children (Basel)*, 4(72):1–17, August 2017.

[18] M. Nakatani, S. Okada, S. Shimizu, I. Mohri, Y. Ohno, M. Taniike, and M. Makikawa. Body movement analysis during sleep for children with adhd using video image processing. *Proceedings of the IEEE 35th Annual International Conference on Engineering in Medicine and Biology Society*, pages 6389–6392, July 2013. Osaka, Japan.

[19] S. Okada, Y. Ohno, Goyahan, K. Kato-Nishimura, I. Mohri, and M. Tanike. Examination of non-restrictive and non-invasive sleep evaluation technique for children using difference images. *Proceedings of the IEEE 30th Annual International Conference on Engineering in Medicine and Biology Society*, pages 3483–3487, August 2008. Vancouver, BC.

[20] M. Piccardi. Background subtraction techniques: a review. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 4:3099–3104, October 2004.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, editors, *Parallel distributed processing: explorations in the microstructure of cognition*, volume 1, pages 318–362. MIT Press, Cambridge, MA, 1986.

[22] A. Sadeh. The role and validity of actigraphy in sleep medicine: An update. *Sleep Medicine Reviews*, 15(4):259–267, August 2011.

[23] A. Sadeh. III. sleep assessment methods. *Monographs of the Society for Research in Child Development*, 80(1):33–48, February 2015.

[24] A. Sadeh, K. M. Sharkey, and M. A. Carskadon. Activity-based sleep-wake identification: An empirical test of methodological issues. *SLEEP*, 17(3):201–207, April 1994.

[25] A. J. Schwichtenberg, J. Choe, A. Kellerman, E. Abel, and E. J. Delp. Pediatric videosomnography: Can signal/video processing distinguish sleep and wake states? *Frontiers in Pediatrics*, 6(158):1–11, May 2018.

[26] Sleep Research Society. *Basics of Sleep Behavior*. UCLA, Edinburgh, UK, 1993.

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, December 2015. Santiago, Chile.