Digging Deeper: Towards a better Understanding of Transfer Learning for Human Activity Recognition

Alexander Hoelzemann
Ubiquitous Computing, University of Siegen
alexander.hoelzemann@uni-siegen.de

Kristof Van Laerhoven
Ubiquitous Computing, University of Siegen
kvl@eti.uni-siegen.de

ABSTRACT

Transfer Learning is becoming increasingly important to the Human Activity Recognition community, as it enables algorithms to reuse what has already been learned from models. It promises shortened training times and increased classification results for new datasets and activity classes. However, the question of what exactly is transferred is not dealt with in detail in many of the recent publications, and it is furthermore often difficult to reproduce the presented results. Therefore we would like to contribute with this paper to the understanding of transfer learning for sensor-based human activity recognition. In our experiment use weight transfer to transfer models between two datasets, as well as between sensors from the same dataset. As source- and target- datasets PAMAP2 and Skoda Mini Checkpoint are used. The utilized network architecture is based on a DeepConvLSTM. The result of our investigation shows that transfer learning has to be considered in a very differentiated way, since the desired positive effects by applying the method depend very much on the data and also on the architecture used.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**; • **Computing methodologies → Machine learning algorithms**; **Neural networks**.

KEYWORDS

Human Activity Recognition; Transfer Learning

ACM Reference Format:
Alexander Hoelzemann and Kristof Van Laerhoven. 2020. Digging Deeper: Towards a better Understanding of Transfer Learning for Human Activity Recognition. In *Proceedings of the 2020 ACM International Symposium on Wearable Computers (ISWC '20), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3410531.3414311

ISWC '20 , September 12–16, 2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410531.3414311

1 INTRODUCTION

The recording of datasets is always associated with a very large investment of time and energy and is also always accompanied by significant costs. For this reason, the community has relatively few datasets at its disposal that are suitable for training neural networks due to their nature, in respect to scope, quality and reliability. Therefore, algorithms have been increasingly in the focus of research in recent years, which either allow to enrich datasets with information at low cost or to reuse information from already learned models, like Transfer Learning.

Transfer Learning is a Machine Learning technique with which we are able to transfer knowledge from one previously-trained model to another and therefore use this knowledge to solve a similar problem. Many of the already published papers in which Transfer Learning is used for Human Activity Recognition focus mainly on the feasibility of the methodology or on improving the classification results on the target dataset by adapting the used network architecture. As a result, Transfer Learning for Human Activity Recognition still contains many unknown aspects. However, since this technique has a great potential to improve classification results and to reduce the computational time for training neural networks, it is necessary to do more research on this topic and put the spotlight on the mechanism details. We think that the definition of when, where and how to use Transfer Learning should be called into question when it comes to Human Activity Data. Therefore this paper concentrates more on understanding the source and target datasets, as well as understanding the process of weight transfer between models. We want to encourage researchers to take a closer look on these aspects and to dig deeper into the mechanism of Transfer Learning.

2 RELATED WORK

Deep learning is becoming increasingly important for machine learning applications. Where traditional methods reach their limits, either because of the way classification algorithms work or because of the required feature engineering steps of handmade features [1], neural networks manage to close the gap and outperform classical ML approaches regularly [21].

Deep Learning of Human Activity. While in the beginning it was still tried to use handmade features as input data [20], since [7] at the latest, the data are now treated directly as time-discrete signals and thus can be used as input for the neural networks. The work of Hammerla et al. [7] shows, that the most important advantage of a Deep Neural Network against traditional Machine Learning approaches, e.g. [10], is that the traditional approaches work with hand crafted features [1], whereas neural networks learn their features dynamically through optimizing their connecting weights. One promising and often used neural network architecture for Human Activity Recognition is the DeepConvLSTM [16]. This architecture combines Convolutional and Long-Short-Term-Memory (LSTM) layers and is therefore able to model signal patterns as well as classify them related to their appearance on the time axis. A variation of this architecture inspired by the McFly-Project for Deep Learning [19] is used in our experiments.

Transfer Learning. Transfer Learning has become an increasingly important subtopic of Deep Learning in recent years. Therefore it was only a question of time until it was investigated whether this technology can be transferred to time-discrete sensor signals and thus also to Human Activity Recognition. The number of published papers in this discipline has increased rapidly, e.g. [11], [22], [3], [2], [15] or [13], especially in the last two years [9]. [15] showed a setup that we build up on and expanded with tests that artificially mapped the sensors placement and orientation to each other, according to the results of [12] and [24]. Here it is shown, that only after the sensors have been brought into alignment, the classifier is achieving the best results. [6] showed, that cross-dataset transfer learning is possible, if source and target dataset are coming from the same domain. By using an architecture called MultiResNet [5], that transfers the data into frequency domain and uses residual blocks they achieved promising results when transferring from Skoda Mini Checkpoint [23] to OPPORTUNITY [18], PAMAP2 [17] or JSI-FOS [4]. It seems like due to the transformation into frequency domain the trained filters are not class specific anymore and the orientation and location of the sensors axes loses its importance for the success of Transfer Learning.

3 METHODOLOGY

Two publications that influenced our choice of datasets are [6] and [15]. The results presented here show, that Skoda Mini Checkpoint is basically suitable as a source dataset and PAMAP2 as target-data. PAMAP2, on the other hand, already proved in previous publications, for example [16] or [8], to be suitable for use with neural networks.

Datasets. We have chosen to evaluate on these two publicly-available activity recognition datasets as the type of sensors, the sampling rate, and the location at which the sensor was worn matches particularly well:

PAMAP2: The PAMAP2 dataset consists of 19 different classes of activities of daily living and is recorded with a sampling rate of 100Hz and a sensitivity of ±16g. 9 subjects participated in the experiment. To train the PAMAP2 models we used data that has been recorded following the experiment protocol. We also concentrate on activities that are performed by every subject. With these conditions the used data is reduced to 7 subjects performing 8+1 (null class) different activities of daily living. Activities that are taken into account are: null (0), lying (1), sitting (2), standing (3), walking (4), ascending stairs (12), descending stairs (13), vacuum cleaning (16) and ironing (17).

Skoda Mini Checkpoint: The Skoda Mini Checkpoint dataset is recorded by 1 subject, performing 10 different activities, with a sampling rate of approx. 98Hz, and a sensitivity of ±3g. Classes used from this dataset are restricted to the ones, where the activity is performed equally by both hands. Hence classes that were taken into account are: null (32), open hood (49), close hood (50), check gaps on the front door (51), close both left door (54), check trunk gaps (55), open and close trunk (56) and check steering wheel (57).

While corresponding with the authors of PAMAP2 and Skoda Mini Checkpoint we realized that these two datasets were recorded with different sensor orientations. Figure 1 illustrates this problem.

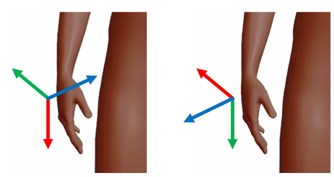

Figure 1: Default Sensor Orientation of PAMAP2 (left) and Skoda Mini Checkpoint (right). The X-axis (red) and Y-axis (green) are switched and the Z-axis (blue) is inverted.

Preprocessing. We used the same preprocessing steps for the baseline model and the transferred model. **(1)** concatenate the data into an array with one channel per sensor-axis, **(2)** delete all synchronization gestures from the dataset, **(3)** scale all axes of the data at ones between -1 and 1, **(4)** apply a jumping window with a length of 50 samples and an overlap-ratio of 50%, **(5)** shuffle the windows with a fixed random seed. For defining the label of the current window we followed the approach used in [16], where the label of the last sample defines the label of the window. Early tests showed, that the classification results between the default

Table 1: Parameters used for the baseline model as well as the different transfer methods. Fixed parameters for all models are: Batch-Size (64), Conv. Kernel-Size (5x3), LSTM-Cells per layer (128), Learning-rate (0.001). After the transfer the trainable parameters were either: frozen (f), trainable (t), or reinitialized and trainable (lecun_uniform)

Parameter	Baseline Model	Transfer Method 1	Transfer Method 2	Transfer Method 3	Transfer Method 4
Training Epochs	1000	30	30	30	30
Weight Init. Conv.-Layers	lecun uniform	pretrained (f)	pretrained (f)	pretrained (f)	pretrained (t)
Weight Init. LSTM-Layers	lecun uniform	pretrained (f)	pretrained (t)	lecun uniform	lecun uniform
Optimizer	Adadelta	RMSProp	RMSProp	RMSProp	RMSProp

98Hz and a resampling to 100Hz for the Skoda Mini Checkpoint dataset are marginal and therefore negligable.

Baseline Model. In order to investigate the effects of transfer learning between different types of sensors, sensors mounted on different body parts, as well misaligned axes, we had to train two different baseline models. One trained on the wrist-worn PAMAP2 accelerometer, and one on Skoda Mini Checkpoint, using only the data from the accelerometer of the right wrist. Instead of using RMSProp as the optimizer, as proposed by [16], we switched to Adadelta, which performs slower, but more stable. RMSProp showed an unstable behavior regarding the classification performance with massive negative peaks in longer training periods, but seems to be good choice for fine-tuning operations.

Transfer Learning. We have applied four different methods of transferring the model. All methods follow the weight transfer method, e.g. used in [14] and [15], to transfer the pretrained model. We transferred the pretrained weights from the baseline model and replaced the classification layer by an untrained one, which fits the number of classes of the target dataset. We also switched the optimizer of the transferred model to RMSProp, since we only fine-tune our model for 30 epochs. Following we distinguish between different levels of post-transfer trainable layers: **(1)** All layers are frozen after transfer, except the classification-layer, **(2)** Only the ConvBlocks are frozen after transfer, LSTM-Layers stay trainable, **(3)** the ConvBlocks are frozen after transfer, LSTM-Layers stay trainable, but are reinitialized with lecun-uniform initialization and **(4)** the Conv.- and LSTM-Layers are trainable, but LSTM-Layers are reinitialized with lecun-uniform initializiation. Figure 2 depicts the used architecture and transfer method. To evaluate the results, we determined the respective Training F1-Score. In order to simulate all possible orientations of a sensor relative to the baseline model, we decided to permute and invert the position of the sensor axes. This results in 48 possible combinations. Thus, all models were transferred 48 times in each test that was not transferred back to the source dataset. A transfer back to the source data was done as a sanity check. These sanity checks, as well as transfer within the dataset, but to another sensor worn at the same position, are done with a leave-one-fold-out cross-validation with 4 folds.

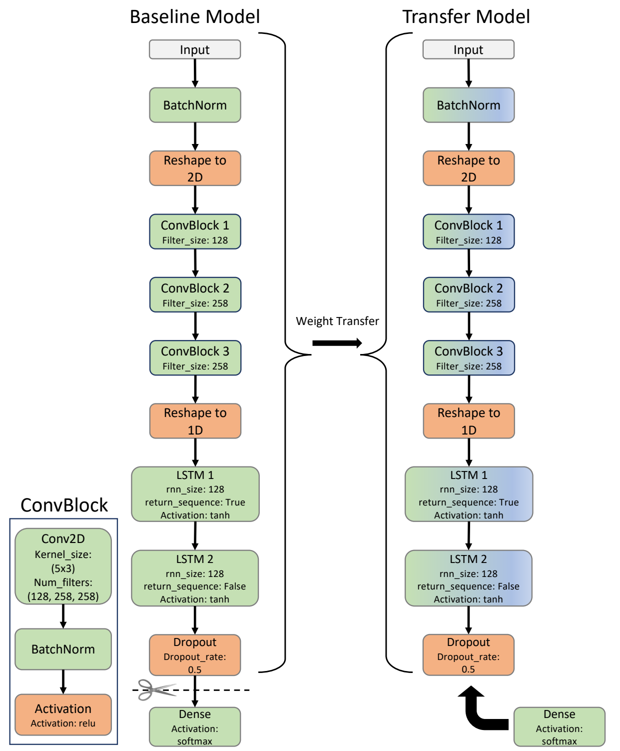

Figure 2: DeepConvLSTM [16] architecture. Red blocks do not have trainable parameters, whereas green blocks are trained, transferred, and frozen (represented as blue blocks) in the target model, depending on the used transfer method. The original last dense layer is replaced with a new output layer during transfer, with a size according to the number of classes of the target dataset. One ConvBlock consists of three layers, a convolutional layer, a batch normalization layer and an activation layer with ReLu activation function.

4 RESULTS AND EVALUATION

We tested Transfer Learning between different sensor locations, different sensor types and different sensor orientations for intra-, as well as inter dataset transfer. Transfer back to the source dataset was performed as a sanity check, see experiment **(1)**, and **(5)**. Similar results after transfer with method 1 ensure that no errors occurred during transfer or data preprocessing. If irregularities occur in the preprocessing or transfer process, the after-transfer performance of method 1 would decrease significantly. The result of **(1)** with method 2, 3 and 4 shows that transfer learning harms the classifier in general and needs to be fine-tuned to perform reliable. The result of experiment **(2)** with method 1 must be

Table 2: Train-F1-Score in %, given as minimum, maximum and mean. PAMAP2 (P), Skoda (S), Accelerometer (A), Gyroscope (G), Magnetometer (M), Wrist (W), Chest (C).

	Source -> Target	Method 1			Method 2		
		Min	Max	Mean	Min	Max	Mean
1	S (W, Right) → S (W, Right)	97.5	98.5	97.8	93.1	93.5	93.2
2	S (W, Right) → S (C, Right)	43.3	64.4	54.8	84.1	93.6	93.0
3	S (W, Right) → S (W, Left)	50.8	61.0	54.6	89.1	93.4	91.1
4	S (W, Right) → P (A, W)	12.1	26.6	20.7	54.9	69.4	63.6
5	P (W, A) → P (W, A)	82.3	82.5	82.4	92.8	93.0	93.0
6	P (W, A) → P (C, A)	23.3	42.6	35.2	59.1	73.4	65.8
7	P (W, A) → S (W, Right)	38.1	46.3	42.8	73.0	86.4	79.4
8	P (W, A) → P (W, G)	03.2	03.4	03.3	21.6	23.6	22.4
9	P (W, A) → P (W, M)	36.6	37.4	37.1	66.6	68.1	67.5
	Source -> Target	Method 3			Method 4		
1	S (W, Right) → S (W, Right)	96.0	96.6	96.2	60.9	64.2	63.0
2	S (W, Right) → S (C, Right)	85.5	94.4	89.0	42.0	69.8	63.4
3	S (W, Right) → S (W, Left)	90.1	94.4	92.0	63.4	77.9	72.6
4	S (W, Right) → P (W, A)	54.5	70.7	64.6	11.0	63.7	52.1
5	P (W, A) → P (W, A)	94.7	95.4	94.9	54.9	55.9	55.1
6	P (W, A) → P (C, A)	59.3	73.5	66.7	57.1	64.4	60.7
7	P (W, A) → S (W, Right)	77.3	87.7	82.3	34.2	72.8	61.5
8	P (W, A) → P (W, G)	21.5	23.2	22.5	05.4	07.1	06.0
9	P (W, A) → P (W, M)	67.6	68.0	67.5	09.8	19.8	15.5

subjected to closer examination. The best result is achieved, after the X-axis is first inverted and then swapped with the Z-axis, which results in an Z, Y, -X orientation. Whether this orientation corresponds to the actual position of the axes relative to the sensor worn on the wrist cannot be said with certainty at this point, but it is evident, that this result deviates from the average by about 10%. Remarkable is experiment **(3)** with method 1, in which the model trained on the data of the right wrist of Skoda Mini Checkpoint, is applied to the data of the left wrist. The best result was obtained after leaving the axes in default position, but inverting the X-axis. Thus the left hand data was artificially mapped to the orientation of the trained model. Experiment **(6)** resulted in the highest F1-Score, when the axes were left in the original position, but differs by up to 19.3% during the permutation test. This means that the alignment of the chest sensors matches the alignment of the one worn at the wrist. Inter-Dataset Transfer Learning, experiment **(4)** and **7** with method 1 always resulted in very low F1-Scores. However, the transfer from PAMAP2 to Skoda **(7)** had better results than from Skoda to PAMAP2 **(4)**, which might be a direct result of the bigger size and variability of the PAMAP2 dataset. A transfer between types of sensors is in general not recommendable. **(8)** and **(9)** show that a model trained on accelerometer data is not capable to classify the same activities recorded by a gyroscope or magnetometer.

5 DISCUSSION AND CONCLUSION

We started this experiment in assuming that by properly adjusting the position and orientation axes of the inertial data along the sensor axes we could significantly increase the classification results. We could demonstrate this with the results of method 1, but these results did not reach the significance as initially expected and are therefore not an acceptable final state for a classifier. Matching the alignment of the sensor axes results in a more adapted classifier, but it is not possible to achieve the classificative properties of the baseline model. Due to the mostly frozen architecture, the adaptation process of fine-tuning the classification layer reaches its limits very quickly.

The results of method 2 and 3 are very similar. However, these experiments show that it is basically advisable to reinitialize the LSTM-Layers to default, since the F1-Score is on average 3.4% higher with method 3 than with method 2. The experimental results of method 4 demonstrates that convolutional layers should not be fine-tuned after model-transfer. The comparatively worse results of this method are caused, since the outputs of the convolutional layer are fed as input to the LSTM-Layers due to their position in the architecture. By re-initializing these layers, but also keeping the convolutional layers trainable, the pre-trained data dependent link between these layers is lost.

The datasets used in this paper share many modalities, such as the position of the sensors on the body, the sampling rate, and the sensor technology used, but differ fundamentally in the underlying classes. Thus, we assume that the features of the filters trained in the convolutional layers are very dataset dependent and thus class specific. Using pretrained weights can provide a speed advantage and thus lead to faster network convergence, due to less trainable parameters, but we consider the impact on the final classification performance, even with artificially adjustments of the orientation and position of the sensors, to be marginal. These results largely correspond to those of [15].

It is surprising that although the modalities of both datasets are largely identical, a transfer between them is always accompanied by strong performance losses. This observation leads us to the following research challenges, which we leave open at this point for the research community to address:

(1) Under which exact conditions is Transfer Learning recommendable for wearable-based activity data?
(2) How transferable are the pretrained convolution filters between inertial activity datasets?
(3) Which preprocessing steps are suitable to make models transferable, regardless of their architecture?

We argue that the three questions for designing transfer learning – What? When? How? – are hard to adapt from other disciplines and should be reconsidered for Transfer Learning with inertial sensors-based signals.

ACKNOWLEDGMENTS

We thank our anonymous reviewers and paper shepherd for their many insightful suggestions and feedback.

REFERENCES

[1] Ling Bao and Stephen Intille. 2004. Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing*. Springer, 1–17.

[2] Yiqiang Chen, Yang Gu, Xinlong Jiang, and Jindong Wang. 2016. Ocean: A new opportunistic computing model for wearable activity recognition. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 33–36.

[3] Tom Diethe, Niall Twomey, and Peter A Flach. 2016. Active transfer learning for activity recognition.. In *ESANN*.

[4] Hristijan Gjoreski, Boštjan Kaluža, Matjaž Gams, Radoje Milić, and Mitja Luštrek. 2015. Context-based ensemble method for human energy expenditure estimation. *Applied Soft Computing* 37 (2015), 960–970.

[5] Martin Gjoreski, Vito Janko, Gašper Slapničar, Miha Mlakar, Nina Reščič, Jani Bizjak, Vid Drobnič, Matej Marinko, Nejc Mlakar, Mitja Luštrek, et al. 2020. Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. *Information Fusion* (2020).

[6] Martin Gjoreski, Stefan Kalabakov, Mitja Luštrek, Matjaž Gams, and Hristijan Gjoreski. 2019. Cross-Dataset Deep Transfer Learning for Activity Recognition. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 714–718.

[7] Nils Yannick Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Plötz. 2015. PD disease state assessment in naturalistic environments using deep learning. In *Twenty-Ninth AAAI conference on artificial intelligence*.

[8] Nils Yannik Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* (2016).

[9] Netzahualcoyotl Hernandez, Jens Lundström, Jesus Favela, Ian McChesney, and Bert Arnrich. 2020. Literature Review on Transfer Learning for Human Activity Recognition Using Mobile and Wearable Devices with Environmental Technology. *SN Computer Science* 1, 2 (Mar 2020), 66.

[10] Alexander Hoelzemann and Kristof Van Laerhoven. 2018. Using wrist-worn activity recognition for basketball game analysis. In *Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction*. 1–6.

[11] Derek Hao Hu, Vincent Wenchen Zheng, and Qiang Yang. 2011. Cross-domain activity recognition via transfer learning. *Pervasive and Mobile Computing* 7, 3 (2011), 344–358.

[12] K. Kunze and P. Lukowicz. 2014. Sensor Placement Variations in Wearable Activity Recognition. *IEEE Pervasive Computing* 13, 4 (2014), 32–41.

[13] Frédéric Li, Kimiaki Shirahama, Muhammad Adeel Nisar, Xinyu Huang, and Marcin Grzegorzek. 2020. Deep Transfer Learning for Time Series Data Based on Sensor Modality Classification. *Sensors* 20, 15 (2020), 4271.

[14] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. 97–105.

[15] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. 92–99.

[16] Francisco Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* 16 (01 2016), 115.

[17] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring.

[18] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 233–240.

[19] D. van Kuppevelt, C. Meijer, F. Huber, A. van der Ploeg, S. Georgievska, and V.T. van Hees. 2020. Mcfly: Automated deep learning on time series. *SoftwareX* 12 (2020), 100548.

[20] P. Vepakomma, D. De, S. K. Das, and S. Bhansali. 2015. A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. 1–6.

[21] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* 119 (2019), 3–11.

[22] Josh Jia-Ching Ying, Bo-Hau Lin, Vincent S. Tseng, and Sun-Yuan Hsieh. 2015. Transfer Learning on High Variety Domains for Activity Recognition. In *Proceedings of the ASE BigData & SocialInformatics 2015 (ASE BD&SI '15)*. Association for Computing Machinery, New York, NY, USA, Article 37, 6 pages.

[23] Piero Zappi, Daniel Roggen, Elisabetta Farella, Gerhard Tröster, and Luca Benini. 2012. Network-Level Power-Performance Trade-Off in Wearable Activity Recognition: A Dynamic Sensor Selection Approach. *ACM Trans. Embed. Comput. Syst.* 11, 3, Article 68 (Sept. 2012), 30 pages.

[24] Tobias Zimmermann, Bertram Taetz, and Gabriele Bleser. 2018. IMU-to-segment assignment and orientation alignment for the lower body using deep learning. *Sensors* 18, 1 (2018), 302.