Analysis of Deep Transfer Learning Using DeepConvLSTM for Human Activity Recognition from Wearable Sensors

Stefan Kalabakov
Department of Intelligent Systems, Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
International Postgraduate School, Jamova 39, Ljubljana, Slovenia
E-mail: stefan.kalabakov@ijs.si

Martin Gjoreski
Faculty of Informatics, Università della Svizzera Italiana (USI), Lugano, Switzerland
E-mail: martin.gjoreski@usi.ch, martingjoreski.github.io

Hristijan Gjoreski
Faculty of Electrical Engineering, University Ss. Cyril and Methodius, Skopje, Macedonia
E-mail: hristijang@feit.ukim.edu.mk, dis.ijs.si/hristijan

Matjaž Gams
Department of Intelligent Systems, Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
E-mail: matjaz.gams@ijs.si, dis.ijs.si/mezi

Keywords: transfer learning, deep learning, human activity recognition, accelerometer data

Received: July 15, 2021

Human Activity Recognition (HAR) from wearable sensors has gained significant attention in the last few decades, largely because of the potential healthcare benefits. For many years, HAR was done using classical machine learning approaches that require the extraction of features. With the resurgence of deep learning, a major shift happened and at the moment, HAR researchers are mainly investigating different kinds of deep neural networks. However, deep learning comes with the challenge of having access to large amounts of labeled examples, which in the field of HAR is considered an expensive task, both in terms of time and effort. Another challenge is the fact that the training and testing data in HAR can be different due to the personal preferences of different people when performing the same activity. In order to try and mitigate these problems, in this paper we explore transfer learning, a paradigm for transferring knowledge from a source domain, to another related target domain. More specifically, we explore the effects of transferring knowledge between two open-source datasets, the Opportunity and JSI-FOS datasets, using weight-transfer for the DeepConvLSTM architecture. We also explore the performance of this transfer at different amounts of labeled data from the target domain. The experiments showed that it is beneficial to transfer the weights of fewer layers, and that deep transfer learning can perform better than a domain-specific deep end-to-end model in specific circumstances. Finally, we show that deep transfer learning is a viable alternative to classical machine learning approaches as it produces comparable results and does not require feature extraction.

Povzetek: V prispevku je raziskan vpliv števila učnih primerov pri prenesenem učenju, ki kaže izboljšano delovanje pri malem številu primerov.

1 Introduction

With numerous healthcare, smart-home and security applications on the horizon, human activity recognition (HAR) is a field which has gained significant traction in the past two decades. Most of the research done on this topic has been aimed at understanding human activity using data from wearable sensors, primarily inertial measurement units (IMUs). This is in large part due to the rapid development of wearable devices, allowing researchers to perform complex tasks on them, as well as their ubiquitous and unobtrusive nature.

Until recently, researchers in the field of HAR were mainly focused on using traditional pattern-matching methods as well as machine learning to detect activities [1]. In order to work properly, these methods require the extraction of features which in turn requires considerable domain knowledge in order to extract a diverse and information-rich feature set. However, in recent years, as the benefit of using deep neural networks is apparent in domains such as computer vision and NLP [2], research has shifted towards deep learning [3]. The focus has mostly been centered around Convolutional Neural Networks (CNNs). These networks are capable of automatically capturing hierarchical feature representations of the data [4], i.e. they produce features which range from general to application specific as one goes deeper in the network. Another emerging trend in HAR is the use of LSTM cells which are able to better capture the temporal dependencies between sensor readings [5][6]. Finally, an interesting approach which yields arguably the best result is the creation of hybrid models such as the one proposed by Ordóñez et al., where they combine convolutional layers with LSTM cells in order to exploit both the spatial and the temporal analysis which those types of layers provide [7].

Nevertheless, as is the case in many other fields where it is applied, aside from the benefits, deep learning also brings certain challenges. These challenges are usually even more emphasized when working on HAR as opposed to fields such as image classification. For example, large amounts of (diverse) data are required to train a deep end-to-end classifier that can accurately predict human activities. These large amounts of data are usually difficult to collect as it takes a lot of time and sometimes money to do so. In addition, deep neural networks require a lot of time to train in order to reach their full potential, which hampers the ability to quickly create prototypes that can be further built upon. Finally, the source and target data in HAR can be very different, as different users perform the same activities differently depending on their personal preferences. This makes building end-to-end deep learning models a difficult task.

Given these challenges, it is clear that providing a solution to them could accelerate the development of HAR models for specific activity domains and make these models more adaptable to users and data that they have not previously seen. To this end, we explore transfer learning, a learning paradigm that deals with transferring knowledge acquired in one (source) domain to another related (target) domain, as a method that could help mitigate these issues.

There have been several works in the past which showed transfer learning to be beneficial in the HAR domain, but most of them used classical ML methods [8][9][10]. An extensive analysis of conventional transfer learning methods can be found in [11]. However, in contrast to this, there aren't many works addressing deep transfer learning. Morales et al. presented the pioneering deep transfer learning approach for HAR in [12]. In this paper the authors worked with the PAMAP2 and Skoda Mini Checkpoint datasets and investigated the transfer of weights of the DeepConvLSTM model between the two domains. Unfortunately, they get negative transfer results and conclude that the two domains are just too different from each other. In addition, Hoelzemann et al. have a similar transfer learning setup to the one presented in [12] and in their experiments they conclude that transfer between sensor locations in the same domain is feasible, but transfer between datasets is accompanied with significant performance losses. The authors in [13] propose a method which is able to achieve good results when transferring between HAR datasets which have the same set of tasks. Finally, in our previous work [14] we showed promising results for a transfer learning system using the MultiResNet architecture [15] at different adaptation set sizes.

In order to provide more detailed insights into some of the open questions, this paper is going to explore the performance of a transfer learning system, using two intuitively similar datasets which consist of activities of daily living (ADL). The deep learning architecture we are going to be using in this work is the DeepConvLSTM architecture. In more detail, we are going to explore: (i) the performance of transfer learning when transferring the weights of different numbers of convolutional layers; (ii) the performance of transfer learning when using different sizes of labeled adaptation sets; (iii) how transfer learning performs in comparison to domain-specific classical machine learning approaches and domain-specific end-to-end learning.

The rest of this paper is organized as follows. In Section 2 we introduce the datasets which are used in our experiments. Section 3 describes the preprocessing and feature extraction steps which were performed on the raw signal data before it was presented to the algorithms. The following section (Section 4) briefly describes the deep learning architecture that we chose to use. Following that, in Section 5 we give an introduction into our experimental setup, and in Section 6 we present and discuss the results. Finally, our work is concluded in Section 7.

2 Datasets

For the experiments we chose two datasets that are intuitively similar to each other. Both the Opportunity [16] and the JSI-FOS dataset [17][18], consist of activities which users commonly perform in their daily routines. Both of these datasets contain at least one 3D accelerometer worn on the right wrist, which allows us to discard the sensor modality and location as variables in our analysis. Another important characteristic of these datasets is the fact that they consist of data from several different users, which allows us to test the generalization capabilities of our models by using a Leave-One-Subject-Out (LOSO) evaluation.

Although they have a lot of similarities, the JSI-FOS and Opportunity datasets differ in several areas: (i) the number of activities, with JSI-FOS having 18 and Opportunity having 21 distinct activities (reduced to 10 and 14 after the preprocessing steps described in Section 3); (ii) sampling rate, which is equal to 50Hz and 30Hz, respectively; (iii) the overall duration of the data, which is around 3 times larger in the JSI-FOS dataset and amounts to around 20 hours (after the preprocessing steps described in Section 3). Furthermore, the two datasets also differ slightly in the types of activities, with Opportunity focusing on gestures and not just locomotion activities. Figure 1 and Figure 2, show the distribution of the activities we selected from both datasets.

Finally, a summary of the information about both datasets is given in Table 1.

Dataset	Type	#Subjects	Sampling rate	#Activities	#Selected activities	# of examples
JSI-FOS	ADL	10	50Hz	18	10	36060 [20h]
Opportunity	ADL	4	30Hz	21	14	10822 [6h]

Table 1: Overview of the two datasets used in our experiments

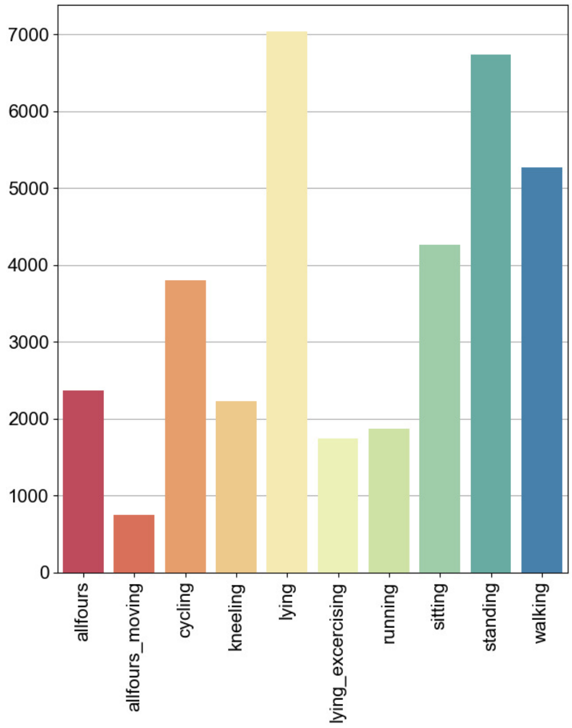


Figure 1: The number of examples per selected activity in the JSI-FOS dataset.

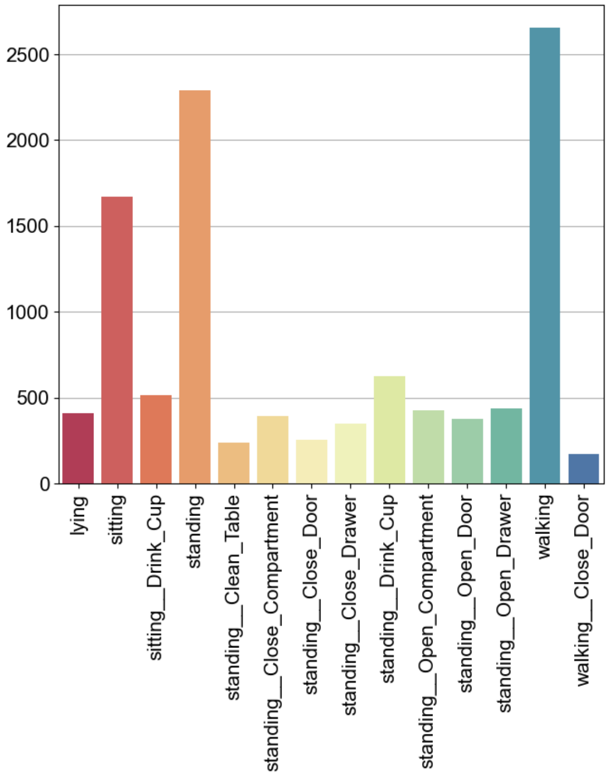


Figure 2: The number of examples per selected activity in the Opportunity dataset.

3 Preprocessing and feature extraction

In order to unify the way data is represented in both datasets, we constructed a fairly simple preprocessing pipeline. The first step in this pipeline is the selection of data which comes from a 3D accelerometer, worn on the right wrist. In addition to the three channels of data that these accelerometers produce, we also calculate the magnitude as a virtual fourth channel. Although it is available, we disregarded data from other sensors in order to simplify the model and make the analysis easier.

The second step in the preprocessing pipeline is the resampling of the data from the accelerometers to a sampling rate of 25Hz. We chose this relatively low sampling rate to make our transfer learning setup more suitable for potential use with wearable devices since a lower sampling rate results in better energy efficiency. Furthermore, the authors in [15] show that there are no significant performance differences when using sampling rates between 25Hz and 100Hz. The data from both datasets is downsampled to a common sampling rate to ensure that the filters we transfer from the source model to the target model work as intended. Otherwise, if the source and target data have different sampling frequencies, each of the transferred convolutional filters would work on a piece of data that has a different temporal length in comparison to what it was trained on. Next, we also convert the units of the measurements to a common unit, "g" (9.81 m/s^2). After we take care of the raw data format, we turn our attention to the activities in each dataset. In the JSI-FOS dataset, we only perform some simple aggregations of the original activities. For example, the activities *lying_back*, *lying_left_side*, *lying_right_side*, and *lying_stomach* are all aggregated to one activity, *lying*. This is why, from the original 18 activities, we end up selecting only 10 distinct ones. Table 2 shows all the aggregations used for this dataset.

On the other hand, when working with the Opportunity dataset, whenever a locomotion and gesture label are simultaneously available we create a new activity label. This new activity is simply the concatenation of the locomotion

Original	Aggregation
lying_back	lying_back
lying_left_side	
lying_right_side	
lying_right_stomach	
allfours	allfours
allfours_still	
standing	standing
standing_leaning_still	
transition_up	Null
transition_down	

Table 2: Aggregations of activities used for the JSI-FOS dataset

Original	Aggregation
*_open_door1	*_open_door
*_open_door2	
*_close_door1	*_close_door
*_close_door2	
*_open_fridge	*_open_compartment
*_open_dishwasher	
*_close_fridge	*_close_compartment
*_close_dishwasher	
*_open_drawer[1/2/3]	*_open_drawer
*_close_drawer[1/2/3]	*_close_drawer

Table 3: Aggregations of activities used for the Opportunity dataset

and gesture label, for example, walking (locomotion) while drinking from a cup (gesture). After this, we perform the same aggregation process as with JSI-FOS. Table 3 shows the aggregation rules for the Opportunity dataset. An asterisk is used in this table as a placeholder for a potential locomotion label.

The penultimate step of the pipeline is segmenting the raw data into windows of fixed size. Each window in this work contains 100 sensor readings, which represents 4 seconds of data at a sampling frequency of 25Hz. There is a 50% overlap between two windows. Windowing is performed for each channel separately, which means that after this step, both datasets are represented as sets of quadruples (4 channels).

Finally, as the last step, from both datasets we remove the windows with a Null activity label and disregard all activities with fewer than 100 windows (3.3 seconds). At this point, the preprocessing steps for the DeepConvLSTM model end, and the quadruples, stacked vertically, can be fed into the model in order to be processed.

3.1 Feature extraction

In order to be able to use classical machine learning algorithms we need to further change the form of the aforementioned windows by extracting features. In order to provide the algorithms with an information-rich representation, from each quadruple of windows (4 channels), we extract around 2400 features based on the related work on HAR. The majority of the features come from the TSFRESH package, which allows for the extraction of general-purpose time-series features. On top of the features extracted with TSFRESH, we also extracted a set of frequency-domain features which was previously shown to work well in other HAR applications [19][20]. This set of features is based on the Power Spectral Density (PSD) of the signal and include its binned distribution, entropy, energy, magnitude, and first four statistical moments of the PSD, among others.

4 Model architecture

In this work we chose to use the DeepConvLSTM framework, proposed by Ordóñez et al. in [7]. This architecture consists of stacked convolutional layers which are followed by LSTM cells. This allows the network to extract hierarchical feature representations and model the temporal dependencies between them. The network architecture was chosen primarily for its simplicity, which allows for an easier evaluation of how some changes affect the transfer learning performance, as well as its frequent use in other deep transfer learning studies in the field of HAR [12] [21].

The input, in our implementation of the network, is expected to consist of 4 stacked windows of signal data (one per sensor channel). This input is then processed through 4 convolutional layers, each with 64 feature maps. The convolutional layers use the ReLU activation function to compute their output.

Following them are two pairs of dropout and LSTM layers. Each dropout layer has a rate of 0.5 and each LSTM layer has 128 cells. Finally, a softmax layer is attached to the last LSTM layers in order to produce the final predictions. A diagram of this architecture can be seen on Figure 3.

5 Experimental setup

In this paper we adopt the following transfer learning approach: (i) train a (source) model on the Opportunity dataset; (ii) transfer the weights of all layers (except the softmax layer) of that model to a new (target) model in which the softmax layer fits the number of classes of the JSI-FOS dataset; (iii) freeze a certain number of convolutional layers and allow all the rest to be fine-tuned; (iv) fine-tune the rest of the layers using some number of instances (adaptation set) from the JSI-FOS dataset. Both the source and target models in this approach are trained using a batch size of 64 and a learning rate of 0.001. However, there is a difference in the number of training epochs between the source and target models and those numbers of epochs are

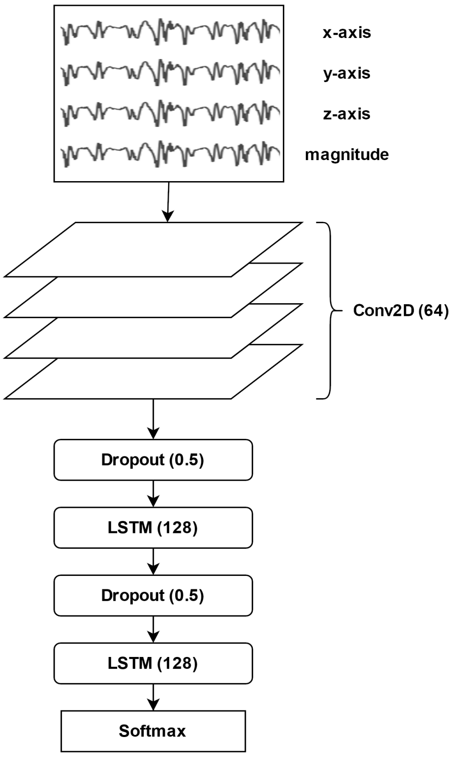


Figure 3: A diagram of our implementation of the DeepConvLSTM architecture.

100 and 70, respectively. These numbers were determined experimentally.

Another important detail to explain, before we move on to the experiments, is the adaptation set. This is the set of instances from the JSI-FOS dataset we use to either train or fine-tune a model. Since we wanted to explore the efficacy of transfer learning at different amounts of labeled data from the target domain, we repeat each of our experiments several times, using different sizes of the adaptation set. The adaptation set sizes range from 100 instances to 12000 instances, that is, between 3.33 minutes and 6.66 hours of labeled data.

Furthermore, it is important to note that, since we use Leave-One-Subject-Out (LOSO) evaluation, the adaptation set is produced in a stratified manner, using all subjects except for the one which is selected for testing and the two subjects (randomly chosen from the remaining set of subjects) which are selected for validation. This means that an adaptation set of 100 instances, will be produced at least 10 times, once in each iteration of the LOSO evaluation. In order to make the results more relevant and minimize randomness, we repeat the LOSO evaluation several times at each adaptation set size and report the average of these results. For example, given an adaptation set of 100 instances, we repeat the LOSO evaluation 4 times, which means that a random adaptation set will be produced a total of 40 times (JSI-FOS has 10 subjects).

Finally, we should note that when we train models on the target dataset (JSI-FOS) we use early stopping based on the loss value on the validation set (two randomly selected train-users in each iteration).

Experiment 1
The first experiment is aimed at exploring the optimal number of convolutional layers to freeze in step (iii) of the transfer learning approach. To this end, we repeat the transfer learning approach 4 times, and in each of them we freeze a different number of convolutional layers, ranging from 1 to 4.

Experiment 2
The second experiment is aimed at comparing the performance of deep transfer learning, deep end-to-end models and classical ML. As an example of a classical ML algorithm we chose Random Forest, as it often shows state-of-the-art results and does not require extensive hyper-parameter tuning [15]. This model is trained only on the instances from the adaptation set, using the features extracted in Section 3.1. The end-to-end model is also trained using only the examples in the adaptation set and uses the same architecture as the transfer learning model, but its weights were initialized randomly and no transfer of knowledge has taken place. Finally, the transfer learning model is trained using steps (i) through (iv) and the number of convolutional layers transferred between models is based on the results from the first experiment.

6 Results and discussion

The results from ***experiment 1*** can be seen on Figure 4. The x-axis of that graph, shows the number of instances in the adaptation set, while the y-axis of the graph, shows the macro F1-score. Based on the results from this experiment, it seems that there isn't a huge difference in performance when freezing different numbers of convolutional layers from the DeepConvLSTM architecture. However, the setup in which we only freeze the first convolutional layer and allow all others to be fine-tuned, performs marginally, but consistently, better than the rest. This finding seems to be in line with what was concluded by [12] and supports the claim that convolutional layers deeper in the model, extract features which are just too dataset (domain) specific.

The results from ***experiment 2*** can be seen on Figure 5. Here we compare the performance of a RF classifier, an end-to-end (E2E) DeepConvLSTM model and a DeepConvLSTM model trained using transfer learning. As is expected, the E2E model shows very poor performance when the adaptation set size is very low, and gradually, improves as the adaptation set grows. It is interesting to note that although it comes close, the E2E model never really matches the performance of the Random Forest (RF) classifier. On the other hand, the RF classifier produces strong results on all adaptation set sizes, except the smallest one. This is probably due to the fact that relevant features were extracted by hand. Lastly, it is interesting to see that the DeepConvLSTM model trained using transfer learning produces

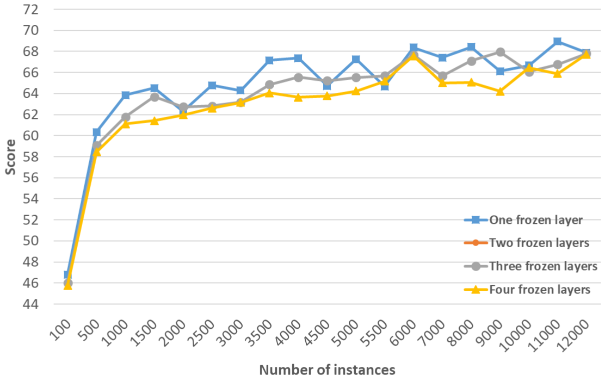


Figure 4: Performance of the DeepConvLSTM architecture when freezing different numbers of transferred layers.

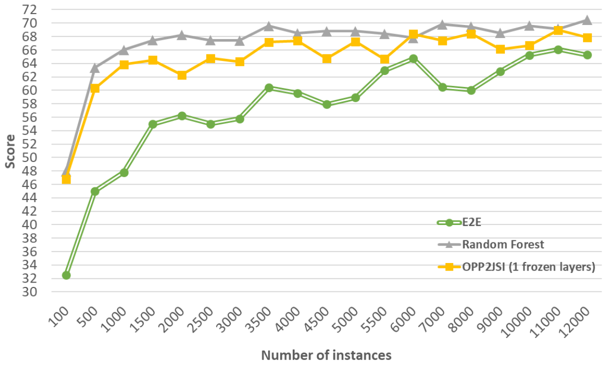


Figure 5: A comparison between the performances of a classical ML algorithm, an end-to-end deep learning model and a transfer learning model, at different adaptation sizes.

results which are quite similar to the ones produced by the RF classifier and that it manages to beat the results of the E2E model across all adaptation set sizes. This seems to support the idea that relevant features were already extracted in the first convolutional layer (trained on the source domain) and that even small adaptation sets contain enough data for the model to make sense of those features.

Finally, Figure 6 shows a per activity comparison between the performances of the deep E2E model and the deep transfer learning model. Each row represents a different adaptation set size, while the columns represent the different activities in the target dataset (JSI-FOS). The value in each of the cells, is the difference in activity F1-score between the deep transfer learning model and the domain-specific deep E2E model. This produces positive values, whenever the deep transfer learning model is better and negative values whenever the opposite is true. As we can see, there are very few situations in which the deep E2E model manages to perform better than the deep transfer learning model, which is to be expected based on the results shown on Figure 5. This figure also, quite clearly shows the gradual decline in performance gains (as we increase the size of the adaptation set) for the *allfours*, *allfours_moving*, *cycling*, *standing*, *walking* activities.

7 Conclusion

In this paper we use the DeepConvLSTM architecture to explore the benefits of transferring knowledge (represented by model weights) from the Opportunity dataset, to the JSI-FOS dataset. Unlike in several previous works, we explore transfer learning between datasets which come from intuitively similar domains and both contain activities from the daily lives of users. In this work we aim to create a head-to-head comparison of classical ML, end-to-end deep learning and deep transfer learning. From the results, we can conclude that it is better to transfer the weights of fewer convolutional layers, as there was already extracted a set of diverse features which only get more domain specific as we transfer more layers. Furthermore, we also show that deep transfer learning is able to produce better results in comparison to a deep end-to-end model trained on the same amount of labeled data. Finally, we show that with the use of deep transfer learning one can produce results comparable to those of a RF classifier without the need for feature extraction done by hand.

Acknowledgement

Part of the study was supported by the project WideHealth - Horizon 2020, under grant agreement No 95227. Additionally, Stefan Kalabakov would like to thank the Slovene Human Resources Development and Scholarship Fund (Ad futura) for the support.

8 References

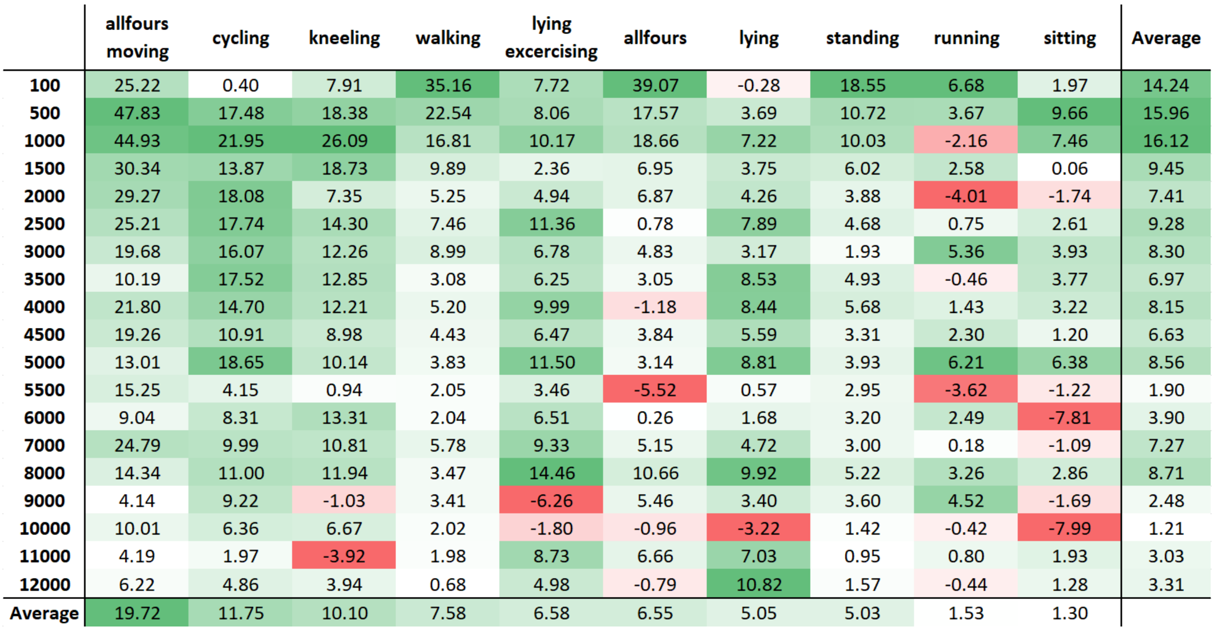
References

[1] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.
https://doi.org/10.1109/SURV.2012.110112. 00192.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
https://doi.org/10.1038/nature14539.

[3] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.
https://doi.org/10.1016/j.patrec.2018.02.010.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
https://doi.org/10.1109/5.726791.



	allfours moving	cycling	kneeling	walking	lying excercising	allfours	lying	standing	running	sitting	Average
100	25.22	0.40	7.91	35.16	7.72	39.07	-0.28	18.55	6.68	1.97	14.24
500	47.83	17.48	18.38	22.54	8.06	17.57	3.69	10.72	3.67	9.66	15.96
1000	44.93	21.95	26.09	16.81	10.17	18.66	7.22	10.03	-2.16	7.46	16.12
1500	30.34	13.87	18.73	9.89	2.36	6.95	3.75	6.02	2.58	0.06	9.45
2000	29.27	18.08	7.35	5.25	4.94	6.87	4.26	3.88	-4.01	-1.74	7.41
2500	25.21	17.74	14.30	7.46	11.36	0.78	7.89	4.68	0.75	2.61	9.28
3000	19.68	16.07	12.26	8.99	6.78	4.83	3.17	1.93	5.36	3.93	8.30
3500	10.19	17.52	12.85	3.08	6.25	3.05	8.53	4.93	-0.46	3.77	6.97
4000	21.80	14.70	12.21	5.20	9.99	-1.18	8.44	5.68	1.43	3.22	8.15
4500	19.26	10.91	8.98	4.43	6.47	3.84	5.59	3.31	2.30	1.20	6.63
5000	13.01	18.65	10.14	3.83	11.50	3.14	8.81	3.93	6.21	6.38	8.56
5500	15.25	4.15	0.94	2.05	3.46	-5.52	0.57	2.95	-3.62	-1.22	1.90
6000	9.04	8.31	13.31	2.04	6.51	0.26	1.68	3.20	2.49	-7.81	3.90
7000	24.79	9.99	10.81	5.78	9.33	5.15	4.72	3.00	0.18	-1.09	7.27
8000	14.34	11.00	11.94	3.47	14.46	10.66	9.92	5.22	3.26	2.86	8.71
9000	4.14	9.22	-1.03	3.41	-6.26	5.46	3.40	3.60	4.52	-1.69	2.48
10000	10.01	6.36	6.67	2.02	-1.80	-0.96	-3.22	1.42	-0.42	-7.99	1.21
11000	4.19	1.97	-3.92	1.98	8.73	6.66	7.03	0.95	0.80	1.93	3.03
12000	6.22	4.86	3.94	0.68	4.98	-0.79	10.82	1.57	-0.44	1.28	3.31
Average	19.72	11.75	10.10	7.58	6.58	6.55	5.05	5.03	1.53	1.30	

Figure 6: A per activity comparison of the F1-scores achieved by the end-to-end model and the model trained using transfer learning. Each cell shows the difference between the F1-score achieved but the transfer learning model and the score achieved by the end-to-end model. A positive value means that the transfer learning model performed better, while a negative one suggest better performance by the end-to-end model.

[5] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” *arXiv preprint arXiv:1604.08880*, 2016.

[6] Y. Guan and T. Plötz, “Ensembles of deep lstm learners for activity recognition using wearables,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.
https://doi.org/10.1145/3090076.

[7] F. J. Ordóñez and D. Roggen, “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, p. 115, 2016.
https://doi.org/10.3390/s16010115.

[8] A. Calatroni, D. Roggen, and G. Tröster, “Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion,” in *Eighth international conference on networked sensing systems (INSS’11)*. Eighth International Conference on Networked Sensing Systems (INSS’11), 2011.

[9] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, and G. Tröster, “Real-time transfer and evaluation of activity recognition capabilities in an opportunistic system,” *machine learning*, vol. 1, no. 7, p. 8, 2011.

[10] S. Inoue and X. Pan, “Supervised and unsupervised transfer learning for activity recognition from simple in-home sensors,” in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2016, pp. 20–27.
https://doi.org/10.1145/2994374.2994400.

[11] D. Cook, K. D. Feuz, and N. C. Krishnan, “Transfer learning for activity recognition: A survey,” *Knowledge and information systems*, vol. 36, no. 3, pp. 537–556, 2013.
https://doi.org/10.1007/s10115-013-0665-3.

[12] F. J. O. Morales and D. Roggen, “Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations,” in *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, 2016, pp. 92–99.
https://doi.org/10.1145/2971763.2971764.

[13] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, “Deep transfer learning for cross-domain activity recognition,” in *proceedings of the 3rd International Conference on Crowd Science and Engineering*, 2018, pp. 1–8.
https://doi.org/10.1145/3265689.3265705.

[14] M. Gjoreski, S. Kalabakov, M. Luštrek, M. Gams, and H. Gjoreski, “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, 2019, pp. 714–718.
https://doi.org/10.1145/3341162.3344865.

[15] M. Gjoreski, V. Janko, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Luštrek *et al.*, "Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors," *Information Fusion*, vol. 62, pp. 47–62, 2020.
https://doi.org/10.1016/j.inffus.2020.04.004.

[16] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 2010, pp. 233–240.
https://doi.org/10.1109/INSS.2010.5573462.

[17] H. Gjoreski, B. Kaluža, M. Gams, R. Milić, and M. Luštrek, "Context-based ensemble method for human energy expenditure estimation," *Applied Soft Computing*, vol. 37, pp. 960–970, 2015.
https://doi.org/10.1016/j.asoc.2015.05.001.

[18] S. Kozina, H. Gjoreski, M. Gams, and M. Luštrek, "Three-layer activity recognition combining domain knowledge and meta-classification," *Journal of Medical and Biological Engineering*, vol. 33, no. 4, pp. 406–414, 2013.
http://dx.doi.org/10.5405/jmbe.1321.

[19] V. Janko, M. Gjoreski, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Gams *et al.*, "Winning the sussex-huawei locomotion-transportation recognition challenge," in *Human Activity Sensing*. Springer, 2019, pp. 233–250.
https://doi.org/10.1007/978-3-030-13001-5_15.

[20] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua science and technology*, vol. 19, no. 3, pp. 235–249, 2014.
https://doi.org/10.1109/TST.2014.6838194.

[21] A. Hoelzemann and K. Van Laerhoven, "Digging deeper: towards a better understanding of transfer learning for human activity recognition," in *Proceedings of the 2020 International Symposium on Wearable Computers*, 2020, pp. 50–54.
https://doi.org/10.1145/3410531.3414311.