Cross-dataset Deep Transfer Learning for Activity Recognition

Martin Gjoreski
Mitja Luštrek
Matjaž Gams
Jožef Stefan Institute, Department of Intelligent Systems, Jožef Stefan Postgraduate School, Slovenia,
martin.gjoreski@ijs.si

Stefan Kalabakov
Hristijan Gjoreski
Faculty of Electrical Engineering and Information Technologies,
University Ss. Cyril and Methodius, N. Macedonia

ABSTRACT

Convolution Neural Network (CNN) filters learned on one domain can be used as feature extractors on another similar domain. Transferring filters allow reusing datasets across domains and reducing labelling costs. In this paper, four activity recognition datasets were analyzed to study the effects of transferring filters across the datasets. A spectro-temporal ResNet was implemented as a deep, end-to-end learning architecture. We analyzed the number of transferred CNN residual blocks with respect to the size of the target-adaptation data. The analysis showed that transferring layers from domains with fine-grained activities might be more useful than transferring layers from domains with high-level activities. Furthermore, transferring three out of four residual blocks is the most robust choice for the specific architecture. The most successful transfer achieved an F1-score of 74%, which is an increase of 6 percentage points compared to a domain-specific baseline model.

CCS CONCEPTS

• Human-centered computing • Human computer interaction (HCI) • Ubiquitous and mobile computing

KEYWORDS

Activity recognition, Deep learning, End-to-end learning, Transfer learning

HASCA'19, October, 2019, London, United Kingdom

https://doi.org/10.1145/1234567890

1 Introduction

Deep learning represents a class of machine learning algorithms that use a cascade of multiple layers of non-linear processing units [1]. The first layer receives the input data, and each successive layer uses the output from the previous layer as the input. The basic idea dates from 1943 when McCulloch and Pitts created the first computational model of neural networks (NN) based on threshold logic [2]. Fast forward to the modern era, where large processing power and memory storage are relatively affordable, deep learning architectures are used to solve complicated AI tasks (e.g., in computer vision, language, biomedicine, etc.) by learning high-level abstractions [3][4].

In the last decade, many attempts at activity recognition (AR) were made with DL end-to-end architectures [8][9]. The focus was on Convolutional Neural Networks (CNNs). The CNNs are a type of NNs that are specially designed to ensure some degree of shift, scale and distortion invariance [5][6][7]. In addition, the CNNs can automatically capture hierarchical feature representations of the data [7]. Thus, CNNs, in combination with subsampling layers (e.g., pooling layers) and fully connected layers, are a very powerful end-to-end learning architecture [10][11][12]. However, because of the specific architecture (parameter sharing and local connections), the CNNs have much fewer connections and parameters to train compared to other types of NNs, e.g., fully connected NNs.

Since classical (feature-based) AR requires substantial amounts of labeled training data to perform well under diverse circumstances [16], end-to-end learning architectures require even bigger amounts of labeled data. One such example is the DeepConvLSTM introduced by Ordonez et al. [13] in 2016. The DeepConvLSTM uses several CNN layers stacked over a long short-term memory (LSTM) recurrent neural network (RNN) for AR [14]. To reduce the costs of acquiring labeled data, transfer learning can be employed. In the AR domain, transfer learning techniques provide mechanisms for transferring

instances, feature-representations, model-parameters and/or transfer or relational-knowledge [17]. Although transfer learning has been demonstrated to be feasible [18][19], it still remains a challenging task. Evidence from other domains (e.g., computer vision), where CNNs have been successfully applied for supervised tasks, suggest that the filters learned by the CNNs on one domain can be used as feature extractors on another similar domain. By analogy, this form of transfer learning has been also analyzed in the AR domain in the recent years. Ding et al. [20] analyzed deep transfer learning between users with unlabeled data using an end-to-end domain-adversarial neural network based on CNNs. Their architecture used shared filters across all sensor modalities, i.e., acceleration, magnitude and gyroscope data, applied over the temporal representation of the signals. Morales et al. [21] analyzed the DeepConvLSTM for transfer learning across mobile AR datasets, sensor modalities and sensor locations. Among other things, the authors have analyzed whether the features learned by the CNN's filters on the OPPORTUNITY dataset [22] can be re-used for building a model for the Skoda [23]. Their study confirmed that transferring filters across datasets is quite challenging and in most of the cases the domain-specific baseline model outperformed the models build using transfer learning. Similarly as Ding et al. [20], the DeepConvLSTM uses shared filters across all sensor channels, i.e., acceleration data, applied over the temporal representation of the signals.

Compared to the related work, our study provides several novelties: (i) to the best of our knowledge, this is the first study that analyzes four different datasets for transfer learning in the AR domain; (ii) our DNN architecture provides the possibility to learn CNN filters individually for each channel and thus to transfer channel-specific CNN filters; (iii) our DNN architecture provides the possibility to learn CNN filters for both temporal and spectral representations of raw signals, and thus it provides the possibility to transfer CNN filters not only in the temporal but also in the spectral domain

2 Datasets

The four datasets used to evaluate the effectiveness of the transfer learning method are: Skoda [23] OPORTUNITY [22], JSI-FOS [24][25] and PAMAP2 [26]. All datasets, except for Skoda, are comprised of activities of daily living, whereas Skoda is comprised of activities performed by a worker at a quality control checkpoint in a car factory. All of them contain information captured by a 3D accelerometer on the dominant hand of the wearer, which makes it ideal to use in this analysis since it focuses our transfer learning methods to the domain of the task, rather than the modality or location of the sensors used. More detailed information on the datasets can be found in Table 1.

Table 1. Overview of the experimental datasets.

Dataset	Type	#Subjects	Sampling Rate	#Activities	Duration [h]
OPPORTUNITY [22]	ADL	4	~30Hz	20	10.6
Skoda Mini Checkpoint [23]	Factory	1	96Hz	10	2.9
PAMAP2 [26]	ADL	9	100Hz	11	15.1
JSI-FOS [24][25]	ADL	10	100Hz	8	40.2

3 Deep Learning Architecture

The DNN architecture (Figure 1) used in this study is a deep multimodal spectro-temporal ResNet (Multi-ResNet). The Multi-ResNet has already been proved successful for AR in our previous study [27] by achieving comparable accuracy to state-of-the art feature-based models on one of the largest AR datasets, the SHL dataset [29]. The structure is based on an idea for training very deep end-to-end networks for image recognition: it uses shortcut (residual) connections to fight the gradient-vanishing problem [28]. Additionally, the network has two key factors for a successful AR system – it utilizes multimodal and spectro-temporal information. For each sensor channel, the network extracts channel-specific spectro-temporal information: the spectral information is extracted by calculating a spectrogram of the input signal, which is then used as input to a 1D convolutional layer; the temporal representation is extracted by residual blocks that contain 1D CNN filters. To reduce internal covariance shift, each CNN layer is followed by a batch normalization (BN) layer [30]. To speed up the training process, ReLU activation layers are used [31]. For dimensionality reduction, each residual block ends up with a maximum pooling layer. The output of the channel-specific layers, i.e., channel-specific spectro-temporal information, is then merged by two dense (fully connected) layers. To avoid overfitting, L2 regularization and dropout were used for the dense layers. The final output of the Multi-ResNet is provided by a softmax layer, which outputs probabilities for each class.

4 Experiments

4.1 Experimental Setup

Four datasets were used in this study and the effects of transfer learning were tested on each combination of these datasets. Firstly, all datasets were downsampled to a frequency of 25Hz and segmented to windows of 4s with an overlap between windows of 2s. Furthermore, all datasets were split into a train, validation and test subsets, which comprised 40%, 10% and 50% of the overall duration of each dataset, respectively.

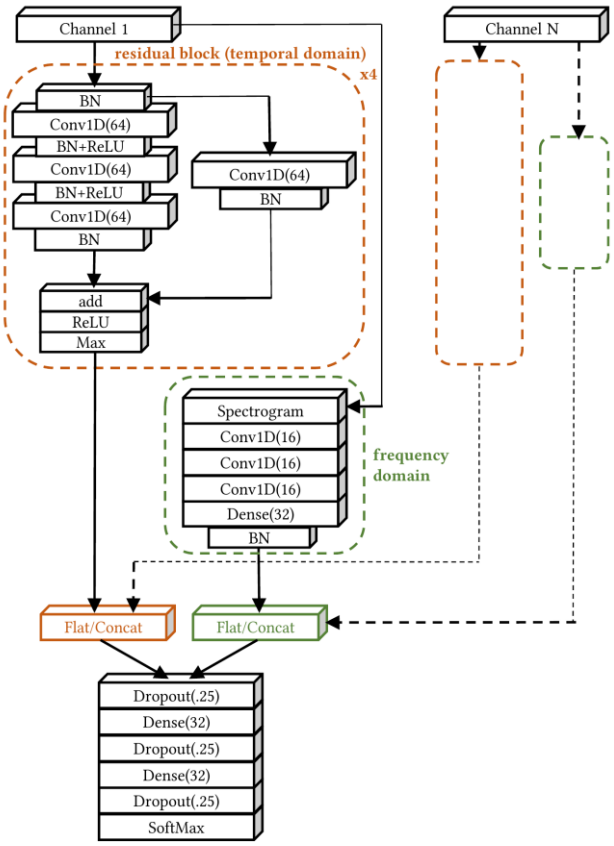

Figure 1. The deep multimodal spectro-temporal ResNet (Multi-ResNet) used in the study.

To get more realistic evaluation of our work, the split was done in such a way that every subject would appear in only one of the subsets. In the case were there was only one subject in the dataset, such as in the Skoda, the split was done per activity occurrence so that neighboring instances could occur in only one of the subsets.

In each experiment, one of the datasets served the purpose of a source domain and another dataset served as a target domain. At the start of each experiment, a model (source model) was trained on the training subset of the source dataset and validated on the validation subset of that same dataset. The training was fully supervised, by back-propagating the gradients through all the layers. In the next step, the model weights were transferred from the source model to a new model (target model), which would then go on to be adapted (trained) on a portion of the training subset of the target dataset (adaptation subset). In this case, all weights of the source model were transferred to the target model and, depending on the experiment, some of them were frozen during the training of the network. Thus, the gradients were back-propagated only up to a certain layer of the target model.

The experiments were carried out in such a way that the weights of whole residual blocks were transferred from the source model to the target model and not just individual convolutional layers. Also, several different sizes of the adaptation subset were tested in these experiments. However, the maximum number of instances never exceeded the full size of the training subset of the Skoda dataset, which is the smallest dataset of all four. Thus, in summary, two parameters were quantified in the experiments:

(i) the number of residual blocks whose weights would be transferred from the source model to the target model;

(ii) the size of the adaptation subset on which the target model would adapt to the new domain.

In addition, each experiment included the results of a model which was trained solely on the adaptation subset and had no weights transferred to it from the source model. This domain-specific model served as a baseline to which we compared our results. Finally, the results were scored using the F1-score with micro average.

All models were trained by minimizing the cross-entropy loss function using the Adam optimizer, using a learning rate of 10^{-3} and a decay of 10^{-3}. The batch size was set to 256 and the maximum number of training epochs was set to 70. The network parameters, including the number of residual blocks, the number of CNN layers per block, the size of the CNN filters, the learning rate and the batch size, were determined expe rimentally.

4.2 Experimental Results

The results of our experiments can be seen in Figure 2. The title of each heatmap represents: which dataset was used as the target domain (T); in the brackets is the F1-score achieved by a majority classifier for the specific target domain (24 for Skoda, 19 for OPORTUNITY, 26 for PAMAP and 24 for JSI-FOS); which dataset was used as the source domain (S). The vertical axis of every heatmap represents the number of instances in the adaptation subset. The horizontal axis represents the number of residual blocks whose weights were transferred from the source model to the target model. The numbers in each cell represents the micro F1-score achieved for the specific combination of number of instances in the adaptation set and the number of residual blocks whose weights were transferred. The leftmost column shows the F1-score of the domain-specific baseline model which was trained on that same adaptation subset of the target domain.

When Skoda is the target dataset (first-row heatmaps in Figure 2) and JSI-FOS or PAMAP are the source datasets individually, we can observe a degradation of performance compared to the baseline. On the other hand, an increase of performance can be observed when OPPORTUNITY serves as

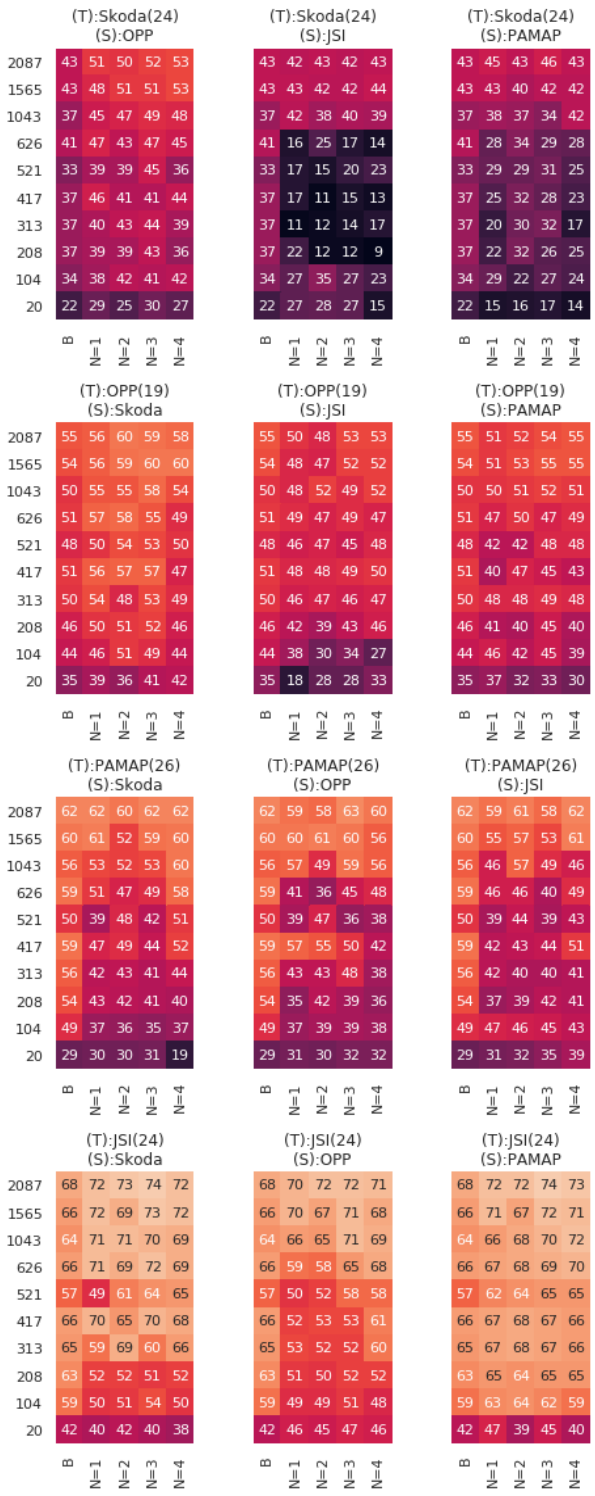

Figure 2. F1-score for each Target (T) - Source (S) domain combination. The y-axis represents the number of instances in the adaptation subset. The x-axis represents the number of residual blocks transferred from the S model to the T model. The leftmost column shows the F1-score of the domain-specific baseline model.

the source dataset, with the case of transferring the weights of three residual blocks appearing to be the most robust one.

When OPPORTUNITY is the target dataset (second-row heatmaps in Figure 2) and Skoda is the source dataset, the same increase of performance can be seen. Furthermore, when the source dataset is JSI-FOS or PAMAP, it either brings a decrease of performance or in the best case is on par with the baseline score.

When PAMAP is the target dataset (third-row heatmaps in Figure 2) and we use the maximum or near to the maximum number of instances in the adaptation subset, the scores for the baseline and transfer learning, with any number of weights transferred, are within a few percentage points of each other. But, a decrease of the F1 score can be noticed in cases of smaller adaptation subsets, no matter which dataset acts as the source.

Finally, when JSI-FOS is the target dataset (last-row heatmaps in Figure 2), another increase of performance can be seen when Skoda or PAMAP are used as the source datasets. More specifically, the transfer from PAMAP to JSI-FOS provides more consistent results. When OPPORTUNITY is the source dataset and the adaptation dataset is large enough, some increases of performance can also be observed.

5 Conclusion and Discussion

In our study, we looked at four different domains as sources and targets of our transfer learning experiments. The results show that the choice of the source domain is very important. This was particularly true in the case of OPPORTUNITY and Skoda, where transfer was successful only between them, and not when JSI-FOS or PAMAP were used as the source. This may be due to the fact that OPPORTUNITY and Skoda as datasets are comprised of fine-grained gestures and contain a lot of diversity, as opposed to JSI-FOS and PAMAP, which are comprised of more general, high-level activities. When JSI-FOS and PAMAP were the target, the choice of the source was less important, although the most successful transfer was still from PAMAP to JSI-FOS. We can conclude that transfer of convolutional layers between significantly different domains does not yield good results. This is especially true when weights are learned on a high-level and low variability source domain and transferred to a target domain with more fine-grained activities. We speculate that this is due to the fact that it is far easier to remove details during the adaptation process, as has to be done when transferring from fine-grained gestures to high-level activities, rather than learn new details on a small adaptation dataset.

Finally, there seems to be an underlying pattern in the scenarios where we noted an increase of performance which suggests that, for this specific network architecture, transferring three residual blocks proves to be the most robust

choice, as it commonly provides the best scores regardless of the number of instances in the adaptation subset.

Our experiments focused on building end-to-end models with very little data, i.e., starting from 20 instances (near 1 minute of data) to 2087 instances (near 75 minutes of data). As a consequence, the overall accuracy varies between 53% (on the Skoda) to 74% on the JSI-FOS dataset. Also, the experimental results are quite related to the specific DNN architecture, the Multi-ResNet. In future, we plan to test additional end-to-end DL architectures and to compare their performance. Also, we plan to use multiple datasets as a source, and this way to learn more general model. Finally, we plan to provide a comparison of transfer learning performed in the spectral domain as opposed to a transfer learning in the temporal domain.

REFERENCES

[1] L. Deng, D. Yu. (2014). Deep Learning: Methods and Applications (PDF). Foundations and Trends in Signal Processing. 7 (3–4): 1–199. doi:10.1561/2000000039.

[2] W. McCulloch, W. Pitts (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. 5 (4): 115–133. doi:10.1007/BF02478259.

[3] Y. Bengio (2009). "Learning Deep Architectures for AI." Technical Report 1312. [Online. Accessible at: https://www.iro.umontreal.ca/~lisa/pointeurs/TR1312.pdf]

[4] A. Krizhevsky, I. Sutskever, G. E. Hinton (2012). ImageNet classification with deep convolutional neural networks. in International Conference on Neural Information Processing Systems, pp. 1097-1105.

[5] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 609–616. ACM.

[6] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun (2009). What is the best multi-stage architecture for object recognition? In International Conference on Computer V sion, pages 2146–2153. IEEE.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324.

[8] H. Gjoreski, J. Bizjak, M. Gjoreski, M. Gams (2016). Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer, in: Proceedings of the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence, New York, NY, USA, Vol. 10.

[9] D. Ravi, C. Wong, B. Lo, G.-Z. Yang (2017). A deep learning approach to on-node sensor data analytics for mobile or wearable devices, IEEE journal of biomedical and health informatics 21 (1) 56–64.

[10] S. Bhattacharya, N. D. Lane, Sparsification and separation of deep learning layers for constrained resource inference on wearables, in: Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, ACM, 2016, pp. 176-189.

[11] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, J. Zhang (2014). Convolutional neural networks for human activity recognition using mobile sensors, in: 6th International Conference on Mobile Computing, Applications and Services, IEEE, pp. 197-205.

[12] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, S. Krishnaswamy (2015). Deep convolutional neural networks on multichannel time series for human activity recognition, in: Twenty-Fourth International Joint Conference on Artificial Intelligence.

[13] F. Ordonez, D. Roggen (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (1) 115.

[14] S. Hochreiter, J. Schmidhuber (1997). Long short-term memory, Neural computation 9 (8) 1735-1780.

[15] K. Krishna, D. Jain, S. V. Mehta, S. Choudhary (2018). An LSTM based system for prediction of human activities with durations, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1 (4) 147.

[16] D. Cook, K. D. Feuz, N.C. Krishnan (2013). Transfer learning for activity recognition: A survey. Knowledge and information systems, 36(3), pp.537-556.

[17] S. Pan, Q. Yang (2010). A survey on transfer learning. Knowledge Data Entering IEEE Trans 22(10):1345–1359

[18] A. Calatroni, D. Roggen, and G. Tröster (2011). Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion. In Proc. INSS, Vol. 6.

[19] M. Kurz, M. Hölzle, A. Ferscha, A. Calatroni, and others (2011). Real-time transfer and evaluation of activity recognition capabilities in an opportunistic system. In Proc. ADAPTIVE. 73–78.

[20] R. Ding et al. (2019). Empirical Study and Improvement on Deep Transfer Learning for Human Activity Recognition. Sensors, 19(1), 57.

[21] F. J. O. Morales, D. 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 (pp. 92-99). ACM.

[22] D. Roggen, et al. (2010). Collecting complex activity datasets in highly rich networked sensor environments. In 2010 Seventh international conference on networked sensing systems (INSS) (pp. 233-240). IEEE.

[23] P. Zappi et al. (2008). Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In European Conference on Wireless Sensor Networks (pp. 17-33). Springer, Berlin, Heidelberg.

[24] H. Gjoreski, B. Kaluža, M. Gams, Radoje Milić, M. Luštrek (2015). Context-based Ensemble Method for Human Energy Expenditure Estimation. Applied Soft Computing, Pages 960–970.

[25] S. Kozina, H. Gjoreski, M. Gams, M. Luštrek (2013). Three-layer activity recognition combining domain knowledge and meta-classification. Journal of Medical and Biological Engineering, vol. 33, no. 4. p.p. 406-414.

[26] A. Reiss, D. Stricker (2012). Introducing a new benchmarked dataset for activity monitoring. In 2012 16th International Symposium on Wearable Computers (pp. 108-109). IEEE.

[27] M. Gjoreski et al. (2019). Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. Information Fusion, Under review.

[28] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies

[29] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, D. Roggen (2018). The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics with Mobile Devices." IEEE Access, 2018, DOI: 10.1109/ACCESS.2018.2858933

[30] S. Ioffe, C. Szegedy (2015). Batch normalization: Accelerating deep network train- ing by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.

[31] V. Nair, G. E. Hinton (2010). Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.