Robust Human Activity Recognition based on Deep Metric Learning

Mubarak G. Abdu-Aguye[1] and Walid Gomaa[1,2]
[1]*Computer Science and Engineering Department, Egypt-Japan University of Science and Technology, Alexandria, Egypt*
[2]*Faculty of Engineering, Alexandria University, Egypt*

Keywords: Activity Recognition, Deep Metric Learning, Convolutional Neural Networks.

Abstract: In the domain of Activity Recognition, the proliferation of low-cost and sensor-enabled personal devices has led to significant heterogeneity in the data generated by users. Traditional approaches to this problem have previously relied on handcrafted features and template-matching methods, which have limited flexibility and performance with high variability. In this work we investigate the use of Deep Metric Learning in the domain of activity recognition. We use a deep Triplet Network to generate fixed-length descriptors from activity samples for purposes of classification. We carry out evaluation of our proposed method on five datasets from different sources with differing activities. We obtain classification accuracies of up to 96% in self-testing scenarios and up to 91% accuracy in cross-dataset testing without retraining. We also show that our method performs similarly to traditional Convolutional Neural Networks. The obtained results indicate the promise of this approach.

1 INTRODUCTION

Activity Recognition is aimed towards determining the particular action a user is carrying out or the manner in which said action is being performed. This is primarily done for therapeutic purposes (Liu et al., 2016), (De et al., 2015) and for other intelligent applications where such information may be necessary for some action to be taken, e.g., Smart Homes (Mehr et al., 2016), (Hoque and Stankovic, 2012), etc.

With the myriad devices carried or worn by users today, activity data may be easily collected from and possibly processed on the devices themselves, allowing for pervasive adoption of this domain. At the same time, this ubiquity gives rise to significant heterogeneity in the data generated by users. One source of this is differences in the properties (e.g device placement, sampling rates, sensor biases, etc) of the devices used to collect the data (Banos et al., 2014), (Stisen et al., 2015). Another significant source of this arises due to differences in the wearers' mannerisms while performing such actions (Barshan and Yurtman, 2016). Such heterogeneity, while expected, is undesirable in practice. It complicates the problem of activity recognition and must therefore be taken into consideration when attempting to solve this problem. Therefore, methods capable of performing well even in the presence of such heterogeneity are of no small import for real-world scenarios.

Activity recognition is essentially a classification problem, and as such necessitates feature extraction. Manual feature extraction relies on the estimation of statistical, structural or transient features of the given data. In general, several types of features may need to be combined to achieve good performance. More recently, much emphasis has been placed on Deep Learning-based methods, as they are capable of automatic and problem-specific feature extraction, which ultimately leads to significant improvements over non-deep methods together with considerably less manual input. Given the complexity of the activity recognition problem, deep methods are virtually a necessity as a result of these desirable properties.

One intuitive way of tackling this heterogeneity is to embed the activity samples into a semantic space where samples of the same label are "close" to each other as quantified by some metric, and are "distant" from samples of different labels by the same metric. One of the most successful approaches which relies on this metric/distance approach is Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978). DTW has shown good results on many problems (Sempena et al., 2011), (Singhal and Dubey, 2015), (Cui and Zhu, 2013). However, it suffers from a number of issues, chief amongst which are its generally high computational requirements and the difficulty in gener-

Abdu-Aguye, M. and Gomaa, W.
Robust Human Activity Recognition based on Deep Metric Learning.
DOI: 10.5220/0007916806560663
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 656-663
ISBN: 978-989-758-380-3

alizing to multivariate data, a category under which most activity recognition data falls. It does not explicitly attempt to solve the problem of heterogeneity and may require complex steps in order to derive suitable class or group templates for future matching/classification. In addition, it does not yield an intermediate representation of the input data which may be used for any other purpose.

Metric Learning (Xing et al., 2002) aims to learn a suitable projection of the features of some input data into some abstract space such that data items in the same group are similar as quantified by some mathematical distance function, and data items in different groups are dissimilar by the same function. The reformulation permitted through Metric Learning is a natural and powerful paradigm which offers a number of benefits. It radically simplifies the classification problem within the learned feature space by emphasizing separability between classes without relying on strict cohesion or similarity of intra-class elements. This implicitly allows for heterogeneity between intra-class samples such as is found in this domain. In addition, its potential for generalization can be expected to be better than regular methods which become inherently biased/attuned to the training data in the course of minimizing the classification error. It also yields an output "embedding" which is a compact, fixed-length representation of the input data, and may be used for any purpose as desired.

Traditional metric learning methods e.g (Weinberger and Saul, 2009) rely heavily on the solution of nontrivial convex optimization problems, which are computationally intensive and assume that the transformation is linear. Subsequent work has extended metric learning with nonlinear methods (Kedem et al., 2012), leading to significant improvements in performance due to their inherent flexibility. Most recently, nonlinear metric learning via deep neural networks has been explored (Hoffer and Ailon, 2015), enabling the learning of both suitable features and corresponding nonlinear mappings for best results. Some results achieved through this method (Huang et al., 2012), (Hu et al., 2014) indicate that such *Deep Metric Learning* shows immense potential in other different scenarios.

In this vein, we explore the application of deep metric learning to the domain of activity recognition. We consider five diverse and publicly-available datasets of different sizes collected from different devices with different characteristics i.e device placement, sampling rate, etc. This is done in order to examine the robustness of this approach in the presence of device and data heterogeneity which is a reality in practical scenarios. We show that deep metric learning provides good classification accuracy and generalization ability out of the box (i.e without any retraining). The following sections introduce deep metric learning and Triplet Networks and our experimental methodology.

2 RELATED WORK

In this section we briefly discuss similar work that has been done in this domain. In (Li et al., 2018), the authors evaluate both traditional and deep methods in activity recognition. They consider CNN, LSTM and CNN+LSTM architectures and perform evaluations on two publically-available datasets. They find that deep architectures significantly outperform traditional approaches by a significant margin. In (Zeng et al., 2014) the authors adopt a convolutional neural network to perform activity recognition using accelerometer signals only. They consider three public datasets. Their method was found to outperform traditional methods. Similar findings were obtained in (Ha et al., 2015) where the authors used convolutional neural networks in a multimodal approach. Their method was found to be comparable to state of the art methods.

In (Margarito et al., 2016) the authors consider a template-matching approach to sports activity recognition based on accelerometer data. Several distance measures, including Euclidean and DTW distance were used, as well as different classification methods. Their proposed matching index was found to give the best results, suggesting that general purpose distance measures may not be well suited to all problems. Another approach is taken in (Seto et al., 2015) where the authors use multivariate DTW for template-matching in activity recognition for real and simulated datasets. They report results comparable to those obtained with traditional feature extraction methods. However, their method is highly susceptible to the manner in which the templates are constructed, showing significant variance in performance based on different approaches.

In (Che et al., 2017) the authors propose a method for metric learning for multivariate time series. They formulate an optimized method for aligning two multivariate time series inputs based on the average best warping length. Subsequently they utilize a 2-layer neural network to perform metric learning using the largest margin approach (Weinberger and Saul, 2009) to minimize the distance between the aligned series. In comparison with other methods such as multivariate DTW, their proposed method reportedly provides best-in-class performance. However, their method re-

lies on DTW distance during its training phase, and computes the learned distance over every timestep of the aligned series, both of which generally add non-trivial computational overhead to their method.

In this work we apply deep metric learning specifically to activity recognition data. Our method, in contrast to (Seto et al., 2015) does not rely on the derivation of any templates and avoids the computational overhead of DTW-based approaches. In contrast to (Che et al., 2017), our proposed method does not rely on pre-alignment of the training data. Additionally, our method yields an output representation upon which the similarity may be directly computed for any desired purposes. This saves the computational requirement of per-timestep similarity computation. Finally, we use convolutional filters for feature extraction and train our network as a single unit, allowing for end-to-end optimization of the metric learning objective.

3 THEORETICAL BACKGROUND

3.1 Metric Learning

Metric learning as a concept relies very heavily on the notion of similarity between pairs of entities. From a human point of view, this similarity is implicitly defined in terms of the closeness of sensory inputs induced by two observed phenomena. In more concrete terms, we will introduce the concept of a *distance function* or *metric*, which numerically quantifies the distance/dissimilarity between two elements of some set. To be admissible as a metric, a function must fulfill certain axioms. Considering the metric as a mapping D over two members - G_1 and G_2 - of a set, then we have that D: $G_1 \times G_2 \longrightarrow R^+$ for every x_i, x_j, x_k which exist in some set. We may express the axioms as:

1. Non-negativity: $D(x_i, x_j) \geq 0$
2. Identity: $D(x_i, x_i) = 0$
3. Symmetry: $D(x_i, x_j) = D(x_j, x_i)$
4. Triangle Inequality: $D(x_i, x_k) \leq D(x_i, x_j) + D(x_j, x_k)$

In this work we consider the Mahalanobis Distance Function(Mahalanobis, 1936), which takes the following form:

$$D = \sqrt{(x_i - x_j)^T M (x_i - x_j)} \tag{1}$$

This function is parameterized by M, which is a positive semidefinite matrix and permits the function fulfill the fourth condition described above. Therefore, depending on M, the computed distance may be considered to be obtained in a space whose projection is defined by M. It is noteworthy at this point to state that Euclidean distance is a special case of Mahalanobis distance when M is the identity matrix. Without a loss of generality however, we may consider an alternate form of the preceding function by incorporating a function F(.), which performs some transformation on the input space. We then obtain the following:

$$D_L = \sqrt{(F(x_i) - F(x_j))^T)(F(x_i) - F(x_j))} \tag{2}$$

We can consider the transformation by the M matrix from before to be a special case of the above where F(.) is a linear function (since matrix multiplication is equivalent to the application of some linear function to a given input vector). The metric learning problem as defined by (Weinberger and Saul, 2009) can then be considered to be the learning of a suitable function F(.) such that, given an element x_i and two others x_j and x_k such that $x_i, x_j \in A$, $x_k \in B$ (where A and B are two different classes/groups) then:

$$D_L(x_i, x_j) < D_L(x_i, x_k) \tag{3}$$

It can also be seen from this new form that such a formulation imposes this condition in Euclidean space, as there is no longer a need for a separate projection matrix M explicitly. The element x_i is called the *anchor*, while x_j and x_k are called the *positive* and the *negative* respectively. In this vein, it can be seen that such a transformation would lead to a feature space where similar classes occupy the same region and are distinct from entities from differing classes. This implies that the decision surface becomes much simpler to intuit as the constraint defined in (3) implicitly emphasizes discriminability between classes. In deep metric learning, a type of deep neural network is used to find a suitable approximation for F(.). The details by which this is achieved are discussed in the proceeding subsection.

3.2 Triplet Networks

Triplet networks were first introduced in (Hoffer and Ailon, 2015) with particular application to deep metric learning. In practice, the distinctive feature of such a network is not in its structure but the manner by which it is trained and type of loss function which is used in its optimization. Since they are designed specifically for deep metric learning, a Triplet Loss (Schroff et al., 2015) function is used which aims to learn a nonlinear mapping from the input features to the network output (which is a fixed-size vector called an embedding) that enforces the constraints

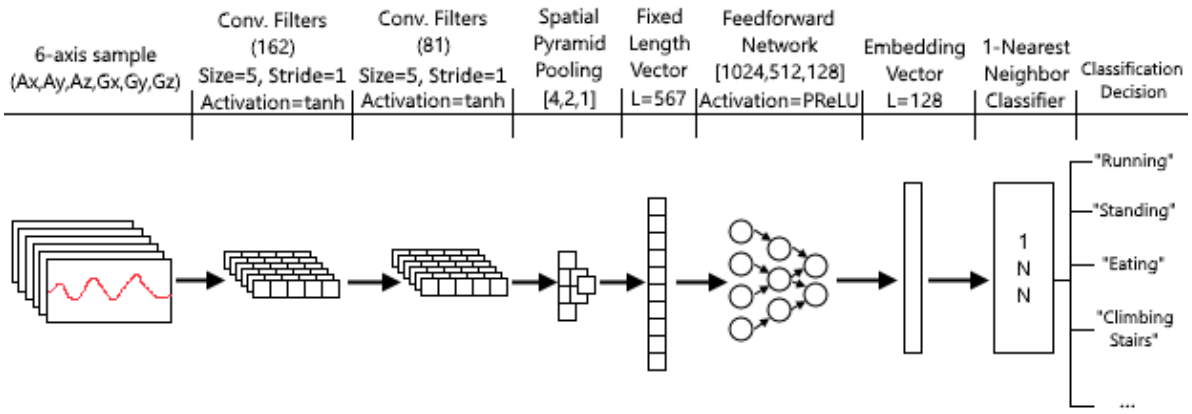

Figure 1: Structure of Proposed Method.

as described in (3). After training, the embeddings correspondingly generated from the network inputs are similar for data from the same class and dissimilar for data from different classes, using Euclidean distance as the metric.

During the training of this sort of network, an anchor is randomly selected, together with a positive and a negative. A single tuple consisting of these three inputs is called a triplet. The manner in which the triplets are selected influences the eventual performance of the network, and must be carefully taken into account for best results. For further information regarding triplet selection strategies the reader is referred to (Hermans et al., 2017). In mathematical terms, triplet loss is defined as follows:

$$L_t = argmin \sum max(||E_a - E_p||_2^2 - ||E_a - E_n||_2^2 + \alpha, 0) \quad (4)$$

Where E_a represents the embedding of the anchor sample, E_p represents the embedding of the positive and E_n represents the anchor of the negative (where anchor, positive and negative retain their meanings as described in the previous section) and the summation of the loss is taken over all the triplets in the training set. α represents a margin parameter that describes the minimum desired spacing between entities of differing classes. This can be seen to be tailored achieving the objective set forth in (3).

Therefore, once suitable triplets are selected, each triplet is passed through the network to obtain its constituent embeddings. The loss is then computed as defined by (4) and is subsequently backpropagated through the network using standard methods.

4 PROPOSED METHODOLOGY

In this section we describe the details of our proposed method. We construct a triplet network consisting of two convolutional layers and three feedforward layers. We include the convolutional layers and train the network as a whole in order to achieve optimized feature extraction and transformation in an end-to-end way. As the network inputs do not have the same length, we include a 1-D Spatial Pyramid Pooling layer (He et al., 2014) between the convolutional and the feedforward layers. This layer converts the varying-length outputs from the convolutional layers into fixed length vectors which are required by the feedforward layers. Batch Normalization is used in between successive feedforward layers as it was found to speed up training convergence significantly. The network structure is shown in Figure 1.

The network is trained using the triplet loss function as described previously, fixing the α (i.e margin) parameter at 1. In this work Random Negative Triplet Selection (Hermans et al., 2017) was used as the triplet selection strategy as it was found to achieve the best results relative to the other methods. Stochastic Gradient Descent with a Nesterov Momentum value of 0.90 was used as the optimizer as it was found to give the best performance in our evaluations.

During testing, a sample is passed through the network, yielding a 128-feature embedding which is can then be considered to be its representative feature vector/encoding. This embedding vector can then be used as a feature vector for classification or other purposes.

Table 1: Summary of Datasets Considered.

Name	Samples	Activities
Gomaa-1	603	14
HAPT	1214	12
Daily_Sports	9120	19
HAD-AW	4344	32
REALDISP	1397	33

5 EXPERIMENTAL SETUP

In this section we describe the experimental setup and methodology which was used to investigate this method. A brief description of the datasets considered is provided as follows.

5.1 Datasets Considered

We considered five datasets collected from different sources with different activity sets. This was done with a view to evaluating our proposed techniques in terms of its flexibility and adaptability. The details of the datasets considered as summarized in Table 1.

The Gomaa-1 dataset (Gomaa et al., 2017) consists of 603 samples spread over 14 different activities. The dataset was collected from an Apple SmartWatch at a sampling rate of 50Hz. The data was collected from three volunteers wearing the Smartwatch on their right hands. The dataset consists of accelerometer, gyroscope, magnetometer and rotary (i.e roll, pitch and yaw) readings.

The HAPT dataset (Reyes-Ortiz et al., 2016) consists of 1214 samples spread over 12 activities: 6 static activities and 6 postural transitions. It was collected from waist-worn Android smartphones at a sample rate of 50Hz. The samples were collected from 30 volunteers and include only accelerometer and gyroscope readings.

The Daily and Sports Activities dataset (Altun et al., 2010) consists of 9120 samples spread over a mixture of 19 daily and sports activities. It was collected using Xsens IMU units at a sample rate of 25Hz. The samples were collected from 8 volunteers and consist of accelerometer, gyroscope and magnetometer readings.

The HAD-AW dataset (Ashry et al., 2018) consists of 4344 samples spread over a diverse set of 31 activities. It was collected using an Apple SmartWatch at a sample rate of 50Hz. The samples were collected from 16 volunteers and are composed of accelerometer, gyroscope, magnetometer and rotary readings.

The REALDISP dataset (Baños et al., 2012) consists of 4000+ samples distributed over 33 activities. It was collected using Xsens IMU units at a sampling rate of 50Hz, and consists of accelerometer, gyroscope and magnetometer data as well as orientation quaternions. The data was collected from 17 subjects in three different device-placement scenarios. We consider data from all the available scenarios and ignore all samples with indeterminate labels. After preprocessing 1397 samples were found to be usable.

Due to the varying sensor modalities available from these datasets, we use only the accelerometer and gyroscope data, yielding a total of six axes from each dataset. This is because the accelerometer and gyroscope are common across all the datasets and therefore models constructed from such data can easily be applied across datasets.

5.2 Experimental Evaluations

In order to illustrate the efficacy of our proposed method, we carry out a number of experiments on the datasets described previously in order to determine the classification accuracy obtainable from the embeddings generated by our method. We adopt 1-NN classification due to its simplicity and its applicability to similarity-based applications such as metric learning. In addition, it is the most popular benchmark used in evaluating similarity-based techniques (Ding et al., 2008). The PyTorch library (Paszke et al., 2017) was used for the practical implementation of all the experiments.

We construct and train a triplet network as described in Section 4. 75% of the considered datasets were used for training. After training, the embeddings of the training samples are used as exemplars for a 1-Nearest Neighbor classifier. This classifier is then used to obtain the classification accuracy on the embeddings of the remaining unseen 25% of the dataset.

We also perform cross-testing experiments to determine the generalizability of our proposed method. In this case the triplet network is trained on one of the described datasets as before. *Without any retraining*, we use the pretrained network to generate embeddings from the other (cross-testing) datasets. 65% of the embeddings are used as exemplars for a 1-Nearest Neighbor classifier while the classification accuracy is evaluated on the remaining 35% of the embeddings.

In both cases, the evaluations are each repeated 15 times i.e the network is trained and tested on different data for 15 cycles. The mean and standard deviation of the accuracies obtained from each cycle are then computed. In order to provide a sense of the efficacy of our method compared to traditional methods, we construct a Convolutional Neural Network using an

identical structure to our triplet network. However, we omit the 1-NN classifier and replace the last layer with a layer with K neurons, where K is the number of classes in the dataset being evaluated. This way the CNN produces a classification decision directly. We then train the CNN in a similar way i.e using 75% of the dataset considered for training and the remaining 25% for testing. The CNN is trained using the Adam optimizer and Cross Entropy as the loss function. The training and testing cycles of the CNN are also repeated 15 times and the mean and standard deviation recorded.The results obtained from these evaluations and their discussion are provided in the following section.

6 RESULTS AND DISCUSSION

In this section we present the results from the experimental evaluations as described previously. The results are given in Table 2 and are all reported as percentages. The accuracies are shown together with their standard deviations to give a sense of the stability of the obtained results.

Accuracies obtained from training on some dataset and testing on the same dataset are shown in grey in the table. As can be observed, our proposed method gives good performance on all five considered datasets, regardless of the data size or number of classes available. The standard deviation of the figures also indicate that our proposed method gives consistently stable results, even with the use of a simple 1-NN classifier. This underlines the potential of this metric learning approach in this domain.

In terms of the system's cross-testing performance, we see that good results can be obtained with our proposed method for datasets with a moderate number of classes. This is reflected in the cross-testing results for the Gomaa-1, HAPT and Sports dataset, where our proposed method yields a classification accuracy averagely around 85% without any retraining at all, even though these three datasets are collected from different devices in different scenarios. A decline in performance is observed when cross-testing on datasets with significantly more classes than seen during training, as reflected in the results of cross-testing on the HAD-AW dataset (32 classes) and the REALDISP dataset (33 classes). However, it can be observed that when training is carried out on these datasets with many classes, the cross-testing performance is generally good, though their performances on each other show significant degradation.

The performance degradation in the former case can likely be attributed to the fact that when this method is trained on datasets with a moderate number of classes, its discriminatory ability is limited to some degree. In this situation, testing on datasets with many more classes would pose a significant challenge as the model is not powerful/discriminative enough. However, when it is trained on a dataset with many classes, its discriminatory ability is much better and therefore its cross-testing performance on smaller datasets is expectedly better. This can be observed from the results. When training on and testing on datasets with large number of classes (i.e training on HAD-AW, testing on REALDISP and vice-versa) degraded performance is also observed. This is likely because of the severe difference in the types of activities that constitute these datasets. In spite of this, it can be observed that the cross-performance of these two datasets on each other still exceeds the cross-testing performance obtained when training on the less-diverse datasets. As such the benefit on training on datasets with many classes is clearly illustrated. However, from the results obtained in the cross-testing scenarios, it can generally be surmised that the proposed method shows the system's robustness.

Table 3 also shows the performance of an identically-structured CNN on the same datasets. It can generally be seen that our method provides results comparable to the CNN in general. Additionally, the CNN cannot be used for cross-testing purposes without network redesign and retraining. This further helps to underline the efficacy, competitiveness and wide applicability of our method.

7 CONCLUSION

In this work we introduce the use of Triplet Networks in the domain of human activity recognition. Accelerometer and gyroscope data from different datasets were used to train a triplet network, which was subsequently used to generate fixed-size vectors (embeddings) from the varying-length multivariate samples. A 1-Nearest Neighbor classifier was then used to evaluate the classification performance using the generated embeddings as feature vectors. Additionally, cross-testing was carried out whereby the network was trained on some dataset and used to generate embeddings from the other datasets without any retraining being carried out.

Our proposed method was found to yield classification accuracies of up to 96% in the self-testing (i.e training and testing on same dataset) scenarios. In the cross-testing scenarios, our method showed good performance (up to 91% accuracy) on datasets with

Table 2: Classification Accuracy of Embeddings generated by Proposed Method.

Training Dataset	**Testing Dataset**				
	Gomaa-1	HAPT	D_Sports	HAD-AW	REALDISP
Gomaa-1	96.69 ± 1.23	89.60 ± 1.72	91.34 ± 0.41	68.67 ± 1.09	59.48 ± 2.77
HAPT	83.27 ± 2.16	96.25 ± 1.09	89.98 ± 0.74	64.46 ± 1.46	50.61 ± 2.19
D_Sports	86.16 ± 2.69	88.67 ± 1.17	96.59 ± 0.39	68.62 ± 1.14	56.59 ± 2.27
HAD-AW	90.75 ± 1.42	89.10 ± 1.41	91.35 ± 0.56	85.28 ± 1.19	60.60 ± 2.00
REALDISP	91.16 ± 2.14	91.40 ± 1.38	91.63 ± 0.47	70.10 ± 1.65	75.01 ± 1.92

Table 3: Classification Accuracy using CNN.

Dataset	**Accuracy**
Gomaa-1	96.99%
HAPT	96.00%
Daily_Sports	95.31%
HAD-AW	88.36%
REALDISP	76.60%

a small to moderate number of classes even without retraining. When trained on datasets with many classes, the method shows the best performance on both types of datasets (i.e moderate classes and many classes), indicating the improved discriminative ability imparted by training on such type of datasets. In both self- and cross-test scenarios, the standard deviations are below 3% for all datasets. These results underscore the robustness of the proposed method.

In the future we intend to investigate the effects of the embedding size on the classification accuracy obtained, as well as evaluate the topological properties of the embeddings with a view to dimensionality reduction. We also intend to consider network optimization (i.e finding the best deep network architecture and performing hyperparameter tuning for the network and loss function) as we believe that a carefully-designed network may yield further performance gains. Additionally, we intend to explore the efficacy of fine-tuning of pretrained triplet networks. This follows from the cross-testing performance observed thus far, which suggests that additional dataset-specific training may yield better cross-testing performance.

REFERENCES

Altun, K., Barshan, B., and Tunçel, O. (2010). Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recogn.*, 43(10):3605–3620.

Ashry, S., Elbasiony, R., and Gomaa, W. (2018). An lstm-based descriptor for human activities recognition using imu sensors. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, pages 494–501. INSTICC, SciTePress.

Baños, O., Damas, M., Pomares, H., Rojas, I., Tóth, M. A., and Amft, O. (2012). A benchmark dataset to evaluate sensor displacement in activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 1026–1035, New York, NY, USA. ACM.

Banos, O., Toth, M. A., Damas, M., Pomares, H., and Rojas, I. (2014). Dealing with the effects of sensor displacement in wearable activity recognition. *Sensors*, 14(6):9995–10023.

Barshan, B. and Yurtman, A. (2016). Investigating inter-subject and inter-activity variations in activity recognition using wearable motion sensors. *The Computer Journal*, 59(9):1345–1362.

Che, Z., He, X., Xu, K., and Liu, Y. (2017). Decade: a deep metric learning model for multivariate time series. In *KDD workshop on mining and learning from time series*.

Cui, H. and Zhu, M. (2013). A novel multi-metric scheme using dynamic time warping for similarity video clip search. In *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, pages 1–5.

De, D., Bharti, P., Das, S. K., and Chellappan, S. (2015). Multimodal wearable sensing for fine-grained activity recognition in healthcare. *IEEE Internet Computing*, 19(5):26–35.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552.

Gomaa, W., Elbasiony, R., and Ashry, S. (2017). Adl classification based on autocorrelation function of inertial signals. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 833–837.

Ha, S., Yun, J., and Choi, S. (2015). Multi-modal convolutional neural networks for activity recognition. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3017–3022.

He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 346–361, Cham. Springer International Publishing.

Hermans, A., Beyer, L., and Leibe, B. (2017). In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737.

Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer.

Hoque, E. and Stankovic, J. (2012). Aalo: Activity recognition in smart homes using active learning in the presence of overlapped activities. In *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, pages 139–146.

Hu, J., Lu, J., and Tan, Y. (2014). Discriminative deep metric learning for face verification in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1882.

Huang, C., Zhu, S., and Yu, K. (2012). Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. *CoRR*, abs/1212.6094.

Kedem, D., Tyree, S., Sha, F., Lanckriet, G. R., and Weinberger, K. Q. (2012). Non-linear metric learning. In *Advances in Neural Information Processing Systems*, pages 2573–2581.

Li, F., Shirahama, K., Nisar, M. A., Köping, L., and Grzegorzek, M. (2018). Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, 18(2).

Liu, X., Liu, L., Simske, S. J., and Liu, J. (2016). Human daily activity recognition for healthcare using wearable and visual sensing data. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 24–31.

Mahalanobis, P. C. (1936). On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55.

Margarito, J., Helaoui, R., Bianchi, A. M., Sartor, F., and Bonomi, A. G. (2016). User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach. *IEEE Transactions on Biomedical Engineering*, 63(4):788–796.

Mehr, H. D., Polat, H., and Cetin, A. (2016). Resident activity recognition in smart homes by using artificial neural networks. In *2016 4th International Istanbul Smart Grid Congress and Fair (ICSG)*, pages 1–5.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Reyes-Ortiz, J.-L., Oneto, L., Samà, A., Parra, X., and Anguita, D. (2016). Transition-aware human activity recognition using smartphones. *Neurocomput.*, 171(C):754–767.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832.

Sempena, S., Maulidevi, N. U., and Aryan, P. R. (2011). Human action recognition using dynamic time warping. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pages 1–5.

Seto, S., Zhang, W., and Zhou, Y. (2015). Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 1399–1406.

Singhal, S. and Dubey, R. K. (2015). Automatic speech recognition for connected words using dtw/hmm for english/ hindi languages. In *2015 Communication, Control and Intelligent Systems (CCIS)*, pages 199–203.

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. (2015). Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 127–140, New York, NY, USA. ACM.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2002). Distance metric learning, with application to clustering with side-information. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 521–528, Cambridge, MA, USA. MIT Press.

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J. (2014). Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pages 197–205.