Pattern Recognition Letters 146 (2021) 90–99

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Modality-wise relational reasoning for one-shot sensor-based activity recognition

Panagiotis Kasnesis*, Christos Chatzigeorgiou, Charalampos Z. Patrikakis, Maria Rangoussi

University of West Attica, P.Ralli & Thivon 250, Egaleo 12241, Greece

ARTICLE INFO

Article history:
Received 28 June 2020
Revised 20 January 2021
Accepted 1 March 2021
Available online 17 March 2021

MSC:
41A05
41A10
65D05
65D17

Keywords:
Deep learning
One-shot learning
Human activity recognition
Relational reasoning
Self-attention

ABSTRACT

Deep learning concepts have been successfully transferred from the computer vision task to that of wearable human activity recognition (HAR) over the last few years. However, deep learning models require a large volume of annotated samples to be efficiently trained, while adding new activities results in training the whole network from scratch. In this paper, we study the use of one-shot learning techniques based on high-level features extracted by deep neural networks that rely on convolutional layers. Using these feature vectors as input we measure the similarity of two activities by computing their Euclidean distance, cosine similarity or applying self-attention to perceive the relations between the signals. We evaluate four different one-shot learning approaches using two publicly available HAR datasets, by keeping out of the training set several activity classes. Our results demonstrate that the model relying on modality-wise relational reasoning surpasses the other three, achieving 94.8% and 84.41% one-shot accuracy on UCL and PAMAP2 dataset respectively, while we demonstrate the model's sensitivity on fusing sensor modalities and provide explainable attention maps to display the modality-wise similarities.

1. Introduction

Wearable human activity recognition (HAR) can be used to enhance wellbeing and health status [23], facilitate smart environments [14] and improve physical security in public spaces [10]. In contrast to other HAR methods relying on sensors that suffer from privacy concerns (e.g., camera), wearable activity monitoring is unobtrusive. Moreover, similarly to computer vision, natural language processing, and speech recognition, wearable HAR has not remained unaffected by the rise of deep learning (DL) [7]. DL algorithms such as Convolutional Neural Networks (ConvNets or CNNs), have been proven to be capable of automatically extracting features from almost raw motion signals [11]; these high-level features are fed, afterwards, to fully connected (FC) layers or Recurrent Neural Networks (RNNs) enhanced with the Long Short-Term Memory (LSTM) mechanism, to fuse the multimodal features and classify incoming sensor channels into an activity [20].

Moreover, HAR DL algorithms outperform the performance of the standard machine learning classifiers [10], such as Support Vector Machines (SVM), which are fed with hand-crafted features of time and frequency domain [2]. Nevertheless, DL algorithms have a huge drawback; they require huge volumes of labeled data in order to be efficiently trained, while motion signal annotation is a labor-intensive and time-consuming procedure. What is more, if an activity is not included in the training dataset, the engineers must retrain the DL model from scratch to include it. Thus, there is the need of finding ways to alleviate these issues.

Few-shot learning aims at recognizing similar activity classes in a set of data, where we have few instances of the same classes; the case of having only one instance of the same class with the anchor signal in the comparison set is called one-shot learning [26,31]. One-shot learning is achieved by extracting features from the training samples that can generalize, called embeddings. An embedding is a mapping of a discrete variable (e.g., a word) to a low-dimensional vector of continuous numbers, which is useful in neural nets since we can measure its norm distance to other embeddings that we are aware of their class [26].

In this paper, we investigate whether DL architectures can be used to apply relational reasoning [25] on the extracted embeddings to measure the class-wise relevance between activity samples that have only one labeled sample. The contributions and innovations of the current work can be summarized in the following:

1. We examine deep learning model architectures applied to one-shot sensor-based HAR.

* Corresponding author.
E-mail address: pkasnesis@uniwa.gr (P. Kasnesis).

https://doi.org/10.1016/j.patrec.2021.03.003
0167-8655/

2. We introduce a modality-wise relational network to discover activity similarities.
3. We extensively investigate multi-head self-attention performance on one-shot HAR.
4. We construct explainable attention maps to display the modality-wise similarities.

The rest of this paper is organized as follows. Section 2 describes related works on DL-based HAR focusing on few-shot learning. Section 3 elaborates on all the examined one-shot learning network architectures, while Section 4 describes the processed datasets and the experimental setup. The results of our experiments and the comparative analysis with existing works are presented in Section 5. Finally, Section 6 concludes the paper and presents future steps.

2. State of the art

The first CCN approach to wearable HAR was introduced in [37]. The authors used as input a 1D array representation of the motion signals (i.e., tri-axial accelerometer), stacking them into channels (channel-based stacking), just like RGB images use three channels (i.e., Red, Blue, Green channels). Moreover, the authors of [24,38] proposed a similar network architecture; all these HAR ConvNets fuse the motion data in the first hidden layer and, follow an early fusion strategy. Later works show that stacking the input signals vertically is more efficient, enabling the network to fuse the extracted features later using a dense layer [22], a 2D convolutional layer [11], or a LSTM layer [20].

Despite their high performance, DL algorithms that have been applied to wearable HAR have a drawback; they demand large volumes of annotated data for their training. To this end, transfer learning techniques have been studied. Transfer learning refers to the technique of passing the learned parameters from a classification model to a model applied to a different but related classification task. [21] tried to transfer the extracted knowledge by convolutional layers across users, HAR domains, sensor modalities and locations. Due to the fact that the results were not very encouraging, [32] followed another approach based on a cross-domain learning framework capable of exploiting the intra-affinity of classes to perform intra-class knowledge transfer, called Stratified Transfer Learning (STL). In [1] a DL generative cross-domain adaptation technique is proposed, capable of training new HAR models for heterogeneous wearable sensors by using a small amount of new unlabeled data and exploiting the knowledge from an old model. This domain adaptation method aligns the distribution of the features between two heterogeneous sensors through the combination of a generative autoencoder with a typical HAR CNN.

A branch of transfer learning is that of few-shot, which aims at learning a classifier to recognize unseen classes (target domain) with only a small amount of labeled training samples by reusing knowledge from existing models on relevant classes (source domain) [6]. Even though, few-shot learning techniques have been widely used in computer vision tasks (e.g., face recognition [26,29]), where the classifier tries to measure the similarity between two objects [31], it has limited use in wearable HAR. Feng and Duarte [6] utilized a deep LSTM network for this purpose, examining three possible strategies based on cosine similarity, sparse reconstruction and semantic distance between the word embeddings of the activities. Moreover, they alleviated negative transfer, by measuring the cross-domain class-wise relevance so that knowledge of higher relevance is assigned larger weights during knowledge transfer. Similar techniques to those of few-shot learning have been efficiently applied to measure the similarity between pairs of motion data without knowing the explicit labels [16,27]. Specifically, in [16] the authors examined the use of Matching Networks and Triplet Networks based on a ConvNet, while in [27] Siamese Networks exploiting a convolutional LSTM architecture were chosen.

Finally, zero-shot learning has, also, been examined for HAR to transfer information from seen to unseen classes via semantic space. The first DL-based zero-shot application was introduced in [33] using a nonlinear compatibility-based method, while [35] used a multi-nonlinear layers model to project features to semantic space and combined mean square and cross entropy loss to achieve better results. In addition to this, [34] evaluated the use of Matching Networks for this task and [15] exploited Word2Vec word embeddings [18] to represent the semantic space of unseen activities. Word2Vec embeddings were also exploited in [19], which are combined with a one-hot encoding matrix and a matrix containing semantic attributes provided by human experts (e.g., speed, capacity, power), a concept introduced for zero-shot learning HAR in [3]. The produced attribute matrix is fused with the features extracted by a DL model to recognize unseen locomotion modes. Finally, the fusion of semantic vectors is, also, studied in [17], where the authors introduce an expanded word embedding vector (i.e., takes into consideration word embeddings that are synonyms with the activity label), to produce a more generalizable approach and achieve higher zero-shot accuracy.

Our literature survey has shown that Siamese and Matching Networks have been successfully used in one-shot learning computer vision tasks, but have not been broadly examined in HAR, even though they outperform the vanilla transfer learning approach [9]. Moreover, new algorithmic concepts such as self-attention mechanism [30], which have been proven to be efficient to several machine learning tasks, such as natural language processing [5], image similarity [28] and reinforcement learning [36] in the form of relational reasoning, have not yet been applied to sensor-based HAR. Thus, in this paper we evaluate the use of Siamese and Matching Networks and propose the use of multi-head self-attention to discover generalizable patterns in motion signals.

3. One-shot learning methods

3.1. Problem definition

We consider the task of one-shot learning for classification and use three datasets: a training set, a validation set, and a testing set. The training set and validation set share the same label space (i.e., same activities), but the testing set has its own label space that is disjoint with training/validation set (i.e., contains more activities). If the testing set contains N labelled examples for each of C unique classes, the target few-shot problem is called C-way N-shot. Since there are unobserved activities in the testing set, the model's performance on classifying them differs a lot (i.e., is lower) when compared to its performance on the training and validation sets. Thus, our objective is to extract transferable knowledge from the observed activities.

In the current section we introduce the four investigated DL architectures for one-shot learning, shown in Fig. 1. All of the architectures are given as input an anchor signal and a (set) candidate similar signal(s), having as an objective to identify if these signals belong to the same activity class. The input signals are tensors with dimensions $n_c x n_w x n_k$. The number of values per window are denoted by n_w (e.g., a 2 second window of a sensor with sampling rate 100 contains 200 values), while n_c depicts the total size of the sensor channels (e.g., X, Y and Z axes of an accelerometer) that are stacked vertically and n_k the number of kernels (the input n_k is equal to 1). These tensors are given as input to a DL model in order to be processed and produce feature maps. In par-

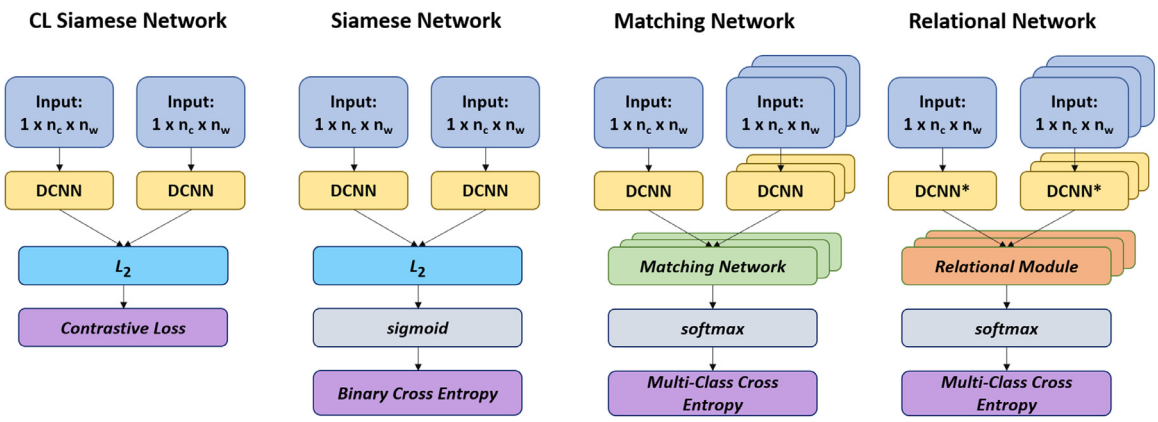

Fig. 1. The four deep learning network architectures used for one-shot learning HAR. An anchor signal and a (set) candidate similar signal(s) with dimensions n_c, n_w, n_k denoting the number of sensor channels, number of values per window and number of kernels, respectively, are given as input to a DCNN. Afterwards, the extracted features go through a similarity function and a loss function to measure the networks' performance.

ticular, all the deep CNN (DCNN) architectures we developed are based on our previous work, which has been successfully applied to HAR, PerceptionNet [11], which has been proven to outperform convolutional LSTM approaches in the one-shot learning setting [9]. It should be noted that all the architectures rely on late multimodal sensor fusion, with the exception of the relational module (see Section 5), since this approach has been proven to be more efficient [11,22], and all the convolutional layers are followed by a ReLU activation function, while all the pooling layers by a dropout. After the input signals are processed the feature maps go through a similarity function (e.g., Euclidean distance) that outputs how similar these signals are.

3.2. Siamese networks

We have implemented a Siamese network [4], which is an end-to-end learning approach; meaning that the HAR ConvNet network is replicated twice (i.e., one for each input activity sample). This way the two network branches share the same filters and their embeddings are compared using the Euclidean distance (i.e., L_2 norm) to directly predict whether the two input samples belong to the same activity. This is accomplished by estimating the absolute difference between the embeddings, that feeds a FC layer followed by sigmoid activation function to map the output into a single logistic unit (i.e., not same equals to 0 and same equals to 1).

The L_2 norm distance between the embeddings M, N is given by:

$$d(M,N) = \|(M_i - N_i)\|_2^2 \tag{1}$$

We used two different loss functions to validate its performance: the contrastive loss function's error [8] and the standard cross entropy loss. The contrastive loss function is given by:

$$L = (1-y)\frac{1}{2}D^2 + y\frac{1}{2}D^2(\max(0, m - D^2)) \tag{2}$$

where D^2 represents the distance $d(M,N)$ and $m > 0$ is a margin that defines a radius (i.e., threshold) where dissimilar pairs contribute to the loss function. During our experiments m was set to 1.2. As it is mentioned above, we developed another Siamese network using a sigmoid activation function and a binary cross entropy at the end, just like the one proposed in [13].

3.3. Matching network

In our Matching Network implementation we used the same ConvNet architecture with that of Siamese but instead of having as input two motion signals, we have an anchor signal as input and a set of motion signals to be compared with, where the set's size is equal to C. We denote as M the embeddings of the anchor signal S^a and N_i are the embeddings of the comparing signal S^c, where $i = 1, ..., C$. The similarity score produced by the Matching Network module for each M, N_i pair is defined as:

$$similarity = \frac{M \cdot N_i}{\|N_i\|} \tag{3}$$

Afterwards, the produced similarities are concatenated and fed to a softmax function to be normalized and compute the multi-class cross entropy loss.

3.4. Relational network

The Relational Networks is consisted of a DCNN and a Relational Module (RM), while its loss is computed using multi-class cross entropy loss, thus, similarly to Matching Network it has as input a query signal S^q and a set of S^c motion signals to be compared with. Firstly, the DCNN processes each S_i^c signal to extract the corresponding feature tensors (maps) N_i which are afterwards queried, in a self-attention manner [30], by the corresponding feature tensor M of the S_q signal to discover whether any similar motion pattern exists.

In particular, as displayed in Fig. 2, for each pair of feature maps (M, N_i) we produce three vectors, Q, K, and V (1). The first one is computed by flattening the query signal and applying a FC layer of size d, while the other two are the outcome of applying a FC layer to N_i. Afterwards, we apply matrix multiplication between Q and K (separately for each sensor modality), to discover patterns between the query vector and the key vector. Their dot product is normalized using a scaled softmax and is multiplied with V (value vector) to produce A, given by the following equation:

$$A(Q,K,V) = softmax\left(\frac{Q \cdot K^T}{\sqrt{d}}\right) \cdot V \tag{4}$$

where d is the dimensionality of the key vectors used as a scaling factor. Similarly to [30], we use a multi-head dot product attention, meaning that this process is executed in parallel h times, where h denotes the number of heads. Thus, the whole RM (Fig. 2) for each (M, N_i) feature maps pair is described from the following equations:

$$Q = M \cdot W_h^Q, K_i = N_i \cdot W_h^K, V_i = N_i \cdot W_h^V \tag{5}$$

$$head_{ih} = A_{ih}(Q, K_i, V_i) \tag{6}$$

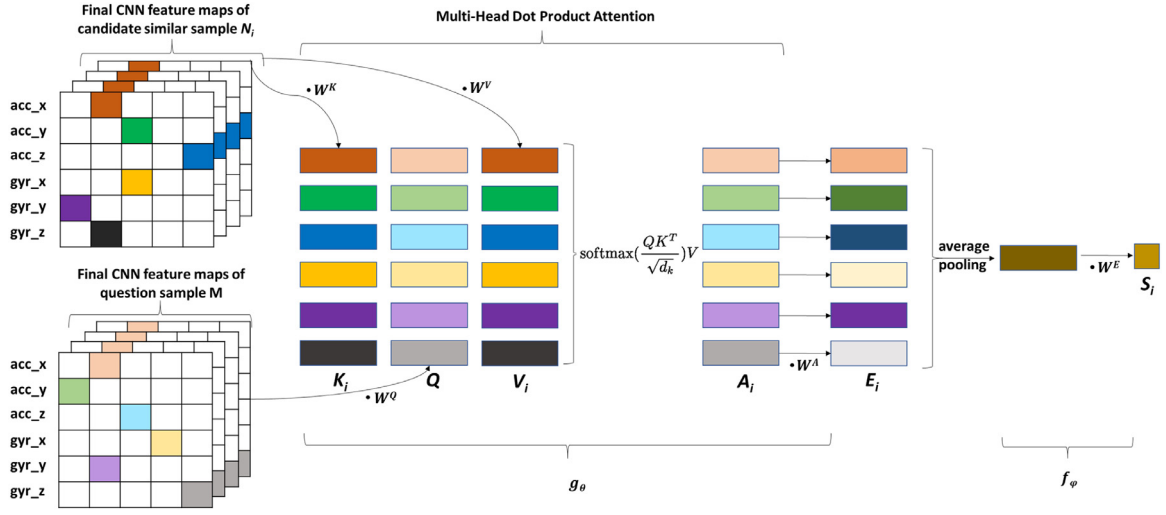

Fig. 2. The architecture of the modality-wise RM module. The extracted high-level modality-dependent features (i.e., actions) N_i of the comparing signal (displayed with dark colors) are queried by those (M) of the query signal (displayed with light colors) to discover similarity patterns between them, using the multi-head attention function (g_θ). The results of each sensor modality are averaged and go through a dense layer (f_ϕ) to produce a similarity value S_i.

$$E_i(Q, K, V) = concat(head_{i1}, \ldots, head_{ih})W^A \tag{7}$$

where $W_h^Q \in R^{d_{model} x d_q}$, $W_h^K \in R^{d_{model} x d_k}$, $WV_h \in R^{d_{model} x d_v}$, $W_h^A \in R^{d_{model} x d_a}$ and $d_q = d_k = d_v = d_a = d_{model}/h$. After computing A_ih, we concatenate the h dot products and transform them into E_i using again a dense layer. Finally, we reduce the dimensionality of E by applying an average pooling function over the sensor modalities and a matrix multiplication with a weight of shape $(d, 1)$, to produce only one similarity value S_i per each M, N_i pair; these S_i similarities are concatenated and given to a softmax activation function for computing the multi-class cross entropy loss (Fig. 1).

$$S_i = mean_{modality}(E_i) \cdot W^E \tag{8}$$

The central contribution of this work is that RMs operate on actions (high-level features, such as a hip movement) captured by a sensor modality, and hence do not explicitly operate on raw data displaying a change on the angular velocity or on the acceleration. This can be represented in an abstract manner by the following equation:

$$RM(A) = f_\phi(\sum_m \sum_{i,j}(g_\theta(a_i^m, a_j^m))) \tag{9}$$

where the input is a set of actions $A = \{a_1^1, a_2^1 \ldots, a_1^2, a_2^2 \ldots, a_n^m\}$, a_i^m is the i-th action conditioned by the m-th sensor modality, and f and g are differentiable functions (FC layers) with parameters θ and ϕ. In particular, the g_θ function is responsible for discovering the relationships between the actions and produces the term E in (7), while f_ϕ produces the similarity term S_i.

As a result, we could consider that in Fig. 2 the displayed features maps M and N_i capture 5 actions in 6 different sensor modalities. For example, if signals S^q and S_i^c belong to a walking activity we could consider that the light green action in M captured by the Y axis of the accelerometer is an arm movement from back to forth that took place during the first milliseconds of the current example, while the corresponding arm movement in the N_i feature map (displayed in dark green) took place during the half duration of this example. Consequently, after multiplying the K_i and Q tensors the RM decides to attend more in this action included, also, in tensor V_i, since this is a similar action pattern included in the M, N_i feature maps.

4. Experimental set-up

For our experiments we used a computer workstation equipped with a NVIDIA GTX 1080 Ti GPU featuring 11 gigabytes RAM, 3584 CUDA cores and a bandwidth of 484 GB/s. Python was used as programming language, and specifically the Numpy library for matrix multiplications, data preprocessing and segmentation, and the Keras high-level neural networks library using as backend the TensorFlow library. The CUDA Toolkit in support with the cuDNN, which is the NVIDIA GPU-accelerated library for deep neural networks, were used to accelerate the tensor multiplications.

4.1. Datasets

We evaluated the four networks architectures on two publicly available HAR datasets, UCL [2] and PAMAP2 [23]. Moreover, we compare the results obtained on the PAMAP2 to those presented in [6].

- **UCL** dataset consists of tri-axial accelerometer and of tri-axial gyroscope sensor data. A group of 30 volunteers, executed six daily activities (standing, sitting, laying down, walking, walking downstairs and upstairs), wearing a waist-mounted smartphone. The sensors' sampling rate is equal to 50 Hz, while the samples are provided preprocessed (i.e., no missing values) and segmented into time windows of 128 values (2.56 sec), having a 50% overlap. What is more, the dataset is separated into train and test data. In particular, the UCL dataset contains 10,299 samples, which are partitioned into two sets, where 70% of the volunteers (21 volunteers) participate in the training set (7,352 samples) and 30% (9 volunteers) in the test set (2,947 samples). Moreover, for hyper-parameter tuning, during the validation phase we followed a Leave-3-Subject-Out approach, where the volunteers (27, 29 and 30) were used as validation set. To evaluate the developed one-shot learning techniques, we selected two activities (walking upstairs and lying) to be left out of sight during the training phase. The 1-shot learning task for UCL was 6-way.
- **PAMAP2** dataset contains 12 lifestyle activities (such as walking, cycling, ironing, etc.) executed by 9 participants; they wore a heart rate monitor and 3 inertial measurement units (IMUs) with a sampling frequency of 100Hz, placed on the dominant

arm, on the chest and on the dominant side's ankle, producing tri-axial accelerometer, gyroscope and magnetometer data. We downsampled the PAMAP2 dataset to 50Hz and selected only the accelerometer and gyroscope data, to obtain the same sampling rate with the UCL dataset and the same sensor signals. Moreover, like UCL, we used a time window equal to 2.56 seconds and 50% overlap. It should be noted that we discarded segments containing ovelapping activity labels and more than one consecutive missing values, otherwise the missing values were filled using linear interpolation. We split PAMAP2 using a Leave-1-Subject-Out approach, for the test and the validation set. Specifically, the samples (1,926) of subject 1 were used for the test set and the samples (2,102) of subject 5 for the validation set, leaving the rest samples (10,940) for the training set. The activities and their indices that were excluded during training were: sitting, cycling, nordic walking, descending stairs and ironing, which are the same with [6]. The 1-shot learning task for PAMAP2 was 12-way.

4.2. Metrics

Once we had optimized the network to master the verification task for the validation dataset, we were ready to demonstrate the discriminative potential of our learned features at one-shot learning. In order to validate the performance of the developed models, we used a C-way one-shot accuracy metric, where C equals 6 for the case of UCL and 12 for the case of PAMAP2. Specifically, the same activity example was compared to C different activity samples out of which only one of them matched the original activity example. It should be noted that activity sampling for the case of the same activity samples was done randomly from a pool of activities executed by the same subject and, for the case of different samples was done randomly (see Section 5). Moreover, it is worth mentioning that all the produced sets for both datasets are balanced (i.e., contain equal number of instances per class class), since we set the number of examples per class equal to that of the activity class that had the less examples. For example, for the PAMAP2, ironing had less examples (equal 97), consulting in having a test set of 485 samples (i.e., we sampled 97 examples for 5 activities).

Given an embedding (feature map) M produced by an anchor (query signal and the embeddings N_c representing the embeddings produced by the signal comparing examples of each C categories, we can now query the network using M, N_c as our input for a range of $c = 1, ..., C$. Then, we proceed with predicting the class y^* corresponding to the maximum similarity for each C-way one-shot learning sample.

$$y^* = argmax_c \mathbf{p}^{(c)} \tag{10}$$

where $\mathrm{p}^{(c)}$ denotes the probability of M be of the same class with embedding N_c. It is noteworthy that for the case of the contrastive loss we use *argmin* instead of *argmax*, since we wish to compute the minimum Euclidean distance. Thus, the each network's one-shot C-way accuracy is given by:

$$acc_{one-shot} = \frac{\sum_{i=1}^{N} (y_i^* == y_i)}{N} \tag{11}$$

where y_i is the true label of the $i-th$ C-way one-shot learning sample and N is the total number of samples.

5. Results and discussion

For all the datasets, we selected the same preprocessing strategy; subtracting the mean and dividing by the standard deviation the motion signals:

$$z_i = \frac{(x_i - \mu_i)}{\sigma_i} \tag{12}$$

where x_i denotes the samples of sensor modality i, while μ_i, σ_i depict their corresponding mean and standard deviation values respectively. The Adam algorithm [12] was selected as network optimizer, having the following hyper-parameters: learning rate equal to 0.001, beta1 equal to 0.9, beta2 equal to 0.999 and epsilon equal to 1e-08. Moreover, we set the batch size equal to 128 and the minimum number of epochs to 1,000, but the training process was automatically terminated if the best validation one-shot C-way accuracy had not improved after 100 epochs. The validation model that had the lowest error rate was saved, and its weights were used to obtain the model's one-shot C-way accuracy on the test set that includes the target (new) activities. It should be noted that we followed a simple grid search approach for tuning the hyperparameters of the DCNN keeping those of the RM stable (d:64, h:1). Afterwards, we tuned the RM's hyperparameters, following the same approach. Appendix B presents the examined set of the hyperparameter values.

Table 1 illustrates the selected size of parameters per layer for the examined network architectures; for layer i we used the same filters' sizes (e.g., number of sensor channels f_c^i and number of values per time window f_w^i) and number of kernels for the two datasets f_k^i : {24, 48, 64}, with the exception of Relational Network (RL) that uses only 1D convolutions. It should be noted that all the convolutional layers are followed by a ReLU activation function. As may be seen, the RL contains fewer parameters from all the other networks, for both datasets. For PAMAP2 we achieved better results using 2 heads and for UCL using 3 heads (Fig. 4). Moreover, it is noticeable that the developed one-shot learning model architectures are lightweight, since they consist of approximately 119,000 parameters, while existing HAR networks [20,21] contain over 1,000,000.

Table 2 presents the performance results that we obtained (average values of 10 runs). Like [6], we measured the algorithms' one-shot C-way accuracy in the case were the subject was included in the training set or was absent from it, in order to check how dependent are the algorithms' predictions to a subject's moving patterns. Of course, in both cases, the target activities were not included during the training process. The Relational Network achieved the best results for both datasets (UCL: 94.80% and PAMAP2: 84.41%), with Matching Network coming second (around 1.5% lower for both datasets). Unexpectedly, for some cases, there is a decrease in the algorithms' classification performance when the moves of the validation subject are included in the training set. This reveals that there is no overfitting when it comes to validating the algorithm's performance on new activities (i.e., the algorithm's performance on one-shot learning can be considered subject independent).

A more intuitive performance representation is shown in Fig. 3, where the one-shot accuracy per activity class of the all the evaluated networks is displayed. It is worth noticing, that while the networks' performances per class is analogous to their overall performance, when it comes to the nordic walking class in the PAMAP2 dataset the Relational Network has the worst one-shot accuracy performance. However, having a look at the false negatives (Appendix C) the algorithm missclassified many examples of nordic walking as walking, which is expected since both of these activities are consisted of very similar actions and the RM attends to sensor modalities that reveal back to forth hand and ankle movement and forward chest movement. This results into acquiring similar attention maps for these activities, where the actions that display their dissimilarities are weighted by very small numbers (close to zero), a feature that does not exist in the other algorithms that take equally into consideration similarities and dissimilarities. In general terms, in both datasets all the networks, including Relational Network (Appendix C), missclassified a lot of walking-alike activities (walking, walking downstairs, walking upstairs) mostly to

Table 1
Used hyperparameters for each network.

	Parameters	Siamese CL	Siamese	Matching Net	Relational Net
1st Convolutional block	$f_k^1 \times f_c^1 \times f_w^1$	32x1x11	32x1x11	32x1x11	32x1x11
	max pool	1x3	1x3	1x3	1x3
	dropout	0.5	0.5	0.5	0.5
2nd Convolutional block	$f_k^2 \times f_c^2 \times f_w^2$	48x1x11	48x1x11	48x1x11	48x1x11
	max pool	1x3	1x3	1x3	1x3
	dropout	0.5	0.5	0.5	0.5
3rd Convolutional block	$f_k^3 \times f_c^3 \times f_w^3$	64x3x11	64x3x11	64x3x11	64x1x11
	max pool	1x2	1x2	1x2	1x2
	dropout	0.5	0.5	0.5	n/a
Relational module	h, d	n/a	n/a	n/a	2–3, 64
	W_Q, W_K, W_V, W_A	n/a	n/a	n/a	hx64x64
	dropout	n/a	n/a	n/a	0.5
	W_E	n/a	n/a	n/a	64x1
Classifier	FC	n/a	2x1	n/a	n/a
Total parameters		118,768	118,770	118,768	84,016- 100,400

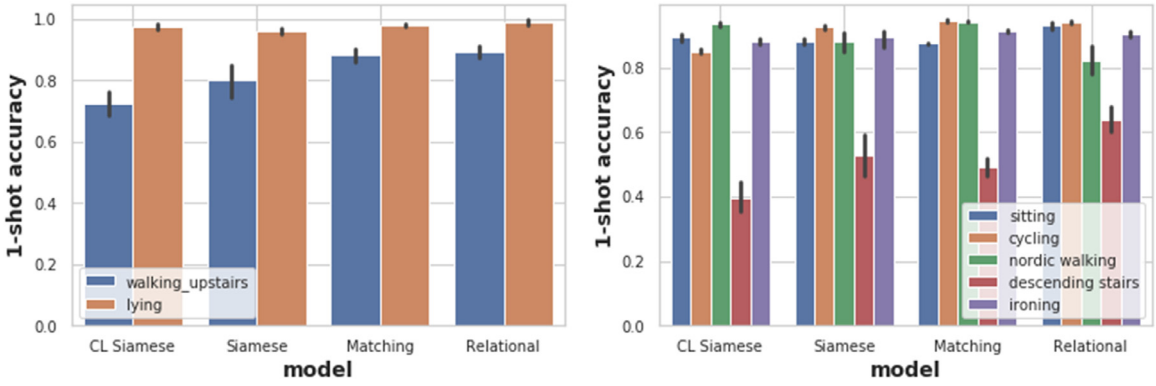

Fig. 3. The one-shot C-way accuracy of all the examined networks per each class for the UCL test set (left) and the PAMAP2 test set (right).

Table 2
One-shot accuracy performance of the one-shot learning techniques on the UCL and PAMAP2 datasets.

Dataset	Network	Test 1-shot acc	Train 1-shot acc
UCL	CL Siamese	85.04%	87.24%
	Siamese	88.06%	87.06%
	Matching	93.04%	92.56%
	Relational	94.80%	92.65%
PAMAP2	FSHAR-NGD [6]	63.00%	58.98%
	FSHAR-Cos [6]	62.82%	56.83%
	FSHAR-SR [6]	63.70%	56.62%
	CL Siamese	79.09%	80.13%
	Siamese	81.49%	86.25%
	Matching	83.26%	85.81%
	Relational	84.41%	88.93%

activities that the algorithm has been trained on, revealing an overfitting towards them. This is noticeable by the fact that none of the nordic walking examples was missclassified as ascending stairs and vice versa.

Furthermore, it is noticeable that the performance on PAMAP2 of the algorithms introduced in [6] is much lower than ours. This is due to the fact that the authors used LSTM as feature extraction layers and not ConvNets, which have been widely used even in works that exploit LSTM architectures [20,27]. Another possible reason may rely on different sampling techniques. In our work, we sampled same activity signals produced by the same subject, while we sampled on different activity signals from all the subjects, due to the fact that the algorithm converged much faster. This sampling strategy can be considered as selecting "easy" samples for the case of same since activity patterns are subject-dependent.

What is more, Fig. 4 presents the sensitivity of the proposed approach regarding hyperparameter h for the UCL case. Both the training and testing sets produced better results when using 3 heads. It should be noted that for the PAMAP2 we used 2 heads. As a result, this indicates that the performance of the RM depends a lot on the hyperparameter h. Moreover, it has a lot of hyperparameters to be tuned (Table 1), while the other networks (e.g., Matching Network) use no parameters and can be tuned more easily. This is of course a disadvantage of the proposed approach, since it has to configure a similarity metric of its own and not use existing ones (i.e., cosine similarity and Euclidean distance). However, as shown in the current paper, when tuned correctly it appears to be more generalizable to recognize new activities.

Furthermore, we evaluated three different modality fusion strategies for the relational network. In particular, apart from the modality-wise relational network architecture (1) we trained a model whose modalities were fused by the DCNN before given to the RM module (prefused model) by using a 64x3x11 filter in the 3rd convolutional block and a non modality-wise model where the values of each modality (e.g., X-axis gyroscope) could attend to activity patterns (actions) captured by different sensor modality (e.g., Z-axis accelerometer). Fig. 5 displays the impact of modality fusion

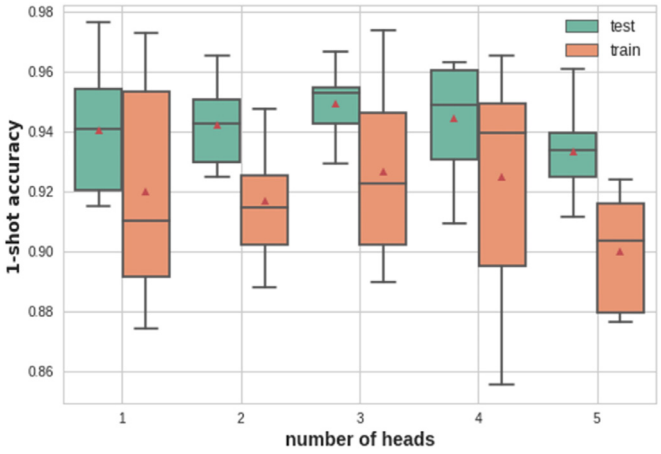

Fig. 4. The impact of the number of heads h in the performance (one-shot six-way accuracy) of the modality-wise RM on UCL test set.

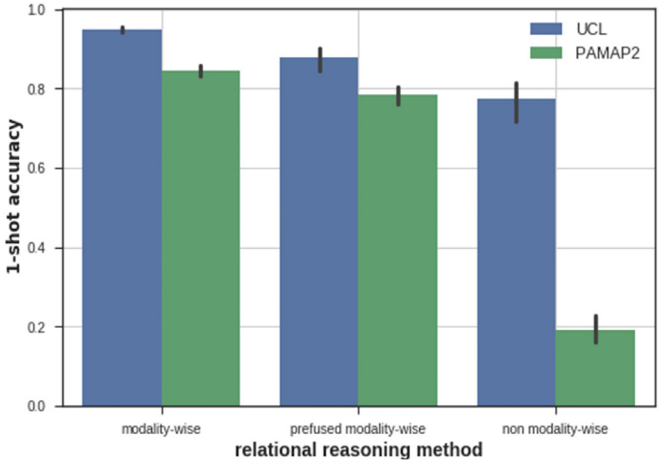

Fig. 5. The impact of the different relational reasoning fusion approaches on the performance of the RM in the UCL and PAMAP2 test sets.

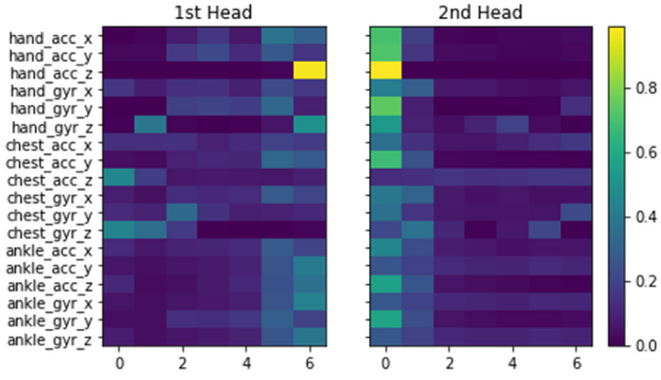

Fig. 6. Visualization of the 2-head attention maps of the ironing activity in PAMAP2 test set. The RM attends more on the last (1^{st} head) and first (2^{nd} head) the Z axis accelerometer action values of the IMU placed on the hand.

on the relational reasoning-based model. As shown, the modality-wise approach without using any prefusion in the ConvNet (i.e., using 2D convolutions at the 3rd layer) exceeds for both datasets the performance of the prefusion approach and the non-modality wise relational reasoning by more than 5% and 15% respectively. It should be noted that for the case of non-modality reasoning in the PAMAP2, the model struggled to discover any generalizable patterns amongst new activities. Finally, as aforementioned, the proposed similarity model exploits the multi-head self-attention mechanism, thus, its attention maps can be visualized for interpretability. Fig. 6 displays that for the ironing activity most attention was paid to the sensor signals produced by the hand-placed IMU and particularly its z-axis. Moreover, the per unseen activity average values of all the multi-head attention maps of the model's predictions are provided in Appendix A.

6. Conclusion

In this paper, we proved that one-shot learning techniques can be applied to wearable HAR. The acquired knowledge from processed motion sensors has proven to be transferable across the same wearable sensors in order to detect new activities, which we have only one labeled sample of them in the testing set. Moreover, the evaluated techniques did not have strong dependency on the subject that performs the activity. The more efficient and robust module for recognizing new activities was the one based on modality-wise reasoning without applying multi-modal sensor fusion in the previous steps.

We advocate that one-shot learning techniques will be widely applied in wearable HAR, since this way users will be given the ability to add their own moves to smartwatches and smartphones, a feature that could be exploited in various domains, such as smart environments to control IoT (Internet of Things) devices. Future steps will be done towards exploiting modality-wise reasoning on activity embeddings for domain-adaptation (over different sensor modalities and on-body placements) and exploring the use of existing active learning techniques for effective activity sampling. Finally, we will build a dataset to be used as benchmark for few-shot learning wearable HAR.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme <<Human Resources Development, Education and Lifelong Learning 2014-2020>> in the context of the project "On applying Deep Learning techniques to insufficient labeled sensor data for gesture recognition" (MIS 5050324).

Appendix A

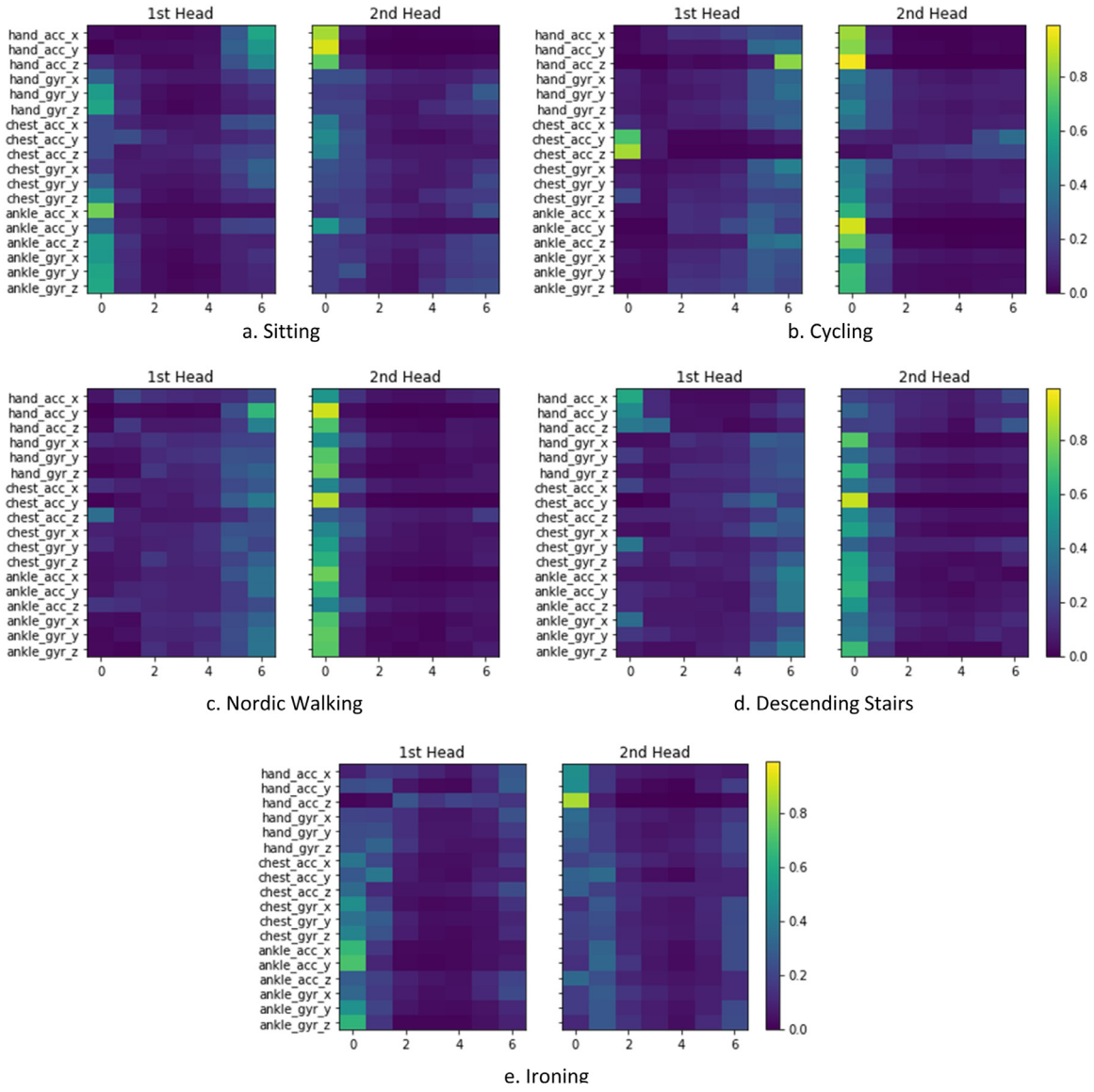

Fig. A1. Averaged attention maps for the PAMAP target activities.

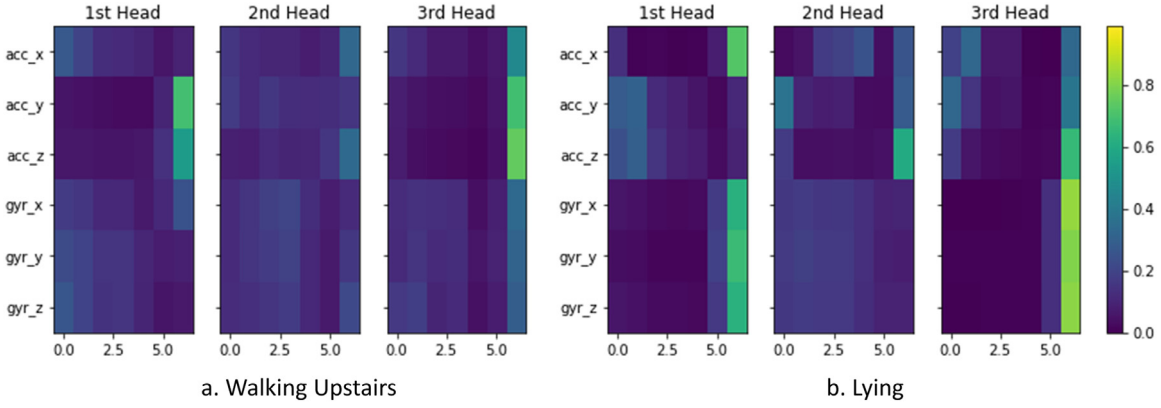

Fig. A2. Averaged attention maps for the UCL target activities.

Appendix B

Table B1
Examined hyperparameters.

Name	Symbol	Range
batch size	-	128
learning rate	α	1e-03
beta1	β_1	0.9
beta2	β_2	0.999
epsilon	ϵ	1e-08
filter height	f_h	1–3
filter width	f_c	7–15
filter channels	f_c	16,24,32,48,64,80
dropout probability	-	0.1-0.5
number of heads	h	1–5
K, Q, V vectors size	d	32, 64, 128
maximum epoch	-	1000
early stopping criterion	-	100

Appendix C

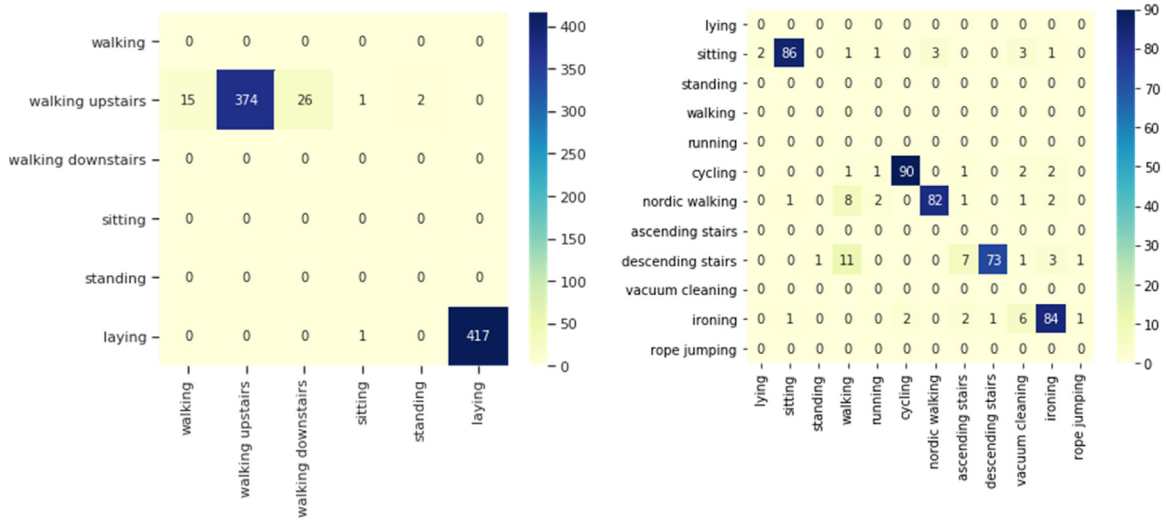

Fig. C1. Confusion matrices of the Relational Network for the UCL (left) and the PAMAP2 (right) test sets.

References

[1] A. Akbari, R. Jafari, Transferring activity recognition models for new wearable sensors with deep generative domain adaptation, 2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) (2019) 85–96.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, ESANN, 2013.

[3] H.-T. Cheng, M. Griss, P. Davis, J. Li, D. You, Towards zero-shot learning for human activity recognition using semantic attribute sequence model, Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (2013).

[4] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) 1 (2005) 539–546 vol. 1.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, ArXiv abs/1810.04805 (2019).

[6] S. Feng, M.F. Duarte, Few-shot learning-based human activity recognition, ArXiv abs/1903.10416 (2019).

[7] I.G. Goodfellow, Y. Bengio, A.C. Courville, Deep learning, Nature 521 (2015) 436–444.

[8] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) 2 (2006) 1735–1742.

[9] P. Kasnesis, C. Chatzigeorgiou, C.Z. Patrikakis, M. Rangoussi, Introducing and benchmarking a one-shot learning gesture recognition dataset, 10th EAI International Conference on Big Data Technologies and Applications (2020).

[10] P. Kasnesis, C. Chatzigeorgiou, L. Toumanidis, C.Z. Patrikakis, Gesture-based incident reporting through smart watches, 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (2019) 249–254.

[11] P. Kasnesis, C.Z. Patrikakis, I.S. Venieris, Perceptionnet: a deep convolutional neural network for late sensor fusion, ArXiv abs/1811.00170 (2018).

[12] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, CoRR abs/1412.6980 (2015).

[13] G.R. Koch, Siamese neural networks for one-shot image recognition, 2015.

[14] G. Laput, R. Xiao, C. Harrison, Viband: high-fidelity bio-acoustic sensing using commodity smartwatch accelerometers, Proceedings of the 29th Annual Symposium on User Interface Software and Technology (2016).

[15] F.A. Machot, M.R. Elkobaisi, K. Kyamakya, Zero-shot human activity recognition using non-visual sensors, Sensors (Basel) 20 (2020).

[16] K. Martin, A. Wijekoon, N. Wiratunga, Human activity recognition with deep metric learners, ICCBR Workshops, 2019.

[17] M. Matsuki, P. Lago, S. Inoue, Characterizing word embeddings for zero-shot sensor-based human activity recognition, Sensors (Basel) 19 (2019).

[18] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, ArXiv abs/1310.4546 (2013).

[19] R. Mishra, A. Gupta, H.P. Gupta, T. Dutta, A sensors based deep learning model for unseen locomotion mode identification using multiple semantic matrices, IEEE Trans. Mob. Comput. (2020). 1–1.
[20] F.J.O. Morales, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors (Basel) 16 (2016).
[21] F.J.O. Morales, D. Roggen, Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations, ISWC '16, 2016.
[22] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, R. Dürichen, Cnn-based sensor fusion techniques for multimodal human activity recognition, Proceedings of the 2017 ACM International Symposium on Wearable Computers (2017).
[23] A. Reiss, D. Stricker, Introducing a new benchmarked dataset for activity monitoring, 2012 16th International Symposium on Wearable Computers (2012) 108–109.
[24] C.A. Ronao, S.-B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert Syst. Appl. 59 (2016) 235–244.
[25] A. Santoro, D. Raposo, D.G.T. Barrett, M. Malinowski, R. Pascanu, P.W. Battaglia, T.P. Lillicrap, A simple neural network module for relational reasoning, NIPS, 2017.
[26] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 815–823.
[27] T. Sheng, M. Huber, Siamese networks for weakly supervised human activity recognition, 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) (2019) 4069–4075.
[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H.S. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018) 1199–1208.
[29] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 1701–1708.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, ArXiv abs/1706.03762 (2017).
[31] O. Vinyals, C. Blundell, T.P. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, NIPS, 2016.
[32] J. Wang, Y. Chen, L. Hu, X. Peng, P.S. Yu, Stratified transfer learning for cross–domain activity recognition, 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom) (2018) 1–10.
[33] W. Wang, C. Miao, S. Hao, Zero-shot human activity recognition via nonlinear compatibility based method, Proceedings of the International Conference on Web Intelligence (2017).
[34] A. Wijekoon, N. Wiratunga, S. Sani, Zero-shot learning with matching networks for open-ended human activity recognition, SICSA ReaLX, 2018.
[35] T. Wu, Y. Chen, Y. Gu, J. Wang, S. Zhang, Z. Zhechen, Multi-layer cross loss model for zero-shot human activity recognition, Advances in Knowledge Discovery and Data Mining 12084 (2020) 210–221.
[36] V.F. Zambaldi, D.C. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D.P. Reichert, T.P. Lillicrap, E. Lockhart, M. Shanahan, V. Langston, R. Pascanu, M.M. Botvinick, O. Vinyals, P.W. Battaglia, Deep reinforcement learning with relational inductive biases, ICLR, 2019.
[37] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, 6th International Conference on Mobile Computing, Applications and Services (2014) 197–205.
[38] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Exploiting multi-channels deep convolutional neural networks for multivariate time series classification, Frontiers of Computer Science 10 (2015) 96–112.