

Consistency Based Weakly Self-Supervised Learning for Human Activity Recognition with Wearables

Taoran Sheng and Manfred Huber

Department of Computer Science and Engineering
University of Texas at Arlington
taoran.sheng@mavs.uta.edu, huber@cse.uta.edu

Abstract

While the widely available embedded sensors in smartphones and other wearable devices make it easier to obtain data of human activities, recognizing different types of human activities from sensor based data, remains a difficult research topic in ubiquitous computing. One reason for this is most of the collected data is unlabeled. However, many current human activity recognition (HAR) systems are based on supervised methods, which heavily rely on the labels of the data. We describe a weakly self-supervised approach in this paper that consists of two stages: (1) In stage one, the model learns from the nature of human activities by projecting the data into an embedding space where similar activities are grouped together; (2) In stage two, the model is fine-tuned using similarity information in a few-shot learning fashion using the similarity information of the data. This allows downstream classification or clustering tasks to benefit from the embeddings. Experiments on three benchmark datasets demonstrate the framework's effectiveness and show that our approach can help the clustering algorithm achieve comparable performance in identifying and categorizing the underlying human activities as pure supervised techniques applied directly to a corresponding fully labeled data set.

Introduction

Many machine learning algorithms have been studied to address a variety of problems associated with wearable sensor-based human activity analysis. While the majority of them used supervised algorithms, largely relying on labeled data, a limited number of works have explored unsupervised or semi-supervised ways to address the recognition problem. However, since labeled data is in short supply and the bulk of acquired data is unlabeled, utilizing supervised techniques to recognize distinct human behaviors remains a difficult research challenge in ubiquitous computing. On the other hand, unsupervised approaches do not need labeled data to train the model; nevertheless, their performance is often inferior than that of supervised methods.

To overcome the aforementioned challenges, we aim at developing an approach that makes use of domain knowledge of human behavior, a limited amount of information about the similarity of data samples, and a significant quantity of unlabeled data. The proposed approach is motivated by two factors: first, the use of domain knowledge and a

large amount of unlabeled data to simplify the model's learning task and force the model to retain useful task-specific features; second, limiting the supervision required to train the model by relying only on the similarity of a small amount of data samples rather than explicitly using the labels. Experiments have been conducted to assess the efficacy of our proposed approach.

Related Work

Human Activity Recognition

There are two primary directions in wearable sensor-based human activity recognition (HAR): approaches based on handcrafted features, and approaches based on deep neural networks (DNN). In the former one, handcrafted features are designed using domain knowledge. For instance, (Anguita et al. 2013; Reyes-Ortiz et al. 2016) included statistical features such as mean, variance, and entropy into their models. In (Tamura et al. 1998), features taken from a wavelet transform were used. He and Jin (He and Jin 2009) extracted features using the discrete cosine transform. These features have the benefit of being readily derived from the signal and have been shown to be pretty successful in the HAR system.

Various HAR models have recently used DNN to enable automatic feature extraction. Morales and Roggen (Morales and Roggen 2016) proposed a model based on convolutional neural networks (CNNs) and long short-term memory (LSTM) components. In the work, CNN is employed to capture local temporal relationships, while LSTM's memory states help in the learning of large time scale dependencies. In (Abu Alsheikh et al. 2016), a hybrid method was developed in which the sequence of human behaviors was modeled using a deep belief network as the emission matrix of a Hidden Markov Model. In (Chen et al. 2021), a deep network architecture using stage distillation with a teacher and a student network are used to allow the model to extract more and more useful features from the raw data. These approaches are capable of extracting features from data automatically and without the need for domain knowledge. However, they continue to need explicit labels to supervise the model's training.

In addition to these pure hand-engineered and purely learned feature representations, recent work has also combined the two techniques. For example, (Qin et al. 2020)

utilized engineered wavelet features arranged in the form of an image but then applied this as the input to a convolutional network to automatically select and abstract these engineered base features to obtain a more powerful feature representation for the problem. This again gives more flexibility to the learning system but, like the other approaches still continues to rely on explicit labels for learning.

In the work presented in this paper, the requirement for explicit labels is dramatically relaxed through the application of self-supervised and unsupervised learning components. To facilitate the architecture, a hybrid approach to input features is taken where statistical measures are derived from data segments and utilized as input to deep learning structures to select and adapt useful features for the activity recognition task.

For a more detailed discussion of the different aspects and methods in HAR, please refer to two recent survey papers in the area, (Minh Dang et al. 2020) and (Wang et al. 2019).

ResNet Autoencoder and Siamese Networks

We take ideas from Autoencoder (Kramer 1991), ResNet (He et al. 2016) and siamese architectures (Bromley et al. 1993) to construct our model that is capable of effectively capturing patterns in data and computing semantic similarity between pairs of data samples.

Autoencoder Autoencoder is a highly effective and extensively used unsupervised deep learning architecture for feature learning. It is divided into two components: an encoder and a decoder. The encoder specifies the nonlinear transformation $E(\cdot)$ that is used to convert the input data sample x to the representation $E(x)$. The decoder specifies another nonlinear transformation $D(\cdot)$ with the goal of decoding $E(x)$ and reconstructing the original input x .

$$\tilde{x} = D(E(x)) \quad (1)$$

Here, x denotes the input data, $E(x)$ denotes the encoded representation from the encoder, and \tilde{x} is the decoded reconstruction from the decoder. The learning target of a conventional autoencoder is to minimize the reconstruction loss:

$$\Phi_{ae}(x) = \sum_{x \in X} \|x - \tilde{x}\|^2 \quad (2)$$

where X denotes the dataset. This loss function is used to ensure that the reconstruction \tilde{x} is as close to the original input x as possible. If a decent reconstruction \tilde{x} can be decoded from the representation $E(x)$, it indicates that the representation $E(x)$ has retained the essential information from the input x , allowing the reconstruction \tilde{x} to be very similar to the original input x . As a result, the representation $E(x)$ may be utilized for further tasks such as classification and clustering.

Yet, often the general reconstruction objective is insufficient. The target of a classic autoencoder is to develop a representation $E(x)$ that has sufficient information to reconstruct the input, which implies that an accurate reconstruction contains noise and all the nuances in the input data. However, not all of the information included in the learnt representation is relevant to the downstream task (e.g. noise

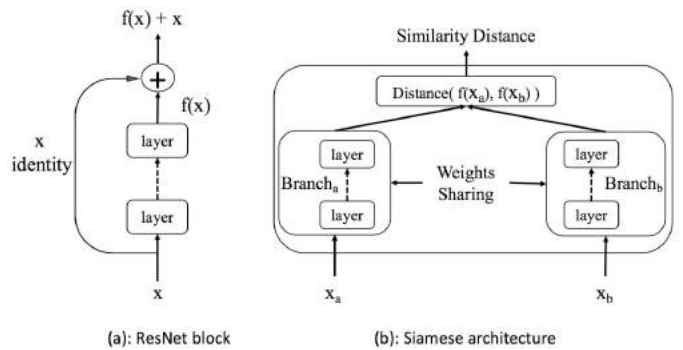


Figure 1: ResNet and Siamese Architecture.

as well as some task-irrelevant details might not only be unnecessary but could even be detrimental, especially in the context of subsequent classification and clustering tasks). To develop an effective representation for categorizing activities effectively, the representation should include just the information necessary to perform the task while mostly disregarding unnecessary details. Based on this observation, our approach incorporates additional task-oriented loss functions to guide the learning process of the autoencoder and improve the efficacy of the learnt representations in subsequent tasks.

ResNet ResNet allows the user, by utilizing skip connections or shortcuts to jump over some layers, as shown conceptually for a single ResNet block in Figure 1(a), to boost network capacity without suffering the expense of feature learning deterioration, when compared to standard deep neural networks. The output of a building block in ResNet is the sum of the result from the last non-linear layer and the input of the building block. However, in residual blocks, the shapes of the input tensor and the output tensor can be different. To address this problem, if the input tensor and output tensor have different shapes, a linear layer with the size of the output tensor will be used as the residual connection. Otherwise, an identity function will be used as the residual connection.

Siamese Networks Siamese Networks, as illustrated in Figure 1(b), are dual-branch neural networks with shared weights such that $Branch_a = Branch_b$. Given an input pair of human activity data samples $\{x_a, x_b\}$, the siamese network learns to project the input pair to the representation pair $\{E(x_a), E(x_b)\}$. The distance between the representation pair is then used as the semantic similarity of the input pair. Mueller and Thyagarajan (Mueller and Thyagarajan 2016) measured the semantic similarity between two sentences using siamese recurrent neural networks. In (Chopra, Hadsell, and LeCun 2005), a siamese CNN is trained for face verification. Other applications include unsupervised acoustic model learning (Zeghidour et al. 2016; Synnaeve and Dupoux 2016; Kamper, Wang, and Livescu 2015), image recognition (Bertinetto et al. 2016), and object tracking (Koch, Zemel, and Salakhutdinov 2015).

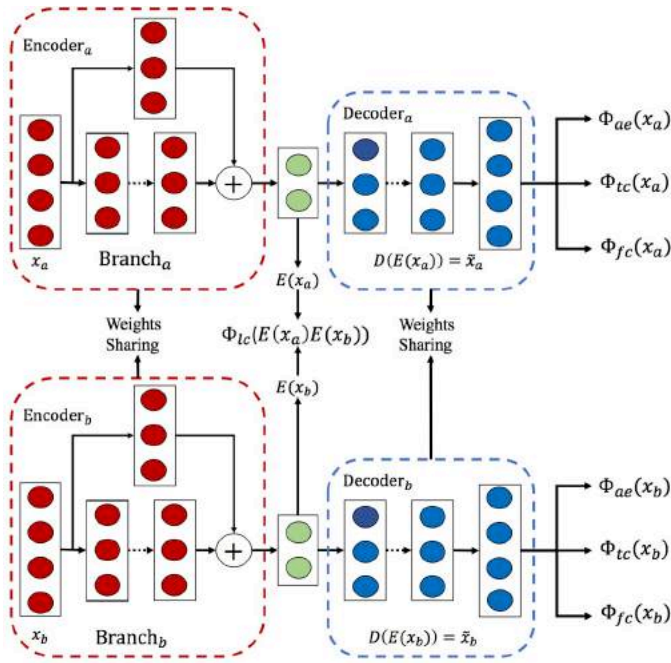


Figure 2: The overall architecture of the proposed approach.

Proposed Approach

Our proposed approach circumvents the lack of labeled data by leveraging properties pertaining to the nature of human activities. Specifically, our approach makes use of two kinds of relationships: temporal consistency of time series of human activity and feature consistency of human activity data in feature space.

In this section, we first introduce the architecture of the proposed model. Then we show the definitions of various consistency criteria that are used in the model, and how the domain knowledge is employed in the model in the form of these consistencies.

Architecture of the Proposed Model

As shown in Fig. 2, the foundation of the overall architecture of this approach is a combination of a ResNet autoencoder and siamese networks. The proposed approach utilizes self-supervised consistency criteria within the autoencoder components. Multiple learning targets, based on the nature of human activity, imposed on the autoencoder allow it to capture information that is relevant to the activity recognition task, and disregard irrelevant details in the data. In addition, the siamese network structure further enhances the recognition performance of the model with only limited amounts of weakly supervised data. Moreover, the residual connections, as mentioned above, increase the learning capacity of the model. These consistency criteria complement each other, obtaining a high accuracy model in a few-shot learning fashion.

Consistency Definition

The consistencies employed in the model are the common sense knowledge or domain knowledge, which can provide useful task-oriented information to the model, guide the model to learn task-relevant information, and ignore task-irrelevant information, hence improving the performance of the model.

Temporal Consistency describes the temporal continuity in human activity. Generally, human activities may be split intuitively into two components: a time variable component and a temporally fixed component. To be more precise, some dynamic features of a particular action may change over time. For instance, during walking, the body’s posture changes with time: the left and right feet alternately advance. This sort of dynamic feature is also captured in sensor data and is referred to as the temporally changing component in this section.

On the other hand, regardless of how the bodily stance changes over time, the activity’s semantic meaning stays constant. Specifically, the left and right feet may move forward alternately, but the action remains walking. This part is referred to as the temporally stationary component.

Due to the nature of human activity, the temporal consistency loss pushes temporally near data samples to be similar to one another and ignores the temporal variable component difference. It is inspired by the goal of maintaining a reasonably stable semantic content, i.e. the sort of action in which we are engaged, across time. Though the data samples are temporally near, they may indicate the same sort of activity, even if they are quite far in the sensor data space in terms of Euclidean distance. The temporal consistency loss retains the sensor data’s temporal continuity.

Formally, let x_i^t denote the i -th data sample that occurs at time t during the course of an activity. Let X_i^t denote the set of m data samples which are the nearest neighbors of x_i^t based on the temporal closeness during the course of an activity, and \tilde{x}_i^t the reconstruction of x_i^t . Then the temporal consistency loss, $\Phi_{tc}(x_i^t)$, is defined as:

$$\Phi_{tc}(x_i^t) = \frac{1}{m} \sum_{x_j \in X_i^t} \|x_j - \tilde{x}_i^t\|^2 \quad (3)$$

Temporal consistency attempts to structure data such that data samples that occur directly after each other are closer together in representation space, representing the intuition that activities are usually longer than a single data sample and thus that consecutive data samples are often still part of the same activity.

Feature Consistency was motivated by the fact that although various individuals conduct the same type of activity in a variety of ways, the variety of ways does not prevent other individuals from identifying the activity type. As a result, we think that the individual level characteristics of the activity data may not be required during the activity clustering stage, and that the characteristics that are consistently present across several data points may represent the activity’s fundamental characteristics. This assumption underpins the Feature Consistency target function.

Previously published research (Sheng and Huber 2020a) has demonstrated that combining carefully designed hand-crafted high level features to represent the primary characteristics of a temporally varying signal value with the k-Nearest Neighbor algorithm is an effective method for classifying sensor data from human activities. It is used in this strategy to define the local neighborhood of a data sample because of its efficacy and simplicity. The Feature Consistency loss then attempts to retain the high-level feature traits that are typically present in a neighborhood.

The Feature Consistency loss compels the decoder to decode a data sample using its neighboring data samples' learnt representation. The reasoning behind this is because if data samples are spatially adjacent to one another in the constructed feature space, they may indicate the same sort of activity. Thus, the characteristics shared by several neighboring data samples should be considered to represent the key characteristics of that type of activity. If the features are not present in all surrounding data sets, they may indicate individual level characteristics, but not activity characteristics.

Formally, let x_i^f denote the i -th data sample in feature space f , X_i^f the set of n data samples which are the nearest neighbors of x_i^f in the feature space f , and \tilde{x}_i^f the reconstruction of x_i^f . Then the feature consistency loss, $\Phi_{fc}(x_i^f)$, is defined as:

$$\Phi_{fc}(x_i^f) = \frac{1}{n} \sum_{x_k \in X_i^f} \|x_k - \tilde{x}_i^f\|^2 \quad (4)$$

Feature consistency aims to keep data samples that have very similar features close together in the embedding space. This reflects the intuition that data segments that exhibit very similar sensor readings are more likely to belong to the same activity.

Label Consistency: With the help of siamese networks and a very small amount of weakly-supervised data, a group of pairwise constraints can be defined on the input data. This group of pairwise constraints can drive the proposed model to map the data into a representation space, in which the clusters satisfy the given constraints.

More formally, given an input pair of human activity data sequences $\{x_a, x_b\}$, the encoder of the siamese networks learns to encode the input pair to the representation pair $\{E(x_a), E(x_b)\}$. Then, a distance function, which measures the similarity between the encoded representations, is calculated. For the case of Euclidean distance, the label consistency loss can then be defined as:

$$\Phi_{lc}(x_a, x_b) = \|E(x_a) - E(x_b)\|^2 \quad (5)$$

The similarity distance $Dist(x_a, x_b)$ indicates if the input pair represents the same type of activity. A large distance indicates two different types of activities, while a small distance indicates the same types of activity.

The aforementioned three types of consistency criteria imposed on the proposed model enable the learning to capture the information that is useful to the activity recognition task.

Joint Loss Function

The training process of the model is divided into two stages. In the first stage the model is trained with the following joint loss function:

$$\min \sum_{i=1}^S (1 - \alpha - \beta) \cdot \Phi_{ae}(x_i) + \alpha \cdot \Phi_{tc}(x_i^t) + \beta \cdot \Phi_{fc}(x_i^f) \quad (6)$$

where i is the index of the sample, S is the size of the dataset, α and β are the parameters to balance the contribution of Φ_{ae} , Φ_{tc} , and Φ_{fc} . While Φ_{tc} and Φ_{fc} retain task-relevant features and disregard the unnecessary task-irrelevant features, the Φ_{ae} component is also necessary in the learning process because without the reconstruction loss Φ_{ae} , the risk of learning trivial solutions or worse representations will be increased (Aljalbout et al. 2018). This represents an initial self-supervised training phase in which the embedding space is trained to reflect the basic structure of the data.

In the second training stage, the label consistency loss Φ_{lc} is added to further improve the performance of the model. Since in the first stage, most clusters are already formed, the second stage imposes a small amount of pairwise constraints on the data using the limited amount of available weakly supervised data to readjust the already formed clusters to adapt to the available information about the intended classes. The joint loss function is defined as follow:

$$\begin{aligned} \min \sum_{i=1}^S & (1 - \alpha - \beta - \gamma) \cdot (\Phi_{ae}(x_a) + \Phi_{ae}(x_b)) + \\ & \alpha \cdot (\Phi_{tc}(x_a^t) + \Phi_{tc}(x_b^t)) + \\ & \beta \cdot (\Phi_{fc}(x_a^f) + \Phi_{fc}(x_b^f)) + \\ & \gamma \cdot \Phi_{lc}(x_a, x_b) \end{aligned} \quad (7)$$

The label consistency supervision Φ_{lc} used in the model is limited, but it is directly related to the activity recognition task, so in the second stage, the label consistency will have a larger weight and dominate the learning process, and the other loss functions will have smaller weights, so that the model can learn directly task-related features. Additionally, the amount of label consistency supervision used in the model is very limited, therefore, the other two self-supervised and one unsupervised loss functions are still needed to work as regularization terms to prevent potential overfitting.

Contrastive Loss Function

Part of the objective of the model is to minimize the reconstruction error and to optimize temporal and feature consistency using Euclidean distance, while a contrastive loss function (Hadsell, Chopra, and LeCun 2006) is used for label consistency.

Given the input pair $\{x_a, x_b\}$, the model outputs a pair of activity representations, $\{E(x_a), E(x_b)\}$. The contrastive loss function used to train the model is also defined on the pairs. The similarity distance $Dist_s$ between the pair of data samples is calculated using the euclidean distance between

their representation pair,

$$Dist_s(x_a, x_b) = \|E(x_a) - E(x_b)\|_2 \quad (8)$$

In the following, $Dist_s(x_a, x_b)$ is rewritten as D to simplify the notation. Then, for each of the categorizations used in training, the loss function is defined as follows:

$$L^i(x_a^i, x_b^i, y^i) = y^i L_s(x_a^i, x_b^i) + (1 - y^i) L_d(x_a^i, x_b^i) \quad (9)$$

$$L(x_a, x_b, y) = \sum_{i=1}^N L^i(x_a^i, x_b^i, y^i) \quad (10)$$

where (x_a^i, x_b^i, y^i) is the i -th sample in the data set. L_d, L_s are the loss terms for the negative pairs of data samples ($y = 0$) and the positive pairs of data samples ($y = 1$) respectively. The forms of L_d and L_s are given as,

$$L_d = \frac{1}{2} \{ \max(0, \delta - D) \}^2 \quad (11)$$

$$L_s = \frac{1}{2} (D)^2 \quad (12)$$

where δ is a hyperparameter for the margin. It states that only if the distance between two negative data samples is less than the margin δ , they contribute to the loss function.

Feature Extraction

Following the description of the proposed model, this part discusses the features used in experiments. The feature extraction step converts the segmented raw sensor inputs to feature vectors.

Let r_i be the *sample_i* in the set of segmented raw sensor signals, x_i represent the converted feature vector, and C denote the feature extraction function. Then, the extraction of features is defined as follows,

$$x_i = C(r_i) \quad (13)$$

The proposed model's input is x_i . Table 1 summarizes the statistical high level features that are used in the proposed model. The approach makes use of the mean, variance, standard deviation, and median, which are the most often utilized features in HAR studies. Additionally, several other features that have been demonstrated to be effective in prior research works (Reyes-Ortiz et al. 2016) are included here as well. For instance, the feature interquartile range (*iqr*) is based on Quartiles (Q_1, Q_2 and Q_3) that divide the time series signal into quarters. Using these, *iqr* is the measure of variability between the upper and lower quartiles, $iqr = Q_3 - Q_1$.

Each of these features is calculated independently for each axis. Because the data from several sensors is synchronized, it is possible to combine data from multiple sensors. During the training process, the proposed model accepts these derived features as input and learns to retain the task-oriented information included in the features while discarding the task-irrelevant information.

Cluster Construction

The network is trained using the standard backpropagation approach with stochastic gradient descent. Each weight is

Table 1: List of the used statistical features.

Feature extraction function	Description
$mean(r_i) = \frac{1}{N} \sum_{j=1}^N r_{ij}$	Mean
$var(r_i) = \frac{1}{N} \sum_{j=1}^N (r_{ij} - mean(r_i))^2$	Variance
$std(r_i) = \sqrt{var(r_i)}$	Standard deviation
$median(r_i)$	Median values
$max(r_i)$	Largest values in array
$min(r_i)$	Smallest value in array
$iqr(r_i) = Q_3(r_i) - Q_1(r_i)$	Interquartile range

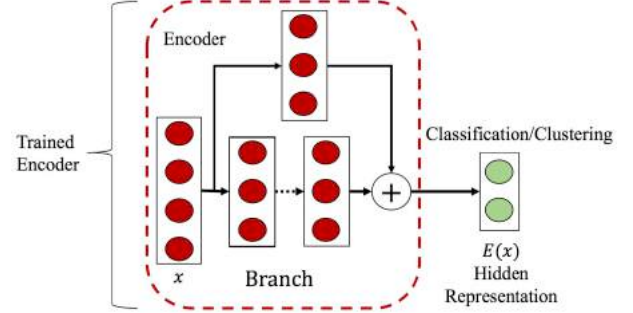


Figure 3: Trained Encoder for Classification/Clustering.

first set to a small value. The initial learning rate is $lr = 0.05$, and after every 10000 steps, an exponential decay function with a decay rate of 0.95 is applied to the learning rate. The network is evaluated using validation data at the end of each epoch. The training procedure is complete when the validation error has not decreased for a predetermined number of epochs.

After two stages of training to build the hidden representation within the siamese and ResNet autoencoder architecture, the trained model (the encoder of the model, as shown in Figure 3) can project the input data sample x into a clustering-friendly embedding space. The learnt representations are evaluated in the subsequent clustering task. The experiments use k -means (KM), arguably the most widely used clustering algorithm. The number of clusters in each dataset is selected to equal to the true number of classes.

Evaluations and Experiments

Three publicly available benchmark datasets containing wearable sensor data from various human activities are utilized to evaluate the proposed model.

Datasets

The three HAR datasets include: PAMAP2 (Reiss and Stricker 2012), REALDISP (Baños et al. 2012), and SBHAR (Anguita et al. 2013). The sliding window technique is used to segment all sensor data sequences.

PAMAP2 is a dataset collected from nine volunteers who completed 12 activities while wearing three inertial measurement devices (IMU) on their wrist, chest, and ankle. The dataset includes data on sporting activities (rope jumping,

nordic walking, and so on) as well as home activities (vacuum cleaning, ironing etc.). Heart rate, accelerometer, gyroscope, magnetometer, and temperature data are collected during the experiments. In accordance with previous studies on this dataset, the data is segmented using a sliding window of 5.12 seconds with a one-second step size.

SBHAR collects data from 30 subjects executing 6 basic activities such as walking and laying, as well as 6 postural transitions such as stand-to-sit and sit-to-lie. In our experiment, we treat all postural transitions as one general transition. The data was collected by strapping a smartphone to the participant’s waist and recording accelerometer and gyroscope data at a sampling rate of 50Hz using the inertial sensors on the smartphone. As in the previous study (Reyes-Ortiz et al. 2016), we employ a sliding window with a step size of 1.28 seconds and a duration of 2.56 seconds.

REALDISP is collected from 17 individuals performing 33 activities with the use of 9 sensors attached to their arms, legs, and back. Each sensor measures acceleration, gyroscope rotation, and the direction of the magnetic field. This dataset provides data on fitness workouts, as well as data from the warm-up and cool-down periods. The sliding window utilized in this case is two seconds in length without overlap.

To evaluate the performance of the proposed model, the original fully labeled data was modified by removing labels and randomly selecting 1%, 5%, and 10% weakly supervised data. The performance of the proposed approach with this information-reduced data was compared to fully supervised methods that had access to the complete original dataset as well as to unsupervised and purely self-supervised approaches. The comparison of purely self-supervised technique here also provides some ablation results for the proposed approach as it is effectively the self-supervised approach with only temporal and feature consistencies (Sheng and Huber 2020a). In all experiments, the total dataset (i.e. unlabeled and labeled) comprised all data segments of the datasets, giving the supervised comparison approaches access to all the labels while our model has access only to similarity information for a very small subset. To apply the weakly self-supervised method, the set of data segments was split according to the above-mentioned percentage and labels were removed from the samples in the data sets. The small percentage of labeled data was then converted into weakly supervised data pairs which only contain information whether they belonged to the same or a different class.

Results and Analysis

To gain a deeper understanding of the model’s learnt representations, we visualize the distribution of the activity vectors in the representation space. We used t-sne (van der Maaten and Hinton 2008) to map the representation vectors to two dimensions. The visualization results for the PAMAP2, REALDISP, and SBHAR datasets are shown in Figures 4, 5, and 6, respectively. The various colors in the figures indicate distinct activity labels in the datasets. The experimental results for the datasets are summarized in Tables 2, 3, and 4.

Table 2: Results on PAMAP2

Methods	ACC
MTLS (Sheng and Huber 2020b)	0.9893
Probability SVM with Filter (Reyes-Ortiz et al. 2016)	0.9433
Boosted C 4.5 (Reiss and Stricker 2012)	0.9969
CNN (Hammerla, Halloran, and Plötz 2016)	0.9370
UEL Autoencoder (Sheng and Huber 2020a)	0.8543
Vanilla Autoencoder	0.7706
Proposed approach with 1% labels	0.9128
Proposed approach with 5% labels	0.9755
Proposed approach with 10% labels	0.9904

Table 3: Results on REALDISP

Methods	ACC
Probability SVM with Filter (Reyes-Ortiz et al. 2016)	0.9952
Adaboost Ensemble Classifier (Subasi et al. 2018)	0.9998
CNN (San et al. 2017)	0.9280
SMLDist (Chen et al. 2021)	0.9400
UEL Autoencoder (Sheng and Huber 2020a)	0.6812
Vanilla Autoencoder	0.6401
Proposed approach with 1% labels	0.7515
Proposed approach with 5% labels	0.8780
Proposed approach with 10% labels	0.9223

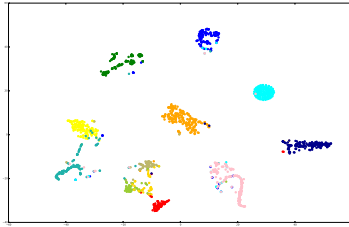
From the figures and tables, we can reach the conclusions that the proposed weakly self-supervised approach achieved significant improvement over the purely unsupervised method with the same number of clusters even with very limited data and improved performance with the increase of available weakly supervised data. Moreover, performance was reasonable good even with only 1% of the labeled training data. With 10% of the labeled data, the proposed approach can achieve competitive performance compared with even the purely supervised methods, despite it having access to only a fraction of the labeled data.

Conclusions

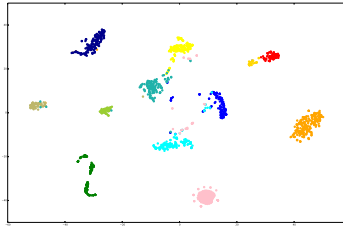
In this work, we have presented a weakly self-supervised embedding learning approach which is based on a ResNet autoencoder and a siamese architecture. The proposed approach uses two generic properties of human activities: temporal consistency and feature consistency, to project the activity data into an embedding space, and then imposes a small amount of pairwise constraints on the data to fine-tune the model in a weakly supervised fashion. We have demonstrated the effectiveness of the approach by applying it to three widely used HAR benchmark datasets where for our approach labels were removed and only a very small percentage of data pairs were used as weakly supervised data by

Table 4: Results on SBHAR

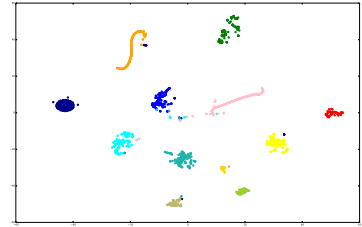
Methods	ACC
MTLS (Sheng and Huber 2020b)	0.9885
Probability SVM with Filter(Reyes-Ortiz et al. 2016)	0.9678
tFFT + CNN (Ronao and Cho 2016)	0.9575
STLS (Sheng and Huber 2019)	0.9268
UEL Autoencoder (Sheng and Huber 2020a)	0.7401
Vanilla Autoencoder	0.6369
Proposed approach with 1% labels	0.8015
Proposed approach with 5% labels	0.8853
Proposed approach with 10% labels	0.9352



(a): PAMAP2 with 1% of labels

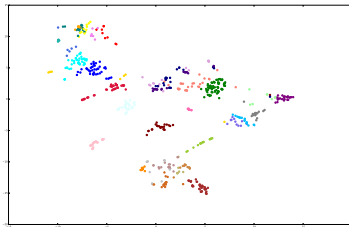


(b): PAMAP2 with 5% of labels

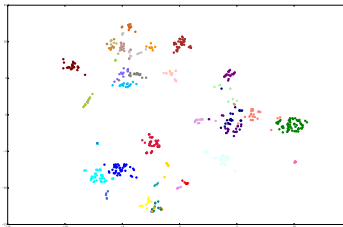


(c): PAMAP2 with 10% of labels

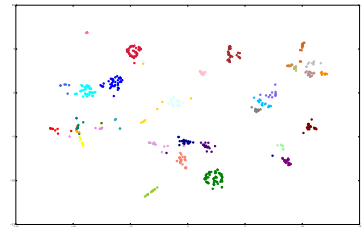
Figure 4: Representation space visualizations on the PAMAP2 dataset.



(a): REALDISP with 1% of labels

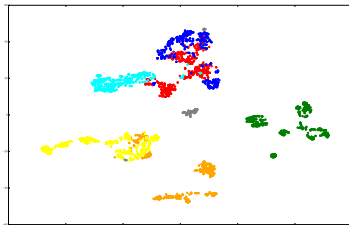


(b): REALDISP with 5% of labels

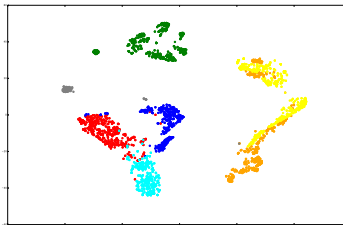


(c): REALDISP with 10% of labels

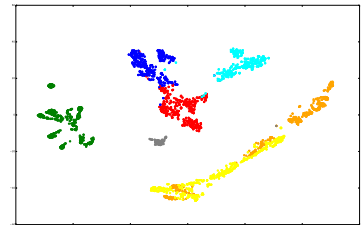
Figure 5: Representation space visualizations on the REALDISP dataset.



(a): SBHAR with 1% of labels



(b): SBHAR with 5% of labels



(c): SBHAR with 10% of labels

Figure 6: Representation space visualizations on the SBHAR dataset.

maintaining whether they were from the same activity class or not. The results of the experiments show that with only ten percent of the labeled data used for fine tuning through weak supervision, the proposed approach can achieve competitive results, which significantly reduced the supervision needed to train the recognition model and can thus dramatically reduce the overhead of obtaining useable datasets and extend the applicability of HAR to a broader range of ubiquitous sensor settings.

References

- Abu Alsheikh, M.; Selim, A.; Niyato, D.; Doyle, L.; Lin, S.; and Tan, H. 2016. Deep activity recognition models with triaxial accelerometers. In *AAAI Conference on Artificial Intelligence*, volume WS-16-01 - WS-16-15, 8–13. United States: AI Access Foundation.
- Aljalbout, E.; Golkov, V.; Siddiqui, Y.; and Cremers, D. 2018. Clustering with Deep Learning: Taxonomy and New Methods. *CoRR*, abs/1801.07648.
- Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; and Reyes-Ortiz, J. 2013. A Public Domain Dataset for Human Activity Recognition using Smartphones.
- 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, 1026–1035. New York, NY, USA: ACM. ISBN 978-1-4503-1224-0.
- Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. S. 2016. Fully-Convolutional Siamese Networks for Object Tracking. volume abs/1606.09549. Cham: Springer International Publishing.
- Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; and Shah, R. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, 737–744. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Chen, R.; Luo, H.; Zhao, F.; Meng, X.; Xie, Z.; and Zhu, Y. 2021. Modeling Accurate Human Activity Recognition for Embedded Devices Using Multi-level Distillation. *arXiv preprint arXiv:2107.07331*.
- Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, 539–546 vol. 1.
- Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742.
- Hammerla, N. Y.; Halloran, S.; and Plötz, T. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 1533–1540. AAAI Press. ISBN 978-1-57735-770-4.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- He, Z.; and Jin, L. 2009. Activity recognition from acceleration data based on discrete cosine transform and SVM. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, 5041–5044.
- Kamper, H.; Wang, W.; and Livescu, K. 2015. Deep convolutional acoustic word embeddings using word-pair side information. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4950–4954.
- Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese Neural Networks for One-shot Image Recognition.
- Kramer, M. A. 1991. Nonlinear principal component analysis using autoassociative neural networks. *Aiche Journal*, 37: 233–243.
- Minh Dang, L.; Min, K.; Wang, H.; Jalil Piran, M.; Hee Lee, C.; and Moon, H. 2020. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108: 107561.
- Morales, F. J. O.; and Roggen, D. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. In *Sensors*.
- Mueller, J.; and Thyagarajan, A. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 2786–2792. AAAI Press.
- Qin, Z.; Zhang, Y.; Meng, S.; Qin, Z.; and Choo, K.-K. R. 2020. Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion*, 53: 80–87.
- Reiss, A.; and Stricker, D. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *2012 16th International Symposium on Wearable Computers*, 108–109.
- 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.
- Ronao, C. A.; and Cho, S.-B. 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.*, 59: 235–244.
- San, P. P.; Kakar, P.; Li, X.-L.; Krishnaswamy, S.; Yang, J.-B.; and Nguyen, M. N. 2017. Chapter 9 - Deep Learning for Human Activity Recognition. In Hsu, H.-H.; Chang, C.-Y.; and Hsu, C.-H., eds., *Big Data Analytics for Sensor-Network Collected Intelligence*, Intelligent Data-Centric Systems, 186–204. Academic Press. ISBN 978-0-12-809393-1.
- Sheng, T.; and Huber, M. 2019. Siamese Networks for Weakly Supervised Human Activity Recognition. In *Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 4069–4075.
- Sheng, T.; and Huber, M. 2020a. Unsupervised Embedding Learning for Human Activity Recognition Using Wearable Sensor Data. In *FLAIRS Conference*.
- Sheng, T.; and Huber, M. 2020b. Weakly Supervised Multi-Task Representation Learning for Human Activity Analysis

- Using Wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4: 1 – 18.
- Subasi, A.; Dammas, D. H.; Alghamdi, R. D.; Makawi, R. A.; Albiety, E. A.; Brahim, T.; and Sarirete, A. 2018. Sensor Based Human Activity Recognition Using Adaboost Ensemble Classifier. *Procedia Computer Science*, 140: 104–111. Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.
- Synnaeve, G.; and Dupoux, E. 2016. A Temporal Coherence Loss Function for Learning Unsupervised Acoustic Embeddings. In *SLTU*.
- Tamura, T.; Sekine, M.; Ogawa, M.; Togawa, T.; and Fukui, Y. 1998. Classification of Acceleration Waveforms during Walking by Wavelet Transform. *Methods of information in medicine*, 36: 356–9.
- van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605.
- Wang, J.; Chen, Y.; Hao, S.; Peng, X.; and Hu, L. 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119: 3–11. Deep Learning for Pattern Recognition.
- Zeghidour, N.; Synnaeve, G.; Usunier, N.; and Dupoux, E. 2016. Joint Learning of Speaker and Phonetic Similarities with Siamese Networks. In *INTERSPEECH*.