# Few-shot Learning Human Activity Recognition

Zaid Bin Tariq

Project Report: EESC 6343

***Abstract*— This project aims to implement the paper titled "Few-Shot Learning-Based Human Activity Recognition" [1] . In Few-shot learning approach, the goal is to learn a model with very few labelled data samples. For the case of human activity recognition using the on-body sensors, learning models using few data samples helps in practical application of human activity recognition given that it is very expensive to obtain the human activity data for each possible activity. The proposed method in [1] utilizes the deep learning model to extract the features with knowledge transfer in the form of model parameters transfer which is to be utilized in classifying new activities.**

.

## I. Introduction

Human Activity Recognition (HAR) is a method to predict the activity of the participants using the data from the on-body sensors. The sensors might vary depending upon the application of HAR. Usually the aim is recognize the physical activity of the participant therefore the sensors used are mainly acelerometer, gyroscope and magnetometer. These sensors can be attached to different body parts of the participant like Arms, legs,chest etc. The time series data from these sensors is used to predict the activity. In order to predict these activities, the data from the on-body sensors needs to be sent to the device for processing. This usually consists of the on-body device like a smart phone. Once the data is received, the signal from different sensors is combined and given as an input to an already trained activity predictor. By predicting the activity, different applications come into play. For example, the Fall detection of elderly patients can be detected for smart elderly patient assistance. In this case, the detected fall can trigger emergency response from the associate authorities. Further more application like indoor localization and smart hospitals can also use HAR systems.

Different methodologies can be utilized for HAR. Data based machine learning approaches have become among the most popular given the advancement in machine learning algorithms and computational capability of the modern computers. Usually, the collected labelled data is used for to train a machine learning model like a artifial neural network. Given the incoming data from the sensors, this trained model can be used for predicting the acitivity. It should be noted that the supervised learning based HAR methods have an inherent problem in terms of annotated data available for training. The model trained on the data is not applicable for predicting new activities. It becomes necessary to come up with models that generalize to new activities given the model HAR applications requirement.

Few-shot learning is framework which helps is to generalize the trained model to newer classes of the data. Few-shot learning trains the parameters for new class using very few examples of new training class(es). A classic example for the applicability of Few-shot learning is the Face Recognition problem. In face recognition, a model is trained based on the available data set of human faces and the corresponding annotation. The data set for training usually consists of a few examples of a person with classification task of the model to tell if the given two images of face are of the same person. The few-shot learning works by easily integrating the person in to the face recognition system using only few images of the person as a support. Similar to the face recognition problem, few-shot learning framework should easily integrate new activities using the support set of the new classes. The model is learnt using the source classes where it is assumed that sufficient source classes annotated data is available for training the intial model. The model is trained in a way that later using the support set of the target classes, the model is able to make accurate enough predictions for the new/target activities.

## II. Methodology

Figure 1 shows the overall framework of the proposed solution for the few shot learning method for HAR. The framework consists of first training the source network using the source activity data set. Afterwards, a knowledge transfer step is used to transfer information from source network parameter to the target network for target activities classification. Although in the paper, it is not suggested what data is utilized for the knowledge transfer of the framework in figure 1, for the purpose of this project, we have utilized the support set for the target class with the details of the support set given in the proceeding sections. In the proceeding sections of the report, we explain each part of this framework which helps in HAR using limited number of target class labels i.e. few-shot HAR (FSHAR) .

### A. Source Network

Figure 2 shows the main features of the source network in the FSHAR framework. The source network takes the input $S_{src}$ from the source activity domain (classes). This input is converted into the features using the source feature extractor $f_{src}(\cdot, \Theta_{src})$ with parameters $\Theta_{src}$. The $d$ dimensional output of the source feature extractor is then passed to the source classifier $C(\cdot, W_{src})$ with parameters $W_{src}$. In this case, $W_{src} \in \mathbf{R}^{c_{src} \times d}$, where $c_{src}$ is the number of source classes.

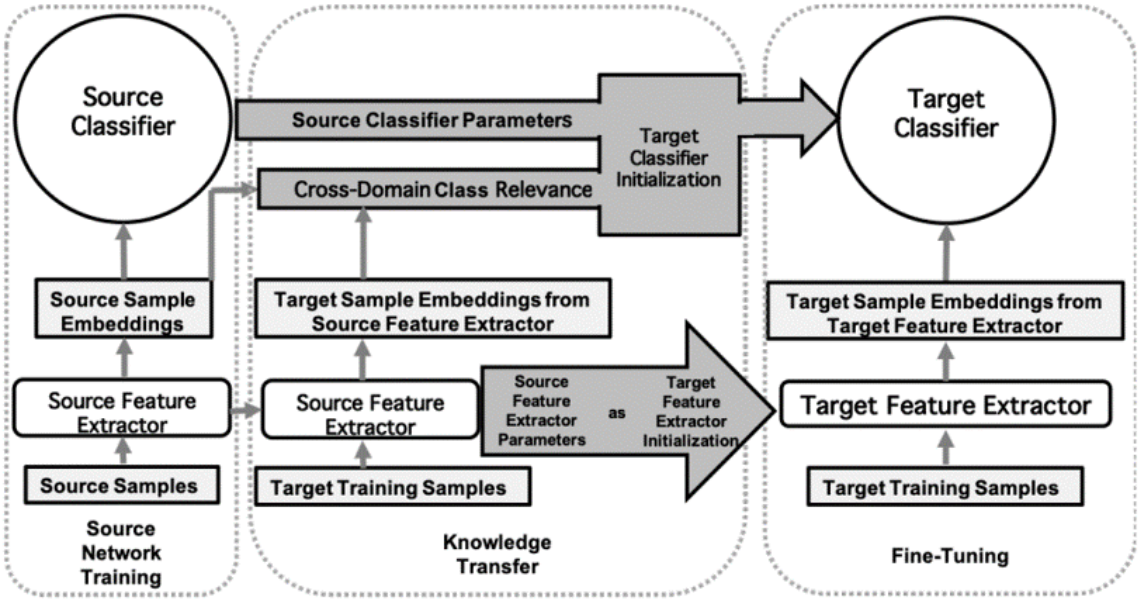


Fig. 1: Graphical summary of FSHAR framework. The framework consists of first training the source network using the source activity data set. Afterwards, a knowledge transfer step is used to transfer information from source network parameter to the target network for target activities classification [1].

Since $X_{src}$ for the case of activity recognition is a time series data of the sensors, we use the stacked LSTM followed by two fully connected layers for $f_{src}(\cdot, \Theta)$. LSTM helps to use the temporal dependence of the input $X_{src}$ for extracting the features since the the current samples in the HAR data is dependent upon the previous time samples [2]. The extracted embeddings of the feature extraction layer passes through the classifier. The output of the classifier is passed through the softmax function [3]. The source network is trained using categorical cross-entropy to learn the parameter $\Theta_{src}$ and $W_{src}$ [4]. Although ot mentioned in the paper [1], we introduce the 20% at each layer of the source network to introduce regularization generalization[5].

### *B. Knowledge Transfer*

The source network is trained using the data from the source activity data. The paper in [1], develops the knowledge transfer step of the overall the framework under the assumption that the source feature extractor extracts "generic information" as it forms the lower layer of the overall source network. In order to transfer this capability to the target network, we extract the feature extractor parameters $\Theta_{src}$ from the source network and use it as an initialization of the target network feature extractor called $\Theta^0_{trg}$. Hence the target feature extractor has the same architecture for the feature extractor in terms of using the same machine learning network like LSTM and dense layers. Although the feature extractor parameters $\Theta_{src}$ are assumed to be more generic, the classifier layer of the source network is more source activities specific and so the classifier $C(\cdot, W_{src})$ can not be directly used as the weights for the target network classifier $W_{trg}$. The authors in [1] develop a relevance measures to pick information to alleviate negative information transfer of the embedded features for the target network. The authors focus on the class-Wise relevance between the source activities and the target activities.

The first step is to calculate the sample-wise relevance between the source activity samples and the target activity samples. It is assumed here that the target activity samples come from the support set for the target classes which usually amount to very few samples depending upon the HAR application. After calculating the sample-wise relevance, we calculate the class-wise relevance. The idea is to find which activity in target activities is more relevant to the source activities. In order to calculate the sample-wise relevance, we use the exponential cosine similarity where the values between the $l^{th}$ source sample and $j^{th}$ target sample are saved in the matrix $A \in \mathbf{R}^{n_{src} \times n_{trg}}$ such that :

$$A^{(i,j)} = e^{[\tilde{f}_{src}(X^{(i)}_{src})]^T \tilde{f}(X^{(j)}_{trg})} \tag{1}$$

whee $\tilde{f}_{src}(\cdot) = f_{src}(\cdot)/||f_{src}(\cdot)||_2$ is the normalized embedded features. Although the authors in [1], suggest other relevance measures, in this project we utilize only the cosine similarity since we wanted to focus on the benefit of using the knowledge transfer framework which will be shown in the proceeding sections.

Given the sample-wise relevance matrix $A$, the class-wise relevance matrix $\mathbf{O} \in \mathbf{R}^{\mathbf{c_{src}} \times \mathbf{c_{trg}}}$, is given by:

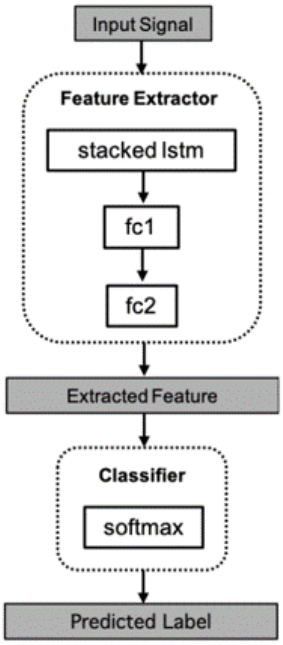


Fig. 2: Structure of the source network for HAR. The input goes through the stacked LSTM followed by two fully-connected neural network layers. The extracted features goes through the classifier softmax function as an output. The network is trained using categorical cross entropy[1].

$$\mathbf{O}^{(p,q)} = \sum_{i \in s_p} \sum_{j \in s_p} A^{(i,j)} \tag{2}$$

where $s_p$ and $s_q$ are the set of indices that correspond to the $p^{th}$ and $q^{th}$ class of source and target activities respectively. This scheme of using FSHAR with cosine similarity is called FSHAR-Cos. The resulting matrix is normalized for better transfer of parameters from the source to target network. For this purpose, we use the either the soft-normalization or hard-normalization to get matrix $W \in \mathbf{R}^{c_{src} \times c_{trg}}$ such that $W^{(p,q)} = O^{(p,q)} / \sum_{p=1}^{c_{src}} O^{(p,q)}$ for soft-normalization and $W^{(p,q)} = 1$ for $O^{(p,q)} = \{max_i O^{(i,q)}\}_i^{c_{src}}$. The final target classifier parameters is initialized as $W_{trg}^0 = W^T W_{src}$.

The final step in the knowledge transfer from source to target activities is the fine-tuning of the target network to get the final target classifier $\Theta_{trg}$ and $W_{trg}$. In order to perform the fine tuning, in this project, we use the support set of the target classes. It should be noted because we are using the Few-shot learning framework, we utilize the M-shot learning where $M$ samples from each target class is used in the support training set. For the purpose of this projec, we demonstrate the framework using M=1 and M=5. We use the same back propagation with categorical cross entropy as the loss function. We also use a low learning rate of 0.001. The overall FSHAR steps can be summarized as follows:

1) Train source network using $X_{src}$ to get feature extractor $\Theta_{src}$ source classifier $W_{src}$.
2) Initialize the target feature extractor $\Theta_{trg}^0$ using $\Theta_{src}$.
3) Calculate the class-wise relevance $W$ and initialize $W_{trg}^0$.
4) Fine tune the target network $\Theta_{trg}$ and $W_{trg}$.

## III. Evaluation and Results

Inorder to test framework of the proposed framework, we utilize the ReadDisp data set [6]. Figure 3 shows an example of the placement of the sensors on the body of the test particpant. The device consists combination of sensors consisting of the accelerometer, gyroscope and magnetometer. Each of these type of sensors measure the X,Y,Z part of the respective measurements. Considering the number of sensors attached to different body parts of the participant, a total of 117 sensor values are recording making a total of 117 features as input size. The data is measured at 50 samples per second. In order to collect the HAR data, 17 subjects perform 33 different activities like walking, running, jumping, squatting and many more.

Evaluating the proposed framework, we need to arrange the incoming data from each subject as a time series input to $f_{src}(\cdot)$. This is because the first layer to the network is a stacked LSTM. We use 5 seconds of the time series which makes the input $X_{src} \in \mathbf{R}^{250 \times 117}$ dimensional tensor. In order to increase the number of samples used for training and validation, we introduce an overlap of 1 second in the time series data from the sensors. We random choose 26 activities as source activities from these 33 activities and utilize the remaining 7 as target activities to be used for testing. As pointed in [6], the corresponding target activities are activities A12,A14, A16, A26, A27, A30 and A33. Further more we also separate the support training set for the target activities which consists of $M$ examples for M-shot learning method. In this case an example consist of 1 $250 \times 117$ tensor and we use $M$ of these as support set from each target activity. As mentioned earlier, $M$ is either 1 or 5 corresponding to 1-shot or M-shot learning approach. The size of M is fair given that in practical scenarios, 5 seconds can be used by the HAR system to gather the support set data from the subject.

Fig. 3: The placement of the on-body sensors for the collection of RealDisp data set[6].

It should be noted that each activity performed by the 17 subjects may not have similar number of resulting training

samples because of difference in the duration of performing a certain activity. For example, walking might be performed for longer duration compared to running. This results in the class imbalance which is a major issue for training a deep network. To overcome this problem, we currently rely upon re-sampling the training and validation data set for the source network. The resulting data is utilized for training the source network of LSTM-LSTM-FC1-FC2 of size 128-100-64-40 respectively. 20% drop out is introduced after every layer for regularization and generalization. For training the source network, we used 300 samples per class for 26 source activities making a total of 7800 samples with 520 samples used for validation of loss getting 98% accuracy.

Along with testing the proposed FSHAR framework, we also test on different baselines to assess the benefit of using the knowledge transfer method proposed in [1]. Following three different baselines that we use:

a) Source Parameter+ Classifier: In this case, we use the source feature extraction parameters $\Theta_{src}$ as the initialization for the feature extractor with MDL+softmax function as classifier. In this case, no knowledge transfer is used except for using the trained source parameters. The classifier parameters are untrained and are fined tuned along with the rest of the parameters using the support set.
b) Features + Nearest Neighbor: In this case, we use the source parameter extractor to get the normalized features of the test example. The step is also performed with the support set to get the features and train the nearest neighbor classifier. The test features are then classified using the nearest neighbor classifier.
c) Fine Tuned Untrained Network: In this case, we do use the parameters of the source network for any layers of the target network. The network is trained from the ground up using only the support set for target activities.

For testing, we use 2636 samples from the target classes. Table I shows the results for the proposed FSHAR-Cos with hard and soft normalization along with the baselines. First we can observe from the results that Soft normalization is better than Hard normalization. FSDHAR-Cos-Soft achieves test accuracy of 93.5% and 72.7% compared to 85.7% and 61.2% for 1-shot and 5-shot respectively. The results indicate that soft normalization better capture the class-wise relevance between source and target activities to help toward the target network. In order to asses the knowledge transfer capability of the FSHAR-Cos, we test the data using feature and softmax classifier (as in item (a) above). In this case the classifier do not use the knowledge transfer method as proposed in this project. The results show approximately 7% lesser accuracy compared to the FSHAR-Cos-Soft which indicates that the knowledge transfer has a significant benefit for few-shot HAR. Similarly using only the source features and the nearest neighbor classifier, we get an approximately 85.2% 5-shot accuracy which is similar to FSHAR-COS-Hard and baseline (a) which reinforces the hypothesis that hard normalization of the target classifier in FSHAR does not significantly helps with knowledge transfer. Finally, we use an untrained network from ground up with the same architecture to train using the support set. For 1-shot case, it give an accuracy of 48.5% and 74.0% for 5-shot learning which far less than FSHAR-Cos-Soft. This shows that using the source network as the initialization for the target network does helps with getting better knowledge transfer.

| Method | 1-shot | 5-shot |
|---|---|---|
| FSHAR-Cos-Hard | 61.2 | 85.7 |
| FSHAR-COS-Soft | 72.7 | 93.5 |
| Features +softmax (a) | 65.3 | 86.6 |
| Feature + NN (b) | 17.0 | 85.2 |
| Fine Tuned Untrained (c) | 48.5 | 74.0 |

TABLE I: Results for different methods for target activity classification.

## IV. CONCLUSIONS

In this project, we have implemented a Few-shot learning based human activity recognition framework. It is assumed that we already have the source activity data to train the source network. The assumption is that the embeddings learned from the source network can be used to generalize to the target activity classification. For this purpose, a knowledge transfer method is proposed which uses the class-wise similarity between the source and target activities for easy transfer of knowledge from source to target classes. We evaluation shows that FSHAR-Cos with soft normalization significantly helps with the knowledge transfer.

## REFERENCES


[1] S. Feng and M. F. Duarte, "Few-shot learning-based human activity recognition," *Expert Systems with Applications*, vol. 138, p. 112782, 2019.
[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[3] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, vol. 2, no. 3, 2016, p. 7.
[4] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.
[5] N. Srivastava, "Improving neural networks with dropout," *University of Toronto*, vol. 182, no. 566, p. 7, 2013.
[6] O. Baños, M. Damas, H. Pomares, I. Rojas, M. A. Tóth, and O. Amft, "A benchmark dataset to evaluate sensor displacement in activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1026–1035.