Comparing CNN and Human Crafted Features for Human Activity Recognition

Federico Cruciani*, Anastasios Vafeiadis†, Chris Nugent*, Ian Cleland*, Paul McCullagh*,
Konstantinos Votis†, Dimitrios Giakoumis†, Dimitrios Tzovaras†, Liming Chen‡ and Raouf Hamzaoui‡
*School of Computing, Ulster University - Belfast, Northern Ireland
Email: {f.cruciani, cd.nugent, i.cleland, pj.mccullagh}@ulster.ac.uk
†Information Technologies Institute - Center of Research & Technology Hellas - Thessaloniki, Greece
Email: {anasvaf, kvotis, dgiakoum, tzovaras}@iti.gr
‡Faculty of Computing, Engineering and Media - De Montfort University - Leicester, UK
Email: {liming.chen, rhamzaoui}@dmu.ac.uk

***Abstract*—Deep learning techniques such as Convolutional Neural Networks (CNNs) have shown good results in activity recognition. One of the advantages of using these methods resides in their ability to generate features automatically. This ability greatly simplifies the task of feature extraction that usually requires domain specific knowledge, especially when using big data where data driven approaches can lead to anti-patterns. Despite the advantage of this approach, very little work has been undertaken on analyzing the quality of extracted features, and more specifically on how model architecture and parameters affect the ability of those features to separate activity classes in the final feature space. This work focuses on identifying the optimal parameters for recognition of simple activities applying this approach on both signals from inertial and audio sensors. The paper provides the following contributions: (i) a comparison of automatically extracted CNN features with gold standard Human Crafted Features (HCF) is given, (ii) a comprehensive analysis on how architecture and model parameters affect separation of target classes in the feature space. Results are evaluated using publicly available datasets. In particular, we achieved a 93.38% F-Score on the UCI-HAR dataset, using 1D CNNs with 3 convolutional layers and 32 kernel size, and a 90.5% F-Score on the DCASE 2017 development dataset, simplified for three classes (indoor, outdoor and vehicle), using 2D CNNs with 2 convolutional layers and a 2x2 kernel size.**

***Index Terms*—Human Activity Recognition, Deep Learning, Convolutional Neural Networks, Free-living.**

I. Introduction

Human Activity Recognition (HAR) finds several real-life applications; in smart home research, for instance, it can be applied to support Ambient Assisted Living (AAL) [1] [2]. AAL is among the application scenarios making this research branch particularly relevant. Its relevance is linked to the potential role AAL could play in dealing with rising healthcare costs associated with an ageing demographic [3]. At the same time, HAR is also among the main fields of application of Machine Learning (ML). As in other cases of ML applications (e.g. speech-recognition or visual object recognition), Deep Learning (DL) has been increasingly employed in recent years, and its adoption has led to a significant improvement in the state-of-the-art performance metrics [4]. HAR is no exception in this sense. DL methods such as Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) have shown good results in terms of recognition accuracy both in the case of simple activities (e.g., 'sitting', 'standing', 'walking') and complex activities (e.g., preparing a meal) [1], [5]. One of the distinctive traits of DL approaches to HAR is that these methods simplify some stages of the conventional approach taken to activity recognition. Namely by automating some of the steps commonly employed in similar classification tasks. In sensor-based HAR, DL allows direct use of raw data as input [4] (e.g., in the case of HAR based on inertial sensors [6]). For instance, CNNs have been successfully employed to extract relevant features from the accelerometer raw data signal, in an automated fashion [1]. The ability of processing data in its raw form represents a disruptive change in comparison to conventional ML approaches. In the past, the step of feature extraction would require (in most cases) an advanced degree of domain specific expertise [4]. Despite the popularity of DL methods, very little focus has been put on automatically extracted features, particularly in comparison with the case of Human Crafted Features (HCF). While in the first case, HCF have been the objective of several studies attempting to identify optimal feature sets for different target activities [7], [8], for automatically extracted features, implementation of the HAR chain focuses on the recognition accuracy performance of different classifiers and less on the impact of the extracted features. Some recent studies have attempted to fill this gap [1], [9]. In [1], 1D temporal convolution has been examined for HAR using inertial sensors. In [9], different feature learning methods have been compared including CNN, LSTM and HCF. The comparison, however, has been measured on the final F1-score obtained on the same dataset using different methods, rather than focused on comparison between HCF and automatic features. In this respect, this works provides the following contributions:

- an analysis of CNN feature extraction is provided, both in the case of inertial and audio sensors
- a comparison between CNN auto features and gold standard HCF is performed on publicly available datasets. For

the inertial sensors, we used the 1D feature vectors from accelerometer and gyroscope. For the audio sensors, we used 2D images, normalized from 0 to 1, of the Mel-Frequency Cepstral Coefficients (MFCCs)

The remainder of the paper is structured as follows. Section II describes relevant work in HAR, giving particular emphasis to feature quality. Section III describes the approach and the final experiment. The evaluation methodology is described in Section IV. Results and discussion are reported in Section V and Section VI respectively. Finally, conclusions are drawn in Section VII.

II. Background

The process of activity recognition usually includes a specific sequence of steps, also known as Activity Recognition Chain (ARC) [10]. The chain describes the process of HAR going from raw data to final classification, and includes the following steps: pre-processing, segmentation, feature extraction, and classification. In the pre-processing step, raw data are processed in order to transform them into a form suitable for processing by a classification model. Typical operations performed at this stage include for instance filtering (e.g., in the attempt or removing noise or not relevant parts of the signal), or re-sampling, as in the case where multiple inputs acquired at different sampling rates need to be put together into a time-series. In the segmentation step, the data sequence is divided into a set of segments. This operation introduces a relevant parameter for classification which is the window size, that will determine the length of each segment [10]. The choice of the window size can influence the ability of extracting informative features from the segment. If, for instance, the window size is too short, relevant features may be missed. Window sizes of 1-4 seconds are often used for HAR of simple activities, while larger window sizes are considered generally for complex activities, as in the case of Activities of Daily Living (ADLs) [6], [11]. The following step in the ARC is the feature extraction step [10], that typically requires domain specific expertise [4]. It is common at this stage to perform an additional step, known as feature selection. Feature selection normally consists in an iterative process, in which different subsets of HCF at each iteration are evaluated based on the final accuracy. This process allows for the identification of an optimal set of features [12]. Although feature selection further exacerbates the complexity of generating a candidate feature set, this process can be automated, as for instance in [12]. In this case, an algorithm has been proposed to discard features with low importance (i.e. not improving classification accuracy), however, the initial set of features prior to selection are still human crafted. DL approaches facilitate automation of both feature extraction and selection [4], and that is also the case of CNN based methods [1], [2], [9]. In this work, two main cases are analyzed: inertial sensors and audio signals for HAR. Consequently, the following subsections will describe common feature extraction techniques for the two cases.

A. Inertial sensor

Inertial sensors are commonly used in HAR as they are less power demanding compared to other sensors, such as Global Positioning System (GPS), and do not pose privacy issues, which occur in the case of video-based approaches. Several studies have investigated different HCF sets as well as feature selection techniques [7]. Common features used for inertial-based HAR usually belong to two main groups, depending on the fact if they are extracted from the time or the frequency domain [6], [7]. Time domain features are more often used and include the statistical moments of the signal (*i.e.* mean, variance, skewness), or other simple features such as max and min values in the interval. Frequency domain features require Discrete Fourier Transform (DFT) computation, and therefore are more rarely employed due to the inherent computational complexity [6]. Identifying an optimal feature set for HAR has been the objective of several studies. Consequently, it is possible to rely on existing literature, that provides a comprehensive analysis of features quality, as in [7]. The UCI-HAR dataset [13] includes a set of 561 features extracted from accelerometer and gyroscope signals (348 considering the accelerometer only), both from the time and frequency domain and both at single axis and at magnitude level. When focusing on DL approaches, and particularly on automating feature extraction, only few studies attempted to do an analysis of produced features [2]. Feature learning strategies including DL has been the objective of [9]. In this case however, the comparison is provided only on the final accuracy of models, that are trained using different approaches (including CNN). In [1], a more detailed analysis of features extracted using CNN is provided, including an insight on the effect of main parameters used for CNN classifiers (e.g., number of convolutional layers and filter size). The objective of the study, however, was to evaluate optimal CNN parameters for HAR, rather than focusing on CNN features. Also in this case, results were provided on the final accuracy of the CNN approach, measured on the UCI-HAR dataset [13]. Moreover, the study analyzed only the case of combined accelerometer and gyroscope signals, while in this work, results obtained using both combined accelerometer and gyroscope, and accelerometer only, are presented.

B. Audio Features

In the field of computational auditory scene recognition, feature extraction remains a fundamental problem. Many types of low-level features such as zero-crossing rate, band-energy ratio, spectral roll-off, spectral flux, spectral centroid, spectral contrast, MFCCs and gammatone frequency cepstral coefficients are commonly used in the literature [14]–[17]. The majority of the features selected within these studies, however, only work well for structured data, such as speech and non-speech separation or genre music classification. Therefore, a more discriminative feature set that captures the spatial and temporal events is required, especially for environmental sounds. Recently, deep CNNs have been successful in many tasks such as, speech recognition [18], audio source separation

[19] and environmental sound recognition [20]. However, the problem of audio-based event recognition remains a hard task. This is because DL approaches that work extremely well for a specific dataset may fail for another. The fundamental difficulty of environmental sound recognition is that the input signal is highly variable due to different environments (indoor, outdoor, vehicle) and acoustic conditions.

In this work, we evaluate the performance of CNNs on a large-scale dataset [21]. The DCASE 2017 development dataset consists of recordings from various acoustic scenes, all having distinct recording locations. For each recording location, 3-5 minute long audio recording was captured. The original recordings were then split into segments with a length of 10 seconds. The total number of recordings were 4680, sampled at 44.1 kHz and were split in four folds (75/25 train/validation split). We compare the recognition accuracy between standard human crafted audio features (MFCCs) and the low-level features that 2D CNNs learn during back-propagation. The MFCCs were selected since they are the most common features used in the fields of speech recognition and environmental sound recognition. Furthermore, since this work used images to train 2D CNNs, it would not be possible to use features such as the zero-crossing rate, spectral centroid, etc.

C. Automatic feature extraction

Among the advantages that DL provides for automatic feature extraction, one of the most relevant is that it does not require domain specific knowledge [4]. In this sense DL provides a standardized way to fulfill the feature extraction step. On the other hand, it introduces some disadvantages. In particular, a training phase is required in order to optimize the weight of the convolutional layers to the characteristic of data from the target domain. This makes DL methods for feature extraction subject to the *cold-start* problem, and potentially also to generalization.

The experiment in Section IV, focuses on feature space rather than on final accuracy. Moreover, the analysis of auto features includes the comparison with gold standard HCF sets.

III. Approach and Experiment

Typically, CNN architectures include several convolutional layers. In most cases convolutions are followed by a max-pooling operation. The first layer can take directly raw data as input, and the convolutional layers play the role of extracting good features for the final classification. Few cases can be distinguished. For instance, when dealing with inertial sensors (such as accelerometer or gyroscope), the input will be a sequence of samples for each channel. Given an accelerometer signal sampled at 40 Hz, and assuming a window size of 3 s for segmentation, this will produce a 3x120 input in the case all 3 channels (X, Y, Z) are considered separately, or a 1x120 input in the case only the 3D magnitude of the acceleration is taken as input. The input is then processed using temporal convolution (Conv1D) as in [1].

In other cases, 2D convolution is used, as for instance in the case of an image used as input. Audio input typically falls into this category where the image representing the spectrogram (or any variations of it e.g., mel-spectrogram) or 2D matrix (e.g., MFCCs) of the audio signal in the time window is used as input. For the case of the MFCC feature extraction, we used the default sampling rate (44.1 kHz) of the DCASE dataset. The number of MFCCs was 13 (including the $0th$ coefficient), the Fast Fourier Transform (FFT) window size was 2048, with a hop length of 1024 (50% overlap). This resulted in a 13x431 matrix. It has been shown that 2D CNNs outperform 1D CNNs in many audio recognition tasks, since they are able to capture the spatio-temporal information of the signal [22]. After the sequence of convolutional layers, the output of last convolutional layer is generally flattened into a 1D vector. The output of this step is the automatic feature vector in our experiment.

In the case of a multi-class problem (where classes are mutually exclusive) the output layer is obtained typically using a softmax activation layer in a dense layer. In some cases, a number of dense layers are added in between the flatten and the softmax operation [23], or in case of multi-label classification (where more than one output class can be active at the same time, e.g., “standing” and “walking”) other activation functions such as sigmoid can be used [24]. Similarly, the loss function used for training varies commonly from mean squared error in the case of multi-class, or binary cross-entropy in the case of multi-label [24]. The entire process is depicted in Fig.1, showing both conventional process with HCF, and automatic features using CNN.

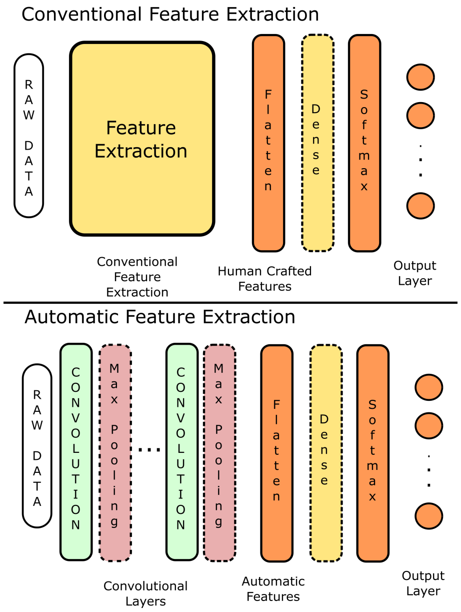


Fig. 1. A typical CNN architecture taking IMU raw data as input will include multiple convolutional layers, with each layer followed by a max-pooling operation. The output of last convolutional layer is then flattened. The vector obtained corresponds to a feature vector automatically extracted. Finally, softmax is generally applied to connect to the output layer in multi-class problems.

The advantages of the CNN approach are (i) the ability of feeding the model directly with raw data, and (ii) that features are extracted within the series of convolutional operations automatically. On the other hand, as mentioned in Sec.II, the CNN layers will not be able to generate good features, until the model is trained on some known data.

To solve the cold start problem a different dataset can be used for initial training. In our case, one for the IMU feature extractor and the second for the audio signal. Our rationale for the training dataset is to consider a dataset with the following characteristics:

1) the dataset shall not be subject to label noise (reducing the risk of overfitting by learning on noisy labels)
2) the input data and the target activities must be similar to the target application scenario (so that a feature extractor trained in a similar dataset can be used on the target dataset, in a similar way to a transfer learning approach).

In order to fulfill the first requirement, only datasets collected in controlled environment have been considered. Similarly, regarding the second requirement we consider only datasets with the same input data and similar target activity sets.

The initial training phase and the comparison with HCF is performed for the inertial sensor using UCI-HAR dataset [13] and for the audio component the DCASE 2017 development dataset [21]. These two datasets match our aforementioned requirements for training purposes. The main reason for using those datasets is to be able to locate the user.

Fig.2 illustrates the training and testing phases of the experiment. UCI-HAR and the DCASE 2017 development dataset are used for training the feature extractor. At this stage the two datasets are used to compare auto features, extracted from the CNNs, with human crafted ones.

Fig.2 depicts the proposed process for cross-validation, where CNN feature extractor is trained on a dataset, then the CNN feature extractor is cross-validated in conjunction with a classifier model for classification on a different dataset.

IV. Evaluation Methodology

In the experimental work two datasets have been used, UCI-HAR dataset [13] for inertial sensors, and the DCASE 2017 development dataset [21] for the audio case. For the audio case, we simplified the 15 classes to 3, based on the location (indoor, outdoor and vehicle). Simplifying the classes helps in a scenario where inertial and audio sensors would be used to approximate the user's location, without a GPS sensor. UCI-HAR dataset fulfill our requirements being a dataset collected under controlled conditions. IMU data have been collected using a Samsung Galaxy S II smartphone, placed on the waist. The dataset includes data from 30 participants and provides benchmark training (70% of participants) and test dataset (30% participants). This type of evaluation enables accuracy performance to be tested on users that have not been part of the training. The dataset provides a benchmark set of 348 features, extracted from time and frequency domains of the accelerometer signal. This benchmark set has been used as the set of HCF in the comparison.

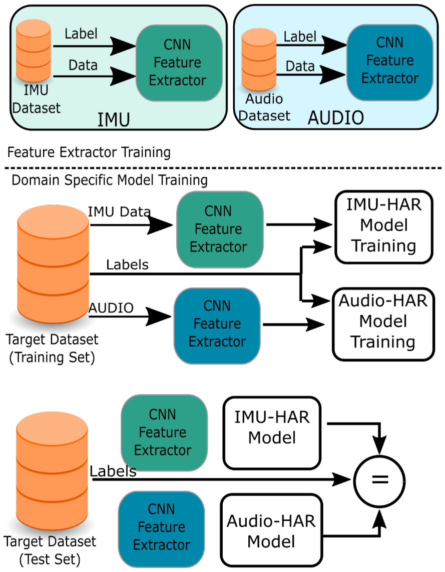


Fig. 2. Cross-validation of a CNN automatic feature extractor: (top) two CNN feature extractors are trained using datasets collected in controlled conditions (UCI-HAR and the DCASE 2017 development datasets), (middle) the CNN model is used as feature extractor, and training data from the final real-world dataset is used to train IMU-HAR and AUDIO-HAR models, (bottom) finally results are evaluated over the final test data.

A. Evaluation of auto features

The aforementioned datasets were used to train the feature extractor, and the features obtained were compared to human crafted ones. The comparison aims also at verifying how parameters affect the quality of automatically generated features. The analysis investigated the following parameters:

1) number of convolutional layers
2) kernel size used for convolution

The final accuracy of the model was complemented with plots, visualizing how activities were separated in the feature space, both in the case of HCF and automatic features. The visual comparison assists to analyze and interpret results obtained in terms of accuracy and provides a more complete evaluation of CNN features.

B. Environment

The experimental framework was implemented using Python. A Keras [25] implementation of CNN was used, with TensorFlow [26] as the backend.

V. Results

A. IMU CNN Features

Regarding automatic generation of CNN features for the IMU, the experiment focused on the set of target activities defined in the UCI-HAR [13] dataset (i.e., 'Laying', 'Sitting', 'Standing', 'Walking', 'Walking Upstairs' and 'Walking Downstairs'). Feature quality has been measured on accuracy performances of CNN models using different layers of convolution (n-CNN where n is the number of layers) and different values of kernel size k. Models were trained for 150 epochs with a batch size of 512 samples. The benchmark train and test sets from UCI-HAR have been used. The training set has been further split using 90% for training and 10% as validation for early stop criterion to avoid overfitting. The UCI-HAR dataset provides IMU raw data for the accelerometer and gyroscope signal. Data are split in segments with a window size of 128 samples and 50% overlap, obtained from IMU signals sampled at 50 Hz. The accelerometer signal is separated into gravity and body components, separated using a Butterworth low-pass filter (cutoff 0.3 Hz). The separation makes a total of 9 channels, 6 for the accelerometer (X,Y,Z for both gravity and body components) and 3 for the gyroscope (X,Y,Z). Consequently, an input of 128x9 or 128x6 was obtained by taking accelerometer and gyroscope signals, or accelerometer only. For multi layers CNN models, the number of filters used were 12, 24, 48 and 96 respectively, each followed by a max-pooling operation. For IMU 1D convolution was used. A flatten layer was used after the last convolutional, followed by a dense layer (64 nodes and relu activation function), and the final output with the 6 classes using a softmax and adam optimizer with learning rate lr=0.001. The comparison with HCF was performed using the same architecture as the CNN case after the flatten layer. The input layer would take HCF (a 348 features vector using accelerometer only, and 561 with both accelerometer and gyroscope) with a dense layer of 64 nodes, and the final output layer. Comparison between HCF and CNN automatic features was performed on the same architecture as in Fig.3.

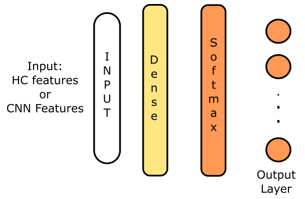


Fig. 3. Architecture used to compare HCF and CNN features: taking features in input, one dense layer (64 nodes) and 6 classes (for the inertial sensors) and 3 classes (for the audio sensors) on the output layer.

Fig.4 (left column) depicts a visualization of the feature space obtained performing Principal Component Analysis (PCA), reducing to three dimensions for visualization purposes. To facilitate visual inspection of data points separation in the feature space, in Fig.4-5 the plot has been obtained excluding data points labeled as ‘Laying’, which were far away in the feature space from all other activities; including them would affect interpretation of visualized data. Training of models using different kernel sizes has been performed. Fig. 4 (right column) depicts data points in the feature space for activities using varying kernel sizes. Fig.5 depicts visual comparison of target activities separation in the feature space using (top) HCF, and (bottom) automatic features obtained with 3-CNN model and using kernel size $k = 32$.

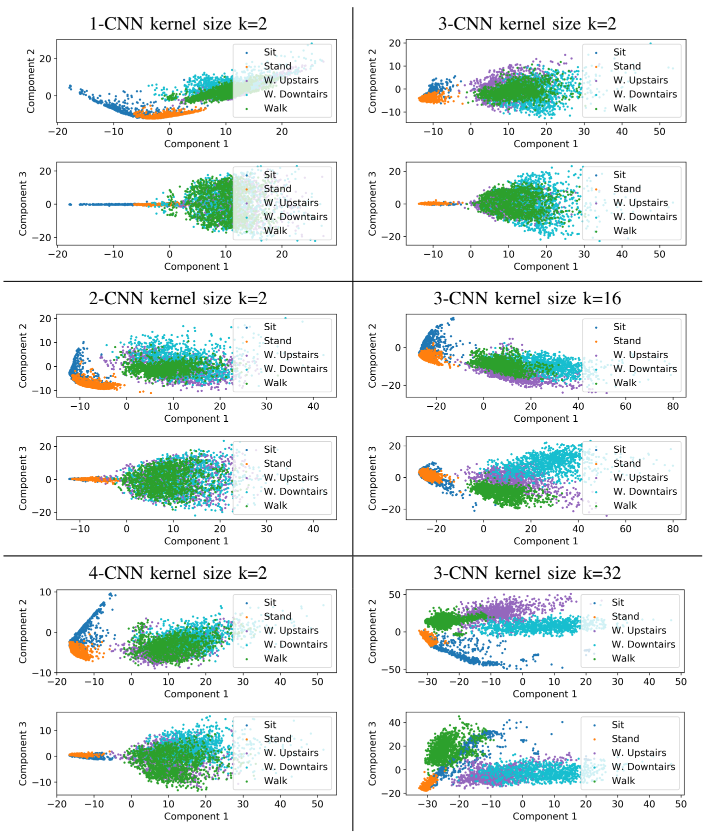
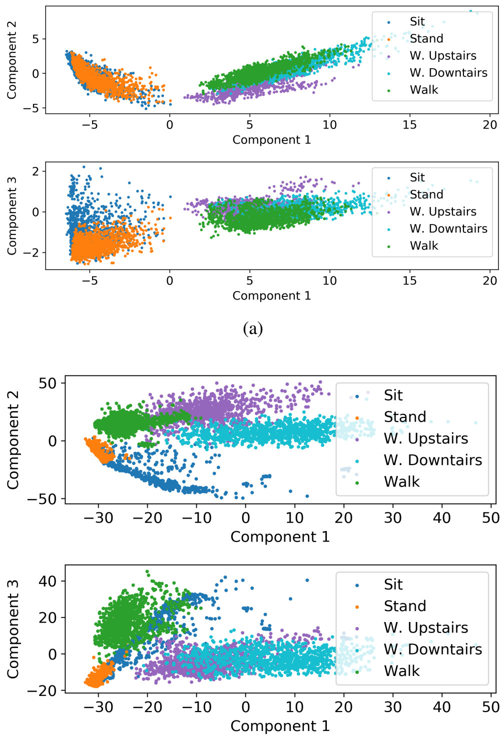


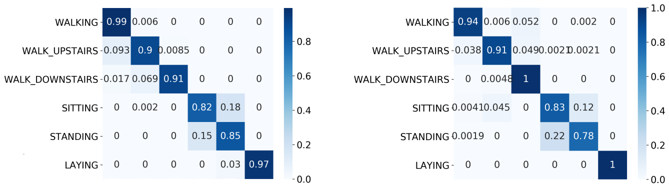
Fig. 4. Visual comparison of 1D CNN architectures using $n = 1, 2$ and 4 layers with a kernel $k = 2$ (on the left), and kernel size $k = 2, 16, 32$ using $n = 3$ layers (on the right).



(a)

(b)

Fig. 5. Visual comparison of (a) HCF, and (b) CNN using 3 layers.



(a) (b)

Fig. 6. Normalized confusion matrices obtained using (a) HCF and (b) using CNN.

Table I and II summarizes average precision, recall and F-score with the tested models, using accelerometer only, and both accelerometer and gyroscope signals.

TABLE I
PRECISION, RECALL AND F-SCORE OBTAINED ON UCI-HAR DATASET USING HCF AND CNN FEATURES OBTAINED WITH DIFFERENT PARAMETERS USING ACCELEROMETER ONLY.

Parameters	Precision	Recall	F-Score
HCF (acc only)[a]	**89.95%**	**89.38%**	**89.58%**
1-CNN K=2	85.31%	84.63%	84.41%
2-CNN K=2	88.26%	87.95%	87.96%
3-CNN K=2	90.73%	90.57%	90.55%
4-CNN K=2	89.62%	89.21%	89.19%
3-CNN K=8	**90.55%**	**90.26%**	**90.20%**
3-CNN K=16	90.71%	90.09%	90.16%
3-CNN K=32	88.24%	87.89%	87.87%
3-CNN K=64	88.17%	87.95%	87.97%

[a] set of 348 accelerometer only.

TABLE II
PRECISION, RECALL AND F-SCORE OBTAINED ON UCI-HAR DATASET USING HCF AND CNN FEATURES OBTAINED WITH DIFFERENT PARAMETERS USING ACCELEROMETER AND GYROSCOPE.

Parameters	Precision	Recall	F-Score
HCF (acc & gyro)[a]	**95.80%**	**95.39%**	**95.50%**
1-CNN K=2	88.84%	89.01%	88.87%
2-CNN K=2	89.54%	89.70%	89.59%
3-CNN K=2	90.51%	90.63%	90.55%
4-CNN K=2	91.84%	91.97%	91.89%
3-CNN K=8	91.44%	91.63%	91.51%
3-CNN K=16	92.96%	93.08%	92.98%
3-CNN K=32	**93.31%**	**93.52%**	**93.38%**
3-CNN K=64	92.03%	92.20%	92.04%

[a] 561 features: accelerometer and gyroscope.

Fig. 6 provides further insight on how error rates were distributed between classes, comparing HCF with a CNN model (3 layers kernel size $k = 64$).

B. Audio CNN Features

Regarding automatic generation of CNN for the DCASE 2017 development dataset, the experiment focused on grouping the 15 given classes (beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train and tram) to three classes, namely outdoor, indoor and vehicle. The raw spectrogram images were used as the input to the CNN. In order to extract the spectrogram of the signal, an FFT size of 512 with a hop length of 512 was used. Furthermore, the original recording was down-sampled to 16 kHz. The reason for down-sampling and using a non-overlapping window for FFT was due to the length of the recording (10 s), which would produce an image size that could not be processed. Therefore, the resulting image after the pre-processing step was 257x313 pixels. Feature quality has been measured on accuracy performances of CNN models using different layers of convolution (n-CNN where n is the number of layers) and different values of kernel size k. The filters used for each CNN layer were 32, 48, 120 and 120 respectively followed by a 2x2 max-pooling layer. The CNNs were trained between 20-30 epochs (for different folds and different network sizes) and the selected batch size was 32. The number of epochs was selected based on the early stopping criterion, in order to avoid over-fitting. The ReLU [27] activation function was used for each convolutional and max-pooling layer and the Adam [28] optimizer was used to train the networks with an initial learning rate lr = 0.001.

For our experiments we used the default 4-fold cross validation that is provided in [21]. However, we show the PCA analysis for the second fold, since it was the most challenging one. The other folds follow a similar trend. Table III summarizes average precision, recall and F-score with the tested models. We notice that the best model consisted of 2 convolutional layers with a kernel size of 2. Fig.7 (left column) depicts a visualization of the feature space obtained performing PCA, reducing to three dimensions for visualization purposes. Fig. 7 (right column) depicts data points in the feature space for activities using varying kernel sizes. The variance of the PCA components increases as the number of kernels increases. The first and third principal components show that for the case of two convolutional layers (kernel of size 2), two classes can be distinguished in the feature space.

TABLE III
PRECISION, RECALL AND F-SCORE (AVERAGED OVER 4-FOLDS) OBTAINED ON THE DCASE 2017 DEVELOPMENT USING DIFFERENT PARAMETERS.

Parameters	Precision	Recall	F-Score
HCF (MFCCs)	85.37%	85.22%	84.75%
1-CNN K=2	51.63%	60.47%	53.72%
2-CNN K=2	**91.02%**	**90.2%**	**90.5%**
3-CNN K=2	90.9%	90.17%	90.45%
4-CNN K=2	90.14%	89.56%	89.74%
2-CNN K=8	89.1%	88.62%	88.78%
2-CNN K=16	49.58%	52.9%	52.12%
2-CNN K=32	11.09%	33.33%	16.59%
2-CNN K=64	12.23%	33.33%	17.86%

VI. DISCUSSION

A. IMU CNN Features

Results obtained using different number of layers of convolution highlights how a model with 3-4 layers outperforms models with 1 or 2 layers in F-score. At the same time, adding more layers of convolution does not improve accuracy, while increasing complexity of the model. The results are confirmed

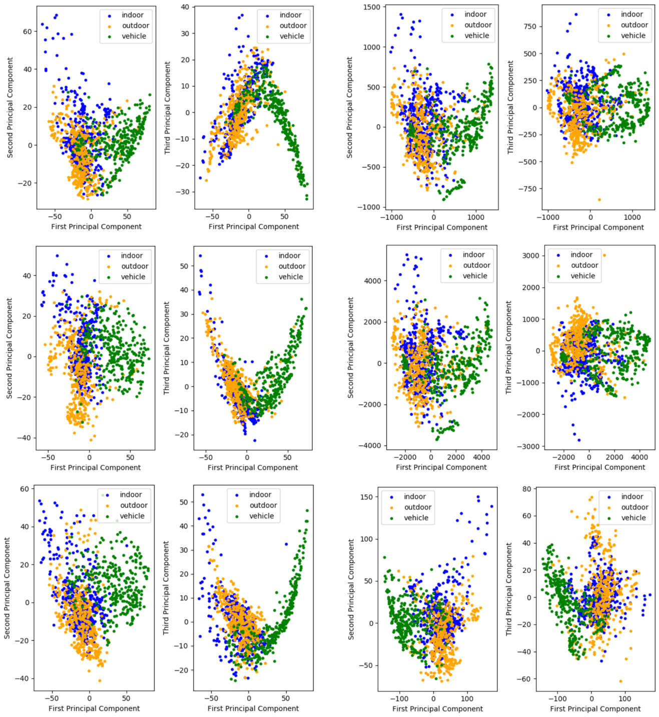


Fig. 7. Visual comparison of 2D CNN architectures using $n = 1, 2$ and 4 layers with a kernel $k = 2$ (on the left), and kernel size $k = 8, 16, 32$ (on the right with 2 layers) for the audio dataset.

with the visualization of data points in Fig.4, showing how a 4 layer model better separates activities in the feature space. Smaller values of kernel size correspond to lower accuracy values; conversely increasing the kernel size over 32, decreased accuracy. When using larger kernel sizes, 3 and 4 layers provide similar results, thus a 3 layers approach is preferable since it reduces model's complexity. It should be noted that data segmentation in this dataset has been performed with window size of approximately 2.5 seconds. With a sampling rate of 50 Hz the best performing kernel sizes (8 and 16) corresponds to 0.3 and 0.6 seconds approximately. Summarizing, increasing the number of layers helps to better separate inter-group variability between static (sitting, standing) and active labels (walking, walking upstairs and walking downstairs). On the other hand, increasing the kernel size helps to better separate data points intra-group for both active and static labels. The insight provided with visualization is confirmed by recognition performance of models measured using precision, recall and F-score. Results confirm that CNN automatic features are able to provide accuracy performances comparable with best known set of HCF, and are in line with performances measured in [1]. In this work, classification using only the accelerometer has also been evaluated. In this case CNN features provided higher precision and recall compared to HCF. When considering both accelerometer and gyroscope, HCF provide about 1-2% higher F-score, although that is including frequency domain features.

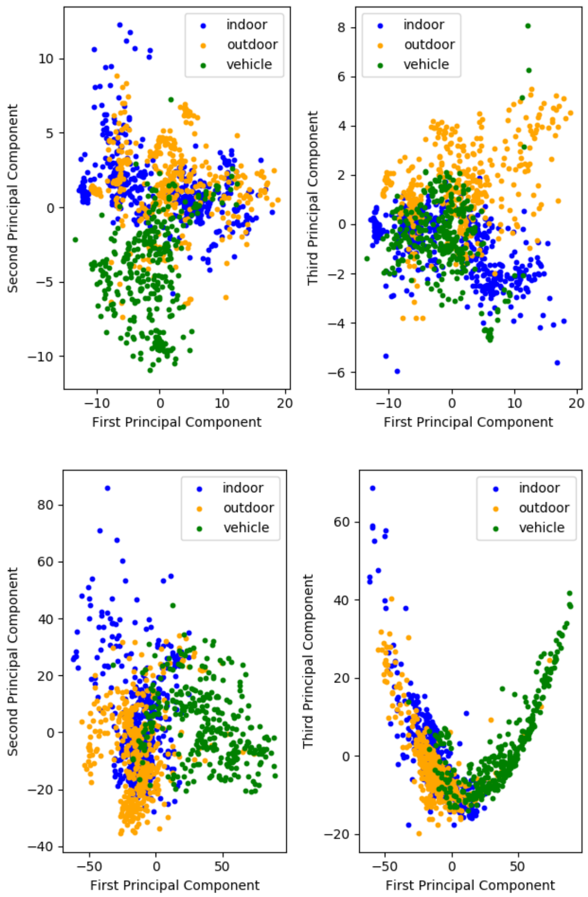
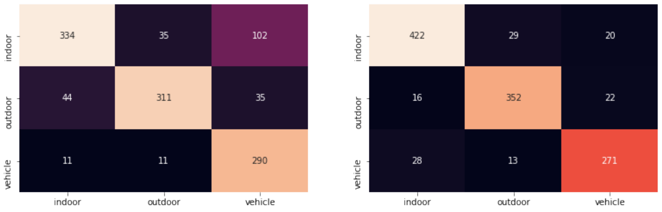


Fig. 8. Visual comparison of HCF (top) and CNN using 2 layers (bottom) for the audio sensor.



(a)

(b)

Fig. 9. Un-normalized confusion matrices obtained using (a) HCF and (b) using CNN for the audio dataset.

B. Audio CNN Features

Good recognition accuracy can be obtained using only two convolutional layers, followed by max-pooling. For the 2D CNN architectures, increasing the kernel size, while keeping a relatively shallow network (two layers), decrease the recognition accuracy performance. The network performance would increase by stacking more convolutional layers, thus increasing the complexity of the model. Furthermore, experiments show that the kernel size of the 2D CNN should be small, in order to capture all the details in the time and frequency domain. Fig.8 shows that a 2-layer 2D CNN can distinguish the three target classes after being trained for 22 epochs from raw spectrogram images. The top part of the figure depicts the human crafted MFCC features. When visualizing the first and third principal components we notice that there is not a clear distinction between the classes. Confusion matrices in Fig.9 show that the CNN can outperform HCF for indoor and outdoor settings. However, for the selected dataset, HCF achieved better classification accuracy in the vehicle environment. This probably occurred since the MFCCs are not robust to noisy environments. The results were promising, especially for training deep networks on device. To ensure privacy of sensitive audio data, networks should be light-weight and able to capture data and adapt (re-train) on the embedded system, ensuring no information is stored on the cloud.

VII. Conclusion

In this work, an analysis of performance of CNN extracted features has been presented. The experiment focused on comparison of automatically extracted and HCF for activity recognition. In particular, the audio signal, accelerometer and gyroscope data have been investigated. Moreover, the effect of important parameters has been evaluated, namely number of convolutional layers, and kernel size used for the convolution. Automatically extracted features achieved comparable results with the HCF on the UCI-HAR dataset. Furthermore, the automatically extracted features of the 2D CNN from the raw-spectrogram outperformed the HCF on the DCASE 2017 development dataset. On the one hand, it must be considered that using CNN features provides a standard way for feature extraction, simplifying the process compared to the human crafted case. On the other hand, a CNN used as feature extractor requires an initial training phase in order to generate good features (cold-start problem). The experiments provide insight on CNN feature performances; however, further work should evaluate performance of CNN, on large real-world datasets. Next steps will include experiments cross-validating a pre-trained CNN feature extractor on different datasets, with different sets of target activities.

Acknowledgment

This research has received funding under the ACROSSING project Marie Skłodowska-Curie EU Framework for Research and Innovation Horizon 2020, under Grant Agreement No. 676157.

References

[1] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[2] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.

[3] S. Blackman, C. Matlo *et al.*, "Ambient Assisted Living Technologies for Aging Well: A Scoping Review," *Journal of Intelligent Systems*, vol. 25, no. 1, pp. 55–69, 2016.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015.

[5] Q. Zhu, Z. Chen, and Y. C. Soh, "A Novel Semi-supervised Deep Learning Method for Human Activity Recognition," *IEEE Transactions on Industrial Informatics*, vol. PP, no. c, p. 1, 2018.

[6] J. Morales and D. Akopian, "Physical activity recognition by smartphones, a survey," *Biocybernetics and Biomedical Engineering*, vol. 37, no. 3, pp. 388–400, 2017.

[7] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors*, vol. 17, no. 3, 2017.

[8] M. Espinilla, J. Medina *et al.*, "Human Activity Recognition from the Acceleration Data of a Wearable Device. Which Features Are More Relevant by Activities?" *Proceedings*, vol. 2, no. 19, p. 1242, 2018.

[9] F. Li, K. Shirahama, M. A. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 2, pp. 1–22, 2018.

[10] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 1, no. June, pp. 1–33, 2014.

[11] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, "Simple and Complex Activity Recognition through Smart Phones," *2012 Eighth International Conference on Intelligent Environments*, no. July 2017, pp. 214–221, 2012.

[12] E. Zdravevski, P. Lameski *et al.*, "Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering," *Ieee Access*, vol. 5, pp. 5262–5280, 2017.

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

[14] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2002, pp. 1941–1944.

[15] A. J. Eronen, V. T. Peltonen *et al.*, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

[16] M. Perttunen, M. Van Kleek, O. Lassila, and J. Riekki, "Auditory context recognition using SVMs," in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08.* IEEE, 2008, pp. 102–108.

[17] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012.

[18] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[19] E. M. Grais, H. Wierstorf, D. Ward, and M. D. Plumbley, "Multi-resolution fully convolutional neural networks for monaural audio source separation," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2018, pp. 340–350.

[20] V. Morfi and D. Stowell, "Deep learning for audio event detection and tagging on low-resource datasets," *Applied Sciences*, vol. 8, no. 8, p. 1397, 2018.

[21] A. Mesaros, T. Heittola *et al.*, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[22] S. Hershey, S. Chaudhuri *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[23] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.

[24] S.-J. Huang, W. Gao, and Z.-H. Zhou, "Fast multi-instance multi-label learning," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[25] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[26] M. Abadi, A. Agarwal *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

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

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations (ICLR-15)*, 2015.