Smart Phone Based Human Activity Recognition

Hongkai Chen
School of Computing
Queen's University
Kingston, ON, Canada
14hc27@queensu.ca

Sazia Mahfuz
School of Computing
Queen's University
Kingston, ON, Canada
sazia.mahfuz@queensu.ca

Farhana Zulkernine
School of Computing
Queen's University
Kingston, ON, Canada
farhana.zulkernine@queensu.ca

***Abstract*— Human Activity Recognition (HAR) is a field that uses collected data to classify different human actions. One simple and general approach to HAR is to use the sensor data from a mobile device to recognize different patterns behind complex motions. Early studies show promising results on simple activities using manually selected features from accelerometer readings. As newer publicly available datasets include more complex data and activities, manual feature selection have become cumbersome, impractical and face limitations in finding the optimal feature sets for HAR. In this paper, we present an empirical approach to defining models of 3D tensor data structures from 2D time series data obtained from multiple sensors on a smart phone, and a new Convolutional Neural Network (CNN) model, which uses the tensor data and performs automatic feature extraction and classification for HAR. We use the public benchmark dataset, MobiAct v2.0, to train and validate our model, which achieved an overall better performance in classifying 11 Activities of Daily Living (ADL) than the state-of-the-art approaches. Compared to the approach presented by Chatzaki et al. which has a very high rate of misclassifications for car-step out (CSO), car-step in (CSI), sit to stand (CHU), and stand to sit (SCH) classes, our proposed approach has 15% higher sensitivity for each of these activities with the optimal number of training epochs being only 25.**

Keywords— Pattern recognition, AI, CNN, IoT, Human Activity Recognition

I. INTRODUCTION

The desire to understand human activities using electronic devices emerged in the late 90s [1]. After more than one decade of technology development, the demand for designing and constructing a portable, unobtrusive and inexpensive data acquisition system is no longer a challenge by virtue of the rise of affordable consumer-grade wearable devices and multisensory integrated smart phones [7][10][11][19]. In recent years, innovations in microelectronics and computer systems have facilitated rapid consumer-grade growth in electronic devices [15]. Tiny devices having high computational power, embedded sensors and affordable pricing have revitalized the research in many long-established fields such as Human Activity Recognition (HAR) [6][12][17].

Accurate comprehension of human activities is an essential part of ubiquitous health monitoring [7]. Many chronic diseases are influenced by people's long-term behaviors and physical activities. Traditionally, it is laborious for a health worker to give suggestions by monitoring patients with chronic diseases. It is also impractical for a patient to be aware of the unhealthy behaviors. Besides medical tests, a patient's self-observation is the only source of information to be provided for a doctor to understand the patient's living habits. Nonetheless, human memory is not always reliable, and people are often unaware of their health condition until the early symptoms start showing up. Furthermore, nearly all internet giants provide advertising through software using recommender systems, which can greatly benefit from the knowledge of users' lifestyles or activities throughout the day. Currently most recommender systems only rely on internet browsing history and search keywords that partially acquire the thoughts of users when they actively use the internet. Ubiquitous activity recognition through the data collected using smart phones can provide detailed information about users' lifestyles to refine the recommender systems.

HAR focuses on comprehending various human movements performed in a continuous series of actions as a human activity and its consequences. The essence of activity recognition using wireless devices is a pattern classification problem with hybrid streaming data [8][14]. However, there are still ongoing challenges that need to be addressed: (1) the selection of sensor data, (2) extraction of relevant feature set, (3) choice of the classifier, (4) general applicability of pre-trained classification models, and (5) power limitations of mobile devices.

Although accelerometer data has been recognized by early research on HAR [4][6][8] to provide the most informative feature sets for classification, recent studies propose that the gyroscope and magnetometer sensors may further increase the accuracy. However, utilization of more sensors also offers new challenges to feature selection and data processing. The second challenge, extracting right features from the data, contributes directly to the classification accuracy. Vavoulas et al. [2] and Pires et al. [4] demonstrate that simply optimization of existing feature sets can greatly increase classification accuracy for the same model. The third challenge has been addressed by many researchers for a variety of datasets, data acquisition and processing methods, and classification models, and are therefore, difficult to compare to decide about the best classifier. For instance, the MobiAct research [2] finds that the K-Nearest Neighbors classifier (kNB) has overall best accuracy while Pires et al. [4] did not consider using kNB for a comparable dataset. The fourth challenge questions the practicality of using pre-trained classifiers as they perform poorly with noisy data from real life environments [11][12]. Also it is infeasible and costly to use professional-grade sensors as used by some of the older studies with large sample sizes and researchers have begun to use off-the-shelf devices [1][15]. This leads to the fifth challenge of power limitations of wireless mobile sensor devices requiring optimal use of devices, frequency of readings and efficient

978-1-7281-1867-3/19/$31.00 ©2019 IEEE
978-1-7281-1867-3/19/$31.00 ©2019 IEEE

processing of high volume and velocity of hybrid streaming data. For commercialization and wide scale application of HAR, further study is needed to innovate a feasible state-of-the-art solution.

The inconsistency between two similar research papers, the MobiAct research [2] and the ALlab research [7], stimulated our interest in finding an optimal feature set and an optimal human activity classification method. In this paper, we present a novel HAR approach by designing several new models of tensor data structures to develop and train a Convolutional Neural Network (CNN) for automatic feature extraction and classification of human activity. Using extensive empirical study, we find the optimal tensor structure to integrate hybrid multi-modal accelerometer, gyroscope, and magnetometer readings, which is then fed into the CNN model. Previous studies typically used manual extraction of feature sets for predefined window sizes [1][2][14], where more than two-thirds of the features were extracted from the accelerometer readings, and some studies never examined the magnetometer readings [5][8]. We illustrate the feasibility of using the gyroscope and magnetometer readings with the accelerometer readings to improve the overall accuracy using the public benchmark dataset MobiAct v2.0 [4]. Several variants of the tensor models are proposed and thoroughly assessed to find the optimal model configuration. Our tensor and CNN models achieved an overall better performance in classifying 11 Activity of Daily Living (ADL) than the state-of-the-art approaches and 15% higher sensitivity compared to previous approaches [5] in classifying car-step out (CSO), car-step in (CSI), sit to stand (CHU), and stand to sit (SCH). Furthermore, our proposed approach only uses 25 training epochs.

The remaining parts of the paper include the following. An extensive discussion and comparison of related work is presented in Section II. Section III describes our approach, dataset used in our study, and the classification model. The results from the experimental studies are presented in Section IV. Finally, Section V concludes the paper with a discussion of future work.

II. Related Work

In this section, we review the previous work on HAR using smart phones and analyze the limitations in these studies. Table I shows the labels of 14 different types of ADL and 4 different types of falls that are frequently used in the HAR research.

TABLE I. Labels for Human Activities

LABEL	ACTIVITY	DESCRIPTION
STD	Standing	Standing with subtle movements
WAL	Walking	Normal walking
JOG	Jogging	Jogging
JUM	Jumping	Continuous jumping
STU	Stairs up	Stairs up
STN	Stairs down	Stairs down
SCH	Stand to sit	Transition from standing to sitting
SIT	Sitting on chair	Sitting on a chair with subtle movements
CHU	Sit to stand	Transition from sitting to standing
CSI	Car-step in	Step in a car
CSO	Car-step out	Step out a car
LYI	Lying	Activity taken from the lying period after a fall
FOL	Forward-lying	Fall Forward from standing, use of hands to dampen fall
FKL	Front-knees-lying	Fall forward from standing, first impact on knees
BSC	Back-sitting-chair	Fall backward while trying to sit on a chair
SDL	Sideward-lying	Fall sideward from standing, bending legs
DRI	Driving	Driving
SLP	Sleeping	Sleeping in the bed

Vavoulas et al. [2] compared selected computational methods to assess accuracy and performances of HAR for fall detection using a MobiFall dataset. The authors manually extracted two optimized feature sets based on their previous study, majority of which were extracted from the accelerometer data except a few that were extracted from the gyroscope and the orientation data. The results showed that kNB achieved a promising result with the highest accuracy of 97.1%. Moreover, the dataset included the subjects' height and body weights, but these were not considered in the feature extraction process. Ten of the subjects had a significantly higher body mass index (BMI) but the factor of overweight was not mentioned in the paper.

Later Vavoulas et al. [3], introduced a benchmark MobiAct dataset containing many ADLs, fall and user data. The authors proposed and tested with 3 different manually extracted feature sets, one from their earlier study using the MobiFall dataset [2], another from WISDM dataset used by Kwapisz et al. [6], and a third set created from the above two feature sets, where the latter provided the most promising results. Majority of the features were extracted from the accelerometer data with a few from the gyroscope and the magnetometer data. Evaluations of assorted classifiers such as kNB, J48 decision tree, Logistic regression and Multilayer perceptron, were carried out with the third feature set to determine the optimal classifier for HAR. The results illustrated that kNB had a comparatively better performance with 99.88% accuracy in classifying six basic ADLs.

Chatzaki et al. [5] extended the MobiAct dataset [2] by adding new user data and 3 more ADLs, and created an optimal reduced feature set to check if the performance and accuracy of two of the earlier best performing classifiers [2] could be maintained for the extended dataset. The kNB classifier still achieved a promising result with the highest accuracy of 97.1%. But the accuracies of recognizing ADLs including Chair-up (CHU), Front-knees-lying (FKL), Forward-lying (FOL), Sit-on-chair (SCH) and Sideward-lying (SDL) were lower than 80%. Frequent misclassifications were observed for CHU and SCH (22.8%) as well as for FKL and FOL (10.66%). In the manually extracted feature set, accelerometer data contributed to a larger portion of the feature set than the gyroscope and orientation data. Artificial neural network (ANN) models were not explored in these studies. We used this extended MobiAct 2.0 dataset from Chatzaki et al. [5] and defined an ANN model as a feature extractor and classifier.

Pires et al. [4] focused on the identification of ADLs using fusion of data from accelerometer, gyroscope and magnetometer on a smart phone. They extracted ten feature sets after cleaning the data where each succeeding set contained more features than the preceding set. Three variances of ANNs were implemented for classification: a multi-layer perceptron with backpropagation learning using the Neuroph framework, a feed forward ANN with backpropagation learning using the Encog framework, and a deep ANN or DNN using DeepLearning4j. The DNN with normalized data always presented a higher accuracy (80%) than the other classifiers. However, the use of a new dataset with only ANNs makes it difficult to compare this work with other research.

Tahavori et al. [7] explored HAR for elderly people with Parkinson's disease during mobility test using an accelerometer and a gyroscope. The authors performed feature selection using the WEKA toolkit and examined several machine learning methods including Naïve Bayes, LogitBoost, Random Forest and Support Vector Machine (SVM) to select the most reliable classifier. Random Forest had the highest average accuracy (92.29%) when both accelerometer data and gyroscope data were used. While this paper illustrates that using both sensor data concurrently can improve the accuracy of HAR, it addresses a specific group of population and ANN is not considered in this study. Also, the dataset is not available for the public.

Zeng et al. [9] developed a CNN to automatically extract features for activity recognition without any domain knowledge using mobile sensors. The authors developed a novel architecture to treat each axis of the accelerometer as one channel of the RGB image. The CNN was applied separately to each data channel to construct a feature set with strong local dependencies. The max-pooling layer proved to be capable of preserving scale invariance, which is crucial for real-world implementations. The CNN partial weight sharing model had the highest classification accuracy of 88.19%, 76.83%, 96.88% on the publicly available Skoda, Opportunity, and Actitracker datasets respectively, which is 4.41%, 1.2%, 9.02% higher than the PCA-ECDF (Principal Component Analysis - Empirical Cumulative Distribution Function) algorithm. Several techniques were suggested for improvement including weight decay, momentum and dropout to solve existing weaknesses of CNN, such as over-fitting and local optimum. Despite the differences between the data acquisition methods of the Actitracker dataset (a public version of the WISDM dataset previously used by this group) used in this research and the MobiAct dataset, the accuracy obtained by Chatzaki et al. using manual feature extraction is comparable, while the CNN allows automatic extraction of the feature set. Therefore, we wanted to explore using CNN with the more popular MobiAct dataset for multiple sensors on the smart phone for HAR.

Hammerla et al. [10] developed and experimented with a feed-forward DNN, a CNN and a recurrent neural network (RNN) classifier for HAR across the Opp, PAMAP2 and DG datasets that contain movement data captured with wearable sensors. Besides illustrating a variety of optimization and regularization techniques to improve the classification models, they explored two variations of RNNs, a deep forward LSTMs and a bi-directional LSTMs. The authors also demonstrated a novel training method for the RNNs for varying sliding window sizes to facilitate the selection of the window size. The CNN had the highest accuracy of 93.7% for the PAMAP2 dataset which has comparable data types as the MobiAct dataset. The LSTM and b-LSTM showed the most promising results of 76% and 92.7% accuracy for the DG and OPP datasets. The research reported that CNNs proved to be more suitable for prolonged and repetitive activities while the RNNs had better accuracies for activities having short duration but a natural ordering.

Yang et al. [16] proposed using a CNN to automatically extract discriminative features from the raw inputs by leveraging the labels in two datasets. The first dataset, Opportunity Activity Recognition, was collected using 15 wireless and wired devices on a single subject. The total number of sensor readings in this dataset was 72 of 10 modalities that is much larger than the MobiAct dataset, which used one mobile device containing multiple sensors to collect data from each subject. The second dataset, Hand Gesture, included data from a three-axis accelerometer and a two-axis gyroscope. The window sizes used in this study for both datasets were 1 second without overlap. The CNN composed of three convolution layers, two subsampling layers, and one fully connected layer in sequence. The results showed that the CNN had at least 5% improvement in accuracy for each activity classification over the best baseline method for both with and without the smoothing settings. Although the datasets used in this study are different from the MobiAct dataset and contains more data, the success of CNN motivated us to use CNN with the MobiAct dataset in this study to explore automatic feature extraction.

Mahfuz et al. [21] and Ajerla et al. [22] mainly focused on the fall activity in the MobiAct dataset. They manually processed the raw data from the accelerometer only to create the feature set to feed into a deep multilayered feed forward network and LSTM classifier to detect fall. Compared to their work, in this work we design a more efficient approach to classify 11 ADL activities excluding fall activities. We propose a CNN model that uses a 3D tensor model containing partially processed data as input, automatically extracts the key features and performs the classification.

A comparison of the 6 commonly used HAR datasets are given in Table II to show the differences since the datasets influence the classifiers and the technology in general.

TABLE II. SIX PUBLIC HAR DATASETS' COMPARISON

Dataset Name	Activities	Sampling Frequency	Sensors	No of Subjects	Dataset Size	File Format
MobiAct v1.0	STD, WAL, JOG, JUM, STU, STN, SCH, CSI, CSO, FOL, FKL, BSC, SDL	100Hz	Accelerometer, Gyroscope, Magnetometer	57	1.24GB	.txt
MobiAct v2.0	STD, WAL, JOG, JUM, STU, STN, SCH, SIT, CHU, CSI, CSO, LYI, FOL, FKL, BSC, SDL	200Hz	Accelerometer, Gyroscope, Magnetometer	67	2.28GB	.csv
ALlab	STD, WAL, JOG, STU, STN, DRI, SLP	100Hz	Accelerometer, Gyroscope, Magnetometer, Microphone, GPS	25	60.1GB	.txt
SisFall	STD, WAL, JOG, JUM, STU, STN, SCH, SIT, CHU, CSI, CSO, LYI, FOL, FKL, BSC, SDL	200Hz	Accelerometer, Gyroscope	15	0.72GB	.txt
WISDM v1.1	STD, WAL, JOG, JUM, STU, STN	10Hz	Accelerometer	26	0.05GB	.txt
UCI HAR	STD, WAL, STU, STN, SIT, LYI	50Hz	Accelerometer, Gyroscope	30	0.27GB	.txt

III. IMPLEMENTATION

In this section we describe the dataset, data processing and the development of the classification model for HAR. First, we split the dataset into training and testing data. Next after necessary preprocessing, we converted the 2D MobiAct time series data containing the timestamp and the X, Y and Z-axis readings for the 3 smart phone sensors namely, accelerometer, gyroscope and magnetometer, into a 3D tensor. The tensor data is then fed into a CNN model to train a HAR classifier.

A. Dataset

We used MobiAct v2.0 HAR dataset from Biomedical Informatics and eHealth Laboratory (BMI lab) in our study as it has been used widely in previous research [5]. It includes 12 different ADLs and 4 types of falls from 67 subjects. There are more than 3200 trials in the dataset. In addition, the dataset includes 5 predesigned real life scenarios involving multiple ADLs of 2 to 3-minutes duration, which can be used to train and test HAR classifiers. We excluded the 4 fall related activities and the LYI activity, which is an activity taken after a fall from this study as we specifically addressed fall detection in our prior studies [21][22].

We explored several other datasets to use in our study. We wanted to use multi-modal data from multiple sensors but the majority of HAR datasets did not record magnetometer readings except the MobiAct and the ALlab dataset [19]. In ALlab dataset, the label for each record is selected by the subject and much of the data include information about the living place besides human activity information. Furthermore, timestamps and frequencies differed for each sensor which made it difficult to integrate data from multiple sensors for each record and the seven human activities in the ALlab dataset were relatively simple compared to the MobiAct dataset. Accordingly, we chose to use MobiAct dataset.

B. Environment

The implementation environment used in our research is discussed here. We used the Ryzen 5 2600k as our CPU; DDR4 16G as the RAM; RTX 2070 8G as the GPU; Windows 10 as the operating system; Anaconda as the software platform; and Python 3 as the programming language.

Based on the comparative study of different deep learning software tools by Kovalev et al. [13], we chose TensorFlow due to its wide spread use, relatively low complexity and training time, fast testing speed, support for GPU and scalability. Python 3 was used with NumPy to develop the CNN model [13]. Pandas, the Python library, was used for advanced data manipulation, analysis, spreadsheet-style file input/output, and dataset integration [25]. NumPy array data structure was used to significantly speed up high complexity matrix computations by reducing memory allocations and usage of computational power [24]. Anaconda was used on Windows 10 operating system to set up the above tools and Anaconda Accelerate was used to get more options to optimize the environment to use Intel CPUs and NVIDIA GPUs [23].

There are 12 columns in the dataset. As shown in Fig. 1, the *timestamp* column in the dataset contains the raw values of time and date from the Android system API. Those values are too complicated to be read by a human, so the *rel_time* column, which is the calculated cumulative time for this single trial, was exploited as the timeline in the data processing step. The next nine attributes represent each axis of one of the sensors used in this study. The last column in Fig. 1, which uses the abbreviation of the activity using three characters, stands for the activity label. One very evident problem was that the maximum and minimum values for each type of sensor differ radically. Since the idea of this approach was to compose all the sensor readings together into one feature map, a suitable normalization equation was required. After a careful examination of the whole dataset, we determined the upper and lower bounds of each type of sensor as presented in Table III. Furthermore, most activities can contain other daily activities during the trial. A simple method was developed as described in the next section to solve the problem caused by the mixture of multiple labels in a single trial.

	A	B	C	D	E	F	G	H	I	J	K	L
1	timestamp	rel_time	acc_x	acc_y	acc_z	gyro_x	gyro_y	gyro_z	azimuth	pitch	roll	label
2	939491959000.00	0	0.553449	10.38883	-1.06807	-1.17775	0.769079	0.037263	320.7457	-96.1361	-5.7735	STD
3	939496910000.00	0.004951	0.508774	10.34415	-1.03829	-1.20188	0.784351	0.037568	320.5204	-95.9281	-5.64653	STD

Fig. 1. Sample raw data from MobiAct v2.0

TABLE III. MIN AND MAX VALUES OF EACH SENSOR

Sensor	Min	Max
Accelerometer	-20 *m/s^2*	20 *m/s^2*
Gyroscope	-180 *rad/s*	180 *rad/s*
Orientation	-360 *degrees*	360 *degrees*

C. Data Processing

To process the raw multi-sensor time series data from the MobiAct v2.0 dataset, we first applied a down sampling process to reduce the sampling frequency and then a linear interpolation to construct the new measure points in the raw data. The original sampling frequency of MobiAct v2.0 dataset was 200Hz, which was too high for HAR research. According to the survey on HAR using mobile phones by Shoaib et al. [15], 20Hz is sufficient, and a common and suitable sampling frequency for a HAR study. For this study, we used 25Hz, 50Hz, and 100Hz to test different approaches based on the structure of our feature map.

Despite the fact that some previous studies have defined their window sizes without using an overlap, the selection of various overlapping window sizes can still have an effect on the HAR classification accuracy [2][14]. Based on the other published work that used the MobiAct dataset [2][5], we chose to construct subsamples using suitable window sizes with 50% overlap to split the whole dataset into smaller sizes. According to Banos et al.'s study on finding the optimal window size for the HAR problem [14], a short window size can result in a nearly precise model. However, a short window size would also increase the total number of records and thereby, require greater computational power. We aimed to build a model with a medium window size, which would be between 2 to 5 seconds. Since the shortest durations of some types of activities were around 6 seconds, we decided to use a window size that was no more than 5 seconds.

The original label of the MobiAct v2.0 dataset was created for each instance. Splitting the time series data into windows could result in more than one class of activities pertaining to a selected window size. A method to define new labels for the subsamples i.e., data partitioned in a window, was needed. The label of each subsample was defined based on the dominance of each type of activity. After careful investigation, we found that, in most cases, there would be at most two different activities in a selected window size. In a very small number of subsamples, there could be more than three different activities in a subsample. Under the circumstances, the most frequent

activity for each new subsample was selected as the new label, which would represent the whole window size. In the cases where the rates of occurrence between two activities in a sample are very close, the subsample would be considered as noisy data and be discarded. We later found out that even without discarding these rare cases, our classifier performed well. Hence, these noisy data were kept in the final test.

We separated our processed datasets into training and testing datasets using a 4:1 ratio. One problem was to evenly separate the dataset by each ADL class. As shown in Fig. 2, the size of data for STD and WAL activities were significantly larger than that of the other activities and the SIT and CHU activities had the smallest data. Therefore, we separated the dataset into testing and training sets by each class instead of the entire dataset.

Unlike the manual feature extraction process used in many previous studies [2][4][4][2][6][7][8][21][22], we developed a CNN to reconstruct our subsamples. We normalized the raw signals to match the range of a single 8-bit byte, which is 0 to 255 using (1) based on the ranges of sensor data values as shown in Table III:

$$x = \frac{(x - x_{min})}{x_{max} - x_{min}} \times 255 \quad (1)$$

Fig. 2. Dataset size for each ADL class

D. Definition of Tensor Models for CNN

We constructed an input tensor of dimension 18*18*3, where '3' stands for the number of channels. If either the window size or the sampling frequency increases, the dimension of the input tensor would have to increase as well. For a better classification accuracy, the selected window size and the sampling frequency should be relatively small. For each sensor we could fit 324 (18*18=324) values. So a total of (324*3=972) 972 readings were collected from the 3 sensors to define the values of the 18*18*3 tensor. One additional value was included which indicated the class label. So for each subsample, we had 973 values in total. The first value indicated the label of the true activity while the rest of them represented a group of sensor inputs. Since to the best of our knowledge this is the first study that feeds normalized data directly into the neural networks (others feed either raw data or do normalization in the network), it is important to investigate an optimal tensor data structure. Therefore, we used different combinations of the three sensor data, each having 3 axes, to fill a predefined tensor data structure and created 8 tensor models to experiment with as shown in Fig. 3. A detailed performance comparison of these models is given in Section IV.

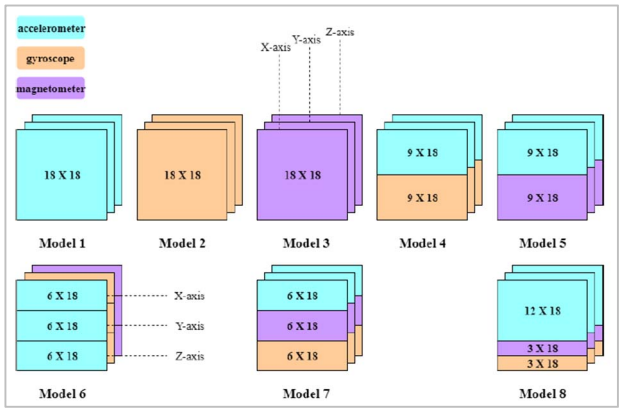


Fig. 3. Tensor data structures for 8 models

As shown in Fig. 3, the first three models were used as the baselines in this study. Each 18x18 matrix represents the sensor readings from one of the X, Y, and Z-axes. Each of the first three models exclusively covers the readings from only one type of sensor. For instance, only accelerometer readings were used in the first model. To fit the tensor (18*18=324), we used a window size of 3.24 seconds with 100Hz sampling frequency (3.24*100=324). Most of the recent studies heavily concentrate on the combination of accelerometer and gyroscope data, and magnetometer data is rarely used in HAR. Therefore, using models 1 to 3, we wanted to verify two hackneyed questions for HAR study: a) usefulness of only the magnetometer readings without accelerometer readings, and b) if accelerometer readings are adequate enough to completely solve the HAR problem. We also wanted to see if the combination of accelerometer and magnetometer would perform well together. Pires et al. [4] suggested that the magnetometer readings can be useful for HAR if used effectively. For these reasons, we developed model 4 and 5. In each channel, half of the matrices are filled with the values from accelerometer readings while another half of the matrices are loaded using either the gyroscope readings or magnetometer readings. The window size remains the same as the first three models while the sampling frequency is changed to 50Hz to maintain the same amount of values in the same matrix size for the two sensors.

The last three models include all the sensors in the dataset but have different structures. In model 6 as shown in Fig. 3, we loaded the same sensor X, Y and Z-axis readings into a single channel (6*18*3=324). For example, all accelerometer readings were in the first channel. In model 7, each channel was loaded using one-axis values from each sensor. As an illustration, the first matrix consisted of the accelerometer X-axis readings, the gyroscope X-axis readings, and the magnetometer X-axis readings. For both models 6 and 7, the window size used was 4.32 seconds and the sampling frequency was 25Hz (4.32*25=108 * 3 sensors=324). After a complete assessment of the first 7 models, we aimed to test the performance of our classification model with more weights given to the accelerometer readings. By lowering the sampling

frequency of the gyroscope and magnetometer, we also save the battery power of the smart phones. The window size for the accelerometer was once again 4.32 seconds and the sampling frequency was 50Hz (4.32*50=216), and the readings were used to fill the 3 tensor channels (12*18=216) with X, Y and X-axis values respectively. For the gyroscope and the magnetometer, we used a sampling frequency of 12.5Hz for the same 4.32 seconds (432*12.5=54) to fill each of the 3 channels (3*18=54) with X, Y and X-axis values respectively.

E. CNN Model

The convolutional neural network has been widely applied in computer vision research especially after the ground-breaking invention of AlexNet [17]. From the perspective of a computer, an RGB image file is, in fact, a tensor, where the size is 3*width*height. Since the computer cannot directly understand the image pattern, the inputs are essentially a series of numbers. Similarly, the values from our time series subsamples are used to construct 3D tensors that are fed into the CNN model as inputs. Each tensor in this new dataset is treated as an RGB image. This also provides a way to visualize the pattern of each activity without using an abstract format.

We created our CNN model based on the architecture proposed by Yang et al. [16] which had five sections. Each of the first two sections contained a convolution layer, a rectified linear unit (ReLU) layer, a max pooling layer, and a normalization layer. The convolution layer of a section performed a convolution operation on the outputs of the previous section or the raw inputs. The sequence of layers in each section reduced the total number of weight-variables in the network. The ReLU layer exploited an activation function to increase the non-linearity in the feature sets. The max pooling layer extracted the maximum feature map in a set of temporally local neighborhoods. The last layer normalized all the features based on the hyper-parameters prior to sending them to the next section. The third section had a similar structure without the max pooling layer since the dimensions of the feature maps from the convolution layer in the second section were already small enough. The fourth section included a custom fully connected layer followed by the ReLU layer and the normalization layer. The fifth section was a standard fully connected layer.

As shown in Fig. 4, the normalization layer was not implemented in our CNN architecture, since the input data in our experiments were already normalized in the data preprocessing stage. We also replaced the fourth section with a standard fully connected network layer due to the insufficient details for this proposed custom layer. The goal of this study is to propose a simple but powerful model for HAR problem.

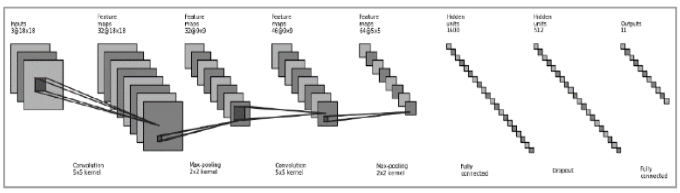
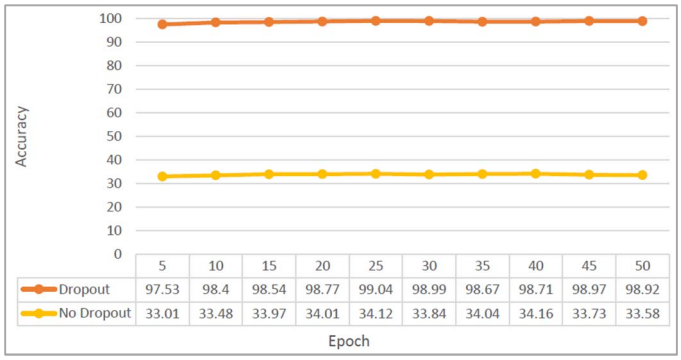


Fig. 4. Architecture of our CNN Model

According to Hinton et al. [18] and Park et al. [19], dropout layers can avoid the overfitting problem. Although the CNN architecture of Yang et al. did not include a dropout layer as the authors claimed a minor performance difference, we included a dropout layer and tested our CNN architecture with a dropout rate of 0, which indicates that none of the neural nets would be dropped in this case. For comparison, we have also assessed the performance of this architecture with a dropout rate of 0.5. For simplicity, we used model 7 as the only tensor data structure in this small experiment. According to Fig. 5, the accuracy of our CNN model drops significantly when the dropout layers are removed. As the size of dataset we used is much larger than those in the previous studies, the overfitting problem could lead to a poorer performance. Although Yang et al. [16] created the training and testing sets from the data containing all subjects, we focused on generating our training and testing sets to include each activity class and not all subjects. For the above reasons, the introduction of dropout layers in our CNN architecture shows notable improvements in classification accuracy.



	5	10	15	20	25	30	35	40	45	50
Dropout	97.53	98.4	98.54	98.77	99.04	98.99	98.67	98.71	98.97	98.92
No Dropout	33.01	33.48	33.97	34.01	34.12	33.84	34.04	34.16	33.73	33.58

Fig. 5. Overall accuracy of different CNN architectures

IV. Experiments and Results

In Table IV, we present the sensitivities of all ADL classes for each tensor model we discussed in Section III computed from the data collected from an intensive experimental study. Since the size of MobiAct v2.0 dataset is extremely large, for all models, the accuracies of most classes are over 98%, which makes it difficult to assess the performance of each tensor model. Chatzaki et al. [5] presented the confusion matrix as their final result, which included the True Positive (TP) rates. Therefore, we chose to present TP rates for each situation.

According to Table IV, model 7 is the most robust one. By examining the weighted average TP rates of tensor models 1, 2, and 3, we can conclude that when using only one of the sensors, the accelerometer reading has the highest discriminative power for HAR. Although tensor model 1 outperforms models 2 and 3 in most ADLs classifications, model 3 which contains only magnetometer readings, has better performance for SCH, CSI, and CSO classification. By comparing tensor models 6, 7, and 8, we observe that the variations in tensor data structures can affect the classification accuracy. The major difference between tensor models 6 and 7 is the local dependencies among different axes and sensors. For model 6, the extracted features would contain local dependency for each sensor while for model 7, extracted features would demonstrate local dependency among

different sensors. The three tensor models containing data from all 3 sensors have superior results in Table IV but from the weighted average of TP rates, tensor model 7 proves to be the optimal tensor data structure to be used with the CNN for HAR.

TABLE IV. TRUE POSITIVE (TP) RATES FOR EACH TENSOR MODEL FOR ALL ACTIVITIES

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
STD	99.95	98.13	98.99	99.95	99.95	99.94	99.94	99.94
WAL	99.82	99.24	99.28	99.28	99.96	99.88	99.94	99.64
JOG	97.43	98.72	96.95	99.68	99.84	99.26	99.99	99.50
JUM	98.87	97.27	94.06	99.68	99.52	96.77	99.50	99.99
STU	73.33	88.33	44.17	95.42	66.67	85.39	91.32	89.04
STN	70.12	76.76	49.38	87.55	82.16	94.06	96.35	86.30
SCH	97.26	90.41	99.99	99.99	82.19	97.26	97.26	93.15
SIT	96.24	28.57	90.23	98.50	93.99	96.84	94.73	95.79
CHU	82.60	39.13	78.26	73.91	95.65	91.30	86.96	95.65
CSI	81.94	81.94	90.28	97.22	99.99	95.83	98.61	97.22
CSO	90.27	79.17	93.06	99.99	95.83	99.99	98.61	93.06
Weighted Avg.	97.66	95.24	94.13	98.89	97.64	98.48	99.11	98.36

The results for each tensor model in Table IV were produced after training our CNN for 25 epochs. The overall accuracy and the weighted average of TP rates for each model could be slightly improved until we reached 25 epochs. After that we observed minor performance difference with the increase in the number of epochs. Because of this and due to constraints in computational power, we only tested the performance of tensor model 7 for up to 100 epochs as shown in Fig. 6, and it reached optimal overall accuracy after 25 epochs. However, since we only tested up to 100 epochs, it is inconclusive whether this accuracy represents a local optimum. However, in terms of the overall accuracy and sensitivity of classification for each activity, with only 25 epochs our tensor model 7 already outperformed the state-of-the-art results reported by Chatzaki et al. for the MobiAct v2.0 dataset [4] for most of the ADLs as shown in Fig. 7.

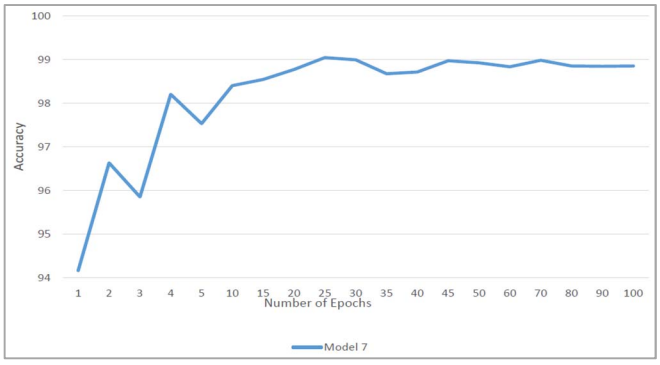


Fig. 6. Overall accuracy of model 7 after different number of epochs

V. CONCLUSION AND FUTURE WORK

We proposed a CNN model for HAR using hybrid multi-sensor multi-modal data from MobiAct v2.0 including the optimal tensor model to use as the input to the CNN that outperforms the state-of-the-art results. Raw 2D time series sensor data are partially processed, normalized and converted into 3D tensor models, which is then fed into the CNN. Instead of manually extracting the features, we designed the CNN to automatically extract features from the tensor model containing the data. Multiple tensor models were constructed to determine the optimal mix of sensor data that gives the highest average accuracy in terms of true positive rates. We designed a seven-layer neural network architecture based on a previous work by Yang et al. [7] but modified the architecture to exclude normalization layers and include dropout layers to achieve greater accuracy.

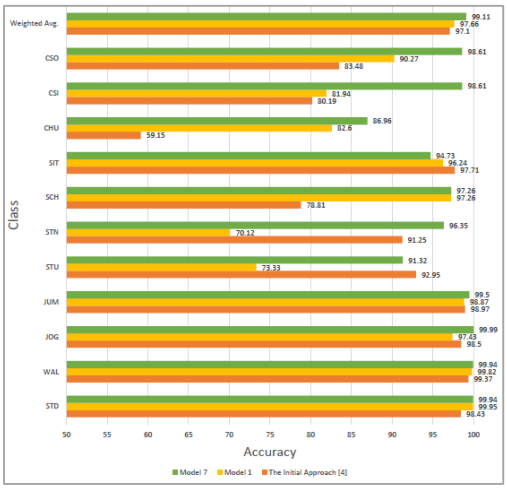


Fig. 7. Comparison of our classification results in % accuracy with the results published by Chatzaki et al. [5] for MobiAct v2.0

We illustrated the experimental evaluation of eight different tensor models and compared the performance of model 7, the optimal model found in our study, with that of Chatzaki et al. [4]. In terms of weighted average TP rate, six of our tensor models with the CNN outperform the state-of-the-art accuracies for HAR reported by Chatzaki et al. for the MobiAct v2.0 dataset. The optimal model 7 shows much better results in recognizing most of the activities. The proposed tensor models represent simple normalized raw data compared to other work that use hand-crafted feature sets as inputs to the classifiers. We down-sampled and normalized the raw data and converted that into 3D image-pattern-like data sets. As shown in Fig. 8, this approach can also significantly reduce the size of the data files for storage.

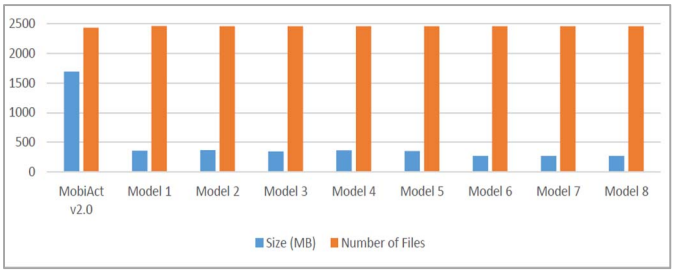


Fig. 8. Preprocessed dataset in comparison to the MobiAct v2.0

The proposed approach provides a solid foundation for multimodal time-streaming data analysis. Even though the initial goal was to solve the HAR problem, it is possible to extend this approach to any other time series data from the Internet of Things (IoT) to transform raw 2D signals into 3D image patterns. The size of the pre-trained model is less than 10 MB, which is suitable for mobile computing. Nonetheless, no real-time data has been tested with this model, so the

performance of this pre-trained model could suffer for noisy data in real-world environment.

Two strategies to further improve this CNN-based method are to increase the number of epochs and to enlarge the dimension of the inputs. Limitations in computational power prevented us from training the model for greater than 100 epochs in a reasonable time, so we could not confirm if the current 25 epochs for model 7 was a local optimum as well as if models 6 and 8 could be more accurate after more epochs. By using higher dimensions of tensor models, one can do more than just increase the sampling frequency. As an example, other type of sensors from smart phones, such as GPS, thermometer, ambient light sensor, and microphone, can also be added into the feature map. By creating a richer feature map, the scope of this study can be extended to more than the just 11 ADL classes and many details in daily life can be included. Thus the proposed approach can become an essential tool for future healthcare research and life style assessment.

One feature we did not exploit from the MobiAct v2.0 dataset is the biometric data of each subject. It would be intriguing to explore the relation between the biometric data and the activity patterns, which may be used to customize the classification model for the 37 subjects included in the dataset.

REFERENCES

[1] O. Lara, and M. Labrador. 2013. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, *15*(3), 1192-1209.

[2] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. Spanakis, and M. Tsiknakis. 2014. The MobiFall Dataset. *International Journal of Monitoring and Surveillance Technologies Research*, *2*(1), 44-56.

[3] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis. 2016. The MobiAct Dataset: Recognition of Activities of Daily Living using Smartphones. *Proceedings of The International Conference On Information and Communication Technologies for Ageing Well And E-Health*.

[4] I. Pires, N. Garcia, N. Pombo, F. Flórez-Revuelta, S. Spinsante, and M. Teixeira. 2018. Identification of activities of daily living through data fusion on motion and magnetic sensors embedded on mobile devices. *Pervasive and Mobile Computing*, *47*, 78-93.

[5] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis. 2017. Human Daily Activity and Fall Recognition Using a Smartphone's Acceleration Sensor. *Communications in Computer and Information Science,* 100-118. DOI: 10.1007/978-3-319-62704-5_7.

[6] J. Kwapisz, G. Weiss, and S. Moore. 2011. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, *12*(2), 74.

[7] F. Tahavori, E. Stack, V. Agarwal, M. Burnett, A. Ashburn, S. Hoseini Tabatabaei, and W. Harwin. 2017. Physical activity recognition of elderly people and people with parkinson's (PwP) during standard mobility tests using wearable sensors. *2017 International Smart Cities Conference (ISC2)*.

[8] I. Pires, N. Garcia, N. Pombo, F. Flórez-Revuelta, and S. Spinsante. 2017. Pattern Recognition Techniques for the Identification of Activities of Daily Living using Mobile Device Accelerometer. Retrieved from http://arXiv:1711.00096

[9] M. Zeng, L. Nguyen, B. Yu, O. Mengshoel, J. Zhu, P. Wu, and J. Zhang. 2014. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. *Proceedings of International Conference On Mobile Computing, Applications and Services*. MOBICASE IEEE. DOI: 10.4108/icst.mobicase.2014.257786

[10] N. Hammerla, S. Halloran, and T. Plotz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. *Proceeding IJCAI'16 Proceedings of The Twenty-Fifth International Joint Conference On Artificial Intelligence*, 1533-1540.

[11] R. Chavarriaga, H. Sagha, A. Calatroni, S. Digumarti, G. Tröster, J. Millán, and D. Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, *34*(15), 2033-2042.

[12] A. Reiss, and D. Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. *2012 16Th International Symposium On Wearable Computers*.

[13] V. Kovalev, A. Kalinovsky, S. Kovalev. 2016. Deep Learning with Theano, Torch, Caffe, TensorFlow, and deeplearning4j: which one is the best in speed andaccuracy? *In: XIII Int. Conf. on Pattern Recognition and Information Processing*, 99-103.

[14] O. Banos, J. Galvez, M. Damas, H. Pomares, and I. Rojas. 2014. Window Size Impact in Human Activity Recognition. *Sensors,14*(4), 6474-6499.

[15] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga. 2015. A Survey of Online Activity Recognition Using Mobile Phones. *Sensors,15*(1), 2059-2085.

[16] J. Yang, M. Nguyen, P. San, X. Li, and S. Krishnaswamy. 2015. Deep Convolutional Neural Networks On Multichannel Time Series for Human Activity Recognition. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015),* 3995-4001.

[17] A. Krizhevsky, I. Sutskever, and G.E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60(6)*, 84-90.

[18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *Technical report, arXiv:1207.0580*.

[19] S. Park, and N. Kwak. 2017. Analysis on the Dropout Effect in Convolutional Neural Networks. *Computer Vision – ACCV 2016 Lecture Notes in Computer Science,*189-204.

[20] I. Pires, N. Garcia, N.Pombo, F. Flórez-Revuelta, and S. Spinsante. 2018. Approach for the Development of a Framework for the Identification of Activities of Daily Living Using Sensors in Mobile Devices. *Sensors,18*(2), 640.

[21] Mahfuz, S., Isah, H., Zulkernine, F., Nicholls, P., 2018. "Detecting Irregular Patterns in IOT Streaming Data for Fall Detection", *IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, BC, Canada, IEEE.

[22] D., Ajerla, S., Mahfuz, Zulkernine, F., 2018. "Fall Detection from Physical Activity Monitoring Data", Intl. SIGKDD workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications (BigMine) at Intl. Conf. on Knowledge Discovery and Data Mining (KDD), London, UK, ACM.

[23] Anaconda Accelerate. Retrieved from https://docs.continuum.io/accelerate/index.html

[24] S.V. Walt, S.C. Colbert, and G. Varoquaux. 2011. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering, 13*(2), 22-30.

[25] F. Nelli. 2015. The pandas Library—An Introduction. *Python Data Analytics*, 63-101.