
Hierarchical Signal Segmentation and Classification for Accurate Activity Recognition

Ali Akbari

Department of Biomedical
Engineering
Texas A&M University
aliakbari@tamu.edu

Jian Wu

Department of Computer
Science and Engineering
Texas A&M University
Jian.wu@tamu.edu

Reese Grimsley

Department of Electrical and
Computer Engineering
Texas A&M University
reesul@tamu.edu

Roozbeh Jafari

Department of Biomedical
Engineering, Computer Science
and Engineering, and Electrical
and Computer Engineering
Texas A&M University
rjafari@tamu.edu

Abstract

The objective of this work is to determine various modes of locomotion and in particular identify the transition time from one mode of locomotion to another as accurately as possible. Recognizing human daily activities, specifically modes of locomotion and transportation, with smartphones provides important contextual insight that can enhance the effectiveness of many mobile applications. In particular, determining any transition from one mode of operation to another empowers applications to react in a timely manner to this contextual insight. Previous studies on activity recognition have utilized various fixed window sizes for signal segmentation and feature extraction. While extracting features from larger window size provides richer information to classifiers, it increases misclassification rate when a transition occurs in the middle of windows as the classifier assigns only one label to all samples within a window. This paper proposes a hierarchical signal segmentation approach to deal with the problem of fixed-size windows. This process begins by extracting a rich set of features from large segments of signal and predicting the activity. Segments that are suspected to contain more than one activity are then detected and split into smaller sub-windows in order to fine-tune the label assignment. The search space of the classifier is narrowed down based on the initial estimation of the activity, and labels are

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

UbiComp/ISWC'18 Adjunct, October 8–12, 2018, Singapore, Singapore
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5966-5/18/10 \$15.00
<https://doi.org/10.1145/3267305.3267528>

assigned to each sub-window. Experimental results show that the proposed method improves the F1-score by 2% compared to using fixed windows for data segmentation. The paper presents the techniques employed in our team's (The Drifters) submission to the SHL recognition challenge.

Author Keywords

Activity recognition; modes of locomotion; motion sensors; deep neural network; adaptive segmentation

ACM Classification Keywords

• Human-centered computing~Ubiquitous and mobile computing • Computing methodologies~Supervised learning

Introduction

Recognizing daily human activities, specifically detecting modes of locomotion and transportation and identifying the transition time from one mode to another, with motion sensors is gaining a bold traction for mobile computing. Many critical applications, such as human-computer interaction, medical services, and context-aware services depend on accurate recognition of modes of locomotion and transportation. In addition, detecting transitions from one mode to another as accurately as possible can empower applications to react in a timely manner. For accurate detection of those human activities and identification of the transition between them, appropriate data segmentation and feature extraction is a crucial step. Activity recognition algorithms usually segment the sensor signals into fixed-size windows for the successive features extraction and classification, and they assign a single label to all samples within a window [1]. Various fixed window sizes have been utilized for signal segmentation targeting specific

activities. An optimum window size depends on the characteristics of activity signals. Intuitively, splitting the whole data into large windows can potentially improve the performance of activity recognition compared to shorter ones as larger windows may include richer information about the activity signal. However, it increases misclassification when signals of two different activities fall into a single window because the classifier assigns only one label to the whole window; this happens at the moment of transition between the activities. Contrary to this, using small window size improves the ability to track the transitions quickly, but it might cause more misclassification, as some critical patterns may not be captured by short windows [2]. Considering this tradeoff, researchers usually determine the best window size by empirical methods and then keep it fixed through their analysis.

To deal with the challenges above, we propose a hierarchical framework to incorporate multiple window sizes for recognizing modes of locomotion and transportation and detecting the transition time from one mode to another as accurately as possible by using a smartphone. We first propose a set of discriminative features to distinguish between modes of locomotion including walk, run, bike, car, bus, train, subway, and still. The feature set includes 1) orientation-independent features, 2) generic time domain features suitable to detecting simple modes of locomotion, such as walking, that have a clear pattern of the signal, and 3) specific frequency domain features to distinguish between challenging modes of motorized transportation that do not have a clear and distinct pattern of the signal in the time domain. A hierarchical approach is utilized to incorporate different window sizes for feature extraction and classification in order to benefit from their strengths. Large window sizes allow more

informative features to be extracted whereas small window sizes allow the transition between activities to be captured more precisely. The process begins by using a large segment of the data and recognizing the target class label using a primary classifier; then in the case that a transition is suspected, each segment is divided into smaller sub-windows, and the label is detected for each sub-window by another classifier. The classifier that runs on the sub-windows looks for a limited number of classes based on the decision made by the primary classifier. The experimental results show that this technique improves the performance of the system compared to using a single fixed window size for data segmentation.

Background

Detecting modes of locomotion and transportation can be considered as a particular case of activity recognition, which is a widely studied field within the wearable and ubiquitous computing communities [3]. Typical locomotion modes include different pedestrian modalities such as walking or running, non-motorized transportation, such as biking, and motorized transportation, such as bus, train or car [4]. Locomotion detection via smartphone sensors has been increasingly explored as they provide an excellent sensing platform for detecting motions ubiquitously and unobtrusively. Accelerometers are the most widely used sensor modality for detecting modes of locomotion. A number of researchers used accelerometers for detecting different pedestrian and non-motorized locomotion types, such as walking, running or biking [5, 6]. A novel accelerometer-based technique was presented for accurate and fine-grained detection of transportation modes using smartphones [4]. The authors improved the algorithm for estimating the

gravity component of accelerometer measurements and presented a novel set of accelerometer features that can capture essential characteristics of vehicular movement patterns. To segment the sensors data, they used a sliding window with duration of 1.2 seconds. They stated that the length of the window was selected to ensure the system can rapidly detect changes in the transportation behavior of the user. However, using such a small window, they got low accuracy in distinguishing between bus, train, metro, and tram.

To overcome the issue with fixed window size, researchers have proposed adaptive sliding window segmentation techniques [1]. This paper presented a novel approach to activity signal segmentation for recognizing human activities. In this study, the window size was adaptively adjusted to achieve the most effective segmentation based on the probability that the signal belongs to a particular activity class. In addition, an activity transition diagram was developed to validate the activity transition and improve recognition accuracy. They started with 3-second windows, and if they detected that a window contains a transitional activity, they expanded the window size until getting the most confident decision from the classifier. They measured the confidence of the classifier based on the likelihood that a signal belongs to a particular activity. In another study, an adaptive time window method was proposed to extract features from quasi-periodic signals for activity recognition [7]. They detect changes in activity and take the beginning and end time as segmentation boundaries. However, their technique worked only for signals that have periodic patterns, such as walking.

SHL dataset and task description

This paper presents the techniques employed in our

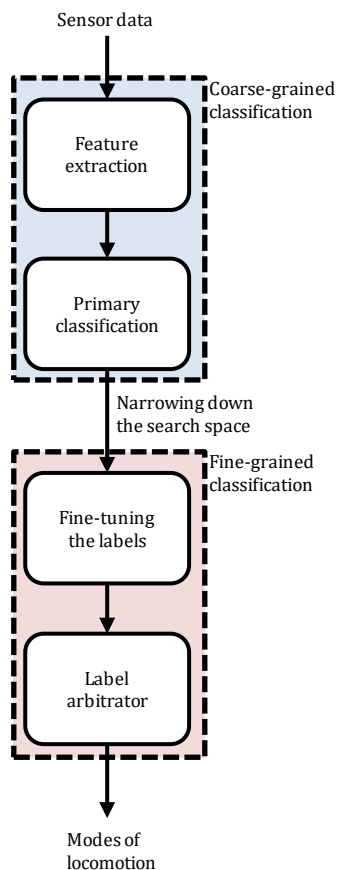


Figure 1: Workflow of the proposed method

team’s (The Drifters) submission to the Sussex-Huawei Locomotion-Transportation (SHL) recognition challenge at the HASCA Workshop at Ubicomp 2018. The goal of this challenge is to recognize eight modes of locomotion and transportation activities from the inertial sensors’ data of a smartphone. The activities that must be recognized are still, walk, run, bike, car, bus, train, subway. The SHL dataset used for this challenge comprises 271 hours of training data and 95 hours of test data [8]. The data is recorded by a Huawei Mate 9 smartphone attached to the right front pocket of a single participant in 4 months. The orientation of the smartphone is not necessarily fixed. The participant performed the activities on a daily basis (approximately 5-8 hours per day) with the phone logging the sensors data. The data include readings from 3D accelerometer, gyroscope, magnetometer, and ambient pressure sensor as well as linear acceleration, gravity, and orientation (represented in quaternion form). Data is collected from all sensors at the frequency of 100 Hz. All data samples are labeled. For both training and testing dataset, the whole data is segmented with a non-overlapped sliding window of 1-minute length. After segmentation, the order of the frames are randomly permuted, so there is no temporal dependency among the frames. The average F1-score over all the activities is used to evaluate models.

Method

Figure 1 shows our proposed framework for recognizing modes of locomotion and transportation and identifying the transition time between them in this study. We start by segmenting the sensors data into relatively large fixed windows, extracting features, and estimating one out of the eight labels for each window (*i.e.* coarse-grained classification). If a transition from

one activity to another one is guessed within a window, we divide the initial window into smaller sub-windows and use a limited version of the classifier to fine-tune the labels for the samples inside that window. Finally, a label arbitrator module modifies the labels that are estimated by the classifiers by taking into account the logical order of the activities (*i.e.* fine-grained classification).

Feature Extraction

To recognize human activities, signal segmentation is a crucial stage because it has a direct impact on the quality of feature extraction and classification accuracy [9]. Windowing approaches are typically used for segmentation. In this approach, a classifier assigns one label to all the samples inside a single window. The optimal window size varies depending on the characteristics of activity signals. Generally, smaller window size allows for faster activity detection and reduces required resources and energy; however, using smaller window size may lead to missing vital information about an activity, especially in case of complex activities such as modes of transportation. Thus, large windows are normally considered for the recognition of complex activities. However, by using a large window size, within a single window, we may capture a signal that belongs to more than one activity. Since the classifier assigns only one label to all samples of the window, this leads to a drop in accuracy.

To find the best window size to start recognizing modes of locomotion and transportation, we extract features from different window sizes and compare the accuracies of the classifiers trained on them as shown in Table 1. It can be seen in Table 1 that 30-second window gives the best performance. It is worthwhile to mention that reducing the window size did not affect the accuracy of

Window size	F1-score
1 minute	94.4 %
30 seconds	95.56%
20 seconds	94.82 %
15 seconds	95.17 %

Table 1: Comparing the effect of window size on performance

Feature name	Dimension
Mean	1
Variance	1
Standard deviation	1
Root mean square	1
Mean crossing rate	1
Zero crossing rate	1
Skewness	1
Kurtosis	1
Entropy	1
AR coefficients (order 10)	10
Integration	1
Signal magnitude area	1
Band power	25

Table 2: Features

detecting activities such as walking, running, and biking that involve significant movement of human limbs. Reduction in the window size mainly decreased the performance of distinguishing between modes of transportation including bus, car, train, and subway. The process starts by extracting features from 30-second windows as this window size gives the best performance. The data is segmented into 30-second windows with no overlap and features are extracted to train the primary classifier.

Afterward, in case of detecting a probable transition in the activity within a 30-second window, the window is divided into 15-second and 5-second sub-windows and again the same features are extracted to further fine-tune the labels (see “*Fine-tuning the labels*” section).

Table 2 lists the features we used in this study, including the feature name and feature dimension. These are all well-known time-domain features for activity recognition [10]. The features are extracted from each axis of each sensor modality (*i.e.*, accelerometer, gyroscope, magnetometer, pressure sensor, linear acceleration, gravity, and quaternion) separately. In addition, the same set of features are extracted from the magnitude of each signal to remove the effect of sensor orientation [11]. For every sample, the magnitude is calculated as L2-norm of 3D signal.

One challenge is that the modes of transportation including bus, car, train, and subway may have similar patterns of time-domain signals, so they cannot be distinguished by aforementioned features. However, given the SHL dataset, we observed that the amount of vibration could vary among different vehicles, and this can be captured by the frequency components of the acceleration signal as shown in Figure 2. The figure illustrates an example of how the power spectrum of

the magnitude of the acceleration for different modes of transportation varies from each other. Therefore, to distinguish between those activities, we added a complete set of frequency-domain features including the power of the signal lying in different frequency bins. As we observe some high-frequency components in activities such as train and bus, we did not filter the signal since it may cause losing valuable information. We take the power of the signal lying in the ranges 0-1 Hz, 1-2 Hz and so on up to 25 Hz as features.

Finally, all the features are normalized to have mean of zero and standard deviation of one.

Classification with Deep Neural Network

Different types of machine learning algorithms have been used for activity recognition, among which neural networks have several interesting properties that motivate us to choose it for the current task. First, the feature selection and classification can be embedded into a single multilayer neural network. Through the training process, the weights of the neurons in the first layers of the network learn to scale the important and informative features and ignore less informative ones. Second, they can estimate the probability of each activity to be true. In fact, to recognize n different activities, a neural network has n neurons in the very last layer, and each of them corresponds to one activity and outputs the probability of that activity to be true. This property can be leveraged to assess the confidence of the classifier. Third, it is fast and efficient even when working with high dimensional data. Feed-forward neural networks, typically with multiple hidden layers, have been widely used for classification and regression tasks [6, 12]. It has been shown that increasing the number of layers (known as the depth of the network) can significantly increase learnability and decrease the

Layer	# of neurons	Activation function
Fully-connected_1	200	ReLU
Fully-connected_2	64	ReLU
Fully-connected_3	8*	Softmax

* It is 2 for two-class classifiers

Table 3: Detail of the neural network

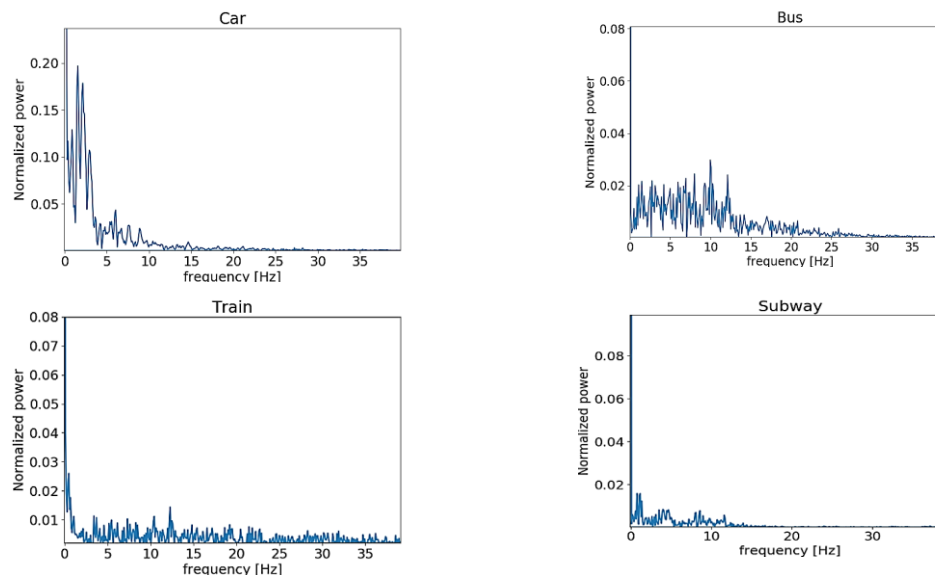


Figure 2: Example shows how power spectrum of magnitude of acceleration varies among different modes of motorized transportation

number of required neuron units (*i.e.*, the number of trainable parameters) in each layer [13]. In the current study, we use a deep neural network (DNN) with three fully connected layers and feed the extracted features to this network. The detail of the neural network can be found in Table 3.

The primary classifier (see Figure 1) is supposed to recognize one of 8 modes of locomotion and works with features that are extracted from 30-second windows. In addition to the primary classifier, we train multiple two-class classifiers to distinguish between every pair of activities. For every possible pair of activities, two two-class classifiers are trained; one with features that are extracted from 15-second windows and one with features that are extracted from 5-second windows. These classifiers operate when the primary classifier

detects two different labels for two consecutive windows (next subsection). They are useful to capture the moment of transition between the activities more precisely. The intuition behind using two-class classifiers instead of eight-class is the fact that by extracting features from shorter windows, it would be very challenging to distinguish between 8 classes; so the decision made by the primary classifier is used to narrow down the search space for the classifier that works with 15-second and 5-second windows. Two-class classifiers have the same architecture as the primary classifier (Table 2) but in the last layer they have 2 neurons corresponding to two activities.

The well-known ReLU function is used as the activation function for the first two layers of the neural network in order to fuse the features and pick important

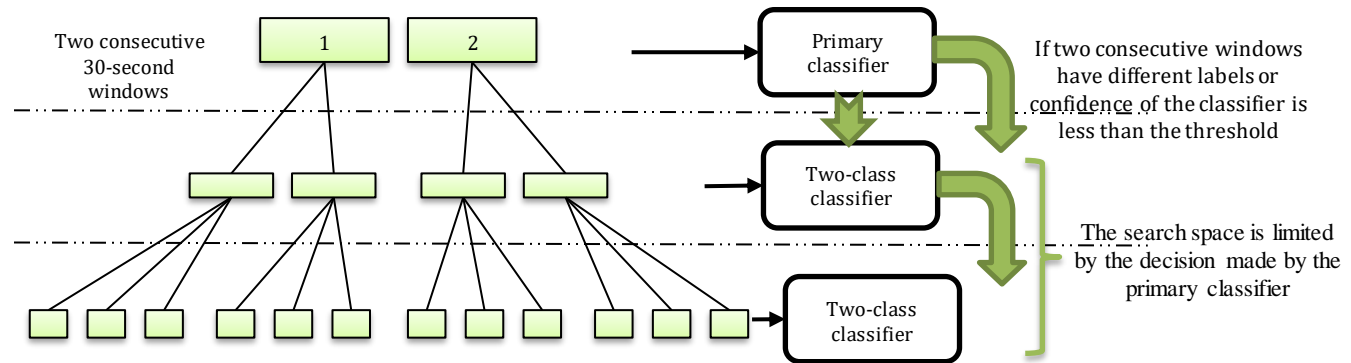


Figure 3: Hierarchical segmentation to fine-tune the labels for windows in which the activity change. The two class-classifier only looks for the labels that are initially determined by the primary classifier.

information out of them [14]. The Softmax activation function is used in the last layer of the neural network to perform classification (Equation 1).

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^C e^{x_j}} \quad (1)$$

Equation 1 shows the Softmax where x_i is the i^{th} input to the function and C is the total number of the inputs, which, in the case of classification, is equal to the total number of classes. The output of the Softmax function is a value between 0 and 1 and the total sum of the outputs is equal to 1. The output of this function is equivalent to a categorical probability distribution. In other words, it indicates the probability that any of the classes are true. Dropout technique is also utilized to reduce the chance of overfitting. The key idea of dropout is to randomly deactivate some units (along with their connections) in the neural network during training [15]. This deactivation prevents the units from co-adapting too much and improves generalization because it forces the layers to learn the same concepts

with different neurons. However, during the prediction phase, the dropout layer is deactivated.

Fine-tuning the labels

Although extracting features from larger windows (30-second in this study) gives better accuracy in recognizing the activities, especially for modes of transportation, it may lead to a higher error for windows that contain more than one activity (*i.e.*, transition from one activity to another). To alleviate this issue and to more accurately label samples near activity transitions, we propose a hierarchical signal segmentation and classification algorithm as shown in Figure 3 that enables a multi-windowing approach for data segmentation.

We divide a one-minute segment of the data into two non-overlapping 30-second segments. It should be mentioned that in the given dataset for this challenge, the whole data is segmented into one-minute windows, and the windows are randomly permuted. As a result, there is no temporal dependency between windows, so one minute is the largest segment of the data we can



Figure 4: An example of a misclassification that can be fixed by label arbitrator

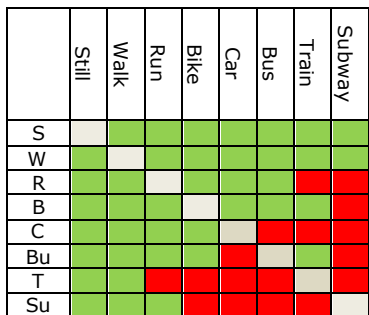


Figure 5: Activity transition matrix

work with. The features are extracted from two consecutive 30-second windows and a primary DNN classifier estimates the labels for each window. Based on the labels and the probabilities that are generated by this classifier, a window is suspected to contain a transition from one activity to another in two cases.

- 1) When the labels of two consecutive segments are not the same.
 - 2) When the probability of the determined activity is less than a threshold. This threshold is determined based on the dataset in hand to get the best accuracy.
- In the first case when there is a change from label A to B for two consecutive windows, the activity might be changed within either of the windows. Thus, 30-second windows are split into two 15-second sub-windows and features are extracted. Each sub-window is then fed to a two-class classifier that is trained and optimized to merely distinguish between the two activities that are initially detected (*i.e.*, A and B) by the primary classifier. If the detected labels for two consecutive 15-second sub-windows are also different, each of them is split into three 5-second sub-windows and features are extracted and fed to the corresponding two-class classifier to recognize the labels for each of them.

In the second case, when the probability of the detected label is less than a threshold for a window of data, the window may contain signals of two distinct activities, so that the classifier is not confident about its decision. In such case, the 30-second window is again split into two 15-second sub-windows and all possible two-class classifiers, which distinguish between the determined activity (*i.e.*, the output of the primary classifier) and any other activity, are used to find the label for each sub-window. The label that has the highest probability is assigned to the whole sub-

window. The threshold for confidence is determined based on the training data to get the best performance, and it was 0.6 in our study.

Finally, if the labels that are estimated for two consecutive 30-second windows agree with each other, that label is assigned to all samples of those windows.

Label arbitrator

To further modify the detected labels, a label arbitrator module is proposed. This module acts when the label detected by a classifier (either of primary or two-class classifiers) changes from one window to the next one. In fact, based on two rules about possible transitions between different activities, this module can modify certain misclassifications. The first rule, which is a general rule, says that one mode of locomotion or transportation cannot happen in the middle of another one. Figure 4 shows an example of such misclassification that may happen due to detecting labels separately for each segment. In this example, the second segment is misclassified because it may have a signal similar to the train activity. In this case, the label arbitrator replaces the label of the middle segment with its neighbors' label. The second rule is data-specific. Based on the training data we create a label transition matrix that shows the possible transitions between activities (Figure 5). In Figure 5, the element at row i and column j is green if a transition from activity of i^{th} row to activity of j^{th} column is possible and is red otherwise. As Figure 5 shows, a transition from activity train to the subway is impossible. Thus, if the classifier detects such a transition due to misclassification, the label arbitrator detects that as an invalid case. In this situation, the label with higher probability wins, and it is assigned to the whole window.

	F1-score
DNN with only 30-second windows	95.56
DNN + Multi-window segmentation	96.71%
DNN + Multi-window segmentation + label arbitrator	97.68%

Table 4: Results of the proposed method and comparing with using fixed windows for feature extraction

Results

The SHL training dataset published by the competition committee is used to train and evaluate the model. This is 271 hours of data collected by a Huawei Mate 9 smartphone placed in the right front pocket of a single participant while he performed eight locomotion and transportation activities including still, walk, run, bike, car, bus, train, subway. To assess the performance of the proposed method, we randomly divide the SHL training dataset into 70% for training the model and 30% for testing its performance. The training data is used to train the primary classifier as well as all two-class classifiers and the transition matrix. Precision and average F1-score metrics are used to evaluate the performance of the classification.

Table 4 shows the results of the cross-validation of the proposed method in comparison with using only fixed 30-second windows for classification. In fact, the first row of the Table 4 shows the result of using a single window size for classification without modifying the labels with dividing approach and label arbitrator. As the table shows, our approach improves the performance by around 2% compared to the simple classifier that uses a single size window for data segmentation. This improvement is associated with two factors. First, by splitting large windows into smaller ones, we can capture the moment of transition between the activities more precisely. This helps to obtain more accurate labels for windows that contain more than one activity; and second, the label arbitrator fixes certain types of misclassifications.

Table 5 shows the confusion matrix of the proposed method. In the table each column corresponds to the true label and each row corresponds to the estimated label. As can be seen in Table 5, most of the

	Still	Walk	Run	Bike	Car	Bus	Train	Sub way
Still	96.9	0.89	0	0.17	0.31	0.65	1.87	1.19
Walk	0.74	98.3	0.24	0.19	0.18	0.22	0.23	0.29
Run	0.11	0.03	98.9	0.18	0	0.42	0	0
Bike	0.18	0.28	0.01	99.3	0.04	0.02	0.07	0
Car	0.14	0.02	0.09	0.01	99.1	0.55	0.08	0
Bus	0.25	0.18	0	0	0.45	97.4	0.03	0
Train	0.5	0	0	0.09	0	0.34	94.9	0.74
Sub way	1.29	0.39	0	0	0	0.31	3.39	97.5

Table 5: Confusion matrix of the proposed method (in percent)

misclassifications occur between the train and subway classes because they have similar signals.

Computational Resources

A high-performance computing server with 8-core Broadwell processors running at 2.4 GHz and 30G RAM is used for extracting the features. The Keras library with TensorFlow backend running on an NVIDIA GeForce GTX 950M GPU is used to train the classifiers. The size of the trained classifiers including the primary classifier and all two-class classifiers is 83.9 MB if it is saved on a hard disc. It takes about an hour to train the whole model for around 190 hours of data.

Conclusion

We proposed a hierarchical framework to recognize modes of locomotion and transportation using data from smartphone sensors. The proposed method can

benefit from using both large window size to increase the general performance and shorter window size to detect transitions between activities faster and more precisely. This approach outperforms base classifiers which use a single window-size for segmenting the whole data. This system can recognize 8 modes of locomotion and transportation accurately. Such system can provide important and useful contextual information about the users to mobile applications and can unlock many new mobile sensing and computing paradigms. The recognition result for the testing dataset will be presented in the summary paper of the challenge [16].

Acknowledgments

This work was supported in part by the National Science Foundation, under grants CNS-1150079 and ECCS-1509063. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

1. Noor, M.H.M., et al., Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive and Mobile Computing*, 2017. 38: p. 41-59.
2. Banos, O., et al., Window size impact in human activity recognition. *Sensors*, 2014. 14(4)
3. Lane, N.D., et al., A survey of mobile phone sensing. *IEEE Communications magazine*, 2010. 48(9).
4. Hemminki, S., et al. Accelerometer-based transportation mode detection on smartphones. in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. 2013. ACM.
5. Iso, T. and K. Yamazaki. Gait analyzer based on a cell phone with a single three-axis accelerometer.

- Proceedings of the 8th conference on Human-computer interaction with mobile devices and services. 2006.
6. Ordóñez, F.J. and D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 2016. 16(1)
 7. Sheng, Z., et al. An adaptive time window method for human activity recognition. in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*. 2015. IEEE.
 8. Gjoreski, H., et al. The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics with Mobile Devices. *IEEE Access*, 2018
 9. Banos, O., et al. Evaluating the effects of signal segmentation on activity recognition. in *IWBIO*. 2014
 10. Wu, J., Sun, L., and Jafari, R. A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors. *IEEE J. Biomedical and Health Informatics* 20, 5 (2016), 1281–1290.
 11. Wu, J., and Jafari, R. Orientation independent activity/gesture recognition using wearable motion sensors. *IEEE Internet of Things Journal* (2018)
 12. Benediktsson, J.A., et al. Neural network approaches versus statistical methods in classification of multisource remote sensing data. 1990.
 13. Mhaskar, H., Q. Liao, and T.A. Poggio. When and why are deep networks better than shallow ones? in *AAAI*. 2017.
 14. Goodfellow, I., et al., *Deep learning*. Vol. 1. 2016: MIT press Cambridge.
 15. Srivastava, N., et al., Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 2014. 15(1)
 16. Lin Wang, Hristijan Gjoreski, Kazuya Muraio, Tsuyoshi Okita, Daniel Roggen. Summary of the Sussex-Huawei Locomotion-Transportation Recognition Challenge. *Proceedings of the 6th International Workshop on Human Activity Sensing Corpus and Applications (HASCA2018)*. Singapore, Oct. 2018.