Deep learning for driving detection on mobile phones

Chetan Ramaiah
Metromile
cramaiah@metromile.com

Allen Tran
Metromile
allen@metromile.com

Evan Cox
Metromile
evan@metromile.com

George Mohler
Metromile
george@metromile.com

ABSTRACT

Sensor based activity recognition is a critical component of mobile phone based applications aimed at driving detection. Current methodologies consist of hand-engineered features input into discriminative models, and experiments to date have been restricted to small scale studies of O(10) users. Here we show how convolutional neural networks can be used to learn features from raw and spectrogram sensor time series collected from the phone accelerometer and gyroscope. While with limited training data such an approach under performs existing models, we show that convolutional neural networks outperform currently used discriminative models when the training dataset size is sufficiently large. We also test performance of the model implemented on the Android platform and we validate our methodology using sensor data collected from over 2000 mobile phone users.

Keywords

activity detection, convolutional neural network, telematics, usage based insurance

1. INTRODUCTION

With the growth of usage based car insurance (UBI) and safe-driving incentives, there is a need for low cost mobile applications that are able to accurately capture vehicle trips, mileage, and driving behavior from sensor data collected on mobile phones[9, 3]. A critical component of such applications is a classifier that takes as input time series from the accelerometer, gyroscope, magnetometer, and GPS radio and outputs a probability or prediction of the current activity state of the phone.

There is a large body of research on mobile phone based activity detection and a good review of the field is given in [19]. The types of models used for classifying sensor segments include decision trees [2, 14], SVM [1, 6], KNN [12], Naive Bayes [13], feed-forward neural networks [10, 11], HMM [16] and ensembles of these approaches. In these studies, accelerometer is almost always used, either alone or in combination with a magnetometer, gyroscope and/or GPS. In all cases, features are engineered from raw time series samples using a variety of techniques including sample statistics over a window [8, 16], sample statistics of the FFT of the data [8, 16], and auto-regressive coefficients [10].

The main contribution of our work here is to show how convolutional neural networks can be used to train an end-to-end activity classifier with no feature engineering. This approach has two advantages, the first being avoiding the time-intensive process of designing features as instead convolution filters are learned during supervised training of the model. The second advantage this approach has is higher accuracy compared to the common approach of inputting engineered features into a discriminative classifier. We note that this is not the first study of activity detection using neural networks and in [10, 11], accelerometer based features (autoregressive coefficients, signal-magnitude area, and tilt angle) are input into a multi-layer perceptron as the classifier. However to our knowledge no work to date has considered convolutional neural networks for driving detection where features are learned from raw accelerometer time series.

Our focus in this work is on battery efficient methods for driving detection that rely on the phone accelerometer and/or gyroscope rather than the GPS radio. A similar goal is considered in [8] where the authors classify activity segments into seven classes including walking, several public transportation categories, and car. Features are engineered from a high frequency, gravity-corrected accelerometer (60hz/100hz) and input into a classifier that combines a kinematic HMM with Adaboost. Using data from 16 individuals and 150 hours of samples, the authors achieve approximately 85% precision and recall, a 20% improvement over GPS based approaches [16] and accelerometer based approaches without gravity correction [23]. We will use the accelerometer based approach of [8] for comparison in this study. However GPS features could also be added to our model either in the initial input layer as an added channel or appended in a dense layer.

The outline of this paper is as follows. In Section 2, we discuss our proposed deep learning model architecture. Our approach makes use of 2D convolutional neural networks by transforming raw sensor time series into spectrograms using a FFT. In Section 3, we discuss our methodology of data collection. Whereas past studies have often focused on 10-20 users, we collected sensor data from over 2,000 mobile phones for our experiments. In Section 4, we present our experimental results that illustrate the advantages of the deep learning approach to activity and driving detection.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. MODEL ARCHITECTURE

Feature engineering based approaches to activity recognition heavily rely on either frequency domain features, such as the FFT procedure[15, 16, 17, 18, 24], or time domain features[4, 7, 12]. In our end to end learning system, we eliminate the process of feature engineering and instead allow the CNN to learn powerful features from different representations of the data.

The frequency domain representation of temporal data has been shown to be useful in discriminatory problems, and has been widely used in literature on various domains ranging from audio waves [5] to accelerometer sensors[19]. The spectrogram is a time-frequency representation of a signal obtained by stacking FFT responses of sliding windows over the signal. Hence it has a good basis to be a useful representation of the data. The spectrograms are generated from accelerometer and gyroscope sensor data. Since both the sensors on a mobile phone are usually tri-axial, the sensor data is a three dimensional temporal stream. The spectrogram is generated independently for each axis, resulting in a three dimensional matrix for each data sample. This matrix is then used as the input to the CNN in Figure 1a. Although spectrograms are usually represented as images, we elect to not transform the matrix into a color space, thus eliminating a lossy, noise-inducing transformation and saving on resources.

In addition to using spectrograms, we have also experimented with 1D convolutions of sensor data. If the spectrogram CNNs can be thought of as analogous to frequency domain feature engineering approaches, the time series CNNs can be considered to be analogous to the time domain feature engineering approaches.

Finally, a multi-stream CNN (Figure 1c), loosely based on [20], is learned, where spectrograms of accelerometer and gyroscope data form the two streams. Each stream's parameters are learned independently before merging the results of the final dense layer and feeding it then to the softmax layer for prediction. This approach is superior to feeding a concatenated version of the two spectrograms in a single stream as it allows for flexibility in model structure by treating the two streams independently.

The architecture of the CNNs is illustrated in Figure 1. All convolution and dense layers were activated with ReLU [25], and dropout [21] was used for regularization. Standard practices were used in training and parameter selection. Training was performed using Lasagne on four NVIDIA GeForce GTX TITAN X GPUs.

3. DATA COLLECTION

Data were collected from 2,133 drivers, using Android platform consumer smartphones. Two minute samples from each device were recorded at random points in time. For each sample, we retained the platform specific activity labels along with the accelerometer and gyroscope measurements. Since a large number of these samples were recorded when the user was likely to be sleeping or inactive, we removed recordings when the activity labels and the sensor measurements suggested the phone was inactive. In total, the data covers roughly 5,400 hours, excluding the inactive recordings.

Each recording was split into 30 second windows with 50 percent overlap. Because the underlying sensor hardware differs across phones, sensor measurements are recorded at time varying frequency both on a particular phone and across different phones. To standardize these recordings, we interpolate sensor measurements to a fixed frequency, 50hz for the accelerometer and 10hz for the gyroscope.

Some of the models tested required converting the sensor measurements to spectrograms, which represent the sensor measurements in time-frequency space. We create spectrograms for each axis at the window level, after interpolation to a fixed frequency. Figure 2 presents two sample spectrograms from each class. The spectrogram is in the *jet* color map, with blue representing a low gray scale value and red representing a high value. Conceptually, when the sensors are stationary, there will be very little response in the frequency domain, as demonstrated by the mostly blue color palette in the still class spectrograms in Figure 2. The automotive class will have responses in the higher frequencies due to vibrations in the vehicle and the walking movement will have high responses across the frequency range.

The final component of data collection relates to labeling each window for the purposes of supervised learning. In each user's primary vehicle, we installed an iBeacon device which allowed us to detect when the phone was in close proximity. Hence portions of recordings were labeled as driving when the smartphone was paired with the iBeacon. To mitigate the risk that driving can occur in other vehicles, we selected drivers that only had a single vehicle listed on their associated insurance policy.

4. RESULTS

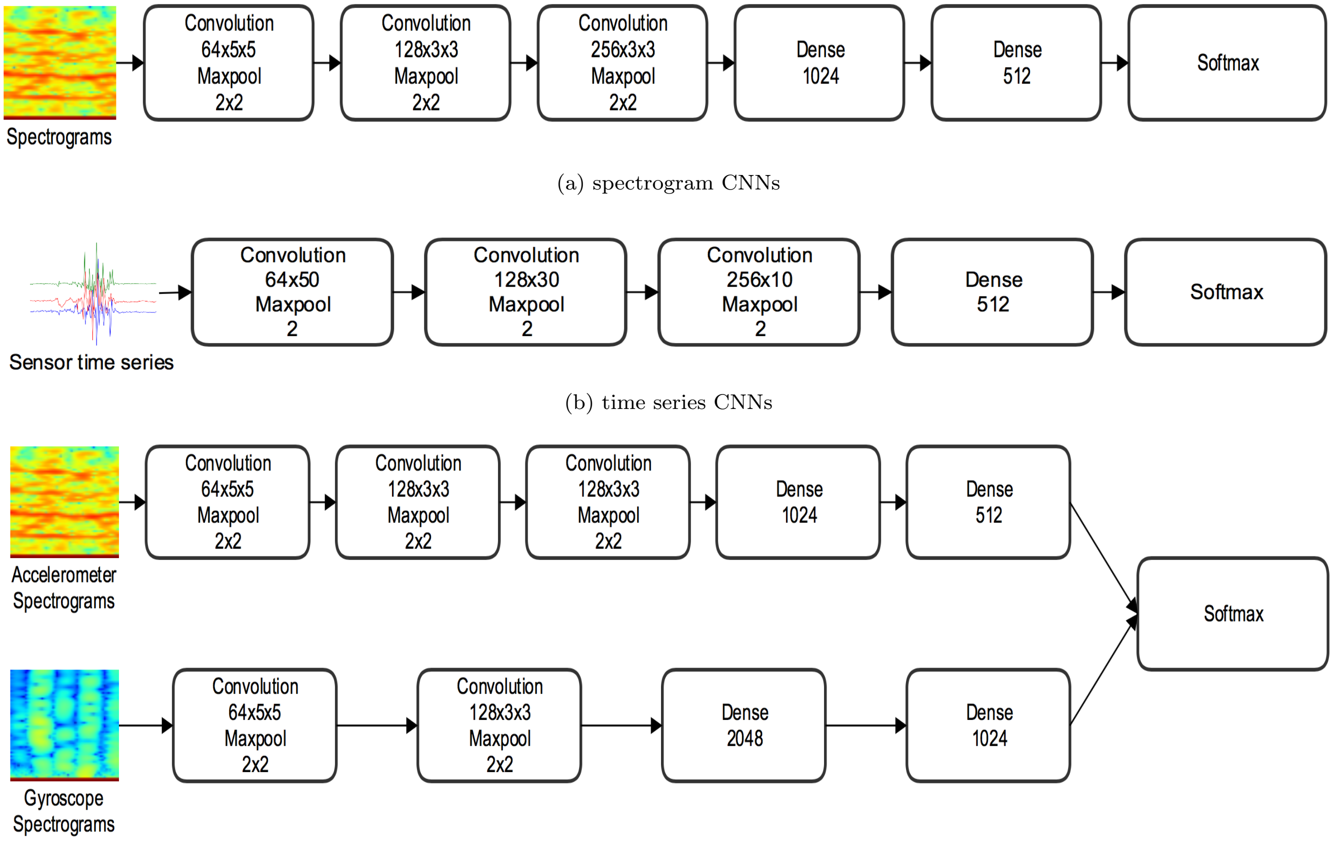
We first evaluate the efficacy of our approach by setting up a transportation mode detection problem, with the approach from [8] as a baseline measure. Next, we evaluate the performance of our deep learning architectures on the novel problem of driving detection. Finally, we describe the android prototype for driving detection in real time with Convolutional Neural Networks.

4.1 Baseline Experiments

Hemminki et. al [8] perform transportation mode classification by modeling three learners, the *kinematic motion classifier* distinguishes between pedestrian and other modes at a coarse level. The *stationary classifier* classifies between stationary and motorized transportation modes and the *motorized classifier* differentiates between various modes of motorized transport. Due to the differences in our dataset and objectives, we have modified their experimental setup as follows. Since the dataset lacks labels for mode of motorized transport, the motorized classifier is no longer necessary. Also, in order to conserve energy on the users phone, the prediction is done exclusively over a short window and not continuously for the duration of the activity.

Table 1: Baseline features

Domain	Features
Statistical	Mean, STD, Variance, Median, Min, Max, Range, Interquartile range Kurtosis, Skewness, RMS
Time	Integral, Zero-Crossing Rate
Frequency	FFT DC first five frequency responses, Wavelet Entropy, Wavelet Magnitude


Spectrograms
Convolution 64x5x5 Maxpool 2x2
Convolution 128x3x3 Maxpool 2x2
Convolution 256x3x3 Maxpool 2x2
Dense 1024
Dense 512
Softmax
Sensor time series
Convolution 64x50 Maxpool 2
Convolution 128x30 Maxpool 2
Convolution 256x10 Maxpool 2
Dense 512
Softmax
Accelerometer Spectrograms
Convolution 64x5x5 Maxpool 2x2
Convolution 128x3x3 Maxpool 2x2
Convolution 128x3x3 Maxpool 2x2
Dense 1024
Dense 512
Softmax
Gyroscope Spectrograms
Convolution 64x5x5 Maxpool 2x2
Convolution 128x3x3 Maxpool 2x2
Dense 2048
Dense 1024

(a) spectrogram CNNs

(b) time series CNNs

(c) multi-stream spectrogram CNNs

Figure 1: CNN architecture. The numbers indicate filter/kernel sizes. All convolution and dense layers are activated with ReLU. The two approaches illustrated here are (a) spectrogram CNNs, (b) 1D convolutions over raw sensor data and (c) multi-stream CNN with accelerometer and gyroscope spectrograms.

For each window in our dataset, we extract the features described in Table 1. The random forest classifier was chosen its accuracy after experiments with alternative discriminative classifiers (logistic regression, SVM, KNN, boosting). The data set is then split into two roughly equal training and testing sets, while ensuring that mobile phone users in the training set are not present in the test set.

4.2 Activity Recognition

The activity recognition problem, also referred to as the transportation mode detection, is a classification problem amongst the three classes in the dataset (i) Walking, (ii) Automotive, and, (iii) Stationary.

Figure 2 presents the t-sne [22] visualization of the spectrogram representation of the dataset. From the figure, it is clear that there is little overlap between the three classes, and that there might be confusion between the automotive and the still class. A hint of which two classes might have confusion is apparent from the overlap of data points from the automotive and the still class. This is corroborated by both the sample images and the experimental result.

The activity recognition problem was modeled by the baseline process and the three CNNs described in section 2. The AUC scores are presented in Table 2. The accelerometer spectrogram appears to be the best representation of the data as it performs well individually or with the gyroscope spectrogram. The gyroscope frequency domain does not contribute to improving accuracy.

Table 2: Activity recognition results

Approach	AUC
Baseline	0.84
Accelerometer Spectrogram CNN	0.89
Accelerometer Temporal CNN	0.83
Accelerometer and Gyroscope CNN	**0.89**

The confusion matrix in Figure 3 is generated from the predictions of the accelerometer spectrogram CNN. On examination, it appears that the automotive class is the hardest class to classify and it is most confused with the still class. This implies that the accelerometer data is insufficient to distinguish between still and automotive classes. In the accelerometer frequency domain, a phone at rest in a stationary location and a phone at rest in an automobile differs only in high frequency vibrations. Additional sensor data, such as GPS, would easily improve the accuracy.

4.3 Effect of training dataset size

Experiments with various dataset sizes are conducted to

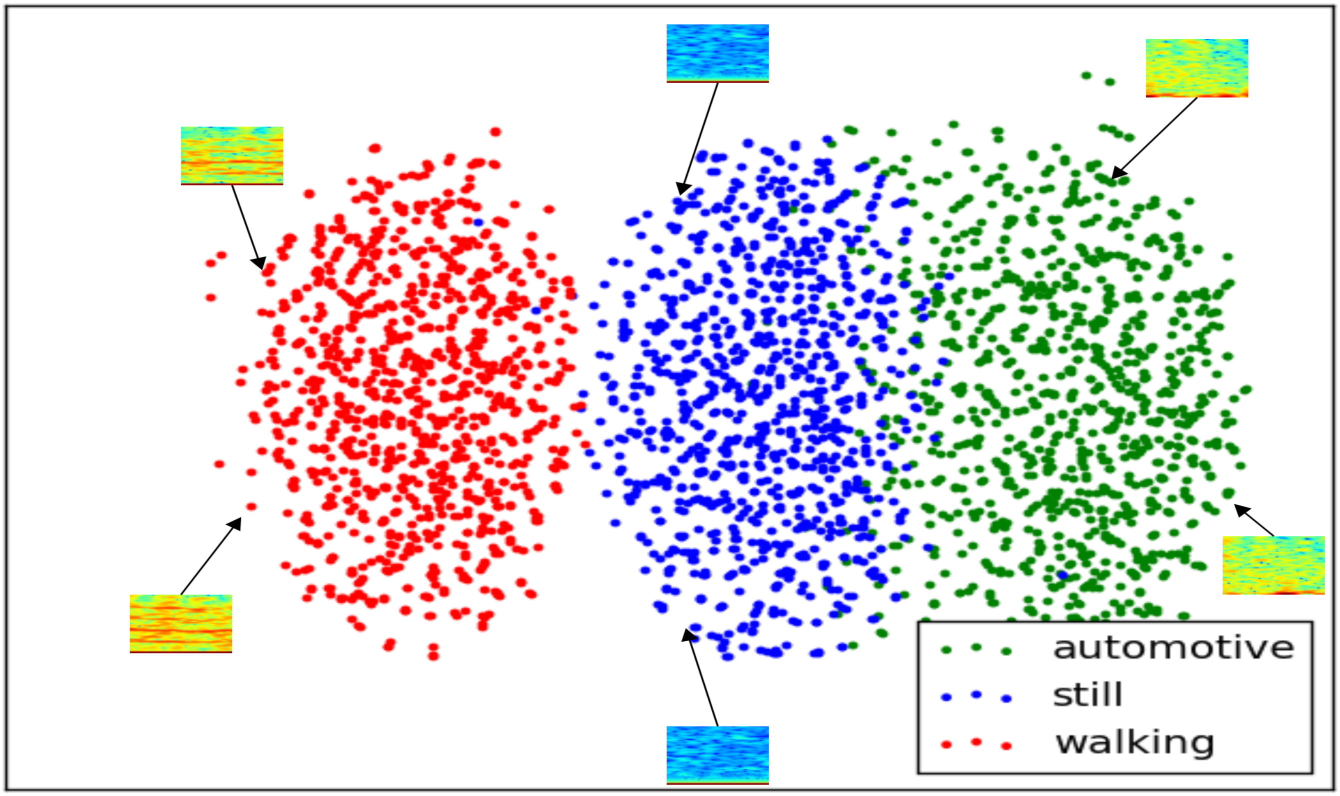


Figure 2: t-sne[22] visualization of the activity recognition problem.

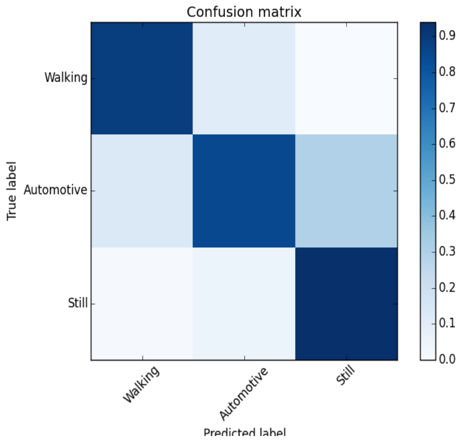


Figure 3: Confusion matrix for activity recognition

evaluate its effect on model performance. In an attempt to replicate the dataset size in [8], we pick a random subset of recordings of approximately equal dimensions.

The results of this experiment are presented in Table 3, where the baseline process is compared with the accelerometer CNN from Figure 1a. The results follow the familiar pattern of deep learning techniques improving significantly with increasing dataset size. For example, with 50 hours of training data the baseline model outperforms the CNN, given the random forests ability to reduce variance and the low dimensionality of the features. However, at 500 hours of training data and above the CNN has the higher performance and at 1300 hours of training data the AUC of the CNN is 0.89 compared to 0.83 for the random forest.

Table 3: AUC scores for activity recognition with varying dataset sizes

Dataset Size	Baseline	Accelerometer CNN
50 hours	**0.82**	0.78
150 hours	**0.82**	0.82
500 hours	0.84	**0.86**
1000 hours	0.84	**0.88**
1300 hours	0.83	**0.89**

4.4 Driving Detection

A second application we consider is driving detection. For usage based insurance, accurately tracking a vehicle and removing trips on public transit and in other vehicles is of high importance.

We restrict our attention to the automotive subset of the data and label each sample as driving vs. other using the presence or absence of the mobile phone within the geo-fence of the iBeacon device in each vehicle. There are 362 unique vehicle models in the dataset, the top ten occurring vehicles are displayed in Figure 4.

Our results are presented in Table 4, where we again see the CNN applied to accelerometer spectrograms are the highest performing models. As discussed in Section 3, negative class samples may include data from other vehicles which likely contributes to the lower overall AUC values.

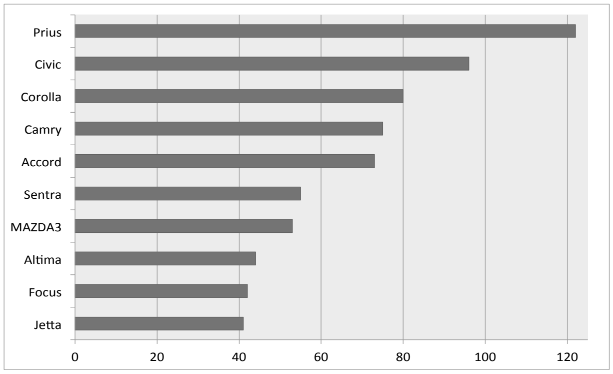


Figure 4: Frequency of top ten vehicles in driving detection dataset.

In practice, low confidence predictions can be supplemented with user generated labels to remove spurious trips. The CNN model discussed here is useful for limiting the number of labels requested from the user and/or incorporating the labels in an active learning framework.

4.5 Android Prototype

In order to study the real world performance of the model, the accelerometer spectrogram CNN was also ported to an Android application. The goal of the app was to perform predictions on the device itself, so as to limit data, memory and processing resource usage. A trained driving detection model, which takes up about 8MB of storage, is packaged into the app. The app collects accelerometer data for the duration of the window length, generates a spectrogram from the collected data, and performs driving detection. In order to support the entire spectrum of devices in the market, the entire prediction operation runs on the CPU. The average run time for the spectrogram generation process is about 0.5s and the prediction process takes about 3.5 seconds.

Table 4: Driving detection results

Approach	**AUC**
Baseline	0.67
Accelerometer Spectrogram CNN	**0.77**
Accelerometer Temporal CNN	0.75
Accelerometer and Gyroscope CNN	0.77

5. CONCLUSION

We showed how deep learning techniques, in particular convolutional neural networks, are well suited for mobile phone based activity and driving detection. With enough training data, the CNN achieves higher accuracy compared to commonly used approaches, requires no engineering of features, and can be implemented efficiently directly on the mobile phone.

While our focus here was on accelerometer based models that limit battery drain, accuracy would be improved by incorporating other data such as GPS. For example, a GPS based classifier could be combined with the CNN through an ensemble approach, or additional time series such as GPS speed could be added as channels in the initial layer of the network.

6. REFERENCES

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. *J. UCS*, 19(9):1295–1314, 2013.

[2] A. Anjum and M. U. Ilyas. Activity recognition using smartphone sensors. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 914–919. IEEE, 2013.

[3] G. Castignani, T. Derrmann, R. Frank, and T. Engel. Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. *Intelligent Transportation Systems Magazine, IEEE*, 7(1):91–102, 2015.

[4] B. Das, A. M. Seelye, B. L. Thomas, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe. Using smart phones for context-aware prompting in smart environments. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 399–403. IEEE, 2012.

[5] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6964–6968. IEEE, 2014.

[6] J. Frank, S. Mannor, and D. Precup. Activity recognition with mobile phones. In *Machine Learning and Knowledge Discovery in Databases*, pages 630–633. Springer, 2011.

[7] J. J. Guiry, P. van de Ven, and J. Nelson. Orientation independent human mobility monitoring with an android smartphone. In *Proceeedings of the IASTED International Conference on Assistive Technologies, Innsbruck, Austria*, pages 15–17, 2012.

[8] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 13. ACM, 2013.

[9] J.-H. Hong, B. Margines, and A. K. Dey. A smartphone-based sensing platform to model aggressive driving behaviors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4047–4056. ACM, 2014.

[10] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *Information Technology in Biomedicine, IEEE Transactions on*, 14(5):1166–1172, 2010.

[11] A. M. Khan, M. H. Siddiqi, and S.-W. Lee. Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. *Sensors*, 13(10):13099–13122, 2013.

[12] M. Kose, O. D. Incel, and C. Ersoy. Online human activity recognition on smart phones. In *Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, pages 11–15, 2012.

[13] D. Lane and B. Mohammod. A smartphone application to monitor, model and promote wellbeing. *IEEE Pervasive Health*, 2012.

[14] Ó. D. Lara and M. A. Labrador. A mobile platform for real-time human activity recognition. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 667–671. IEEE, 2012.

[15] K. Ouchi and M. Doi. Indoor-outdoor activity recognition by a smartphone. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 600–601. ACM, 2012.

[16] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):13, 2010.

[17] J. Ryder, B. Longstaff, S. Reddy, and D. Estrin. Ambulation: A tool for monitoring mobility patterns over time using mobile phones. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 927–931. IEEE, 2009.

[18] C. K. Schindhelm. Activity recognition and step detection with smartphones: Towards terminal based indoor positioning system. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 2454–2459. IEEE, 2012.

[19] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 2015.

[20] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[22] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[23] S. Wang, C. Chen, and J. Ma. Accelerometer based transportation mode recognition on mobile phones. In *2010 Asia-Pacific Conference on Wearable Computing Systems*, pages 44–46. IEEE, 2010.

[24] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 17–24. Ieee, 2012.

[25] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, et al. On rectified linear units for speech processing. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3517–3521. IEEE, 2013.