My Shoe is Smarter than yours!

Artur Balanuta
Carnegie Mellon University
Electrical and Computer Engineering
Pittsburgh, PA 15213
arturb@andrew.cmu.edu

Advait Burde
Carnegie Mellon University
Electrical and Computer Engineering
Pittsburgh, PA 15213
aburde@andrew.cmu.edu

Anchan Saxena
Carnegie Mellon University
Electrical and Computer Engineering
Pittsburgh, PA 15213
anchans@andrew.cmu.edu

ABSTRACT

This project implements the 'Aware Shoe', a shoe capable of detecting seven common activities a person would perform on a daily basis. The activities we targeted are walking, jumping, squatting, standing up, standing, sitting and sitting down. The shoe is able to detect the activity with high accuracy even if it is performed for a short duration of time. The implementation is economical as well as ergonomic. Two pressure sensors in the sole of the shoe and a Particle Photon comprise the core of the sensing system which is powered by a rechargeable battery. Random Forest classifier is employed for detecting the activities. We believe the 'Aware Shoe' can become an integral part of personal activity monitoring that would provide helpful and quick feedback to users regarding the way they walk, exercise and other daily habits. Thus 'Aware Shoe' has the ability to enrich current activity and fitness trackers with activities that go beyond the ones supported by current devices.

KEYWORDS

Pressure Sensor, Particle Photon, Feature Extraction,
Activity detection, Machine Learning

1 INTRODUCTION

The development of power efficient micro controllers with wireless capabilities has led to a surge of innovative wearable devices. Such devices are targeted towards detection, analysis and feedback of various environmental factors to user comfort and lifestyle. The idea of making our daily accessories smarter has become a primary focus of research. Everything from our phones, watches, wallets to our clothes, spectacles and even our shoes are getting smarter.

The motivation behind our project is to help us keep track and improve our health and what better way to do that than working on the unique ability of humans to stand up on two limbs. Our footwear is the most important daily accessory as a correct posture is dependent on how we sit, stand and walk. Therefore, this project keeps track of the duration of an activity and reminds the wearer accordingly.

This project aims to make shoes smart which identify the user's activity to realize the above said model. Using a combination of pressure sensors and a particle photon, the shoe is trained for different activities using machine learning. The Random Forest classifier was used as the main machine learning model. We initially focused on recognizing basic activities of sitting, standing and walking but expanded it to categorize sitting down, standing up, jumping and squatting as well. For sensing, we used a pair of pressure sensors embedded in the shoe sole which are potentiometers and give high values when stressed upon. The particle photon reads values from the pressure sensors and sends the data over WiFi to a UDP socket of a dashboard, in our case, a laptop. We make the assumption that the photon has continuous access to WiFi. A Bluetooth Low Energy module was considered but we decided to focus on the sensing and activity detection aspect.

We trained the model for the above mentioned activities and achieved high training accuracy of over 98% using cross validation. For real time testing it was also accurate in detecting the activities. Such a technology can surely help people to keep track of their general well-being as well as getting rid of undesirable postures of sitting, walking or even running. Athletes can perform quantitative analysis of their performance to improve upon with the incoming data from the sensors in real time. Also it has been scientifically proven that sitting for long periods of time is bad for our posture and health. Therefore such a technology can keep track of how long a user has been sitting and if it detects longer duration than recommended then reminding the user to stand up and walk for sometime. Another important application can be for old people in detecting any anomalous behavior while standing or walking, say if someone falls down, it can alert others about such a mishap.

2 IMPLEMENTATION

In this section we describe the project implementation in detail. We describe the sensors we used, data gathering method, choices we made for feature computation and the machine learning pipeline. We also discuss optimization choices made by observing data and results obtained while training different machine learning models.

2.1 Pressure sensor

The pair of pressure sensors that we used are Force-Sensing Resistors[1] (FSR) shown in Figure 1. They are square shaped strips with each side 43.69 mm and just 0.45 mm thick which are perfect for placing on a flat shoe sole. They work on the principle of Force-Voltage conversion wherein it is connected to a measuring resistor in a voltage divider configuration. We used two of these placing them on the front and back portions of the sole (shown in Figure 4) since generally the foot force is most defined at these spots.

The pressure pads were connected to the Particle Photon using resistors such that we get the maximum voltage value from the sensors when the maximum force is applied. It also should be sensitive enough to output differentiable values for different activities. We obtained a noise-free signal covering almost the whole spectrum of the sensor when used with in series with a 470 Ω resistors in a voltage divider configuration as shown in Figure 2.

[1]http://www.trossenrobotics.com

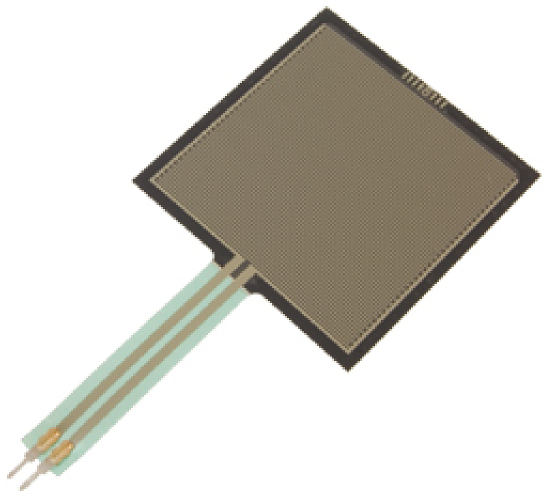

Figure 1: Force Sensing Resistor (FSR)

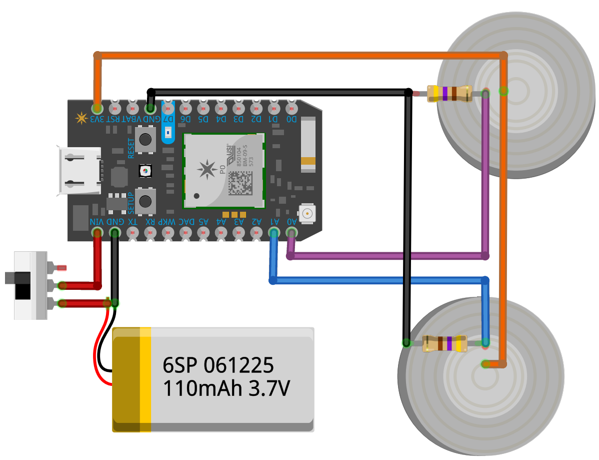

Figure 2: Wiring diagram of the Particle Photon with the FSRs, 470 Ω resistors and 3.7V Li-Po battery

2.2 Data Collection

To collect data from the pressure sensors in the shoe, we decided not to use the Particle Photon cloud which is based on Transmission Control Protocol (TCP) and is limited to one sample per second. Instead, we decided to use Universal Datagram Protocol (UDP) and our personal device as the server. We chose UDP because it is lightweight and faster than Transmission Control Protocol (TCP) as it has no error recovery. Since we can afford to lose out on some samples during transmission, UDP would best fit our needs. The pressure sensor in the front of the sole was connected to the Analog pin A1 of the Photon and the sensor in the back of the sole was connected to the pin A0. Values from the ADC were read and averaged over 5 samples. A 10 ms delay was added to the loop function in the firmware to smooth the transmission and receive data at a rate of 100 Hz. We experimented with a few delay values and found 10 ms best suited our needs. We stored the two sensor values in a 12 Byte buffer along with a sample number. The buffer was sent over a UDP packet to our laptop. The packets were received on the laptop, parsed and stored in plain text files. This process was repeated for all the activities.

We collected at least 10 instances of each activity for all three of the authors. For the walking, squatting and jumping activities we collected data for 20 steps (10 steps for each foot), 10 squats and 10 jumps respectively. For standing, sitting, standing up and sitting down activities we collected the same amount of data as the previous three activities with some variation in the number of samples. For the standing up activity the person would be seated for around 2 seconds before standing up and staying still for approximately 5 seconds. The sitting down activity consisted of the person being in a standing position for 2 seconds and then proceeding to sit on a chair placed in arms reach for approximately 5 seconds. Figure 5 shows the plots for data collected for all activities for one the authors. It might be difficult to see but note that all figures are not at the same range of values. After looking at the plots we decided there was no further need for smoothing or any additional filtering as it would only delay the activity detection pipeline. We also took care to collect sufficient data for each activity so that useful features could be computed from it. Collecting too much data would be wasteful and too little data would not provide enough information to able to classify the activity.

2.3 Feature Computation

In order to train our model we needed to extract meaningful information from the recorded data. A quick analysis of the data showed that some activities like walking, jumping and squatting are periodic in nature and we would have to extract these features. We have used Fast Fourier Transform to compute the three dominant frequencies and their amplitude. We also extracted how well the signal is autocorrelated. Other features include the mean, standard deviation, number of peaks and valleys of the signal and airtime, which is the amount of time the shoe spends in the air (based on empirical threshold). These features where selected in order to minimize the number of features but maintain a high accuracy score.

These features were computed from the data stream obtained from the front and back pressure sensors. In addition we computed the average of front and back sensor values and computed the same features on the average values. In total we ended up with 54 features, 17 each for the front, back and average sensor values.

New features where added incrementally only if they would help to arbitrate between activity selection. For example, we ran into issues while detecting the sitting down and standing up activities. Standing up was detected as the dominant activity over the sitting down ground truth. To solve this issue we created features to detect rising and falling edges in the data. Standing up would have a rising edge while sitting down would have a falling edge. After adding these features in our model, we observed a significant increase in

Figure 3: Particle Photon attached to the Shoe

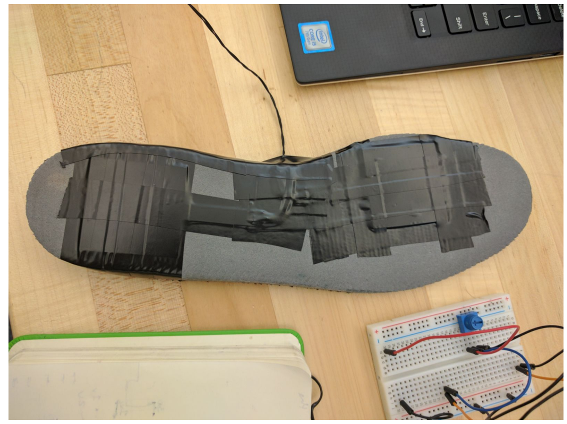

Figure 4: Placement of the pressure pads on the sole

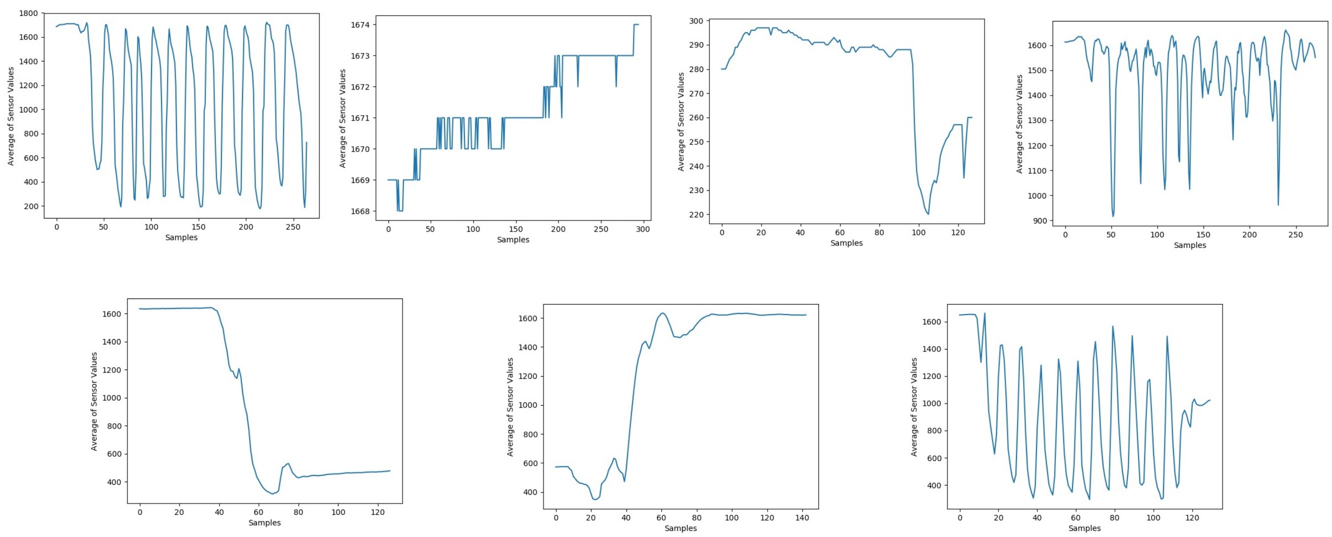

Figure 5: Plots showing data from activities. Top row from left: walking, standing, sitting, squatting. Bottom row from left: sitting down, standing up, jumping

accuracy.

We observed the relative importance of each feature using Scikit Learn (sklearn)[2] while after training the model and observed that there was no dominant feature. All features contributed equally. In some runs, few features were given zero importance while in different runs they had the same importance as other features. This is due to the random nature of the Random Forest classifier. As a result, we decided not to throw away any features as the accuracy was high and experimenting with different permutations would be time consuming.

2.4 Classification Strategies

This section describes the different machine learning algorithms we experimented with to train and test our model and mention the accuracies achieved with each. After computing relevant features from the data we decided to use WEKA [3] to see how descriptive the features were and also give us confidence to move forward in implementing the machine learning pipeline using sklearn. We computed feature from the data using a python script and saved the values in comma separated values format. We then created an Attribute-Relation File Format (ARFF) file from these values to feed

Table 1: Accuracy with Different Classification Models

Model	Accuracy
Random Forest	98.52%
K-Nearest Neighbors	83.30%
Support Vector Machine	94.63%

[2]http://scikit-learn.org/stable/

[3]http://www.cs.waikato.ac.nz/ml/weka/

Table 2: Experiments with Activity Sets and Results

Experiment	Ground Truth	Results
1	standing, jumping, standing	standing, standing up, jumping
2	standing, sitting down, sitting	standing, sitting down, sitting
3	sitting, standing up, standing	sitting, standing up, standing
4	standing, walking, standing	standing, walking, standing
5	standing, squatting, standing	standing, squatting, standing

into WEKA. The testing using WEKA was based on data from two of the authors and used 39 different features. For the final model we used data from all authors and added 15 new features that helped increase the accuracy by almost 5% from the initial 92%.

We used the Multilayer Perceptron (MLP) and Random forest classifier (RFC) in WEKA to train our model and check accuracy. The MLP gave an accuracy of 96.84% and the Random Forest classifier gave an accuracy of 92.63% using WEKA's default values. We used 10 fold cross-validation to calculate the accuracies. We noticed that although the Random Forest classifier had slightly lower accuracy, the training time was much lower than MLP. We tried using a MLP classifier in sklearn with the parameters used by WEKA but did not achieve high accuracy. To achieve high accuracy parameters like learning rate, number of hidden layers and momentum would have to be optimized using validation which would be time consuming. On the other hand, sklearn's RFC gave high accuracy without the need to tune any parameters. The faster prediction time in WEKA and taking instructor feedback to use RFC into account, we decided to go ahead and use RFC in sklearn. Before training, the activities were factorized into integers to facilitate the training process. The factorization occurred based on the order of activity instances fed into the training model. Although it does not affect the overall results, the activities were factorized into integers from 0 to 6 (for 7 activities) with the order being walking, standing, standing up, squatting, sitting down, sitting and jumping.

We used the Random Forest Classifier from sklearn with default parameters to train the model and perform validation. The data was divided into training and testing data randomly with approximately 85% data used for training and the rest used for testing. We trained 10 different models using sklearn's 'fit' method and found average the accuracy to be 98.52%. We also decided to use K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) with linear a kernel with default parameters to train and test our model to observe accuracies obtained from them as well. KNN gave an accuracy of 94.63% and SVM gave an accuracy of 83.30%. The accuracies are summarized are in Table 1. We decided to use RFC as it resulted in the highest accuracy.

For testing the model we trained on all of the data. There was a small chance during the training and testing process because of the random split that the training data did not have enough instances from a particular activity to train on leading to lower accuracy on that activity. By training on all of the data we ensured that all activities would be trained on in the final model. During the prediction phase, the model gave us confidence levels for each activity. The final prediction is then chosen to be the one with the highest confidence level. During testing we observed that the activity of sitting down was sometimes being detected as standing up. We tried solving this issue by looking at predictions from the SVM model and KNN model as well. We hoped that at least one of the models would give us the ground-truth with high confidence level, but we observed that the SVM and KNN models failed to do so and the RFC model was more accurate than the other models almost all of the time. In order to solve this issue we collected more data for the sitting down activity with some variations added which were not present in the initial ground-truth data.

3 EXPERIMENTS AND RESULTS

In this section we describe the validation experiments conducted and results obtained for activity classification in real time.

3.1 Validation with Activity Sets

Before we used our model for real time classification we wanted to test it on static data that consisted of multiple activities performed one after the other. To accomplish this, we conducted 5 experiments each involving 3 activities we wished to detect. Table 2 shows the experiments with ground truth and detected activity for data collected from one of the authors. Out of the 15 total activities, 14 were correctly detected resulting in an accuracy of 93.33%.

We used a window size of 280 samples with an overlap of 130 samples from the previous window to detect the activities from a continuous stream of data stored in a file beforehand. We achieved very good accuracy for the given window size and overlap. This experiment helped us validate that our pipeline should be able to detect activities if they were performed for a long enough duration so that sufficient data could generated for detection. We experimented with different window sizes. A large window size would have a bad resolution and two activities could be detected as one with the dominant one being chosen, while a small window would not be able to compute descriptive features from the data. Based on the results we obtained, we decided to use 300 samples as our window size for real time classification.

3.2 Real Time Classification

An application was developed that reads the values from the UDP socket and processes them trough the classification model resulting in a real time classification of the activity. To test the classifier in real time, we performed 60 activities (20 for each person) consisting of at least two instances of each activity. The activities were continuous with uninterrupted transitions from one activity to the next (standing, sitting down, sitting, standing up, etc.). From the 60 activities, 56 were classified correctly resulting in an accuracy of 93.33%.

4 FUTURE WORK

We now discuss possible improvements and additions that can be made to the sensing system in order to produce better results and extend this project to serve more purposes than detecting 7 activities. An improvement we suggest is the meta-learning of features. We computed features that would be descriptive and able distinguish each of the 7 activities. Although during the course of the project we observed the importance of each feature we did not dig deep in this direction due to time constraints. There are a lot more complex set of features that can be computed that may be more descriptive than the current set. Another improvement that can be helpful is updating the trained model to improve its accuracy. The model would be trained on the cloud and new ground truth data can be supplied by users when feasible using an app on their smartphone. This opens the possibility for crowd sensing. We made the assumption that the photon would have access to WiFi throughout the process. To make the system truly mobile a BLE module can be integrated for data streaming. An addition to this project is to use sensors in both shoes instead of just one. This would help distinguish the activities like jumping more easily and may also give information about gait and balance. An inertial measurement unit (IMU) could also be incorporated that would be helpful in detecting faults in activities and detect more complex activities.

5 CONCLUSION

We were able to successfully implement a smart shoe capable of detecting seven activities with **93.3%** accuracy with different users. The 'Aware Shoe' can enrich current activity and fitness trackers. The system was economical and ergonomic. We experimented with different machine learning models and features and optimized the system based on our observations. There is a lot of scope for improvement as discussed in the previous section but we delivered a sensing system we are really proud of, taking into consideration resource and time constraints.

ACKNOWLEDGMENTS

We would like to thank Prof. Mayank Goel and Prof. Yuvraj Agarwal for providing feedback and guidance during the course of this project. Working on this project was an enriching experience where we picked up new skills that will prove invaluable in the future.