Qeexo Machine Learning Challenge

Abstract: The challenge poses a unique problem of deft touch detection on PDAs. The problem presented with the challenge is a binary classification problem to separate different way humans touch the PDAs in daily lives. The problem specifically poses a challenge in separating pad touch with knuckle touch. With the given data, the proposed solution tries to leverage the information from the current data and use some heuristic in designing feature for classification.

Data Pre-Processing & Feature Engineering.

The current features presented with the dataset include the touch coordinates of the user touch with pressure, orientation and accelerometer vibration data as a16 bit integer array. As part of feature engineering, I use the touch attributes as raw values for all my classification experiment, however I try and extract features from the 256-dimensional accelerometer vibration data. As a part of feature extraction, the signal representation is converted from a time domain to joint time frequency domain using Continuous Wavelet Transform (CWT). “Morlet” is used as the basis wavelet for transformation. The following figure 1 shows sample plot of original signal along with Continuous wavelet transform representation. The scales values for the transform where chosen after trying to experiment with different scales values to understand the nature of the signal. One of the interesting facts that can be observed through wavelet transform was the spread of energy with respect to touch. As seen in figure 2, we see that energy is spread out more in knuckle touch and its vibration data signal while the pad touch (figure1) has more concentrated energy distribution. As a part of the feature matrix for classification magnitude of the CWT matrix is used and with the minmax normalization performed with respect to each sample.

Methodology:

With an idea to learn feature from the continuous wavelet transform of the signal, a convolutional neural network architecture is designed to learn the 2D image equivalent representation of the accelerometer vibration signal. The architecture designed is a combination of convolution layer with depthwise – separable convolution layer. The design of the architecture is done taking into consideration application on mobile devices, hence the use of depthwise-separable convolution layer not only provide speed in convergence of the algorithm but also help in reducing the number of learning parameter which helps in reduce the amount of space required by the trained model. Along with the conv layer, Batch Normalization and Max-pooling layer are used to extract superior invariant feature and reduce internal covariant shift. The design and specification of the architecture can be found in the figure 3.

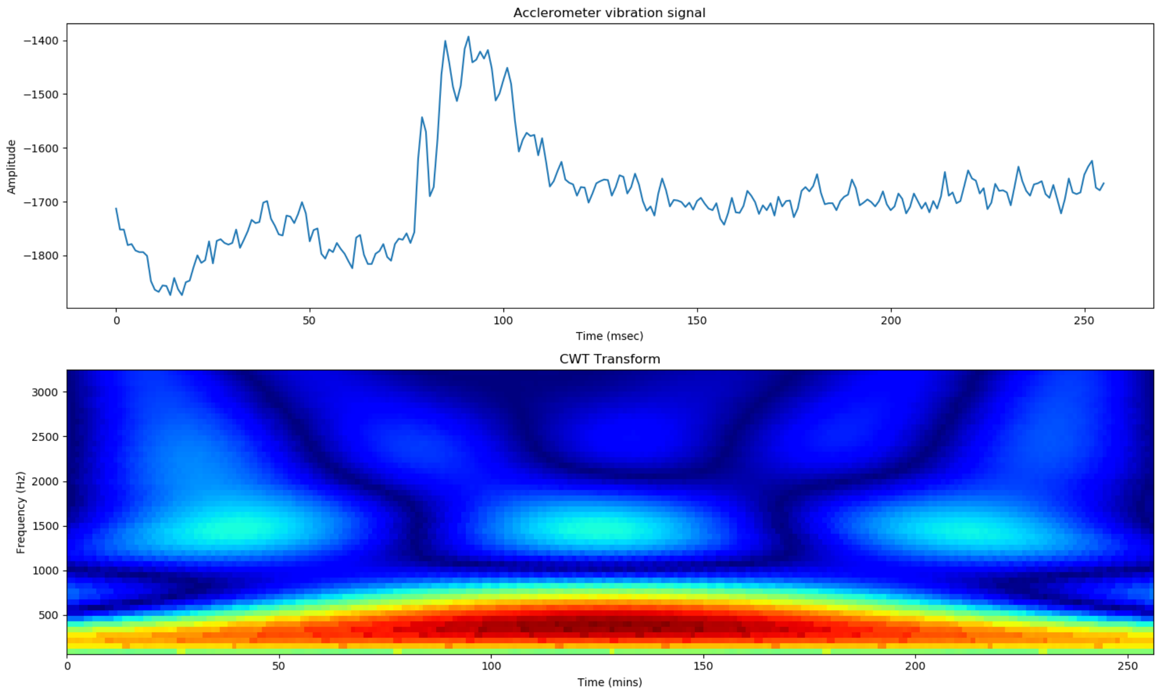

Figure 1. CWT representation of a pad touch accelerometer signal

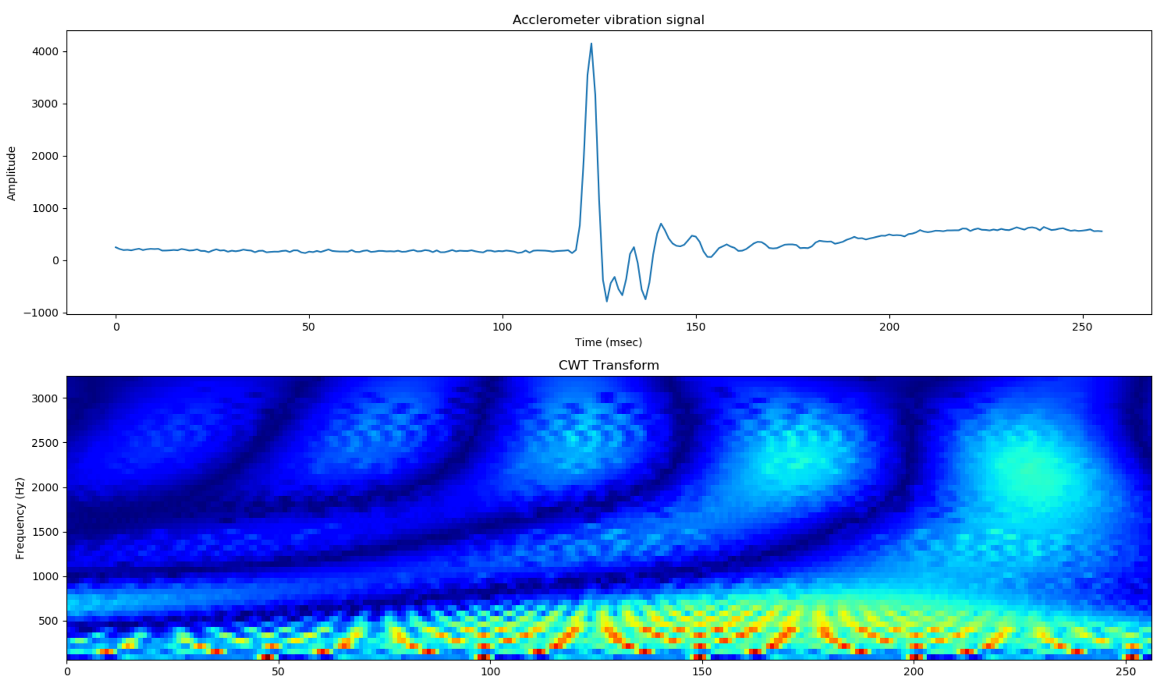

Figure 2: CWT representation of a knuckle touch accelerometer signal

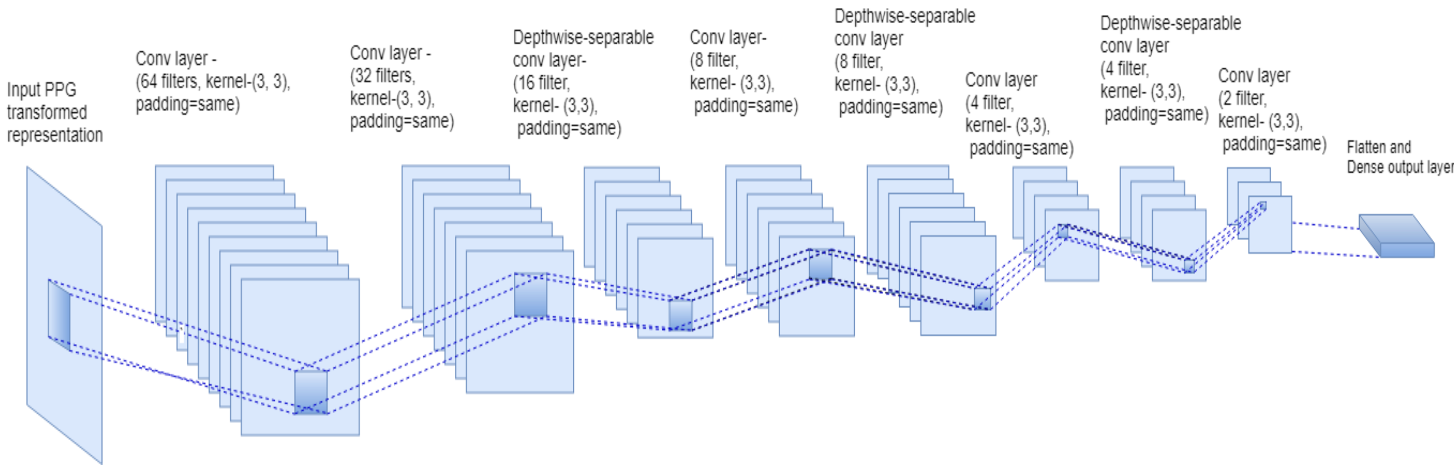

Figure 4: Convolutional Neural Network Architecture

Training:

The convolutional neural architecture is trained to minimize the binary cross entropy loss with L2 regularization employed with penalty of 0.01. The algorithm is trained using the Early Stopping as the training criterion. The training batch size for the algorithm is kept at 64 taking into consideration the memory requirement and the available memory for training the algorithm. The parameters are updated using stochastic gradient descent with Adam as the training optimizer to minimize the generalization error.

Results & Experiments:

The performance of the proposed architecture is compared with baseline statistical machine learning methods, such as Logistic Regression and Random Forest. 10-Fold cross validation with stratified sampling is used as the training strategy for the statistical models. Hyperparameter optimization is performed through grid search. The best estimated parameter from the grid search are then used to retrain the model on the whole dataset to measure the performance of algorithm on the held-out test set. We keep the random seed constant at initial train test split so that the model performances are comparable across experiments. Further we try and experiment with the input feature into the statistical model, the table below provides an overview of experiments performed with different input features. The performance metrics used here are accuracy, AUROC (Area under receiver operating characteristic), F1 score, sensitivity or recall, specificity, PPV or Precision and negative predictive values. Different training strategies are also employed

Algorithm	feature	Training method	acc	AUROC	f1_score	sensitivity	specificity	PPV	NPV
Logistic Regression	touch feature	General method	79.86	0.86	0.81	0.9	0.69	0.74	0.88
Random Forest	touch feature	General method	80.15	0.87	0.82	0.92	0.67	0.73	0.9

Table1: Present the summary of 10-Fold cross validation and Grid Search for statistical methods using a general training set and general test set.

As the table presents results of cross validation on general test set and general training set, we also try and change the training dataset by using user specific information, such that we split the train and test set by random selecting users 5 users to represent test set and rest 37 users represent the training set for the statistical model. The table below provides and overview of the results from the user-based training criterion. We also experiment with the surface base training criterion for the dataset where the training data and cross validation is performed on the touch on surface table and tested on the surface hand across users.

Algorithm	feature	Training method	acc	AUROC	f1_score	sensitivity	specificity	PPV	NPV
Logistic Regression	raw touch feature	user based training	90.27	0.96	0.89	0.86	0.93	0.93	0.87
Random Forest	raw touch feature	user based training	90.34	0.96	0.86	0.86	0.93	0.93	0.87
Logistic Regression	raw touch feature	Surface base training	74.42	0.92	0.67	0.54	0.93	0.89	0.67
Random Forest	raw touch feature	Surface base training	80.34	0.91	0.78	0.71	0.88	0.85	0.80

Table 2: Present the summary of the different training strategies performed with 10-Fold cross validation and Grid Search for statistical methods

Like the experiments performed for the baseline model, same setup is used for the proposed deep learning model. The table below provides and overview of the results from the experiments performed on the convolutional neural network base feature model.

Algorithm	feature	Training method	acc	AUROC	f1_score	sensitivity	specificity	PPV/	NPV
CNN classification	Accel + CWT	General method	87.56	0.95	0.94	0.79	0.82	0.94	0.82
CNN classification	Accel + CWT	Surface based training	73	0.85	0.65	0.51	0.94	0.89	0.66
CNN classification	Accel + CWT	user base training	86.67	0.94	0.85	0.79	0.93	0.92	0.82

Table 3: Presents the summary of the different training strategy on cross validation split for the deep learning proposed algorithm

Discussion:

The experiments performed on the training set using different features suggest that using a user bases training set works best on the statistical machine learning models. Further we see that using the user base training strategy the baseline models seems to be performing better than the deep learning model, as a speculation it could be due to lower amount of variance in the CWT representation that might be the cause of lower performance of the deep learning model. Further, if we can experiment with different proportion of training and test for the user-based training strategies we would be able to conclude with confidence on the performance gap between the two methods. However, the deep leaning model performs quite better than the baseline models on the general training strategy. Using Principal Component Analysis (PCA) as addon feature extraction method over CWT by flattening the CWT array. The feature extraction method did not perform well on the statistical methods and hence the result is not been reported here. Further, from the experiment we understand that training and testing using the surface strategy does decrease the performance of classification. However here the results on show performance of the algorithm by training on the “table” surface and testing on the “hand” surface, it would be interesting to know what happens if we reverse the roles of train and test surfaces. Due to time constraint that experiment has not been reported here. Further also raw signal features experiment was performed before using CWT on statistical methods, however it did not produce good enough results and hence not reported here.

Conclusion:

The study gives some interesting fact about the touch data set and it would be interesting to explore further by using normalized feature to see if that increases the performance on the deep learning model. Also training on larger dataset might help improve the performance on the CNN model and may even outperform the statistical model as there is clear distinction in the CWT outcomes of the two different types of touch.