

Motion detection for ‘Falling Over’

Ruofan Wang

Github link: <https://github.com/TimKingdom666/0018-DLSN.git>

Introduction

This project is aiming to recognise 4 different motions of the Arduino Nano 33, and the 4 motions are defined as idle, walk, run and drop respectively. The principal inspirations of this project are from our smart wearing and especially smart watch. There are researches claims that smart watches have potential to help us monitoring our health conditions by enabling self-monitoring of personal activity; obtaining feedback based on activity measures; allowing for in-situ surveys to identify patterns of behaviour; and supporting bi-directional communication with health care providers and family members (Reeder, 2016). Apart from that, due to the fantastic interactive system between smart watches and smart phone created by technology companies like Apple, Samsung and Huawei, smart watches are becoming familiar to more and more people. Which means that smart watches are becoming common in our lives and becoming convenient for monitoring health conditions of family members instead of going to hospital regularly (Baig, 2020).

As smart wearing is so useful and convenient, I was wondering how to develop more functions to further improve the use of it. The detection and warning of fall over is the answer as it would help to remind parents if their children of old generations fall over by accident. By setting 4 different motions for the Arduino, I have collected the motions of no motions (idle), swinging hands while walking and running, and the freefall from about 1 metres height (Drop).

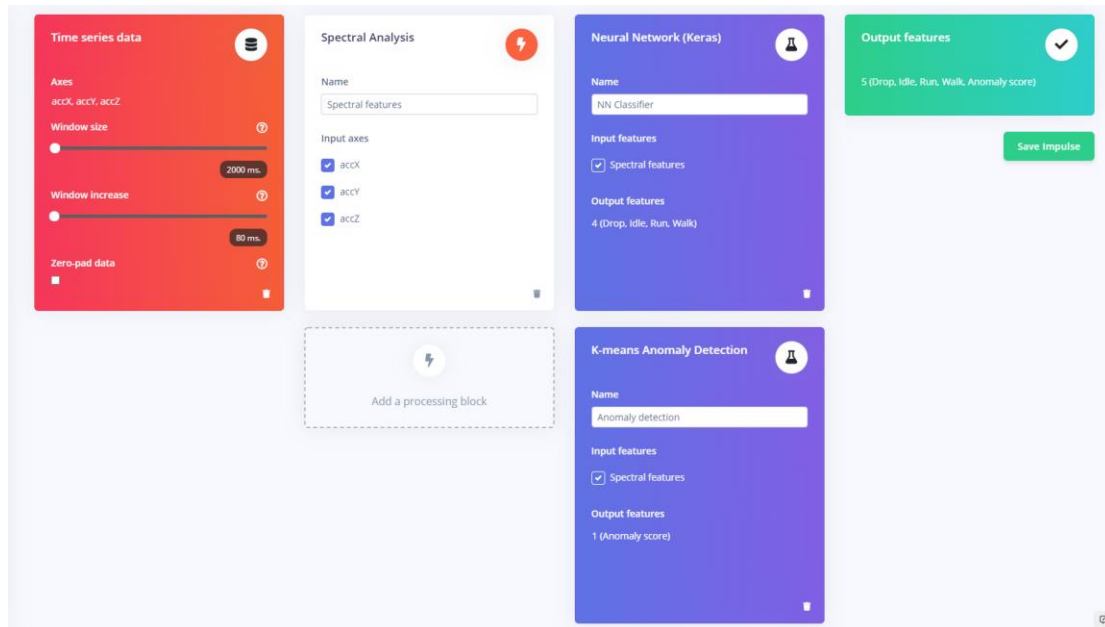
Research Question

The main problem is how to detect different motions and identify fall over correctly and hence be able to send warning messages for further development.

Application Overview

Edge impulse is the tool for me to analysis the motion data collected from the Arduino Nano 33. There are 4 stages for analysis after the data is collected. First of all, we need to create impulse that takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data. For my impulse, I set 3 axes to create a three-dimensional Spectral Analysis, with window size of 2000ms and 80ms increase.

Besides, the neural network (Keras) calling NN Classifier is created to repeat the training to improve the accuracy of prediction, and final stage is the K-means Anomaly Detection, which is used to separate the classified data from the training data via K-means mode.



probably ~200 words and a diagram is usually good to convey your design!

Data

This model is aiming to recognise different types of exercise and fall over motions via the motions collected by accelerometer of Arduino Nano 33. For these four motions, the data for 'idle' is collected by simply setting the Arduino stably on the desk for 1 min, the motion of 'walk' is the motion of the swinging of my hands. I recorded the swinging of my hands by 3 mins as the data of 'walk'. The motion of 'Run' is similar to 'walk', which is also the recording of the swinging of my hands, yet with higher frequency as I was juggling instead of walking. Finally, the motion of 'Drop' is the freefall motion of the Arduino from about 1 m height, as the motion is considerably simple, I repeated the process by 10 times to improve the quality of the data. Among that, for all 4 types of motions, only the 'built-in accelerometer' is used in Arduino, the Frequency of sampling is set at 62,5 Hz.

Collected data



| SAMPLE NAME | LABEL | ADDED | LENGTH | |
|---------------|-------|-----------------|--------|---|
| Drop.21a9el6h | Drop | Today, 02:58:00 | 2s | ⋮ |
| Drop.21a9ea72 | Drop | Today, 02:57:49 | 2s | ⋮ |
| Drop | Drop | Today, 02:55:47 | 2s | ⋮ |
| Run.21a9a7fv | Drop | Today, 02:55:35 | 2s | ⋮ |
| Run.21a99n9m | Drop | Today, 02:55:18 | 2s | ⋮ |
| Run.21a99aed | Drop | Today, 02:55:05 | 2s | ⋮ |
| Drop | Drop | Today, 02:54:53 | 2s | ⋮ |
| Run.21a98ih4 | Drop | Today, 02:54:41 | 2s | ⋮ |
| Run.21a98378 | Drop | Today, 02:54:25 | 2s | ⋮ |
| Drop | Drop | Today, 02:53:33 | 2s | ⋮ |
| Run.21a926db | Run | Today, 02:51:12 | 3m 0s | ⋮ |
| Idle.21a8qvns | Idle | Today, 02:47:16 | 3m 0s | ⋮ |

Model

The Models I have been used are Nearest Neighbour Classifier for Neural Network (Keras) and K-means Anomaly Detection. The 1-N-N classifier is one of the oldest methods known. The idea is extremely simple: to classify X find its closest neighbour among the training points (call it X',) and assign to X the label of X' (Castelli, 1997). On the other hand, K-means model identifies k number of centroids, and then allocate all the other point to its closest centroids, and ensuring the cluster is as small as possible, the K-means stands for the average of the data which is contributed to find the centroid (Garbade, 2018).

The reason I chose NN classifier for my project is due to that NN classifier achieves consistently high performance and involves a training set of both positive and negative cases, and no assumptions of the distribution of the training examples are required. Besides, the advantages of K-means clusters make it become appropriate for my project, the advantages include that it is relatively simple to implement, which means it is easy to adapt to new examples. Apart from that, K-means cluster scales to large data sets,

guarantees convergence, being able to warm-start the positions of centroids, and generalizes to clusters of different shapes and sizes (Sobiech, 2017).



Experiments

After the training data is collected and analysed, the features about the dataset is recorded. Hence, I used the live classification to test my training data.

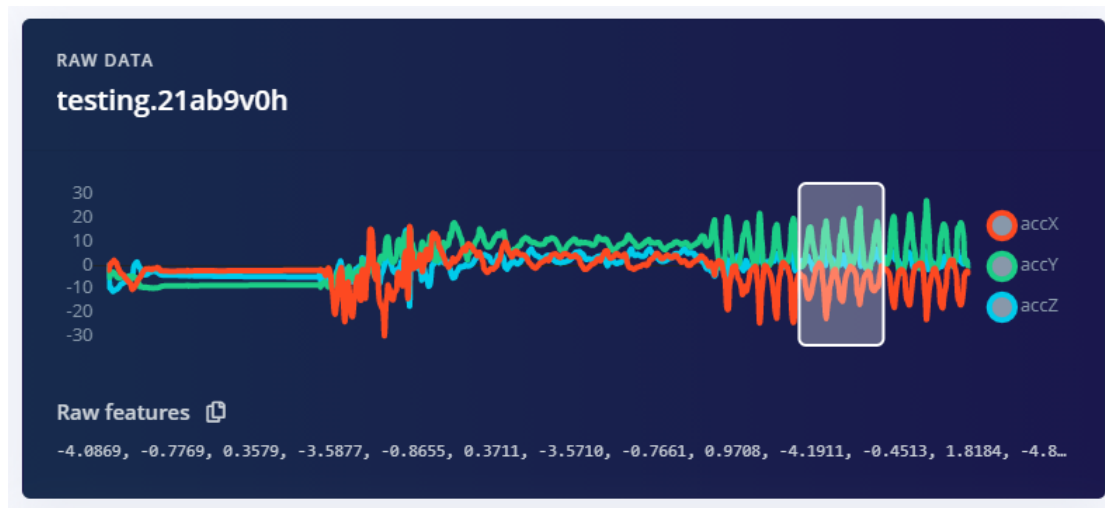
First of all, I recorded 4 different motions and named them testing data. After that, the classification process will predict the motion of the testing data based on the training model I have recorded before, as the result, it will show the percentage of these different motions, if the testing process cannot recognise the motion, it would regard it as

uncertain and anomaly. Furthermore, I have record a combination of 4 different motions randomly for about 1 mins long, the testing process will then demonstrate the motion it recognises at different timestamp, and colour the uncertain motion in red in the detailed result list.

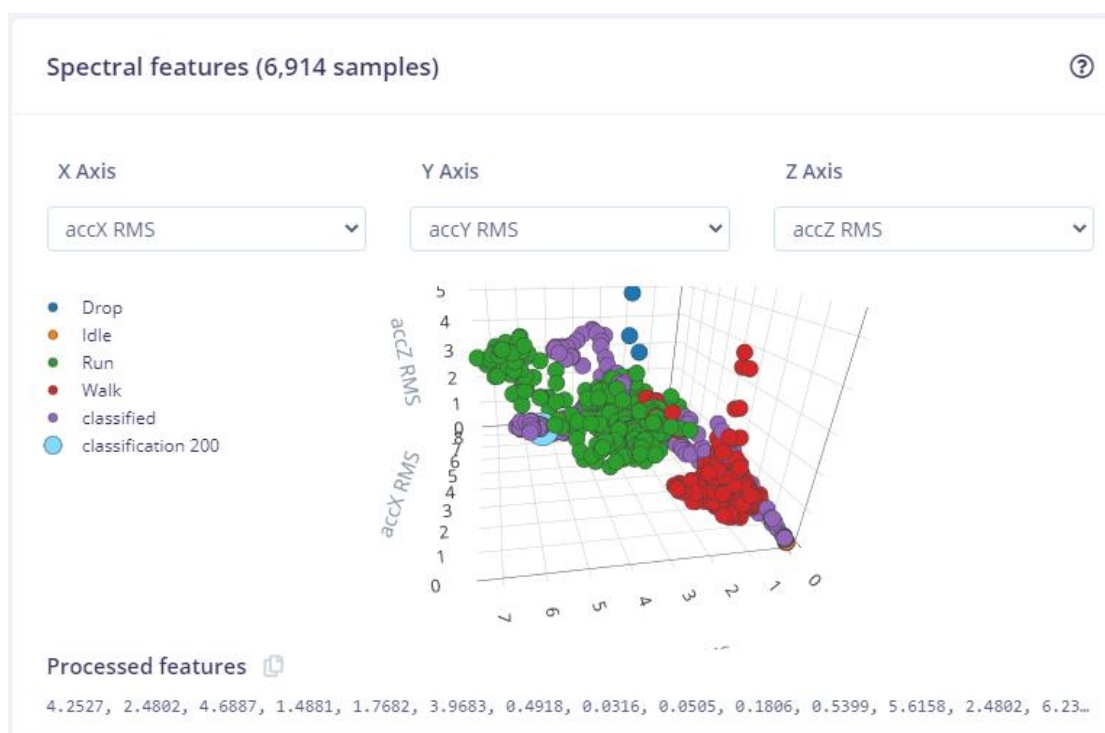
| Detailed result | | | | | | <input checked="" type="checkbox"/> Show only unknowns |
|-----------------|------|------|------|------|---------|--|
| TIMESTAMP | DROP | IDLE | RUN | WALK | ANOMALY | |
| 1120 | 0 | 0.44 | 0 | 0.56 | 0.31 | |
| 3280 | 0 | 1.00 | 0 | 0 | 0.30 | |
| 3600 | 0 | 0 | 0 | 0.99 | 0.41 | |
| 3680 | 0.24 | 0.06 | 0.10 | 0.61 | 0.58 | |
| 3760 | 0 | 0 | 0 | 0.99 | 0.51 | |
| 3840 | 0 | 0 | 0 | 1.00 | 0.32 | |
| 4240 | 0.56 | 0.03 | 0 | 0.41 | -0.39 | |
| 4560 | 0.47 | 0 | 0.53 | 0 | -0.60 | |
| 8640 | 0 | 0 | 0.48 | 0.51 | 0.07 | |
| 12720 | 0.50 | 0 | 0.49 | 0 | -0.39 | |
| 12800 | 0.50 | 0.01 | 0.25 | 0.24 | -0.43 | |

Results and Observations

Synthesis the main results and observations you made from building the project. Did it work perfectly? Why not? What worked and what didn't? Why? What would you do next if you had more time?



This is the testing data of the combination of 4 different motions. The testing process demonstrate a considerably high quality prediction as the result. The number of uncertain values are considerably small, and the accuracy of predicting the motions is significant high.



For the spectral features which contains 6914 samples also shows that the most of the dots with the same colour gather together, which does meet the features of the testing data as my performance of the 4 motions all have similar posture.



The Anomaly explorer also demonstrate the quality of the prediction. As we can see, the blue dots are the training data, and most of the blue dots are inside their clusters (the blue circles), only few dots are outside the dark blue clusters. The orange dots are the classified data been used as testing data. Obviously that most of the classified dots are inside the dark blue clusters and even though there are still some classified dots are close to the edge of the cluster, no orange dots are outside the largest cluster as well.

Last training performance (validation set)

ACCURACY 99.9% LOSS 0.00

Confusion matrix (validation set)

| | DROP | IDLE | RUN | WALK |
|----------|------|------|-------|------|
| DROP | 0% | 0% | 0% | 100% |
| IDLE | 0% | 100% | 0% | 0% |
| RUN | 0% | 0% | 99.8% | 0.2% |
| WALK | 0% | 0% | 0% | 100% |
| F1 SCORE | 0.00 | 1.00 | 1.00 | 1.00 |

On the other hand, just as the K-means cluster graph shows, the training data has significantly high accuracy. From the training model we can see that the accuracy of the training performance has reached 99.9% with no loss. Yet this is not necessarily good as it be overfitting, because it could mean that the neural network is too tuned for the specific test and may perform poorly when it meets new data.

Finally, even though the testing results are considerably good but the accuracy in reality might not perform as well as the testing result. In this project due to the constrain of the

length of wire, I used free fall to simulate fall over. Apart from this, due to that Arduino has to be connected with my laptop though the wire, the motion of walking and running are simply the hand swinging with different frequency as the simulation. Which means that except 'idle', all other three motions have differences to the real performance. In order to improve the quality of the project, making sure the Arduino will not be restricted by the wire, and hence increasing the time of recording of walking and running. For 'drop' motion, setting the freefall from different height with different land materials could also improve the quality of simulation, furthermore, record the real motion of fall over by using cushion to reduce the damage could fundamentally solve the problem.

probably ~300 words and remember images and diagrams bring results to life!

Bibliography

Reeder, B. (2016, October 1). Health at hand: A systematic review of smart watch uses for health and wellness. ScienceDirect. <https://www.sciencedirect.com/science/article/pii/S1532046416301137>

Baig, B. E. C. (2020, September 28). Newest Smartwatches Move From Tracking Fitness to Monitoring Health. AARP. <https://www.aarp.org/home-family/personal-technology/info-2020/smartwatches.html>

Castelli, V. (1997). Nearest-Neighbor Classifier. <https://Www.Ee.Columbia.Edu/~vittorio/>. <https://www.ee.columbia.edu/~vittorio/lecture8.pdf>

Garbade, M. J. (2018, September 13). Understanding K-means Clustering in Machine Learning. Medium. <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

Sobiech, J. (2017, July 26). Observing lake- and river-ice decay with SAR: advantages and limitations of the unsupervised k-means classification approach | Annals of Glaciology. Cambridge Core. <https://www.cambridge.org/core/journals/annals-of-glaciology/article/observing-lake-and-riverice-decay-with-sar-advantages-and-limitations-of-the-unsupervised-kmeans-classification-approach/2B76DB54EA4C9E898F214DE7E400A5CC>

k-Means Advantages and Disadvantages | Clustering in Machine Learning. (2021). Google Developers. <https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages>

