Hand Gesture Recognition for Soldier Support

Siddharth Nayak
Indian Institute of Technology Madras
Chennai, India
ee16b073@smail.iitm.ac.in

Abstract

Soldiers communicate with each other through gestures. But sometimes those gestures are not visible due to obstructions or poor lighting. For that purpose an instrument is required to record the gesture and send it to the fellow soldiers. The two options for gesture recognition are through Computer Vision and through some sensors attached to the hands.The first option is not viable in this case as proper lighting is required for recognition through Computer Vision. Hence the second option of using sensors for recognition has to be used. We present a system which recognises the gestures and sends them to other soldiers.

1 Introduction

In the many dangerous situations our soldiers are made to face, one thing is certain, that effective communication between them is an absolute necessity. Even in circumstances where they may not be able to see or hear one another, interaction between themselves is an absolute must. That is where our prototype comes into action. A lot of work has been carried out in recognising gestures. Most of them use computer vision based approaches. We plan on implementing this technology to make special gloves for our soldiers which recognizes gestures and transmits it between the team. This is particularly helpful during the many stealth missions conducted. Signals like, where one soldier plans to go, to whether an enemy has been spotted etc. are all vital for the success of the operation, and additionally, prevents lapses, ie. lives, time, and property can be saved.

2 Construction

The given gestures(Figure 1) include motions of fingers, wrist and elbow. Hence to detect any changes in them we have used one flex sensors which detects the amount by which it has been bent at each of these joints. To take into account for the dynamic gestures an Inertial Measurement Unit(IMU-MPU-9250) was used.The parameters used from the IMU are Acceleration, Gyroscopic acceleration and angles in all three axes. An Arduino Mega was used to receive the signals from the sensors and send it to the processor.

Figure 1: Gestures recognised by the First Prototype

3 Dataset

Because we made our own design for the sensor positioning on the arms there was no dataset available for training the model. We recorded our own dataset by performing the gestures repeatedly in different sessions. Three different people recording the gestures for variance in the dataset.

3.1 Static Gestures Data:

The static gestures include the gestures which do not include movements of the hands. We have just used the sensor values from the flex sensors and the angles derived from the accelerometer. The reason for this has been explained in the algorithm section. The flex sensor values capture the amount of bend in each of the fingers, wrist and elbow. The other three features used are the angles which the hand makes with the three axes (X,Y,Z) with the Z-axis being perpendicular to the ground.

3.2 Dynamic Gestures Data:

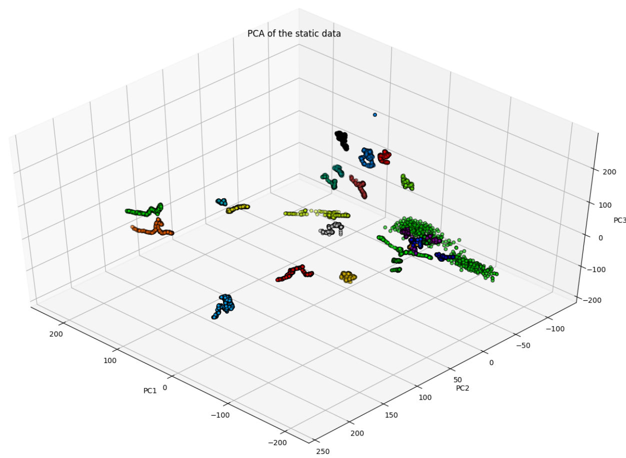
The dynamic gestures include the gestures which do include movements of the hands. For these type of gestures we use the flex sensor values, angles, accelerations in all three axes and the angular acceleration in all three axes.

4 Algorithm

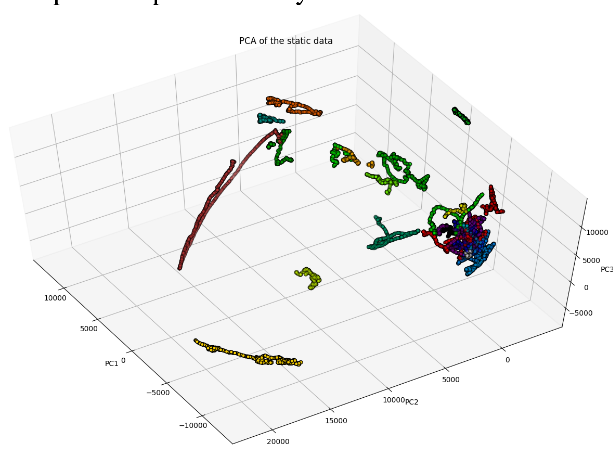
First of all we provide a button to be pressed for the person to specify whether he/she will be making a static of dynamic gesture. The person will then make the gesture and the sensor values will be recorded till the person releases the button. The data recorded will be a '16 x t' vector, where 't' is the number of time steps. The 16 features include: 5 flex sensor values from the fingers, 1 flex sensor value from the wrist, 1 flex sensor value from the elbow, angles made by the palm with all three axes, linear acceleration of the palm in all three axes and the gyroscopic acceleration in all three axes. The data received is filtered using a Kalman Filter to remove the noise from the Inertial Measurement Unit(IMU). After filtering the data, 50 samples are taken from the time series. If the length of the sequence is less than 50 then the values are obtained using linear interpolation. Then, the values are normalised. This '16 x 50' vector is then trained on two different Support Vector Machines(SVM) with a Radial Basis Function(Gaussian Kernel). This method avoids the usage of LSTMs, which require a lot of data for training. The first SVM is activated if the data corresponds to a static gesture and the second one is activated if the data corresponds to a dynamic gesture. The second SVM (dynamic gestures) is nested. This is because the gestures like 'Ammunition', 'Vehicle', 'Column Formation', 'File Formation' have movements in the hand which are similar. And according to our observations by using a non-nested SVM, the model was giving more importance to the acceleration features. For this reason we added another SVM to specifically classify these gestures.

4.1 Reason for using different features for static and dynamic gestures

The features used for the recognition of statics gestures are the flex sensor values and the angles with all three axes. We discard the linear acceleration and the gyroscopic acceleration values. The reason for this being that the acceleration values(both linear and gyroscopic) are not important for the static gestures. Figure 2(a) shows the principal component analysis(PCA) of the datapoints projected from 10 dimensions to 3. Clearly the clusters are separate and we can classify them more accurately. If we use the acceleration values as shown in Figure 2(b) the clusters have a trail which elongates into other clusters. This makes the separation of the clusters with hyperplanes difficult.

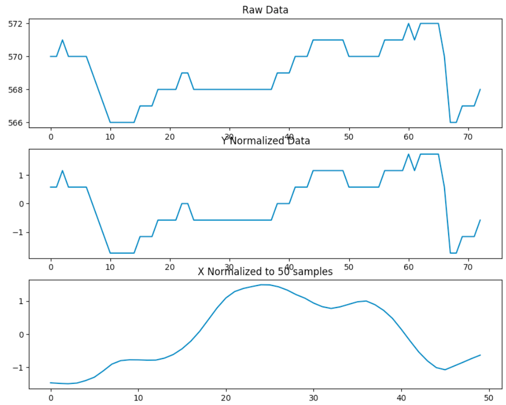


(a) Principal Component Analysis without acceleration values

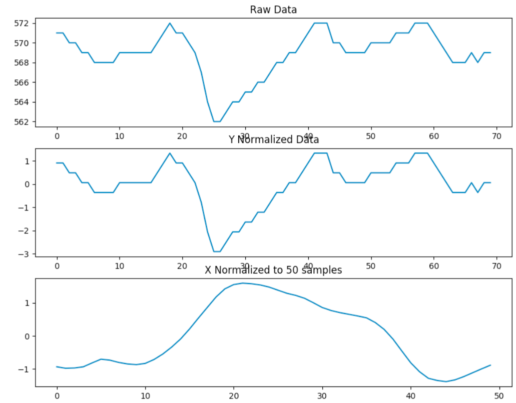


(b) Principal Component Analysis with acceleration values

Figure 2: Principal Component Analysis(PCA) of static gestures.

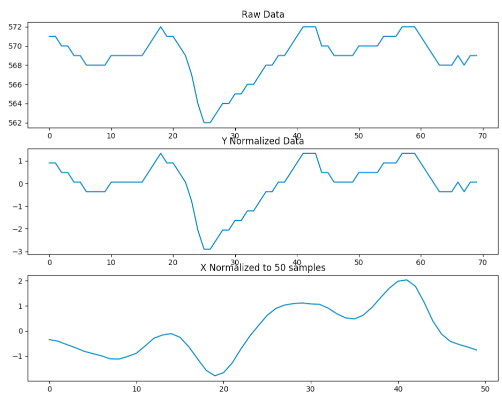


(a) Sample of the gesture 'door'

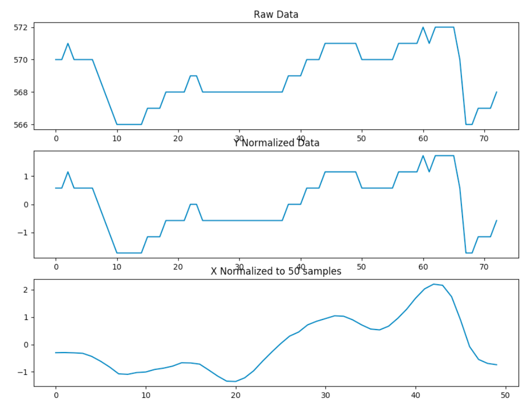


(b) Sample of the gesture 'door'

Figure 3: Two datapoints of the gesture 'Door'



(a) Sample of the gesture 'window'



(b) Sample of the gesture 'window'

Figure 4: Two datapoints of the gesture 'Window'

5 Results

The system recognises the gestures in the test data with 99.2% accuracy and recognised all the gestures we tried in real time without much of a latency.

6 Extensions and Improvements

We want to make an end-to-end system which recognises gestures. We want to make the system cost-effective as well as make it robust to the rough usage in the military. Our main goals for the second prototype are:

6.1 Reducing the wiring along the arm

The first prototype had a lot of wires along the arm for the sensors on the elbow, which hinders the motion of the person. Also, the wiring would come out if the movement was very vigorous. We want to add bluetooth modules to remove the wires along the arms. We also want to remove the usage of the button which signifies the start of a static or dynamic gesture. By doing so, we make a more cleaner, aesthetic model, that allows more freedom of motion and maneuverability, without having to worry about weight, wires getting caught up, and additionally prevents any possible damage.

6.2 Enabling broadcasting of signal between soldiers with encryption

We plan on writing our own protocols to transmit the recognised gestures signal to the other soldiers. With us writing our own protocols, it would be difficult to hack the signals ensuring a safe transmission of data.

6.3 Adding more gestures

As seen in Figure 2, there is room for more gestures to be added. We would like to We also plan to maintain the high recognition accuracy and with low false positives and false negatives.

7 Acknowledgements

I would like to thank Centre for Innovation, IIT Madras for funding the project. I would also like to thank Rohan Rao, Akash Anandan and Rishanth Maanav for their valuable suggestions in the project. Suhas Kumar, Guhan Narayanan and G. Abhilash from IIT Madras are working on the second prototype of the model.

8 References

Bernhard E Boser, Isabelle M Guyon, Vladimir N Vapnik. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory.

Sepp Hochreiter, Jurgen Schmidhuber. Long Short-Term Memory. Journal Neural Computation archive Volume 9 Issue 8, November 15, 1997 Pages 1735-1780 .

Kalman, Rudolph Emil. A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering-1960 Volume 82, Number Series D, Pages 35-45