

Character Recognition through Gestures Using SVM Algorithm

S. Saravanan¹, Dr. S.Govindrajan², Himanshu Bansal³, Archit Singh⁴

¹Assistant Professor(Sr.G), SRM Institute of Science and Technology, Department of Computer Science and Engineering, Chennai, India..

²SRM Institute of Medical Sciences, Department of Electronic data processing (EDP), Chennai, India.

³Final year student, SRM-IST, Department of Computer Science and Engineering, Chennai, India.

⁴Final year student, SRM -IST, Department of Computer Science and Engineering, Chennai, India.

Abstract

The main objective of the paper is to develop a Machine Learning based system that uses gestures to perform functions. There are a lot of gesture based applications existing in today's world but every time we want to use a gesture controlled application we need to learn the predefined gestures and the functionality is also limited to the defaults provided with it. For this project, we aim to build a system which can be trained to recognize the gestures we make and perform the dedicated function we decide for it. We'll be demonstrating this by training the system to recognize letters by the gestures we make in air. To build this system, we'll be using an Arduino Board interfaced with an accelerometer. The device can be attached to the user's hand. The accelerometer will provide input to the microcontroller about the hand's (which is being used to make the gesture) coordinates. The algorithm will pick up this data and maintain a database to recognize each gesture differently. Once we train the system with the same gesture multiple times it will gather enough data to have an estimate of what the gesture should look like. This gesture can then be assigned to perform a task on the computer. We are also aiming to add Bluetooth support so as to make the gesture device wireless and more flexible.

Keywords: ARDUINO, HC- Bluetooth Module- HC-05/06, Accelerometer.

INTRODUCTION

Proposal

Today Machine Learning is taking over everything. This is due to the fact that the machine can be made to learn some amazing things. Gesture controlled applications are already present but they are not user friendly. They want the user to learn the device oriented gestures for optimum working. We can use machine learning to change this situation. We propose to build a system where the user can use their own custom gestures for different applications. To demonstrate this, we will be making a simple example of such a machine which will be using the gestures used by our hand to write in English. We won't be requiring any pen or paper to do so.

What is Machine Learning?

Machine learning is a field of software engineering that enables PC frameworks to "learn" (i.e. logically enhance

execution on a particular errand) with information, without being unequivocally customized.

The name Machine learning was instituted in 1959 by Arthur Samuel. Developed from the investigation of example acknowledgment and computational learning hypothesis in computerized reasoning, machine learning investigates the examination and development of calculations that can gain from and make expectations on information such calculations conquer following entirely static ace gram guidelines by settling on information driven forecasts or choices, through building a model from test inputs. Machine learning is utilized in a scope of figuring undertakings where planning and programming express calculations with great execution is troublesome or infeasible; illustration applications incorporate email separating, location of net-work gatecrashers or malevolent insiders working towards an information rupture, optical character acknowledgment (OCR), figuring out how to rank, and PC vision.

RELATED WORK

Machine Learning

Louridas and Ebert (2016) : In machine learning, a computer first learns to perform a task by studying a training set of examples. The computer then performs the same task with data it hasn't encountered before. This article presents a brief overview of machine-learning technologies, with a concrete case study from code analysis.

A data driven in-air-handwriting biometric authentication system Lu et al. (2017) :

In this paper, a new finger-gesture-based authentication method, where the in-air-handwriting of each user is captured by wearable inertial sensors, is proposed. The approach is featured with the utilization of both the content and the writing convention, which are proven to be essential for the user identification problem by the experiments. A support vector machine (SVM) classifier is built based on the features extracted from the hand motion signals. To quantitatively benchmark the proposed framework, a prototype system is built with a custom data glove device.

An economical air writing system converting finger movements to text using web camera:

Ramasamy et al. (2016) : In this paper, a "Finger Motion Tracking System" is pro-posed that intends to identify the English character written in air using our finger. The identified character is converted into text and displayed on the screen. A web camera is used to capture the LED fitted finger movements and the patterns are identified with one of the

characters available in the database. After identification, the characters are printed onto the display screen. A red-colored LED pointed light source is attached to the finger of the user to ease down the process of finger movement tracking, as it is easier to track down the red color than to track the color of the moving finger. It is assumed that the alphabet drawn is a valid English alphabet and there is no red-colored object other than the LED light within the focus of the web camera.

SYSTEM MODEL

Circuit Diagram

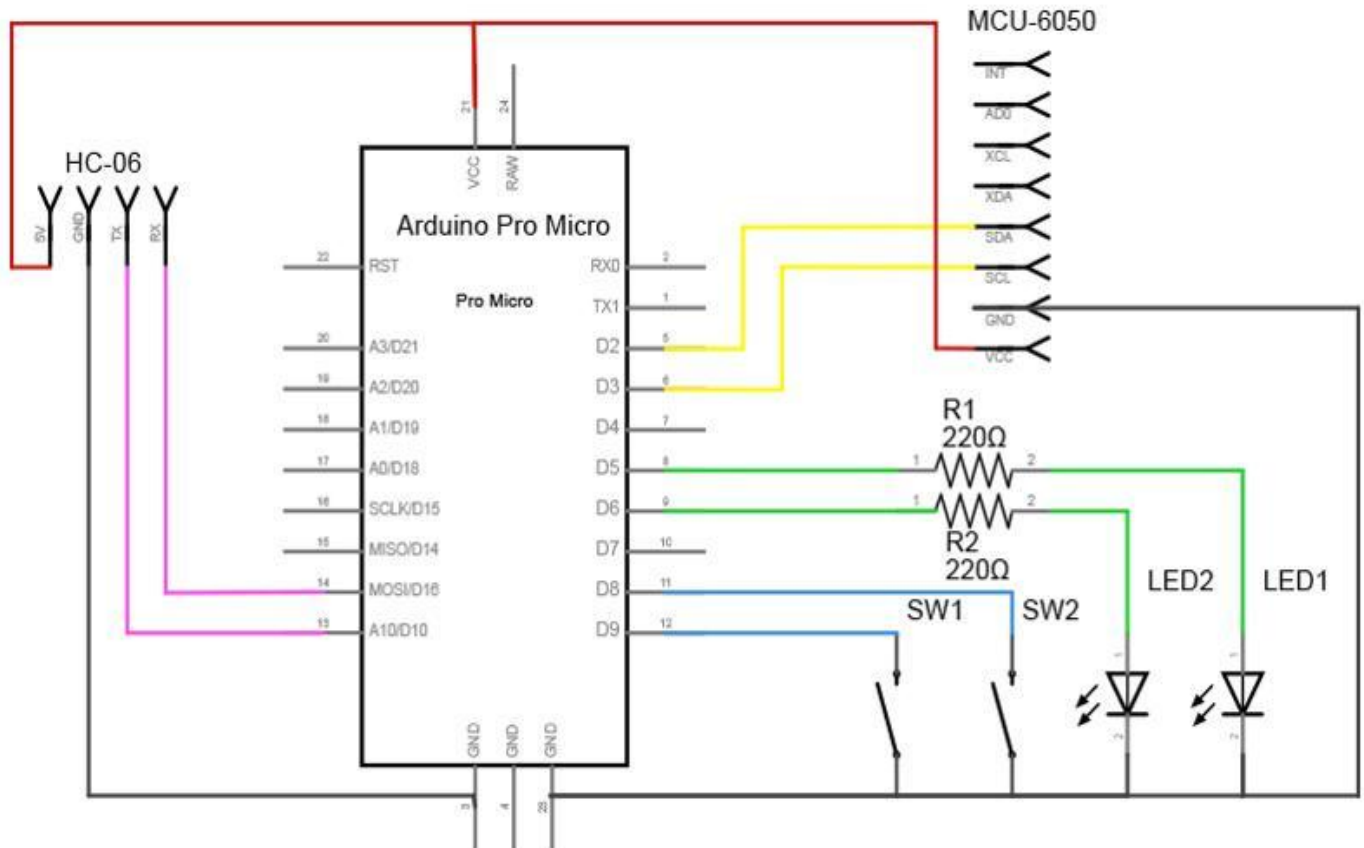


Figure 3.1: Circuit Diagram

Arduino Pro Micro

Arduino is a micro-controller board. The pro micro version helps to reduce system size and adds flexibility to make the device handheld. Arduino receives the signal from the IMU sensor and is coded to use the data to form a gesture database.

Accelerator module- MPU 6050

The MPU-6050 is a 6 DOF (degrees of flexibility) or a six-axis IMU sensor, which implies that it gives six values as yield: three values from the accelerometer and three from the whirligig. The MPU-6050 is a sensor based on (small scale-electro –mechanical- frameworks) innovation.

Bluetooth Module- HC-05/06

HC-05 module is an simple to utilize Bluetooth (Serial Harbour Convention module, planned for straightforward remote serial association setup. Serial harbour Bluetooth module is totally qualified Bluetooth V2.0+EDR (Improved Information Rate) 3Mbps Tweak with complete 2.4GHz radio handset and baseband. This can be utilized to send information to computer over bluetooth.

Switches

The switches are used to control device operation. One switch is for toggling activation of the sensor. The second switch is to enable bluetooth transmission.

LED lights

LED lights are used as status indicators for the switches.

Above mentioned components are mentioned in figure 3.2

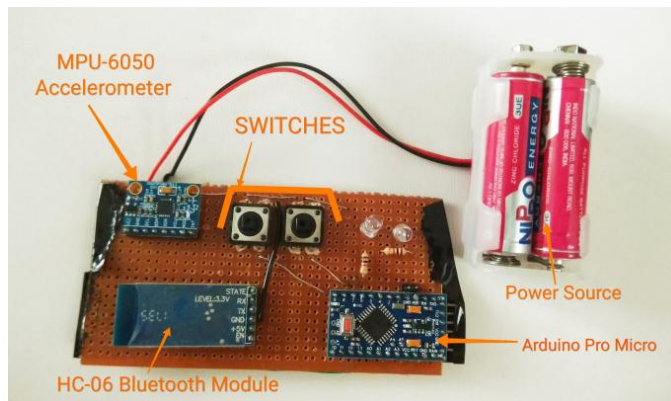


Figure 3.2

Scikit-learn Library

Scikit-learn is a programming library in machine learning for the programming in Python. It highlights distinctive characterization, backslide and clustering calculations counting reinforce vector machines, self-assertive forests, slant boosting, k-means and DBSCAN, and is aiming to interoperate with the Python numerical and coherent libraries NumPy and SciPy. Scikit-learn algorithm cheat sheet is mentioned in figure 3.3.

Introduction

The scikit-learn wander started as scikits.learn, a Google Summer of Code wander by David Cournapeau. Its title comes from the thought that it is a "SciKit" (SciPy Toolkit), an autonomously made and passed on pariah extension to SciPy. The special code-base was afterward adjusted by distinctive engineers. In 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from INRIA ventured up with respects to the undertaking and made the essential open discharge on February the beginning 2010. Of the different scik-its, scikit-learn and too scikit-pictures were portrayed as "especially kept up and discernible" in November 2012. Starting at 2017, scikit-learn is beneath special alter.

Implementation

Scikit-learn is to a extraordinary degree composed in Python, with a few center calculations composed in Cython to achieve execution. Support vector machines are executed by a Cython wrapper around LIBSVM; calculated backslide and coordinate offer assistance vector machines by a comparable wrapper around LIBLINEAR.

Machine Learning

By and expansive, a learning issue considers an course-of action of n tests of data and after that tries to expect properties of darkened data. In the occasion that each illustration is in abundance of a single number and, for illustration, a multi-dimensional area (something else known as multivariate data), it is said to have a few properties or highlights. We can isolated learning issues in a few huge categories: Su-pervised Learning, in which the information comes with extra properties that we need to anticipate. This issue can be either:

Classification: Tests have a place to two or more classes and we need to memorize from as of now named information how to foresee the lesson of unlabelled information. An case of classification issue would be the written by hand digit acknowledgment case, in which the point is to dole out each input vector to one of a limited number of discrete categories. Another way to think of classification is as a discrete (as restricted to nonstop) frame of su-pervised learning where one has a constrained number of categories and for each of the n tests given, one is to attempt to name them with the redress category or course.

Relapse: On the off chance that the pined for abdicate comprises of at slightest one ceaseless variables, at that point the undertaking is called backslide. A case of a backslide issue would be the pre-word utilization of the length of a salmon as a component of its age and weight. Unsupervised learning, in which the arrangement data comprises of an course of action of data vectors x with no comparing target regards. The objective in such issues might be to discover get-togethers of comparative cases interior the data, where it is called gathering, or to choose the allotment of data interior the data space, known as thickness estimation, or to amplify the data from a high-dimensional space down to a few estimations with the conclusion objective of representation.

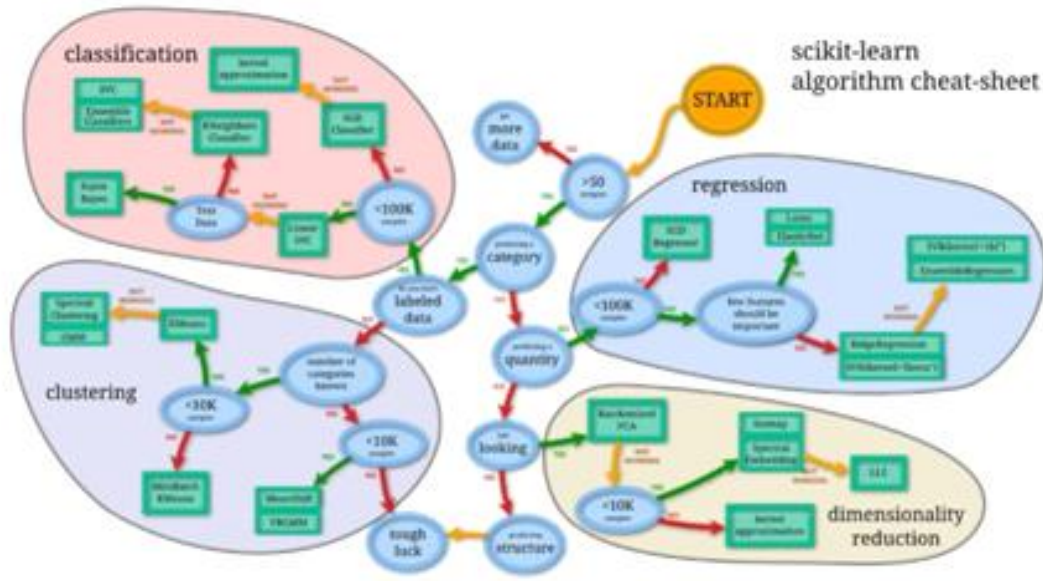
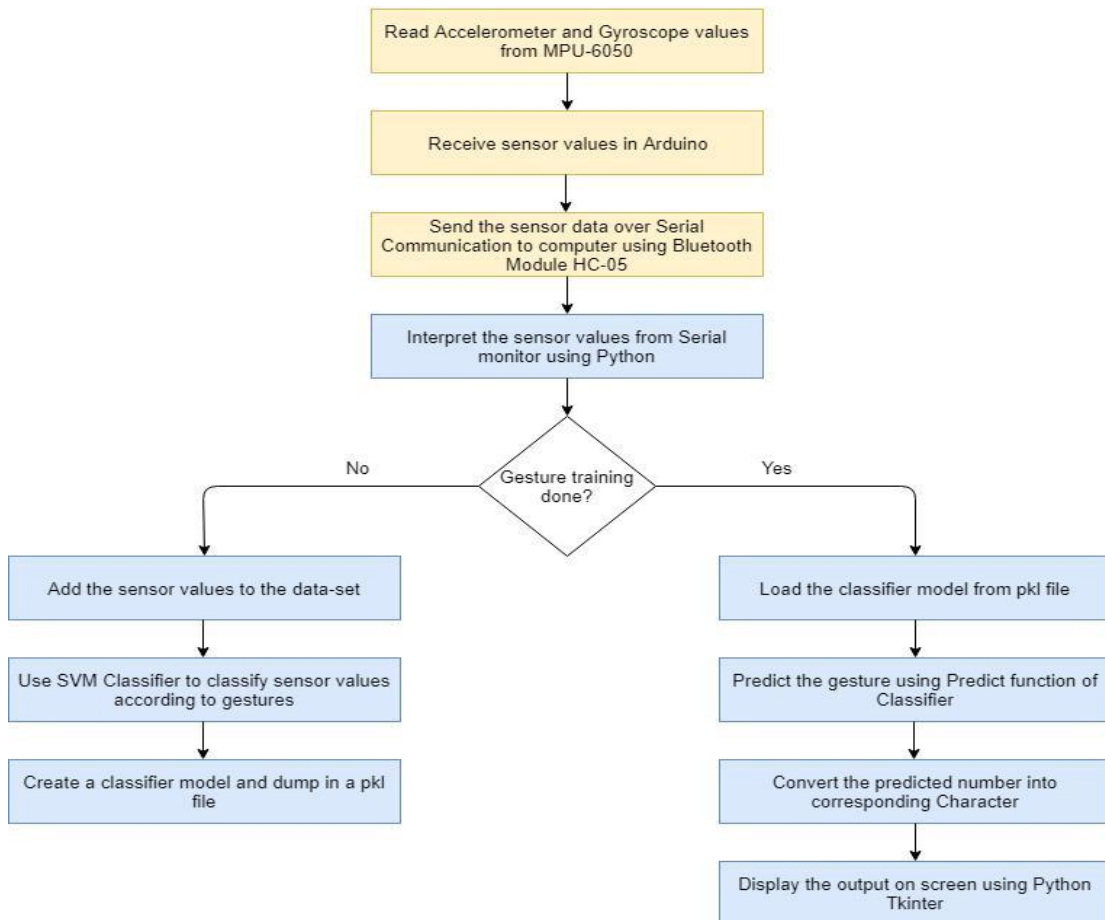


Figure 3.3

METHODOLOGY

Flow Diagram



Working of the system

Initially the machine won't be knowing anything. The machine has to read the gestures which are made by the hardware. The gestures are read through the IMU sensor attached to the Arduino. The sensor values provide the continuous location of the hand used to make the gesture. A combination of the sensor values will be used to estimate the path taken by the hand while making the gesture. The sensor values will be used to form a data-set. The sensor values will be mapped to the corresponding character which was specified by the user while training the machine. This mapping will be used to predict the character from the gesture made in thin air.

For example, making an 'A' in the air with our hand can be mapped to print the letter 'A' on the computer screen. Initially the user has to feed the preferred gesture to the program and after doing that the user has to train the machine by testing it continuously. This way the machine comes to know the gestures required for a particular operation.

How does an IMU sensor work?

An inertial estimation unit (IMU) is an electronic gadget that measures and reports a body's particular power, precise rate, and some of the time the attractive field encompassing the body, utilizing a blend of accelerometers and spinners, some of the time additionally magne-tometers. IMUs are commonly used to move air ship, including unmanned ethereal vehicles (UAVs), among numerous others, and rocket, including satellites and landers. The information announced by the IMU is encouraged into a processor which ascertains state of mind, speed and position. An ordinary usage alluded to as a Strap Down Inertial System coordinates rakish rate from the gyrator to figure precise position. This is combined with the gravity vector estimated by the accelerometers in a Kalman channel to evaluate demeanor. The state of mind assess is utilized to change increasing speed estimations into an inertial reference outline (thus the term inertial route) where they are incorporated once to get direct speed, and twice to get straight position.

Communicating with the computer

The micro-controller receives the sensor values. Arduino communicates with the com-puter using Serial communication. A python script reads the sensor values from Serial Monitor and stores them as data-sets. These sensor values are mapped to targets and can then be used to perform functions on the computer using scripts written in Python. The communication between arduino and a computer can take place through serial commu-nication through Arduino's USB port.

To make the device portable, a Bluetooth module HC-05/06 can be interfaced with the Arduino and the data can then be sent over Bluetooth.

COLLECTING DATA FROM GESTURES

Starting the Python Script

A python script is started by running the following command in the project directory using Command Prompt/Terminal:

```
python start.py target=a:0
```

This command runs the script "start.py" with "target" as 'a'. The target specifies the character that we want to store the gesture for. In this example, since the target is 'a', the sensor values will be stored for the character 'a' in a specified file format.

Interpreting and Storing the data

After starting the script, the device is turned on and gestures are made by holding the device in hand while pressing the Start Button on the device. The job of the script is to interpret values from Serial monitor and store them in a data-set file for the specified target while the button is pressed. Releasing the button on the device, results in storing one set of values. Similarly, we need to make the gesture multiple times by pressing and releasing the button so that the data-set contains sufficient data for prediction. A good number of sets was found out to be 40.

Learning from the Data

Learn.py is the name of the script where the logic to learn is written. It is used to make a classifier and train it according to the data received from the Arduino.

It uses the stored data-set to classify the data into category (on the basis of character) and sensor data. It gets category from file name and sensor data from the file con-tent. The letters are stored in the model as numbers since the classifier model relies on numbers rather than characters. The variable X_Data stores sensor data and variable Y_Data stores category.

Initiate Support Vector Machine

A support vector machine is defined using Scikit-learn library for classification purpose (declared using svm.SVC) The classifier uses GridSearchCV. Matrix look is a way to deal with parameter tuning that will deliberately manufacture and assess a model for every blend of calculation parameters determined in a lattice. It uses exhaustive search over specified parameter values for an estimator. The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid. The classifier is trained using fit() function of GridSearchCV.

Calling .fit()

In a nutshell: fitting is equal to training. Then, after it is trained, the model can be used to make predictions, usually with a .predict() method call. To elaborate: Fitting your model

to i.e. using the `.fit()` method on the training data is essentially the training part of the modeling process. It finds the coefficients for the equation specified via the algorithm being used. Then, for a classifier, you can classify incoming data points (from a test set, or otherwise) using the `predict` method. Or, in the case of regression, your model will interpolate/extrapolate when `predict` is used on incoming data points. Basically, fitting classifies the sensor values according to the category e.g. Stores the sensor values corresponding to the letter 'a'. To use the classifier for prediction the classifier object is dumped into a `model.pkl` file

Predicting the Gesture

After the device is trained for a specific gesture, we can start predicting by running the following command on Computer: `python start.py predict`

The script loads the classifier from `model.pkl` generated in the learning section. Now, a gesture can be drawn in air using the device. The incoming set of sensor values are read by the script and the classifier compares these values to predict the set of sensor values to which it belongs. The number corresponding to the gesture is predicted using "`predict`" function of Classifier. This number is then converted to the corresponding character.

The output is stored in a file from where it is read and displayed on computer screen using Python Tkinter.

CONCLUSION

This paper portrays a framework that have the ability to identify the character drawn through a gesture in air with the assistance of a input device. The strategy proposed here effectively made an acknowledgment framework, that can perceive which motion is performed by the user and precisely play out the usefulness related with it.

FUTURE ENHANCEMENT

The present framework gives best outcomes in a plain foundation and henceforth puts certain imperatives on the user for effective working. The future work will incorporate usage of extra signals which will empower the client to perform more capacities easily. Moreover, foundation subtraction calculation can be utilized for a more compelling execution. The proposed framework utilizes just the correct hand to perform signals. Henceforth, upgrade of the procedure proposed, is conceivable utilizing the two hands for performing diverse PC activities. Examinations should be done on a bigger scale with the goal that outcomes can be more exact.

REFERENCES

- [1] Cost-effective hand gesture computer control interface-7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2016) • October 16-18, 2016 • Wroclaw, Poland.
- [2] Gesture Control of Mobile Robot Based ArduinoMicrocontroller-8th International Conference on Modelling, Identification and Control (ICMIC-2016), Algiers, Algeria- November 15-17, 2016.
- [3] Matthias Rehm, Nikolaus Bee, Elisabeth André, Wave Like an Egyptian - Accelerometer Based Gesture Recognition for Culture Specific Interactions, British Computer Society, 2007.
- [4] Accelerometer based Wireless Air Mouse using Arduino Micro-controller Board - Proceedings of 2015 Global Conference on Communication Technologies(GCCT 2015).
- [5] R. Cipolla and A. Pentland, Computer Vision for Human-Machine Interaction, Cambridge University Press, 1998, ISBN 978-0521622530.
- [6] *Gyro-Accelerometer based control of a robotic Arm using AVR Microcontroller*-The 9th International Forum on Strategic Technology (IFOST), October 21-23, 2014, Cox's Bazar, Bangladesh.
- [7] Chen, Q., Georganas, N., and Petriu, E. Real-time vision-based hand gesture recognition using haar-like features. In Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE (May 2007), pp. 1–6.
- [8] Perng, J., Fisher, B., Hollar, S., and Pister, K. Acceleration sensing glove (asg). In Wearable Computers, 1999. Digest of Papers. The Third International Symposium on (1999), pp. 178–180.
- [9] Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. A hand gesture interface device. SIGCHI Bull. 17, SI (1987), 189–192.
- [10] J. Kela, et al., "Accelerometer-Based Gesture Control for a Design Environment," Personal and Ubiquitous Computing, Vol. 10, No. 5, 2005, pp. 285-299.
- [11] H. Lee, et al., "Select-and-Point: A Novel Interface for Multi-Device Connection and Control Based on Simple Hand Gestures," CHI 2008, Florence, 5-10 April 2008.
- [12] T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character", ACM Transactions on Graphics, vol. 27, no. 5, (2008), pp. 1–9
- [13] S. P. Priyal, P. K. Bora, "A study on static hand gesture recognition using moments", In Proceedings of International Conference on Signal Processing and Communications (SPCOM), (2010), pp. 1-5
- [14] Hand gesture computing, Gaurav Pradhan, Balakirshnan Prabhakaran, IEEE Conference , 2011
- [15] Juha K., Panu K., Jani M., Sanna K., Giuseppe S., Luca J. and Sergio D. M. Accelerometer-based gesture control for a design environment, Springer, Finland, 2005.