**Kathmandu BernHardt**

**College**

(Affiliated to Tribhuvan University)



A Project Proposal

On

“ **Hand Gesture Recognition using Machine Learning**”

**Under the supervision of**

………………………………

**Submitted By**

Anu Hamal

Asmita Acharya

Divya Pandey

Sushant Pandey

**Submitted to**

Department of Computer Science and Information Technology

Kathmandu Bernhardt College

Bafal, Kathmandu, Nepal

April 16, 2018

1

1

# 1. INTRODUCTION

Classification of human movement is a large field of interest to Human-Machine Interface researchers. The reason for this lies in the large emphasis humans place on gestures while communicating with each other and while interacting with machines. Such gestures can be digitized in a number of ways, including both passive methods, such as cameras, and active methods, such as wearable sensors. While passive methods might be the ideal, they are not always feasible, especially when dealing in unstructured environments. Instead, wearable sensors have gained interest as a method of gesture classification.The sign language recognition, human machine interface are used in this system,in order to move a step closer to recognize a number of different gestures classes. It further contrasts the applicability of these tools to noisy data in the form of the Ninapro dataset-Non Invasive Adaptive Hand Prosthetics, a benchmarking tool put forth by a conglomerate of universities.Using this dataset as a basis, this work paves a path for the analysis required to optimize each of the three classifiers. Ultimately, care is taken to compare the three classifiers for their utility against noisy data, and a comparison is made against classification results put forth by other researchers in the field.

# 2. PROBLEM DEFINITION

Hand Gesture recognition is a very important approach to deal with as with the development of ubiquitous computing, current user interaction approaches with keyboard, mouse and pen are not sufficient. Due to the limitation of these devices the useable command set is also limited. Direct use of hands can be used as an input device for providing natural interaction.Similarly with the help of this system it can smoothen the communication between disabled people. Hence, to increase communication between users and machines our system can be very helpful.

# 3. OBJECTIVE

The objective of this Project is to develop a Recognition System which:

* can help to make the talking atmosphere active. With these gestures other talkers can know the feeling directly. This can make the conversation smoothly.

* can help other people understand our main point quickly.

# 4. SCOPES AND LIMITATIONS

# Scopes

* To create a method to recognize hand gestures using camera and using machine learning method.
* To continue the research to improve GCUI(Gesture Controlled User Interface),its outcome and to enhance the communication between machine and users .

**Limitations**

* The algorithm can not detect fast moving hand.
* The system makes an assumption that the hand is the brightest and closest object to the camera.
* The system that is built may not recognize all and complex gestures.

# 5. Methodology

## 5.1 Literature Review

There are a lot of works which concentrate on wearable devices for gesture recognition. These works range in scope from small projects by undergraduate CSIT students, to Master’s Thesis, and much further into million dollar projects such as that conducted by NASA’s JPL. What follows are a few such works that contributed to this thesis, either by offering simple guidelines, explanations and examples, or even aiding in defining the scope.

* One such attempt is that of Driesch and Malsburg how have achieved good recognition rate (86.2%) using the Elastic-Graph matching technique to classify hand postures against complex backgrounds .
* “Hand Tracking and Gesture Recognition for Human-Computer Interaction” is another attempt by Cristina Manresa, Javier Varona, Ramon Mas and Francisco J. Perales. The project was designed for the control of a videogame based on hand gesture recognition and has restriction of real-time response and the use of unconstrained environments. The application uses images from a low-cost web camera placed in front of the work area, where the recognised gestures act as the input for a computer 3D videogame. The first step in the recognition procedure is hand segmentation. The skincolour is choosen as the hand feature. Pixel-based tracking is used for the temporal update of the hand state. In the last step it uses the estimated hand state to extract several hand features to define a deterministic process of gesture recognition. . In order to make the application robust to the segmentation it uses tracking algorithm that tries to maintain and propagate the hand state over time.
* Another similar attempt is “Automatic Hand Gesture Recognition” by Anant Atray .The application is designed to recognizes pre defined hand gestures using various computer vision and machine learning algorithms. Hidden Markov Models method is used for recognition and Viterbi algorithm and forward algorithm are used for evaluation.
* “A System For Real Time Gesture Recognition” is another one by Daniel Persson and Björn Samvik. IT is a complete system which tracks and classifies hands as well as identify if a certain gesture is performed which is able to run on a mobile platform. The application uses Viola-Jones detector and ADA-boost trainer , RAMOSAC tracker , Angular Space transformation and Dynamic Time Warping.
* “Superpixel-Based Human Computer Interface Using Hand-Gesture Recognition” by Abhishek Maheshwari, Anurag Semwal, Susmit Wagle is another attempt .This application implemented Hand Localization and Segmentation, Shape Representation Using Joint Color-Depth Superpixel, Depth Normalization and SP-EMD .Template matching is utilized for hand gesture recognition based on the SP-EMD.

**5.2 Requirement Collection:**

Functional requirements:

* There must not be an object between the mobile device and hand of the user in order for a successful tracking.
* The end position of the hand is must be into the angle of sight.
* Gesture of the hand must be a predefined gesture in order to use the mobile device.
* The end position of the face is must be into the angle of sight.
* Select gesture must match with pre-defined gesture and location of hand must be closer object to phone.
* Quit gesture must match with pre-defined gesture and location of hand must be closer object to phone.

Non function requirements:

* Performance Requirements: - Controlling Mobile Devices via Gesture Recognition should be used by only one hand of the user, indeed. CMDGR system should run on a single core processor, by using 30% capacity of the processor.
* Design Constraints: - Python is the programming language that will be used in the software together with the Open CV and QT libraries. Software architecture will be based on Model-View-Controller (MVC) architecture. MySQL will be the database server keeping the datasets and related information of mobile devices features. The software will include the reliability and portability system attributes.

Algorithms we can use:

Neural Networks:

Back propagation algorithm:

**5.2.1 Implementation of Back propagation algorithm:**

One of the most popular NN algorithms is back propagation algorithm. Rojas [2005] claimed that BP algorithm could be broken down to four main steps. After choosing the weights of the network randomly, the back propagation algorithm is used to compute the necessary corrections. The algorithm can be decomposed in the following four steps:

1. Feed-forward computation
2. Back propagation to the output layer
3. Back propagation to the hidden layer
4. Weight updates The algorithm is stopped when the value of the error function has become sufficiently small.

The algorithm is as follows:

1. First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.
2. Next work out the error for neuron B. The error is What you want – What you actually get, in other words: Error B = Output B (1-OutputB)(Target B – Output B) The “Output(1-Output)” term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be (Target – Output).
3. Change the weight. Let W+ AB be the new (trained) weight and WAB be the initial weight.

W+ AB = WAB + (Error B \* Output A)……..eqn(i)

Notice that it is the output of the connecting neuron (neuron A) we use (not B). We update all the weights in the output layer in this way

1. Calculate the Errors for the hidden layer neurons. Unlike the output layer we can’t calculate these directly (because we don’t have a Target), so we Back Propagate them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A. ErrorA = Output A (1 - Output A)(ErrorB WAB + ErrorC WAC) Again, the factor “Output (1 - Output )” is present because of the sigmoid squashing function.
2. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

## 5.3 Feasibility study

Feasibility study are helpful to expand, build and remodel, change the methods, add new products, or even merge the system.

Types:

1. **Technical feasibility:**

.In our proposed model we have focused on the improvement of the classification accuracy rates using Back propagation algorithm. This work will also be a good example which advocates achieving better results for a data mining mixing up supervised and unsupervised learning.

1. **Schedule feasibility:**

It will be time acceptable web-based system, and can be conducted at a decent time. Estimated time spent in the development of this application is 4 months and scheduled as:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.N | Task description |  |  |  |  | Weeks | |  |  |  |
| **1-2** | **3-4** | **5** | **6** | **7-8** | **9-10** | **11** | **12** | **13-14** |
| 1 | Field Study and  Proposal  Preparation |  |  |  |  |  |  |  |  |  |
| 2 | Analysis |  |  |  |  |  |  |  |  |  |
| 3 | Data Collection |  |  |  |  |  |  |  |  |  |
| 3 | Design |  |  |  |  |  |  |  |  |  |
| 4 | Implementation |  |  |  |  |  |  |  |  |  |
| 5 | Testing\Debugging |  |  |  |  |  |  |  |  |  |
| 6 | Report |  |  |  |  |  |  |  |  |  |

Fig:5.3 Grant Chart

1. **Operational feasibility:**

This system we built will be used efficiently even by the non-technical people. This system can be

used by any person by not even changing the system.

## 5.4 Data Collection

Most of the data will be collected from Nanopro dataset, a benchmarking tool put forth by a conglomerate of universities. Using this dataset as a basis, this work paves a path for the analysis required to optimize the classifiers. Ultimately, care is taken to compare the three classifiers for their utility against noisy data, and a comparison is made against classification results put forth by other researchers in the field.

## 5.5 Tools

### 5.5.1 Design and analysis tools

For this we can use a level-0, DFD (Data Flow Diagram) to show the overall structure of the different processes involved in the web-based system. The architecture of the web based hand gesture recognition system will be based on Backpropagation algorithm.

### 5.5.2 Implementation tools

* **Front-End Tools:**

The front end will be designed using HTML,CSS and Javascript.

▪ This uses windows operating system and with the help of computer vision the data are extracted from the databases.

▪ The Integrated Development Environment (IDE) provides a flexible way for designing the project, coding and CSS is used for style.

▪ We can easily switch over between source code view and designer view for designing the user interface.

* **Back-End Tools:**

Back end will be mainly used to store the data and also retrieve the data using Python along with Django framework and machine learning techniques. Databases are maintained using PostgreSql.

**5.6 Testing Techniques:**

Testing Strategies:

* White box testing

A method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).

* Unit Testing

Unit Test comprises the set of tests performed by an individual programmer prior to integration of the unit into a large system.

Integration of a program unit is usually small enough that the programmer who developed it can test it in great detail and certainly this will be possible when the unit test integrated into an evolving software product.

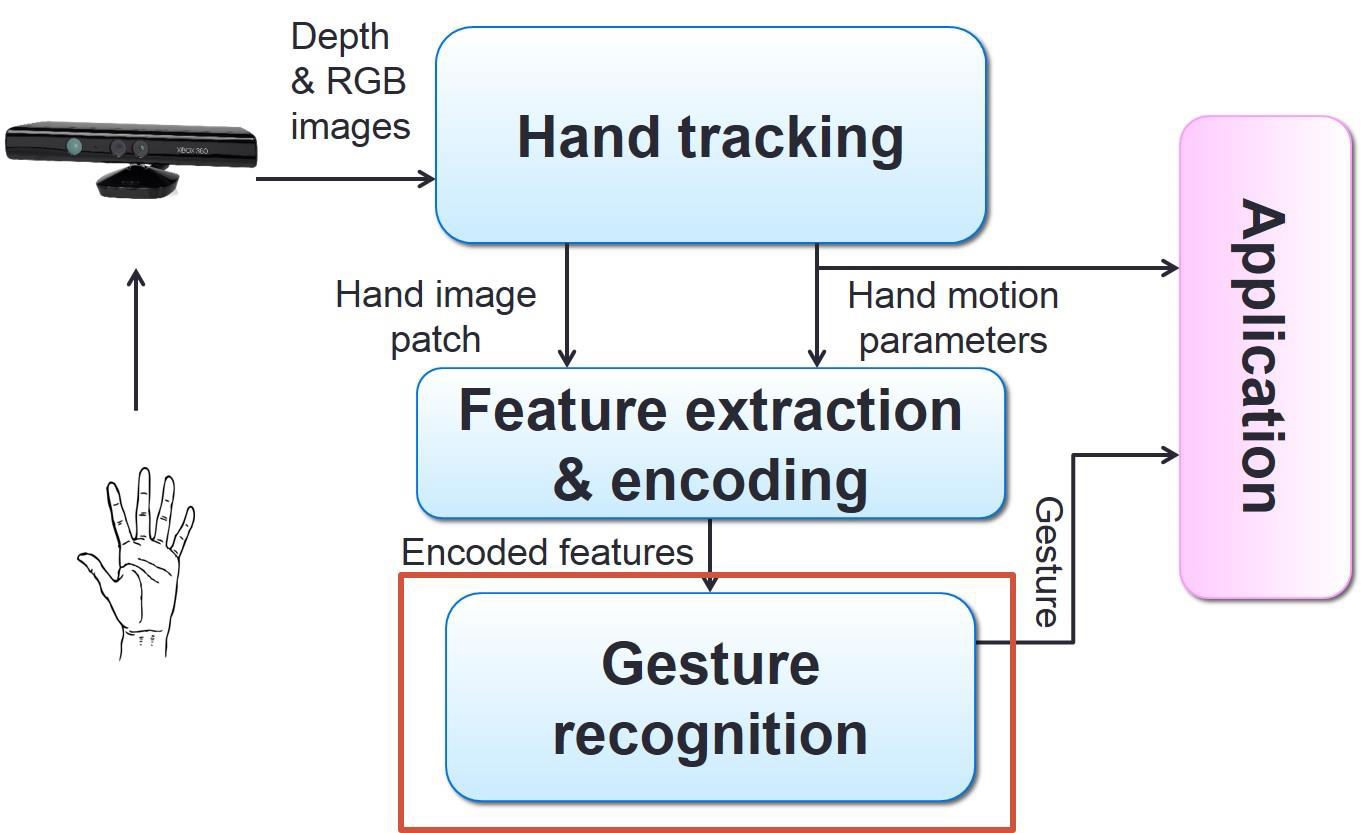
* Integration Testing

Integration testing is the systematic technique for constructihg the program structure while as the same time conducting test to uncover errors associated with interfacing. The objective is to take unit tested module and built a program structure that has been dictated by design.

**TABLE:**

# 6. High Level Design of Proposed System

The high level design of the Hand Gesture Recognition system can be studied from the following flowchart:



**Fig.** System Flowchart ofHand Gesture Recognition System.

# 7. Expected Output

Hand Gesture Recognition System will be developed using Computer Vision and Back Propogation technique. The system extracts data from the databases and provides the outcome by the help of camera.

This model could define the gestures that is trained to the computer, which is defined as machine learning and provides the assumed output which eases the communication between human-computer interpretations.

# REFERENCES

1. T.Acharya , M.Sushmita. “ IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),Volume 3,Issue 3, May 2007.
2. Bowden & Sarhadi, Building temporal models for gesture recognition. In proceedings British Machine Vision Conference, 2000, pages 32-41.

1. S. Naidoo, C. Omlin and M. Glaser, "Vision-Based Static Hand Gesture Recognition Using Support Vector Machines", 1998.pages 88 – 94.

1. Vladimir I. Pavlovic, Rajeev Sharma, Thomas S Huang, “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A review” IEEE Transactions of pattern analysis and machine intelligence, Vole 19, NO 7, July 1997, 677 - 695