PROJECT REPORT

MOTION CAPTURE BASED HAND GESTURE RECOGNITION

"Submitted towards partial fulfilment of the criteria for award of PGPDSE by GLIM"

Submitted by

Student Name	SIS ID
Akshay Diwakar Rautkar	67E3X1A2N5
Anmol Singh	H70A899F71
Ayush Mahendra	Y20S24IQV3
Manish Khapre	SBJGN99N65
Mayur	HS5BT8LMNG
Sneha Pandey	SPP1DOWRLN

Batch: DSE_PUNE_SEPT: 2019-2020

Mentored By: Mr. Shashank Prakash Shirude

GREAT LAKES
INSTITUTE OF MANAGEMENT, CHENNAI

greatlearning
Learning for Life

ABSTRACT

This dissertation focuses on the study and development of algorithms that enable the analysis and recognition of hand gestures in a motion capture environment. Central to this work is the study of unlabelled point sets in a more abstract sense. Evaluations of proposed methods focus on examining their generalization to users not encountered during system training.

In an initial exploratory study, we compare various classification algorithms based upon multiple interpretations and feature transformations of points sets, including those based upon aggregate features (e.g mean) and a pseudo-rasterization of the capture space. We find aggregate feature classifiers to be balanced across multiple users but relatively limited in maximum achievable accuracy. Certain classifiers based upon the pseudo-rasterization performed best among tested classification algorithms. We follow this study with targeted examinations of certain sub problems.

Each unlabelled point is assumed to correspond to a target with independent probability of appearance but correlated positions. We propose replacing the expectation phase of the algorithm to account for the unknown point labels which manifest as uncertain measurement matrices.

ACKNOWLEDGEMENT

At the outset, we are indebted to our mentor **Mr. Shashank Prakash Shirude** for his time, valuable inputs and guidance. His experience, support and structured thought process guided us to be on the right track towards completion of this project.

We are thankful to all the course faculty of the DSE program for providing us a strong foundation in various concepts of analytics and machine learning. Last but not the least, we would like to sincerely thank respective families for giving us the necessary support, space and time to complete this project.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Akshay Diwakar Rautkar
Anmol Singh
Ayush Mahendra
Manish Khapre
Mayur
Sneha Pandey

Date : 26 Feb 2020

Place : Pune

Table of Contents :

Chapter 1

INTRODUCTION

1.1 Overview :

Gesture recognition, as a means of human-computer interaction, provides an intuitive and effective interface for user control, offering the ability to perform complicated tasks with minimal effort. The success of smartphones and tablets with touchscreens supports this hypothesis. A significant amount of research involving gestures has been performed in the past two decades with many methods and solutions offered. Hand gesture recognition is an especially appealing branch of the gesture recognition field because it can offer a more tantalizing avenue for the average end-user, even if only for the visceral thrill of execution. However, there is no current camera-based system that can demonstrate robust and precise finger-based gesture recognition (or even tracking) in a sizable 3D space, although significant strides in finger tracking have been made recently.

We separate our recognition targets into two categories: postures and gestures. A posture, or static gesture, is one in which the hand makes a certain pose, such as holding a closed fist, whereas a (dynamic) gesture involves motion of the hand, arm, or fingers, such as pointing or waving.

There are many different methods by which hand features can be measured. Gloves are sometimes used. It uses wireless magnetic sensors embedded in a glove to detect finger motion and interpret a Braille-like binary code for communication. Vision-based approaches are of particular interest as they do not require any peripheral accessories other than the camera or equivalent sensing device.

The usage of Vicon motion capture cameras is similar to but fundamentally distinct from both depth-based methods and vision-based approaches, which we define to be detection methods based on the visible spectrum of light. Motion capture cameras instead observe infrared (i.e. not visible) light reflected by markers placed at preselected locations on the subject of interest. A noteworthy advantage of motion capture is the low-volume and sparsity of the data. A significant amount of noise that can be introduced by the environment is automatically filtered. Only the coordinates of the markers, inferred by triangulation, are reported for each frame measured. Aside from the exceptionally high costs for the hardware and software involved and the careful camera calibration required to make practical use of the system, motion capture also comes with another major disadvantage encountered repeatedly throughout this research: marker identity is not known except under very limited circumstances.

Marker identity is generally known (or equivalently, markers are labeled) only when part of a rigid pattern or predefined skeleton. A rigid pattern is a configuration of markers such that if each marker is connected by an inflexible rod, then the angle between each pair of rods is constant. Consequently, the rod lengths are also fixed. A skeleton differs from a rigid pattern in that certain rods and angles are explicitly defined whereas others are free to change. Marker identities are often determined by having the subject strike a pose (such as a "T"-pose for a full body skeleton) in order to label markers, after which joint angles and other parameters of the skeleton are determined via inverse kinematics or some other, perhaps probabilistic, method.

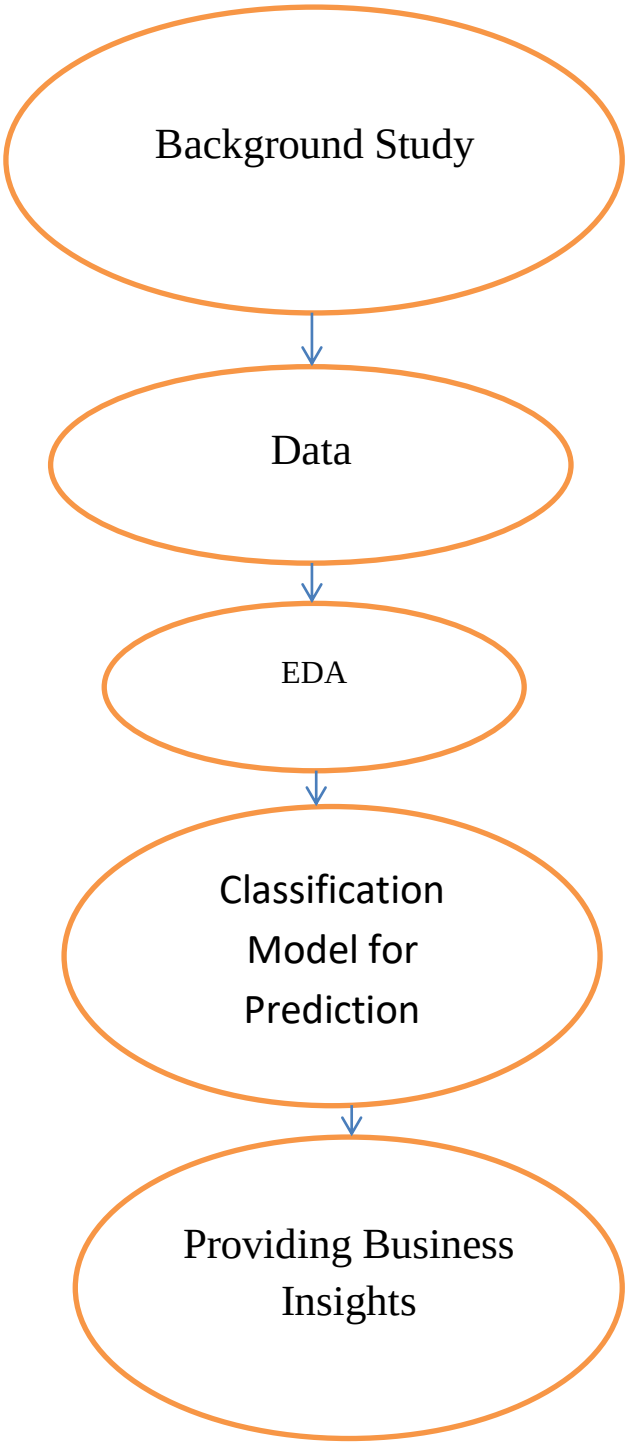
1.2 Objectives :

- The signs language are used from ages to interact with person to person so as to convey message or to assign task related to the sign.
- The hearing and speaking disabled persons are the ones who uses this type of communication.
- The disabled person who can't speak don't have way to communicate through systems
- The aged persons and the ones who are not compatible with the typing input interfaces has difficulty to interact with systems.

1.3 Problem Statement :

In the recent years there have been much development and research in the field of hand gesture recognition system .This has been a major topic in the field of machine learning so as to convey message to the system or have interaction with the systems through the hand gesture. The system detects the hand postures and then take necessary actions or the task is triggered accordingly with specified classes to the gesture. In this project we are classifying the hand gesture classes accordingly with the coordinates data which we obtained by sensors attached to the glove. This classification will lead to the accurate model with reduction in the errors of the recognition system . This model will also lead to the systems where the disabled persons can be precise to use such systems.

1.4 Methodologies of Problem Solving :



1.5 Dataset Information :

The dataset is Multivariate and has missing values.

The data presented here is already partially pre-processed. First, all markers were transformed into the local coordinate system of the record containing them. Second, each transformed marker with a norm greater than 200 millimeters was pruned. Finally, any record that contained fewer than 3 markers was removed. The processed data has at most 12 markers per record and at least 3.

'Class' - Integer. The class ID of the given record. Ranges from 1 to 5 with

1=Fist (with thumb out),
2=Stop (hand flat),
3=Point1 (point with pointer finger),
4=Point2 (point with pointer and middle fingers),
5=Grab (fingers curled as if to grab).

'User' - Integer. The ID of the user that contributed the record. No meaning other than as an identifier.

'Xi' - Real. The x-coordinate of the i-th unlabelled marker position. 'i' ranges from 0 to 11.
'Yi' - Real. The y-coordinate of the i-th unlabelled marker position. 'i' ranges from 0 to 11.
'Zi' - Real. The z-coordinate of the i-th unlabelled marker position. 'i' ranges from 0 to 11.
1st and 2nd columns are categorical and have no missing values. Rest all the columns are continuous numerical.

Columns X0 to Z6 have less than 50% missing values.

Columns X7 to Z11 have more than 50% missing values.

1.5.1 Domain :

A rigid pattern of markers on the back of the glove was used to establish a local coordinate system for the hand, and 11 other markers were attached to the thumb and fingers of the glove. 3 markers were attached to the thumb with one above the thumbnail and the other two on the knuckles. 2 markers were attached to each finger with one above the fingernail and the other on the joint between the proximal and middle phalanx. The 11 markers not part of the rigid pattern were unlabeled, their positions were not explicitly tracked. Extraneous markers were also possible due to artifacts in the Vicon software's marker reconstruction/recording process and other objects in the capture volume due to which the dataset has 78096 rows and 38 columns.

1.5.2 Position of Sensors :

The glove used as the data source for all datasets. The axes of the local coordinate system based upon the rigid pattern are shown.

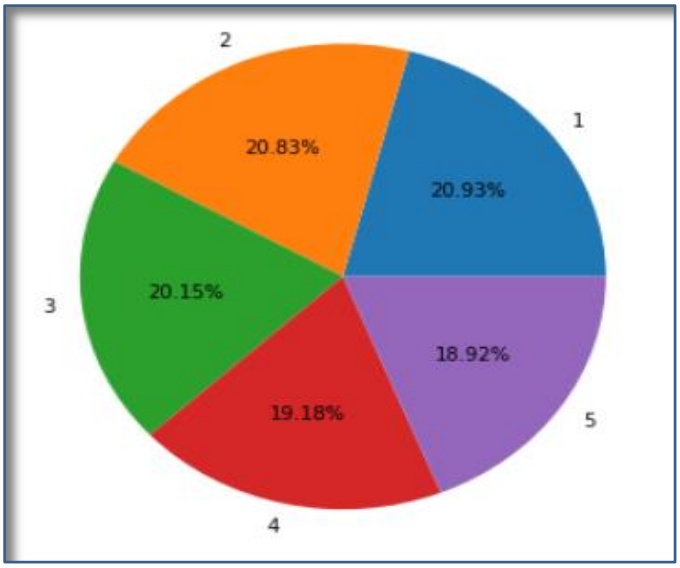
1. Pinky Finger (Joint)
2. Pinky Finger (Nail)
3. Ring Finger (Joint)
4. Ring Finger (Nail)
5. Middle Finger (Joint)
6. Middle Finger (Nail)
7. Pointer Finger (Joint)
8. Pointer Finger (Nail)
9. Thumb (Metacarpophalangeal Joint)
10. Thumb (Nail)
11. Thumb (InterphalangealJoint)

Chapter 2

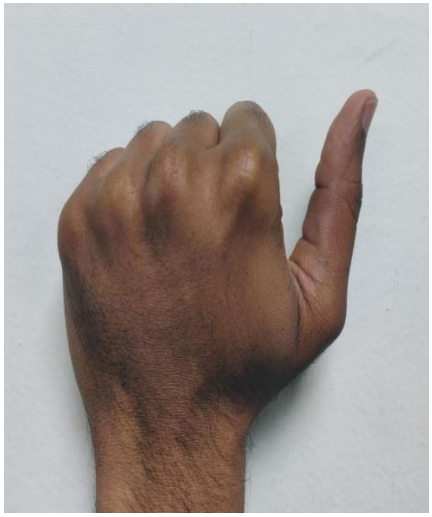
EXPLORATORY DATA ANALYSIS

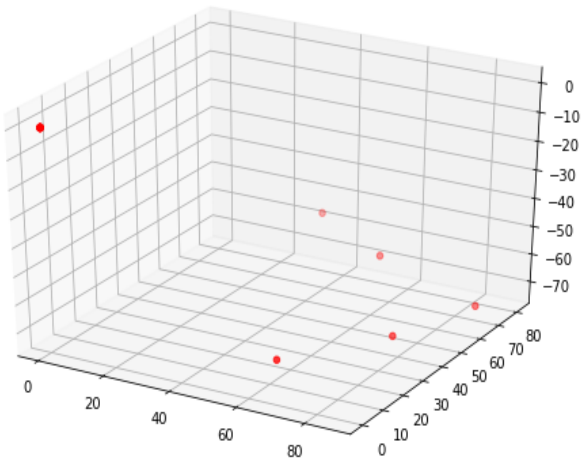
2.1 Data Analysis :

Class wise distribution of dataset

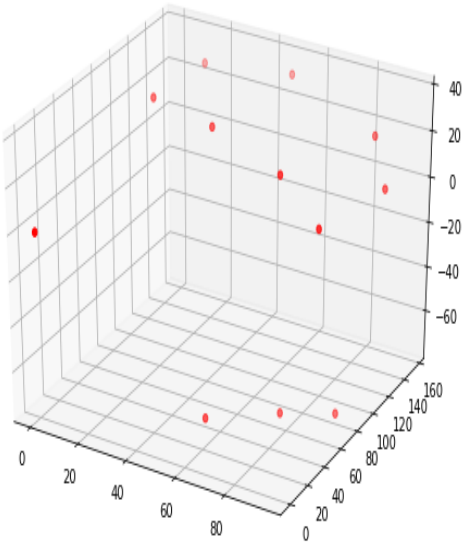


Class 1 : Fist (with thumb out)

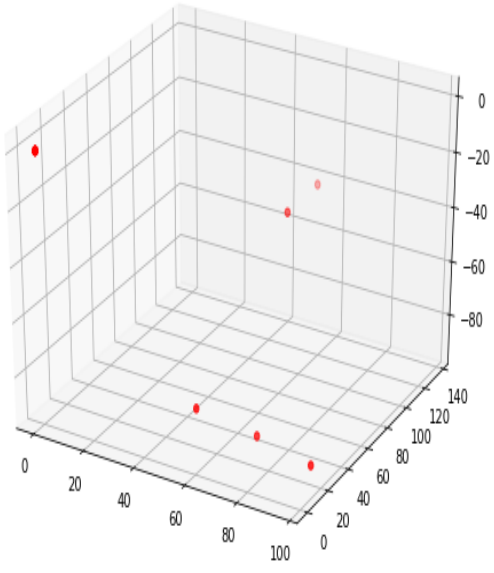




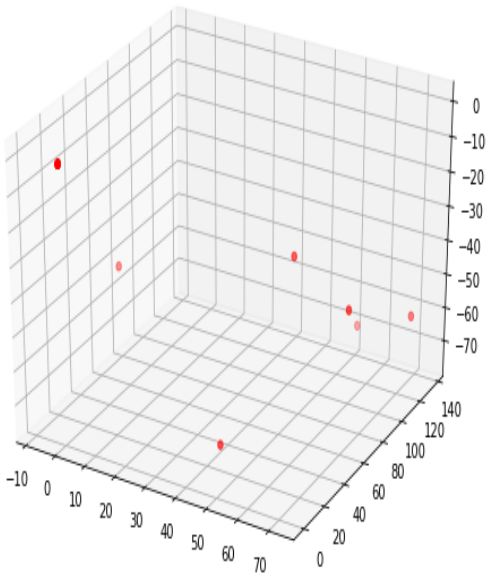
Class 2 : Stop (flat hand)



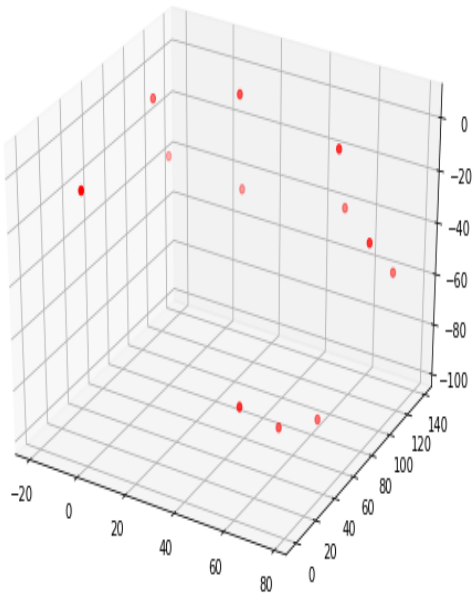
Class 3 : Point 1 (point with pointer finger)



Class 4 : Point 2 (point with pointer and middle finger)



Class 5 : Grab (fingers curled as if to grab)



2.2 Feature Engineering :

Dataset has 38 Columns

1 Column for Class, 1 Column for User, 36 Columns for Co-ordinates

Out of which 9 Columns (X9, Y9, Z9, X10, Y10, Z10, X11, Y11, Z11) has more than 70% of missing values. So we drop them because if we impute them with mean or median than the model would be influence by it.

```
Class 0.000000
User 0.000000
X0 0.000000
Y0 0.000000
Z0 0.000000
X1 0.000000
Y1 0.000000
Z1 0.000000
X2 0.000000
Y2 0.000000
Z2 0.000000
X3 0.008835
Y3 0.008835
Z3 0.008835
X4 0.039951
Y4 0.039951
Z4 0.039951
X5 0.166756
Y5 0.166756
Z5 0.166756
X6 0.330977
Y6 0.330977
Z6 0.330977
X7 0.501332
Y7 0.501332
Z7 0.501332
X8 0.608636
Y8 0.608636
Z8 0.608636
X9 0.693096
Y9 0.693096
Z9 0.693096
X10 0.811091
Y10 0.811091
Z10 0.811091
X11 0.999590
Y11 0.999590
Z11 0.999590 dtype: float64
```

Chapter 3

CLASSIFICATION

3.1 Analysis :

Model Making

Decision Tree :

It is the most powerful and popular tool for classification and prediction. A Decision Tree is a flowchart like a tree structure, where the internal node denotes a test on an attribute, each branch represents an outcome of the test and each leaf node (terminal node) holds a class label.

Training Accuracy : 95.13%
Testing Accuracy : 94.59%
F-1 Score :

KNN (K-Nearest Neighbor) :

It is a simple, easy to implement supervised machine learning algorithm. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. In this algorithm optimal k-value need to be find to select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before.

Training Accuracy : 100%
Testing Accuracy : 97.54%
F-1 Score :

SVM (Support Vector Machine) :

Support vector Machine is another method used for classification. The objective of SVM is to find a hyper plane (A decision boundary separating the tuples of one class from another) in an N-dimensional space (where N represents the number of features). The SVM classifier is a frontier which best segregates the two classes (hyper plane). It works really well with a clear margin of separation It is effective in high dimensional spaces. It is effective in cases where the number of dimensions is greater than the number of samples. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. It doesn't perform well when we have large data set because the required training time is higher.

Training Accuracy : 100%
Testing Accuracy : 65.9%
F-1 Score :

Random Forest :

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model prediction. It is very handy and easy to use algorithm as it's default hyper parameters often produce a good prediction result. The classifier won't overfit the model if there are enough trees in the forest.

Training Accuracy : 89.78%
Testing Accuracy : 89.79%
F-1 Score :

Gradient Boosting :

Boosting is used to create a collection of predictors. In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree.

Training Accuracy :
Testing Accuracy :
F-1 Score :

Cross Validation :

Cross-validation is a statistical technique which involves partitioning the data into subsets, training the data on a subset and use the other subset to evaluate the model's performance.

K fold cross validation

This technique involves randomly dividing the dataset into k groups or folds of approximately equal size. The first fold is kept for testing and the model is trained on k-1 folds. The process is repeated K times and each time different fold or a different group of data points are used for validation.

Confusion Matrix :

The various models built, must be evaluated based on certain model performance measures to identify the most robust models. Model accuracy alone may not be enough to evaluate a model. Hence the following model performance measures have been used to evaluate the models, based on the confusion matrix built for the predictions of training and test datasets:

	Negative (Predicted)	**Positive (Predicted)**
Negative (Observed)	True Negative (TN)	False positive (FP)
Positive (Observed)	False negative (FN)	True positive (TP)

Accuracy :

Accuracy is the number of correct predictions made by the model by the total number of records. The best accuracy is 100% indicating that all the predictions are correct.

Considering the response rate (conversion rate) of our dataset which is ~16%, accuracy is not a valid measure of model performance. Even if all the records are predicted as 0, the model will still have an accuracy of 84%. Hence other model performance measures need to be evaluated.

Precision :

Precision (Positive predictive value) is calculated as the number of correct positive predictions divided by the total number of positive predictions.

Precision tells us, what proportion of customers who generated revenue as customers actually generated revenue. If precision is low, it implies that the model has lot of false positives.

F-1 Score :

F1 is an overall measure of a model's accuracy that combines precision and recall A good F1 score means that you have low false positives and low false negatives. the F1 score conveys the balance between the precision and the recall.

Decision Tree Model :

```
[[4824   17    0    0    0]
 [   0 3971    2   78  420]
 [   0   11 5002    6   12]
 [   0  118    0 4247   78]
 [   0  484    1   40 4118]]
```

KNN (K-Nearest Neighbor) Model :

```
array([[4826,   14,    0,    1,    0],
       [  10, 4200,   36,  114,  111],
       [   2,    2, 4990,   35,    2],
       [   0,   37,   85, 4318,    3],
       [   2,  106,    2,   13, 4520]], dtype=int64)
```

SVM Model :

```
array([[4266, 0, 0, 0, 575],
      [ 1, 839, 0, 0, 3631],
      [ 0, 0, 3464, 0, 1567],
      [ 0, 0, 0, 2230, 2213],
      [ 0, 0, 0, 0, 4643]], dtype=int64)
```

Random Forest Model :

```
array([[4824,   13,    0,    0,    4],
       [   1, 3847,    8,    8,  607],
       [   0,   13, 5004,    2,   12],
       [   0,  380,    4, 3778,  281],
       [   0,  995,   12,   51, 3585]], dtype=int64)
```

Gradient Boost Model :

ROC-AUC Curve :

ROC Stands for Receiver Operating Characteristics Curve. Using Roc one can visually compare classification models. It is a trade-off between true positive rate and false positive rate. The accuracy of the model can be measured by the area under the ROC curve. Vertical axis represents the true positive rate and horizontal axis represents the false positive rate. The model with perfect accuracy will have an area of 1.0.

Decision Tree Model :

KNN (K-Nearest Neighbor) Model :

SVM Model :

Random Forest Model :

Gradient Boost Model :

Hyper-parameter Tuning :

The Machine learning models may have many parameters, a best combination of these parameters is considered to be a search problem and is very important for better performance of a machine learning model.

From the above model making, we can interpret that Random Forest Classifier is giving the best accuracy and prediction amongst other models. Hyper-parameter tuning is also done for this in Random Forest classification model in order to get better results for the same.

Hyper-parameters are selected with the help of Grid search CV algorithm to find best parameters which are as follows:

```
{'bootstrap': True, 'criterion': 'gini', 'max_depth': 4,
'max_features': 'auto'}
```

After using these parameters to tune the Random Forest Classification Model the result are as follows:

```
rfc.score(X_train,y_train)
```

```
0.8978524128343028
```

```
rfc.score(X_test,y_test)
```

```
0.8979469887745956
```

This shows us that after performing hyper parameter tuning and selecting the best parameters with respect to the model, we are getting the accuracy of 89.78% and 89.79% for training and testing data respectively.

Chapter 4

CONCLUSION

4.1 Future Scope :

REFRENCES

1. A. Gardner, J. Kanno, C. A. Duncan, and R. Selmic. 'Measuring distance between unordered sets of different sizes,' in 2014 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), June 2014, pp. 137-143.

2. A. Gardner, C. A. Duncan, J. Kanno, and R. Selmic. '3D hand posture recognition from small unlabeled point sets,' in 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), Oct 2014, pp. 164-169