Abstract

Body gesture recognition is a significant part of human-computer interaction, which is especially useful in robotic vision and the further growing market for a touch less technological experience,intelligent monitoring, virtual reality, human behaviour analysis, and other fields. Our project uses Mediapipe, an open source cross-platform framework for building multi modal applied machine learning pipelines, for accurate data collection and gesture recognition. The project overlays 3 recognition models provided by Mediapipe to recognise gestures created by a combination of hand, face and body and then use machine learning to make their recognition possible and efficient. The second component of the project enables users to maneuver the pointer on their local desktop, click and change volume using hand gestures.This was achieved using Mediapipe for gesture recognition and specialised python libraries like PyAutoGUI and autopy for the command automation corresponding to the gestures recognised. The project aims to show the accuracy of gesture recognition in today's world and its application along with the impact it has on human computer interaction.

Contents

List of Figures

Chapter 1

Introduction

As we know, the vision-based technology of hand gesture recognition is an important part of human-computer interaction (HCI). In the last decades, keyboard and mouse play a significant role in human-computer interaction. However, owing to the rapid development of hardware and software, new types of HCI methods have been required. In particular, technologies such as speech recognition and gesture recognition receive great attention in the field of HCI.

Gesture is a symbol of physical behavior or emotional expression. It includes body gestures and hand gestures. Gestures can be used as a tool of communication between computer and human . It is greatly different from the traditional hardware based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts. In the past decades, many researchers have strived to improve hand gesture recognition technology. Hand gesture recognition has great value in many applications such as sign language recognition , augmented reality (virtual reality) , sign language interpreters for the disabled , and robot control.

We have tried to support the above claims through this project in an effective and efficient method, using pre - existing libraries and open source softwares that empower businesses and projects all around the world for efficient data point collection and cleaning.

The data collected was then used to train to recognise the required gestures, and identify them. The model was trained using 4 machine learning algorithms, the best accuracy was achieved through the random forest algorithm, with a test to train data ratio of 30:70.

The second part of the project uses a custom built hand recognition module, that exploits medipipe and then different actions are associated with the various gestures recognised, the gestures and actions have real time association and effects, providing users with a fresh form of HCI , that has the capability to replace the mouse.

Chapter 2

Literature Review

Gesture recognition and it's Application in Human-Computer interaction

The project aims to create an efficient gesture recognition system that identifies the 5 gestures fed into it. Gesture recognition is a diverse field that and has various methods of functioning, distinction is mostly in the following areas:
1. Data Collection: Few projects take data in the form of direct colored images, and use that for training while some convert it into black and white images for training the model. There are various libraries that help in the data collection and processing phase, such as Mediapipe which is used in this project.
2. Training algorithms: There are various algorithms like xyz that are used for training depending on various factors like data set data type, size of data set, accuracy required etc.

The various approaches used by papers that were as follows:

The hand region is detected through the background subtraction method. Then, the palm and fingers are split so as to recognize the fingers. After the fingers are recognized, the hand gesture can be classified through a simple rule classifier.
The first novelty of the proposed method is that the hand gesture recognition is based on the result of finger recognition. Therefore, the recognition is accomplished by a simple and efficient rule classifier instead of the sophisticated but complicated classifiers such as SVM and CRF.[**5**]

The gesture recognition system is designed using an integrated approach for hand gesture recognition. It recognizes static and dynamic hand gestures.

Once the image of the hand is captured from the camera it is then processed through the following phases/algorithms. The procedure starts by acquisition phase.
This is the reason that our system has to yield better performance even in poor illumination, cheaper cameras and even the variation in the light of the environment. The system has to work within the real time constraints. Hence the procedure has to limit the parameters to the minimal possible level by removing the unnecessary information at the first instance. As the initialization phase is over, the haar cascade classifier is responsible for locating hand position and classifying gestures (open, close, pointing, etc.).
Haar-Like Features are features that digitize images to analyze images in object recognition applications. The tracking of the hand is done through camera shift technique with shifting the region of interest with average shift in the object of interest i.e. hands. As the hand is tracked a contour is mapped with the corresponding hand which further extracts a corresponding convex hull (area).
The recognition has been done through modeling of the hand by mapping it to the number of defects formed in it. Afterwards system tracks the number of defects that have been generated by the hand and maps it to a meaningful command.[**6**]

Our Project's Novelty

Our project mainly differs in the data collection process, and uses an open source library Mediapipe for data collection, unlike raw data intake, and then data processing is done on the coordinate points taken with the help of Mediapipe which becomes the data for the project and hence used for training the machine learning model used for identification.
Our model identifies gestures on the basis of the position of various juncture points in space and their relative alignment, and their dynamic distance in the second part of the project.
With respect to our data set and our accuracy requirements prompted us to use RandomForestClassifier.

Chapter 3

Project Objectives

The Objectives that the project intends on achieving are as follows:

- To successfully train a machine learning model that has the capability to identify 5 gestures(mentioned below) in real time.
- To provide a real-world application of Gesture recognition by enabling users to control their local device's mouse and audio using hand gestures, reinforcing it's significance in modern world's Human-Computer interaction

The 5 body gestures that are to be recognised are as follows:

1. Happiness
2. Drowsiness
3. Victory
4. Salute
5. The thinking stance

Chapter 4

Theory

The project is primarily based on the creation of an image processing machine learning model.
The machine learning model used in the project uses Random Forest Classifier for the generation of the result, i.e to identify the fed gestures.
Some theoretical knowledge about the generation of an efficient machine learning model are as follows:

Machine learning life cycle involves seven major steps, which are given below:

1. Gathering Data
2. Data preparation
3. Data Wrangling
4. Analyse Data
5. Train the model
6. Test the model
7. Deployment

From the above steps, our project doesn't make use of steps 3 and 4 because of us generating the data on our own.

Random Forest Classifier

Random Forest Classifier is the machine learning algorithm used for training and testing the model, because of its extreme accuracy and encouraging results. It was also considered because of its good track record with image processing projects.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

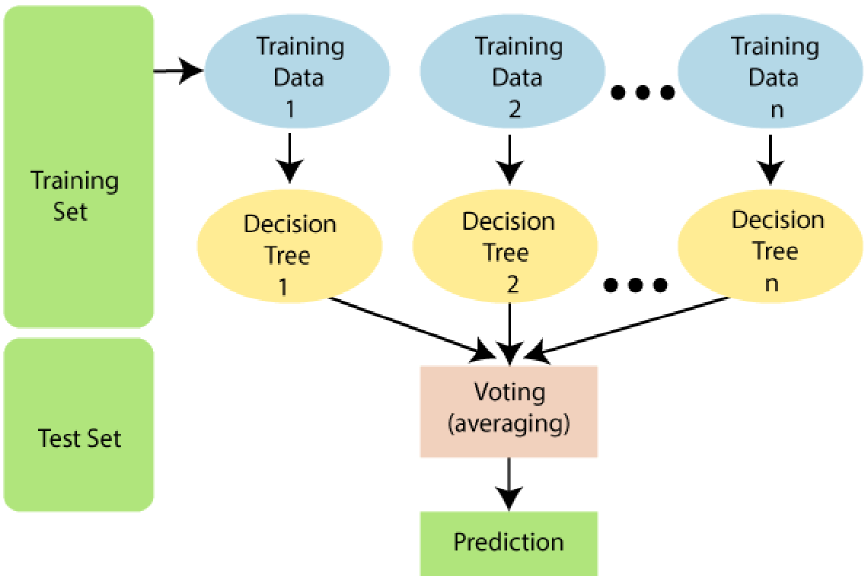

Fig 1

How does the Random Forest Classifier algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.
Step-2: Build the decision trees associated with the selected data points (Subsets).
Step-3: Choose the number N for decision trees that you want to build.
Step-4: Repeat Step 1 and 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Advantages of Random Forest Classifier:

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Because of the above mentioned advantages and suitability with the problem at hand Random forest classifier is ideal and hence gives good results.

Chapter 5

Life Cycle of the Project

1. Body gesture Recogniser

Data Gathering:
Data was gathered in the form of the coordinates of the various juncture points of the gesture in various areas around the camera field view, with the camera in operation.
The coordinates generated by each overlapping feature was stored in a **.csv** file. The webcam was activated and the gesture was made,and hence data points were collected.
A total of 7401 data points were collected for all the gestures combined, each data point having x,y,z (coordinates in space) and v (visibility factor) coordinates for a total of 501 juncture points each.
We took from around 1000-1400 data points for each gesture, the use of video helped us pick data points instantaneously, and collect huge amounts of data in a relatively shorter period of time.

Data Cleaning:
Data points with null values were removed for better accuracy and data was organised to associate with each gesture,i.e each class.

Below is a snippet from the data collection process for the gesture ”Salute” :

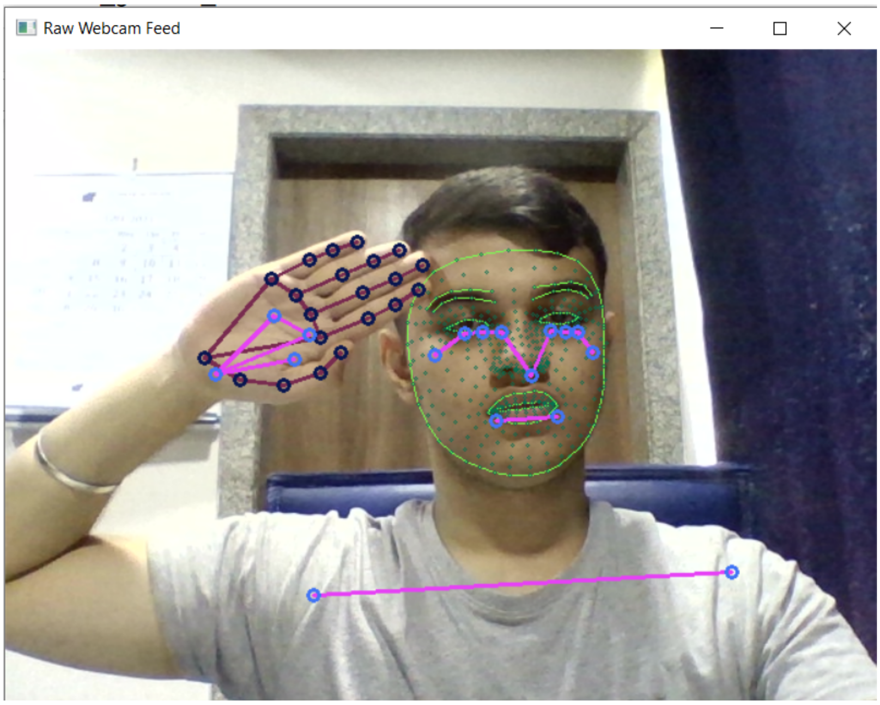

Fig 2

Model Creation:
The data is then processed from the **.csv** file using pandas and then split into testing and training data in 30:70 ratio for each class, making sure that the data point selection is random and not sequential. We then created the required ML pipelines which will standardise the data collected and contain the algorithm that will be used for training from the data that will be provided to it.

We created 4 such pipelines for 4 different algorithms, which are :

1. Logistic Regression
2. Ridge classifier
3. Random Forest classifier
4. Gradient boosting classifier

Then the accuracy for all 4 of the created models were checked, after which Random Forest classifier was chosen for the model.

It was also chosen because of its capability of handling large datasets with high dimensionality and its power of enhancing the accuracy of the model thereby preventing the overfitting issue.
The model created was finally then used in the execution code to create the real time body gesture recogniser.

2. Gesture Controlled Mouse and Audio level Controller

We use the power of mediapipe for locating juncture points on the hand using a hand recognition python module.
It uses Pre - existing python libraries like pycaw (Python Core Audio Windows Library), PyAutoGUI (a cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse keyboard), autopy (AutoPy is a simple, cross-platform GUI automation library for Python).
We define a field of view on the screen and capture the coordinate points of the fingertips, with the index finger corresponding to the mouse pointer and the movement of the thumb corresponding to a click, the transition of the mouse to volume by turning the hand into a **fist** (Transitional phase).
The index finger and thumb are then used to change the volume of the device.
In either case the junction points are recognised and their movements are recorded and corresponding changes are made. The coordinates of the fingertips and their movement in space is captured in real time and HCI takes place.

Chapter 6

Softwares

The Technology stack for this project has been varied, the softwares, libraries and modules used for this project are as follows:

1. **Jupyter Notebook:** The IDE used for creating the project and writing all the fundamental codes.
2. **Pandas:** Data manipulation library used for working on the .csv file.
3. **Numpy:** Math library used to write code to make the data efficient to work on.
4. **Mediapipe:** An open source cross-platform framework for building multimodal applied machine learning pipelines, for accurate data collection and gesture recognition.
5. **Opencv:** A library of programming functions mainly aimed at real-time computer vision.
6. **Pycaw:** Python Core Audio Windows Library.
7. **Autopy:** It is a simple, cross-platform GUI automation library for Python.
8. **Pyautogui:** A cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse keyboard.
9. **Sklearn:** A free software machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

10. **Microsoft Excel:** Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications.

The project has been written in the **Python** programming language.

Chapter 7

Real Time Simulation Results

Snippets from the real time simulation of the first and second part of the project can be found below:

1. Body gesture Recogniser

Happiness:

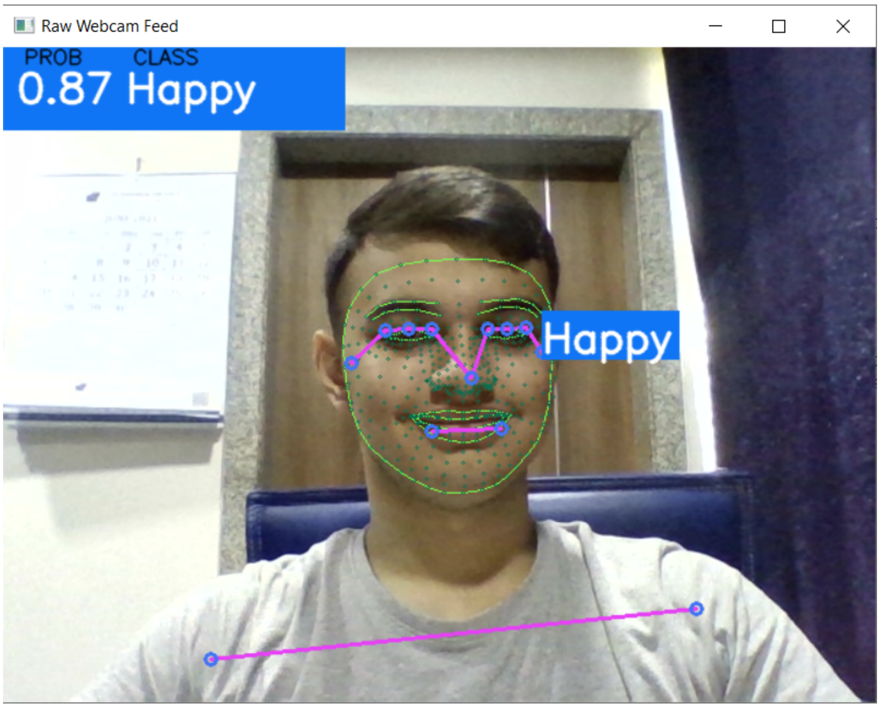

Fig 3

Drowsiness:

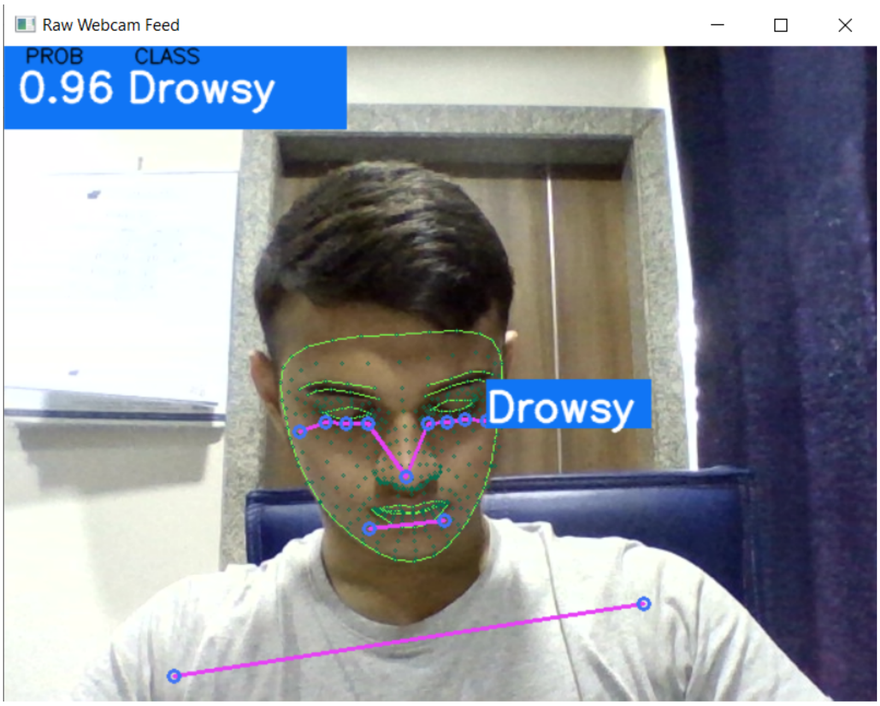

Fig 4

Salute:

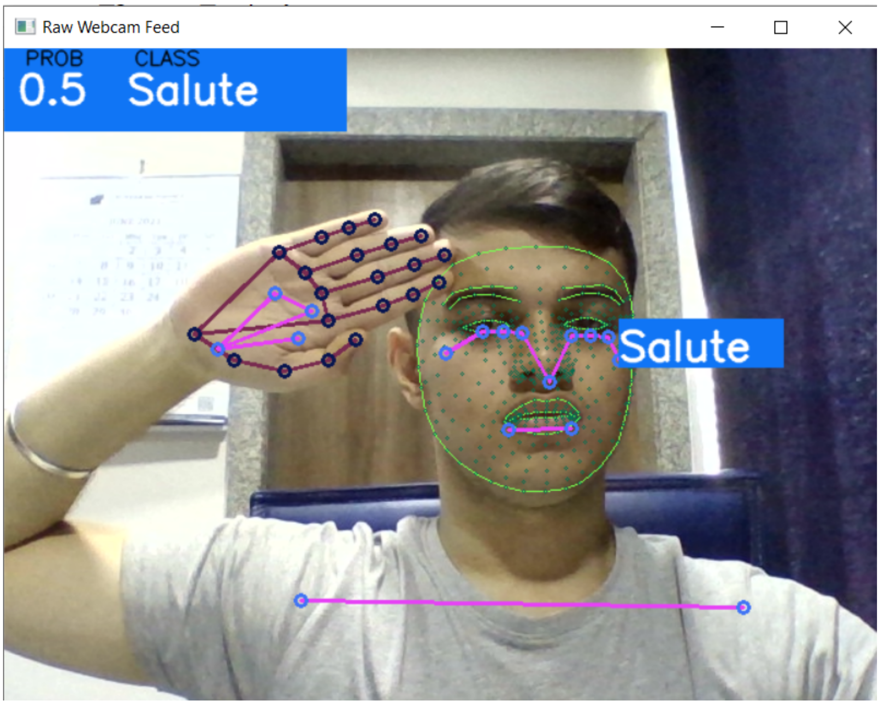

Fig 5

Thinking Stance:

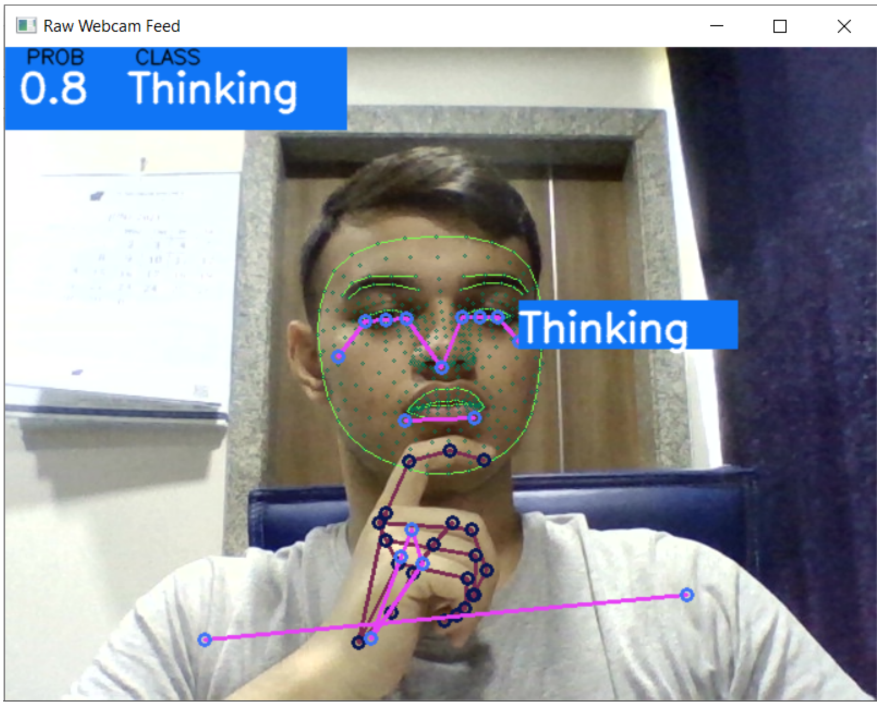

Fig 6

Victory sign:

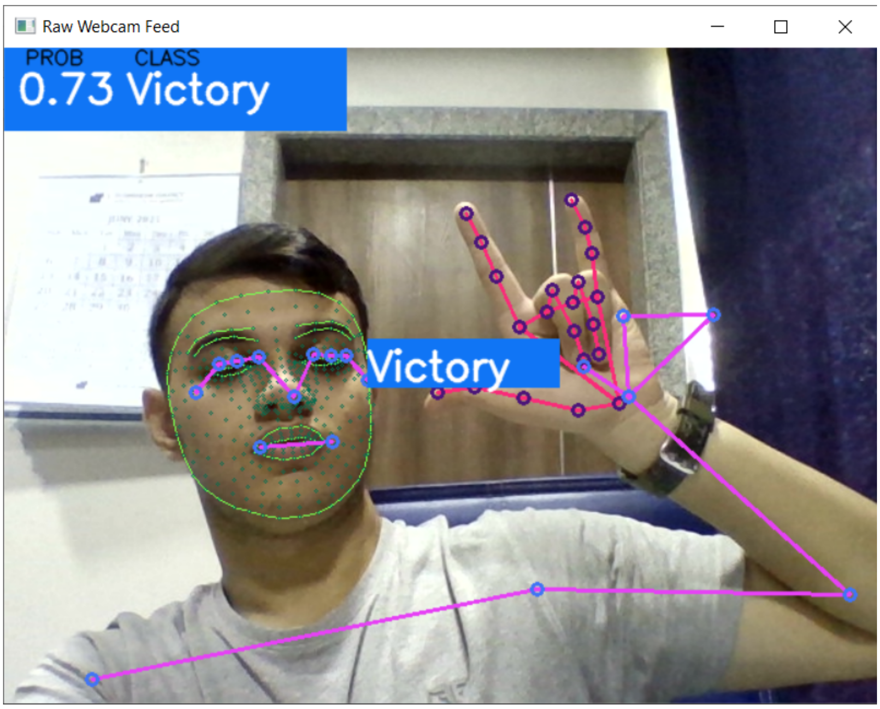

Fig 7

Victory sign with the Right hand:

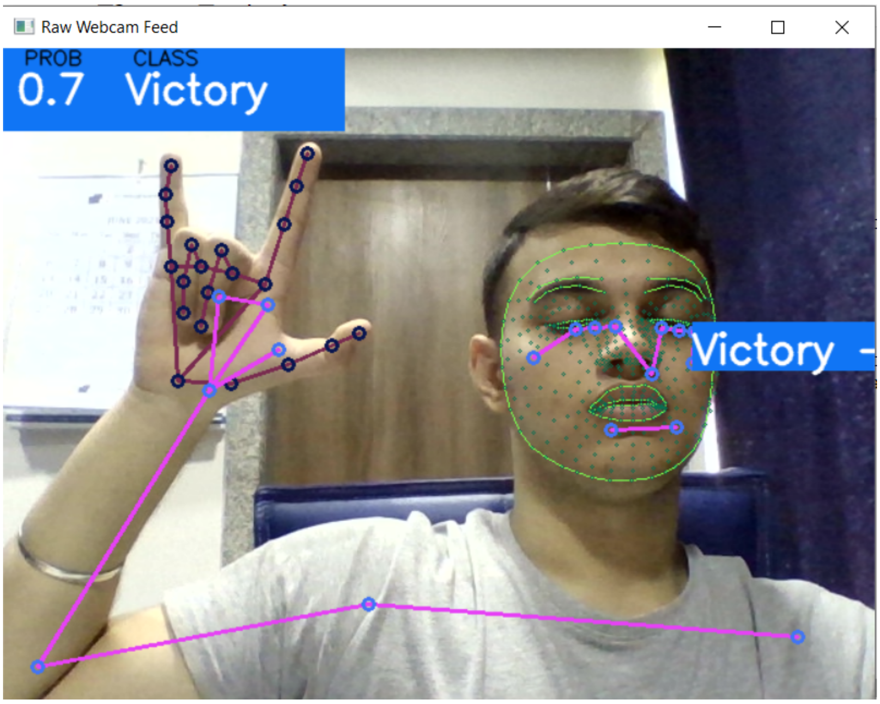

Fig 8

2. Gesture Controlled Mouse and Audio level Controller

Cursor:

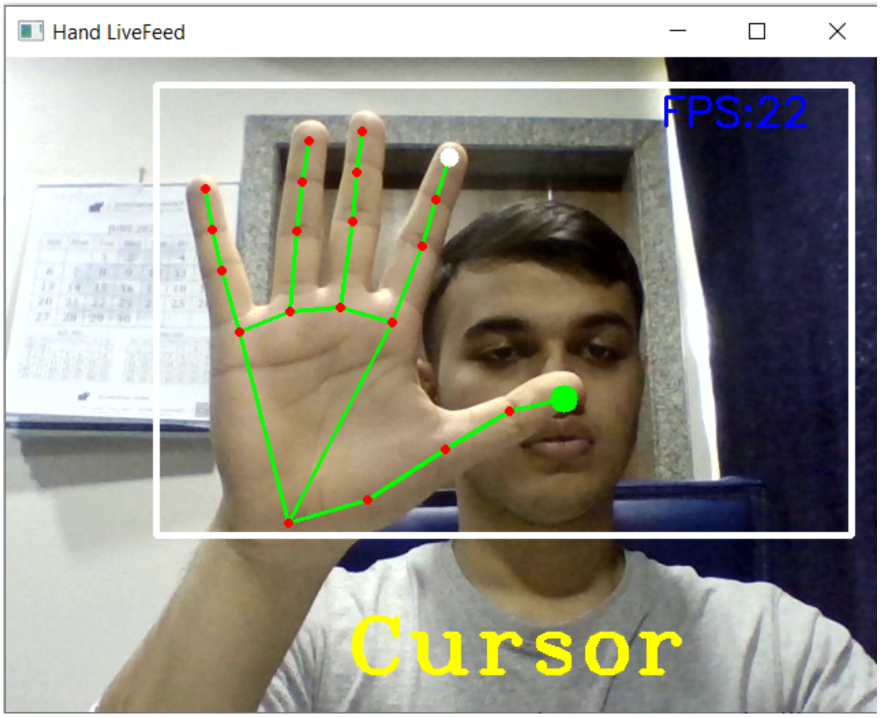

Fig 9

Transitional state :

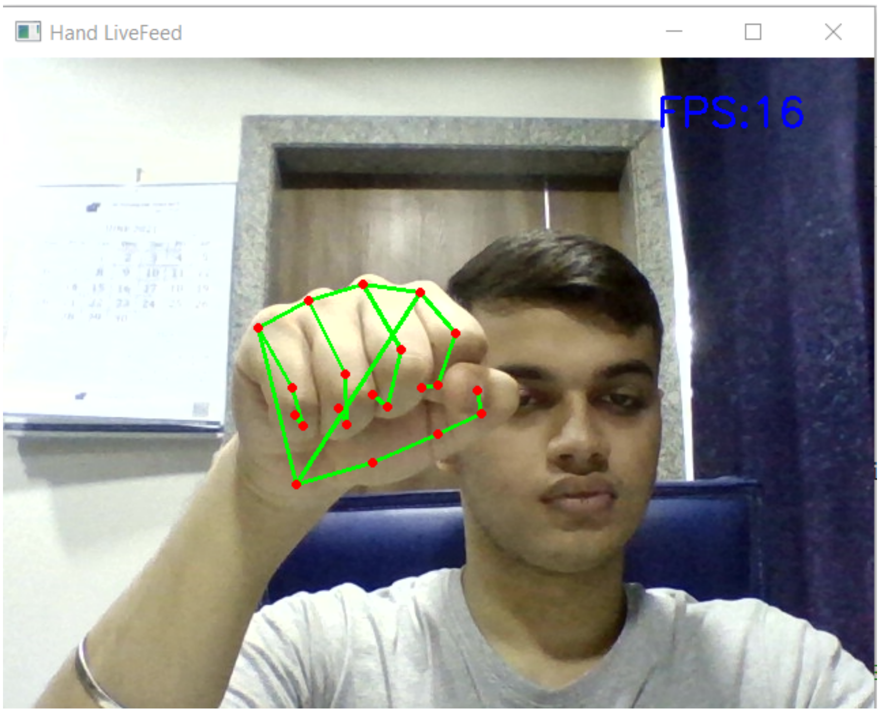

Fig 10

Volume :

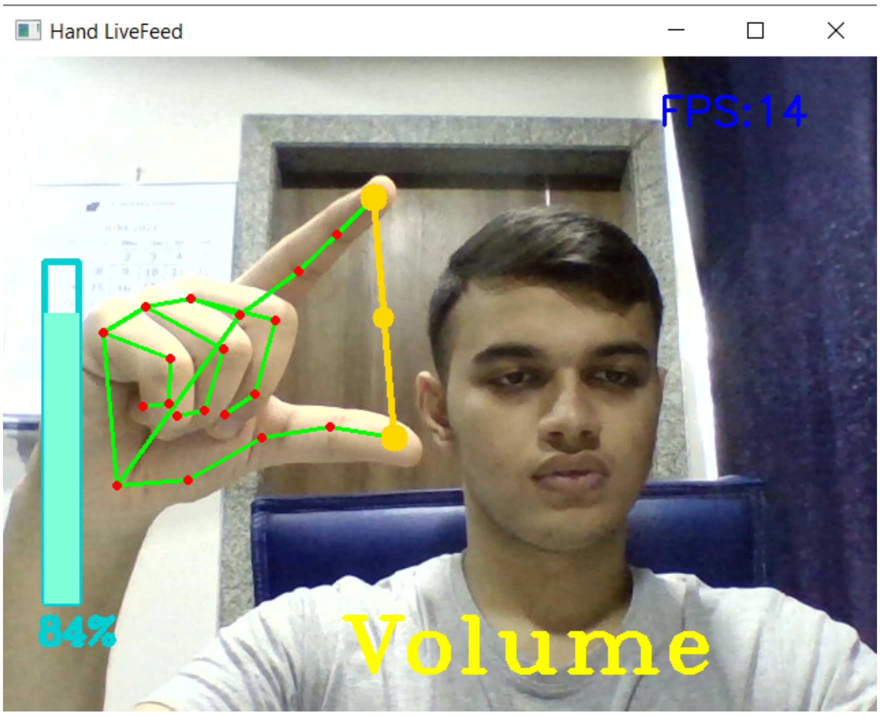

Fig 11

Chapter 8

Conclusion

The objectives of the project were met, the model was successfully able to identify the 5 gestures that it was taught to recognise, thereby confirming the legitimacy of the use of random forest classifier for training the model. The successful functioning of the model also confirms the accuracy of the data collection method. The HCI component of the project(i.e Gesture controlled mouse) also works efficiently with decent stability, fulfilling the project's vision of providing a real-world application of Gesture recognition and reinforce it's significance in modern world's Human-Computer interaction.

Scope for Future Work
The future scope for the project is to create a suitable user interface for the model(i.e the gesture recogniser) and the Gesture controlled mouse to be able to deploy it on the web for everybody's use.

Bibliography

[1] Sklearn Random forest classifier documentation:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[2] **https://www.javatpoint.com/machine-learning-random-forest-algorithm**

[3] Mediapipe documentation :
https://google.github.io/mediapipe/solutions/pose.html
https://google.github.io/mediapipe/solutions/facemesh.html
https://google.github.io/mediapipe/solutions/hands.html

[4] Opencv documentation :
https://docs.opencv.org/3.4/d2/d96/

[5] Zhi-hua Chen and Jung-Tae Kim, 'Real-Time Hand Gesture Recognition Using Finger Segmentation', Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China **https://www.hindawi.com/journals/tswj/2014/267872/**

[6] Siddharth S.Rautaray and Anupam Agrawal, 'Real Time Gesture Recognition System for Interaction in Dynamic Environment'.
https://www.sciencedirect.com/science/article/pii/S221201731200374X