NLP Based Virtual Mouse

Submitted to the
Department of Master of Computer Applications
in partial fulfilment of the requirements
for the Mini Project (MCAP1)

by

Ajay Zad
1MS21MC006

Under the guidance of

Dr. Jagannatha S

Department of Master of Computer Applications

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)
Accredited by National Board of Accreditation & NAAC with ‘A+’ Grade
MSR Nagar, MSRIT Post, Bangalore-560054
www.msrit.edu

2023

RAMAIAH
Institute of Technology

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled NLP Based Virtual Mouse is carried out by

	Student Name	USN
1)	Ajay Zad	1MS21MC006

students of 3rd semester, in partial fulfillment for the Mini Project (MCAP1), during the academic year 2022-2023.

Guide **Head of the Department**

Dr. Jagannatha S

Name of Examiners **Signature with Date**

1.

2.

DECLARATION

I hereby declare that the project report entitled " NLP Based Virtual Mouse " based on study undertaken by me, towards the partial fulfilment for the Mini Project (MCAP1) carried out during the 3rd semester, has been compiled purely from the academic point of view and is, therefore, presented in a true and sincere academic spirit. Contents of this report are based on my original study and findings in relation there to are neither copied nor manipulated from other reports or similar documents, either in part or in full, and it has not been submitted earlier to any University/College/Academic institution for the award of any Degree/Diploma/Fellowship or similar titles or prizes and that the work has not been published in any specific or popular magazines.

Place: Bangalore **Ajay Zad**

Date: **1MS21MC006**

ACKNOWLEDGEMENTS

We take this opportunity to acknowledge the contribution of each individual who has in some way or the other helped us in completing this project successfully.

We express our gratitude to our institute, **Ramaiah Institute of Technology (MCA section)**, **Bangalore** and our Principal **Dr. N.V.R. Naidu** for being the source of encouragement.

We also enhance our gratitude to **Dr.Seema Shedole**, co-ordinator MCA Department and internal guide **Dr. Jagannatha S** for their constant inspiration and necessary resources and working environment in the college who has guided us with their valuable suggestions and guidance. Our expressions extend unbounded to thank our most beloved parents and family members who have always been moral support and strong pillars at every stage of life with cheer enthusiasm. We dedicate our work to them. Last but not the least, we are thankful for Almighty's moral support, which helped us during the successful completion of the project.

ABSTRACT

The project is based on real time tracking of human face for controlling the mouse cursor. This project aims to come up with the system that will eliminate the need of touch to mouse pad to control the cursor. The system will use facial movements as triggers for cursor movement on the screen. The user's real time video for the mouse movement is captured using a webcam. The mouse clicks are done by the commands passed by the user.

In NLP based Virtual mouse clicks are done through voice commands passed by the user. The commands passed could be for a click, right click, left click, double click, scroll up and scroll down, close and minimize. Python's Speech Recognition, OpenCV module, Mediapipe module, pyttsx3 module, pyaudio modules have been used to create this project.

Table of Contents

1. Introduction 1
   1.1 Overview 1
   1.2 Problem Definition 2
2. Literature Survey 3
3. Hardware and Software Requirements 6
   3.1 Hardware Requirements 6
   3.2 Software Requirements 6
4. Software Requirements Specification 7
   4.1 System Features 7
      4.1.1 Speech Recognition 7
      4.1.2 Face Motion Detection 7
      4.1.3 Natural Language Processing 7
      4.1.4 Voice Control 7
      4.1.5 Customizable commands 7
5. System Design Description (SDD) 8
   5.1 System Overview 8
   5.2 Functional Design 9
      5.2.1 Describe the functionalities of the system: 9
      5.2.2 Behavioral design: 10
6. Testing 11
   6.1 Description of Testing 11
   6.2 Test Cases 11
7. Implementation 13
8. Results and Discussion 17
9. Conclusion 20
10. Scope for Further Enhancement 21
11. Bibliography 22

1. Introduction

1.1 Overview

A natural language processing (NLP) based virtual mouse is a system that allows users to control a computer mouse using natural language commands instead of traditional mouse controls. The system utilizes NLP techniques to interpret and understand the user's verbal commands, converting them into actions on the computer screen. This allows users to interact with the computer in a more intuitive and natural way, potentially making it easier for individuals with disabilities or limited mobility to use a computer. In the proposed project we have made use of multiple built-in python modules and packages such as Speech Recognition, OpenCV module, Mediapipe module, pyttsx3 module and pyaudio module. The Python Speech Recognition module is a library that enables users to perform speech recognition tasks using Python programming language. It provides a simple interface for converting spoken words into text and works on most platforms, including Windows, macOS, and Linux. The module is built on the pocket sphinx library, which is an open-source, cross-platform, lightweight, and flexible engine for performing speech recognition. Using this module, developers can build applications that support speech recognition and speech-to-text transcription.

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly used for real-time computer vision. OpenCV provides a large set of computer vision algorithms, including object detection, image processing, camera calibration, and machine learning algorithms. These algorithms can be used to perform various tasks such as face detection, object tracking, background subtraction, and more. The Mediapipe allowing developers to write computer vision and audio processing pipelines in Python. This makes it easier to experiment with and implement computer vision and audio processing algorithms in a high-level programming language, the Python MediaPipe module is a powerful tool for building computer vision and audio processing pipelines, and it can be used for a wide range of applications. pyttsx3 module, developers can add TTS functionality to their applications, allowing users to hear the computer speak. This can be useful in a variety of applications, such as educational software, assistive technology, and more. The library provides a simple interface for converting text to speech, and it supports adjusting various parameters, such as speech rate, volume, and voice. The module pyaudio, the library supports a variety of audio formats and provides a simple interface for working with audio streams, making it an excellent choice for a wide range of audio processing applications, such as speech recognition, audio recording and playback, and more.

1.2 Problem Definition

NLP (Natural Language Processing) based virtual mouse is to develop a system that allows users to control their computer using natural language commands. This involves several challenges, including:

1. **Speech recognition**: Accurately recognizing the spoken commands and converting them into text that can be processed by the system.
2. **Natural language understanding**: Interpreting the spoken commands and determining the intended action, such as moving the mouse or clicking on an item.
3. **Interfacing with the operating system**: Connecting to the computer's operating system to perform the intended actions, such as moving the mouse or clicking on an item.

Overall, the problem definition of a natural language processing based virtual mouse is to develop a system that can accurately recognize and interpret natural language commands and use them to control a computer, in a user-friendly and intuitive way.

2. Literature Survey

1. *Mr.Dhanaraju, Dr.Sreenivas Mekala, A.Harsha Vardhan Rao, CH.Pavan Kumar, R.Lokesh "Human-Eye Controlled Virtual Mouse" IJRASET Journal 2022*

   Methodology: Python language is used to write the code. Python provides a wide variety of libraries for scientific and computational usage. Libraries such as hashlib, rsa. Deep learning models may currently be the most effective for face identification. Face detection, on the other hand, existed before deep learning. Previously, traditional feature descriptors and linear classifiers were a great way to recognize faces .HOG and Linear SVM, to be precise. The HOG (Histogram of Oriented Gradients) feature descriptor and a Linear SVM machine learning approach are used to identify faces. HOG is an easy-to-understand and useful feature description. It is frequently employed in the detection of objects, such as vehicles, dogs, and fruits, in addition to face detection. Because the local intensity is used to characterize the geometry of the item, HOG is reliable for object detection.

2. *Ashish Mhetar, B K Sriroop, Kavya AGS, Ramanath Nayak, Ravikumar Javali, Suma K V "Virtual Mouse" ResearchGate Journal 2014*

   Methodology: A lot of virtual markers are available in market, but it can only function as a marker alone. A high level processor is used to process the data and used as a virtual marker, but it can do even more function like mouse functions. Its capability is not being utilized fully to its ability to function and hence the product becomes under-loaded and hence costlier in terms of market price versus function.

3. *Mr.Gajendra Moroliya, Mr.Sahil Patwekar, Prof.S.P. Gopnarayan "Virtual Mouse Using Hand Gesture" JETIR Journal 2018*

   Methodology: This project presents a new approach for controlling mouse movement using a real-time camera. Major approaches consist of adding more buttons or changing the position of the tracking ball of mouse. Instead, we suggest to change the design of hardware. Our concept is to use a camera and computer vision technology, as image segmentation and gesture recognition, to control mouse tasks (clicking and scrolling) and we show how it can perform everything current mouse devices can. This project shows how to build this mouse control system.

4. *<u>Xuebai Zhang, Xiaolong Liu, Shyan-Ming Yuan, Shu-Fan Lin "Eye Tracking Based Control System for Natural Human-Computer Interaction" Hindawi Journal 2017</u>*

   Methodology: Eye movement can be regarded as a pivotal real-time input medium for human-computer communication, which is especially important for people with physical disability. In order to improve the reliability, mobility, and usability of eye tracking technique in user-computer dialogue, a novel eye control system with integrating both mouse and keyboard functions is proposed in this paper. The proposed system focuses on providing a simple and convenient interactive mode by only using user's eye. The usage flow of the proposed system is designed to perfectly follow human natural habits. Additionally, a magnifier module is proposed to allow the accurate operation. In the experiment, two interactive tasks with different difficulty (searching article and browsing multimedia web) were done to compare the proposed eye control tool with an existing system.

5. *<u>Xiaozhi Yang, Ian Krajbich "Webcam-based online eye-tracking for behavioral research" Cambridge Core Journal 2023</u>*

   Methodology: Experiments are increasingly moving online. This poses a major challenge for researchers who rely on in-lab techniques such as eye-tracking. Researchers in computer science have developed web-based eye-tracking applications (WebGazer; Papoutsaki et al., 2016) but they have yet to see them used in behavioral research. This is likely due to the extensive calibration and validation procedure, inconsistent temporal resolution (Semmelmann & Weigelt, 2018), and the challenge of integrating it into experimental software. Here, we incorporate WebGazer into a JavaScript library widely used by behavioral researchers (jsPsych) and adjust the procedure and code to reduce calibration/validation and improve the temporal resolution (from 100–1000 ms to 20–30 ms). We test this procedure with a decision-making study on Amazon MTurk, replicating previous in-lab findings on the relationship between gaze and choice, with little degradation in spatial or temporal resolution

6. *Tata Jagannadha Swamy, M Nandini, Nandini B, Venkata Karthika K, V Laxmi Anvitha, Ch Sunitha "Voice and Gesture based Virtual Desktop Assistant for Physically Challenged People" IEEE Xplore Journal 2022*

   Methodology: In the modern era computers are becoming faster, smarter and better. Their usage is rising in the fields such as medicine, business administration, education etc. So there is a need for simplifying the operability and usability of a computer. Digital virtual assistants are known for easing the interaction with computers. Since most of the digital virtual assistants use voice as mode of communication deaf and dumb persons are finding it difficult to use virtual assistants on their devices. This research work attempts to propose a voice and gesture based virtual assistant that can be used by disabled as well as non-disabled persons to perform common tasks on their computers. The main aim of this research paper is to develop natural human-machine interaction.

7. *Mohamed Nasor, Mujeeb Rahman K K, Maryam Mohamed Zubair, Haya Ansari, Farida Mohamed "Eye-Controlled Mouse Cursor for Physically Disabled Individual" RearchGate Journal 2018*

   Methodology: The first step was to use a face detection algorithm locate the face on an image frame captured by an ordinary webcam. The next step was to detect only the eyes from this frame. We consider tracking only one eye movement for faster processing time. Then the iris movement was tracked. Since the color of the iris is black, its image has a significantly lower intensity compared to the rest of the eye. This helps us in easy detection of the iris region. Taking the left and right corners of the eye as reference points the shift of the iris as the person changed his eyes focus was determined. The shift was then used to map cursor location on the test graphical user interface (GUI).

3. Hardware and Software Requirements

3.1 Hardware Requirements

- System Processor: i5
- Hard-disk: 500 GB
- Processor Speed: 1.2 GHz
- RAM: 4 to 8 GB
- Monitor: 12” LED
- Microphone: Internal or External
- Webcam: Internal or External

3.2 Software Requirements

- Operating System: Windows 10, macOS
- Coding Language: Python3
- Software used: Spyder 5.0
- Packages: Speech Recognition, OpenCV, Mediapipe, pyttsx3 and pyaudio.

4. Software Requirements Specification

4.1 System Features

The proposed project has multiple static features such as Speech recognition, Face Motion detection, Natural language processing, Voice control and Customable command.

4.1.1 Speech Recognition

Speech recognition, also known as speech-to-text or voice recognition, is a technology that allows computers to transcribe spoken words into written text. It has various applications, including voice-controlled devices, voice-activated personal assistants, and speech-to-text software for people with disabilities. Speech recognition works by first converting spoken words into a digital signal, which is then analyzed by the computer using complex algorithms and machine learning techniques. The computer then compares the digital signal to a large database of recorded speech to find the closest match.

4.1.2 Face Motion Detection

Motion detection of face in AI refers to the process of detecting and tracking the movements of human faces within an image or video stream. The goal of motion detection of face is to identify and track faces in real-time, and is commonly used in applications such as facial recognition and human-computer interaction. The accuracy of motion detection of face can be improved by using deep learning techniques.

4.1.3 Natural Language Processing

The system can understand and interpret natural language commands, allowing for a more intuitive and human-like interaction with the computer.

4.1.4 Voice Control

The system allows for control of the virtual mouse cursor using voice commands, eliminating the need for physical mouse.

4.1.5 Customizable commands

The system allows users to define and customize their own voice commands, enabling a more personalized and efficient experience.

5. System Design Description (SDD)

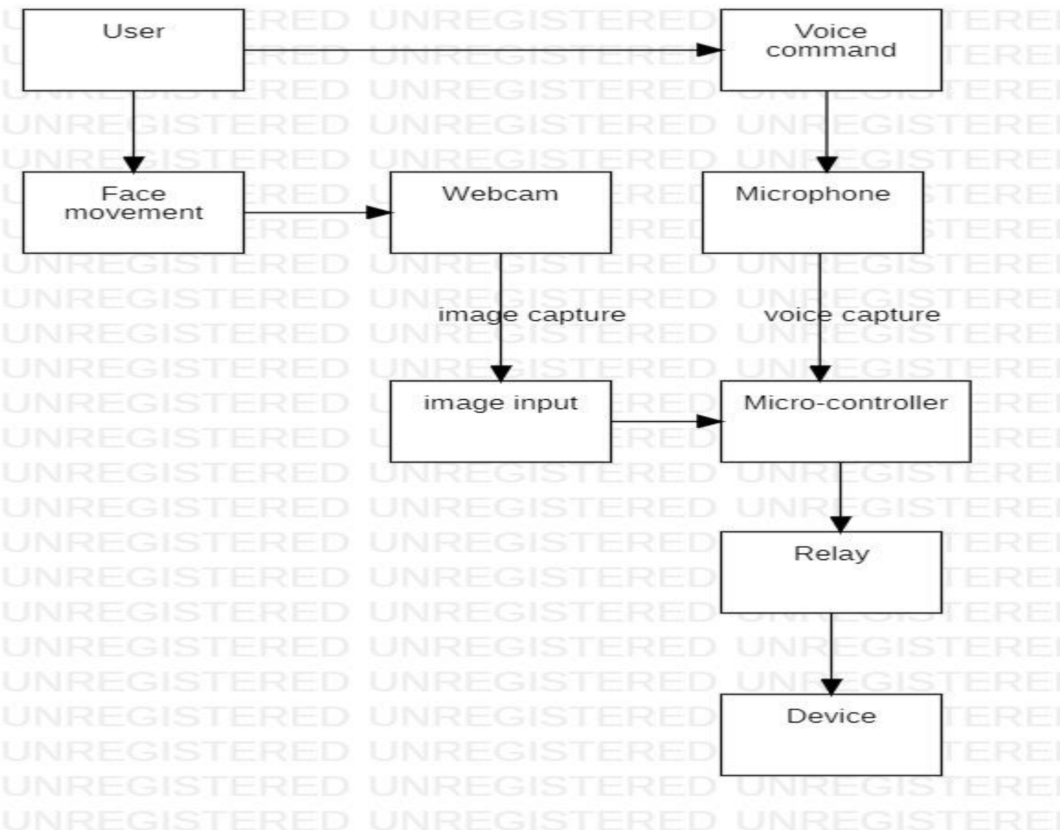
5.1 System Overview

1) **Camera:** In this project, the camera is used to capture/record the head movements done by the user. Any of the internal or external camera can be used for the purpose. The head movements done by the user is been analyzed and with the help of this input mouse cursor movements are performed.

2) **Microphone:** In this project, the microphone is used to capture/record the voice commands given by the user. Any of the internal or external can be used for the purpose. The voice commands given by the user is analyzed and with the help of this input mouse click operations are been performed such as click, right click, left click, double click.

3) **Mouse Event:** In this project, the mouse events are been performed by capturing/recording the images with the help of the camera. We even make use of microphone to capture/record the voice commands from the user. Combining both these processes we are able to perform the required mouse event.

5.1.1 Architecture Diagram



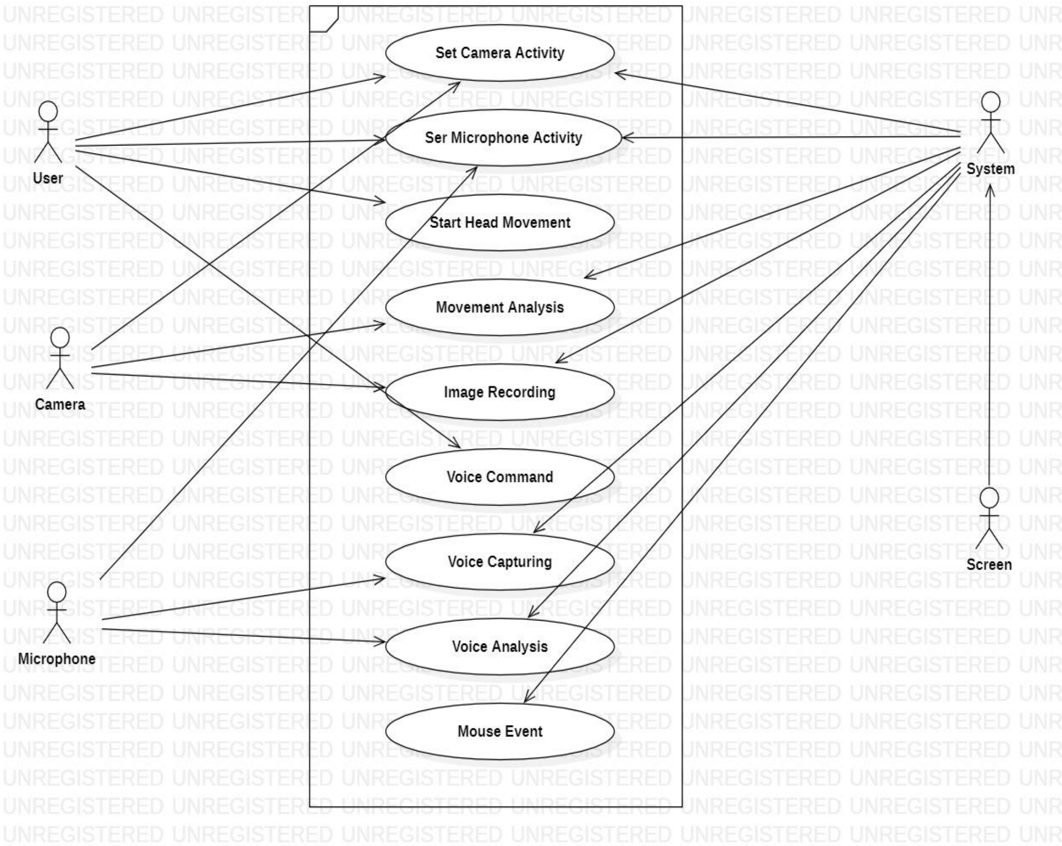
(Diagram 1)

5.2 Functional Design

The functional design of an NLP based virtual mouse system is designed to process speech input, identify the user's intent, map the intent to mouse actions, execute the actions, and provide feedback to the user. The system should be able to handle a wide range of natural language commands and provide a seamless and intuitive experience for the user.

5.2.1 Describe the functionalities of the system:

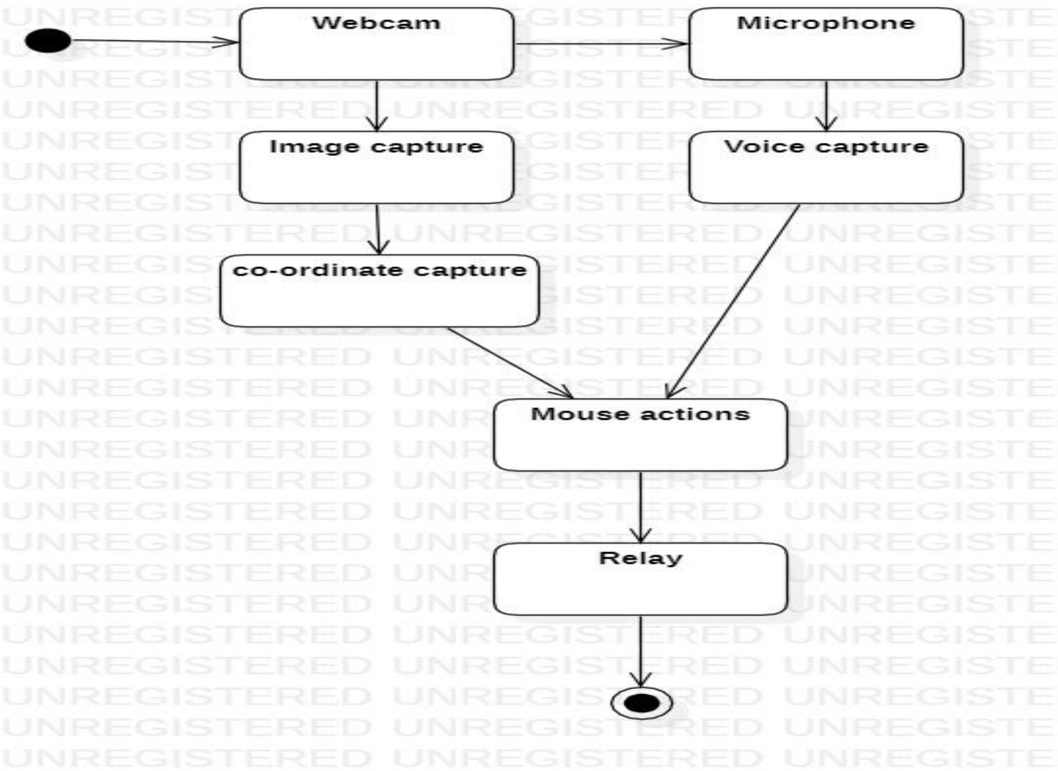
Use-Case Diagram



(Diagram 2)

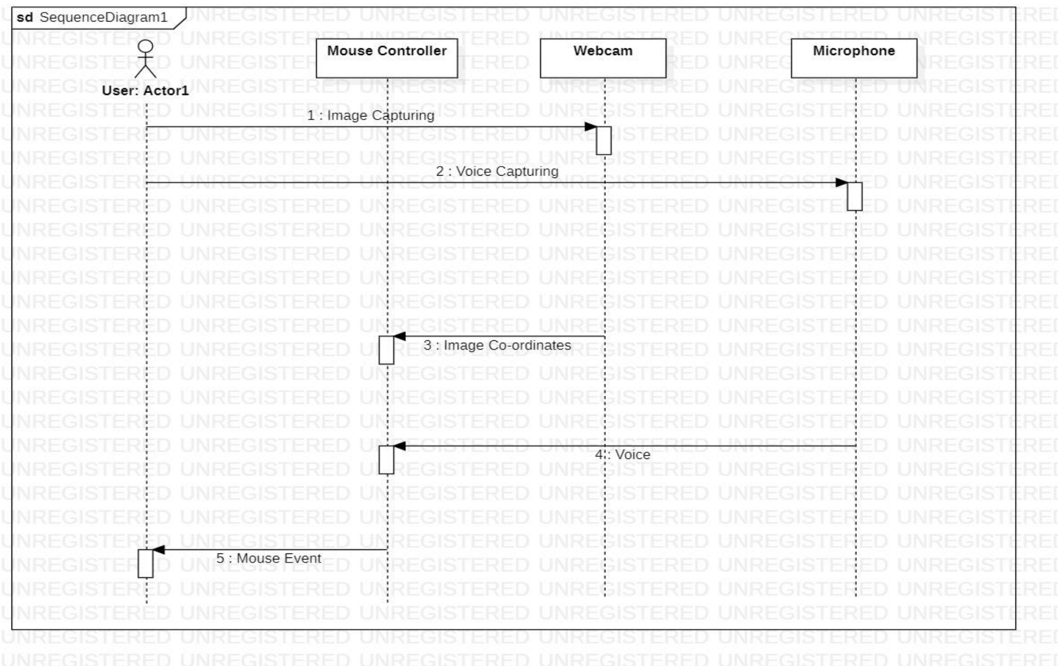
5.2.2 Behavioral design:

Activity Diagram



(Diagram 3)

Sequence Diagram



(Diagram 4)

6. Testing

6.1 Description of Testing

The testing process should be iterative and ongoing, with regular testing and verification to ensure that the system continues to meet the requirements and specifications. The testing process should also be flexible, allowing for changes and updates to the system as needed.

6.2 Test Cases

Test case	Test case Name	Test case Description	Inputs	Expected Output	Actual Output	Status
1	Basic mouse actions	Test the system's ability to recognize and execute basic mouse actions such as moving the cursor, clicking, and scrolling.	Voice command (String) Ex: Left Click Right Click Double Click Scroll-up Scroll-down Close Minimize	Triggering Mouse Events	Triggering Mouse Events	Successful
2	Speech recognition accuracy	Test the accuracy of the speech recognition module by providing a range of speech inputs, including	Voice Command (String)	Recognise the command and trigger the following mouse event	Recognise the command and triggering the mouse event	Intermediate

		different accents and background noise levels.				
3	Intent classification accuracy	Test the accuracy of the intent classification module by providing a range of text inputs that represent different mouse actions.	Voice command (String) Ex: Left Click Right Click Double Click Scroll-up Scroll-down Close Minimize	Different Mouse Event Trigger	Different Mouse event trigger	Successful
4	Cursor Movements	Test the accuracy of the movement of the cursor, following the face detection and leading the mouse to the expected co-ordinates on the screen	Face movements	Moving to expected screen co-ordinates	Moving to expected screen co-ordinates	Successful

7. Implementation

```
import cv2
import mediapipe as mp
import pyautogui
import speech_recognition as sr
import pyttsx3
import pyaudio


ans = int(input("enter 0  for head and 1 for hand: "))
if ans == 0:
    cam = cv2.VideoCapture(0)
    face_mesh = mp.solutions.face_mesh.FaceMesh(refine_landmarks=True)
    screen_w, screen_h = pyautogui.size()
    pyautogui.FAILSAFE = False
    while True:
        _, frame = cam.read()
        frame = cv2.flip(frame,1)
        rgb_frame = cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
        output = face_mesh.process(rgb_frame)
        landmark_points = output.multi_face_landmarks
        frame_h, frame_w, _ = frame.shape
        cnt = 0
        if landmark_points:
            landmarks = landmark_points[0].landmark
            for id,landmark in enumerate(landmarks[0:478]):
                x = int(landmark.x * frame_w)
```

```
            y = int(landmark.y * frame_h)
            cv2.circle(frame,(x,y),3,(0,255,0))
            if id == 1:
                screen_x = int(landmark.x * screen_w)
                screen_y = int(landmark.y * screen_h)
                pyautogui.moveTo(screen_x,screen_y)
                print(screen_x,screen_y)
    cv2.imshow("HEAD controlled mouse",frame)
    cv2.waitKey(1)

    listener = sr.Recognizer()
    try:
        with sr.Microphone() as source:
            print("listing")
            listener.adjust_for_ambient_noise(source)
            voice = listener.listen(source,.5,1)
            command = str(listener.recognize_google(voice))
            print(command)
            c1 = command.lower()
            print(c1)
            if 'left' in c1 :
                print("left click")
                pyautogui.click()
            elif 'right' in c1:
                print("right clcik")
                pyautogui.click(button='right')
            elif 'double' in c1:
                print("Double click")
                pyautogui.doubleClick()
```

```
            elif 'down' in c1:
                print("scrolling down")
                pyautogui.scroll(-1500)
            elif 'up' in c1:
                print("scrolling up")
                pyautogui.scroll(1500)
            elif 'start' in c1:
                print("closing..")
                pyautogui.click(0,1079)
            elif 'close' in c1:
                print("closing..")
                pyautogui.click(1872,24)
            elif 'min' in c1:
                print("Minimize")
                pyautogui.click(1776,-29)
            else:
                pass

        except:
            pass
else:
    cam = cv2.VideoCapture(0)
    face_mesh = mp.solutions.face_mesh.FaceMesh(refine_landmarks=True)
    screen_w, screen_h = pyautogui.size()
    pyautogui.FAILSAFE = False
    while True:
        _, frame = cam.read()
        frame = cv2.flip(frame,1)
        rgb_frame = cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
```

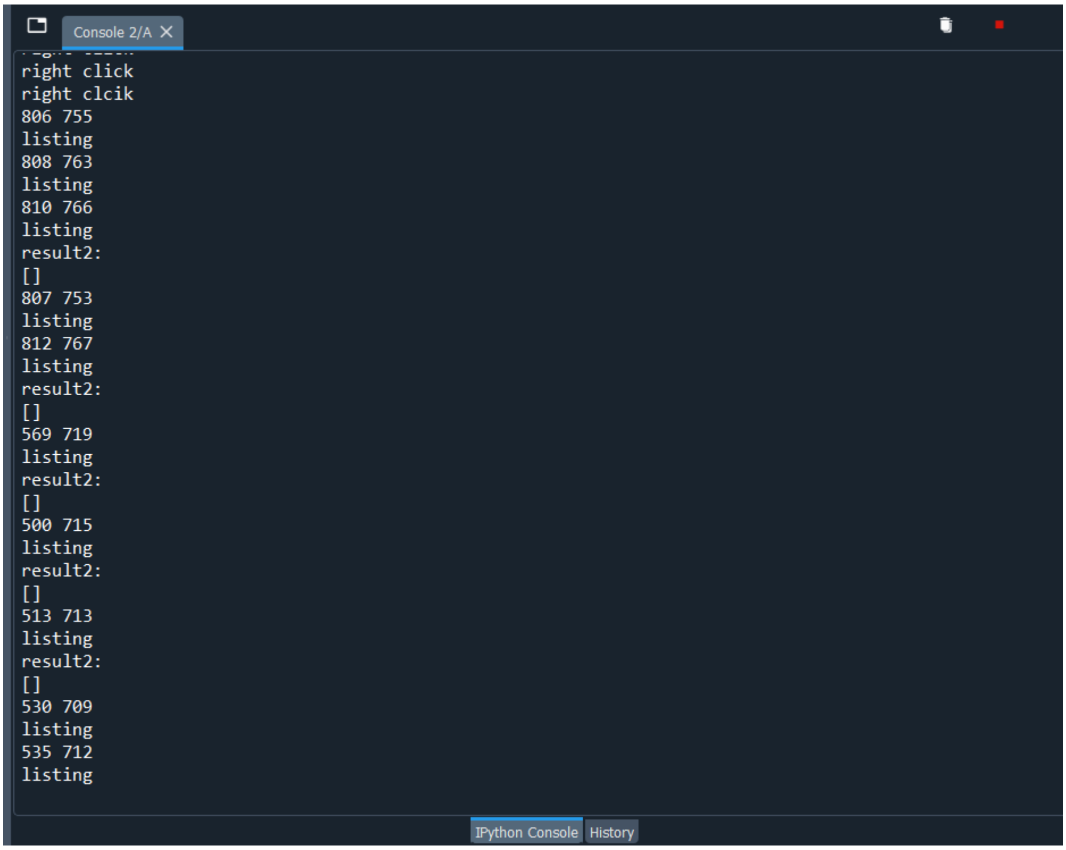
```
output = face_mesh.process(rgb_frame)
landmark_points = output.multi_face_landmarks
frame_h, frame_w, _ = frame.shape
cnt = 0
if landmark_points:
    landmarks = landmark_points[0].landmark
    for id,landmark in enumerate(landmarks[474:478]):
        x = int(landmark.x * frame_w)
        y = int(landmark.y * frame_h)
        cv2.circle(frame,(x,y),3,(0,255,0))
        if id == 1:
            screen_x = int(landmark.x * screen_w)
            screen_y = int(landmark.y * screen_h)
            pyautogui.moveTo(screen_x,screen_y)
    left =[landmarks[145],landmarks[159]]
    for landmark in left:
        x = int(landmark.x * frame_w)
        y = int(landmark.y * frame_h)
        cv2.circle(frame,(x,y),3,(0,255,255))
    l = left[0].y - left[1].y
    if l < 0.004:
        print('click')
        pyautogui.click()

cv2.imshow("HEAD controlled mouse",frame)
cv2.waitKey(1)
```

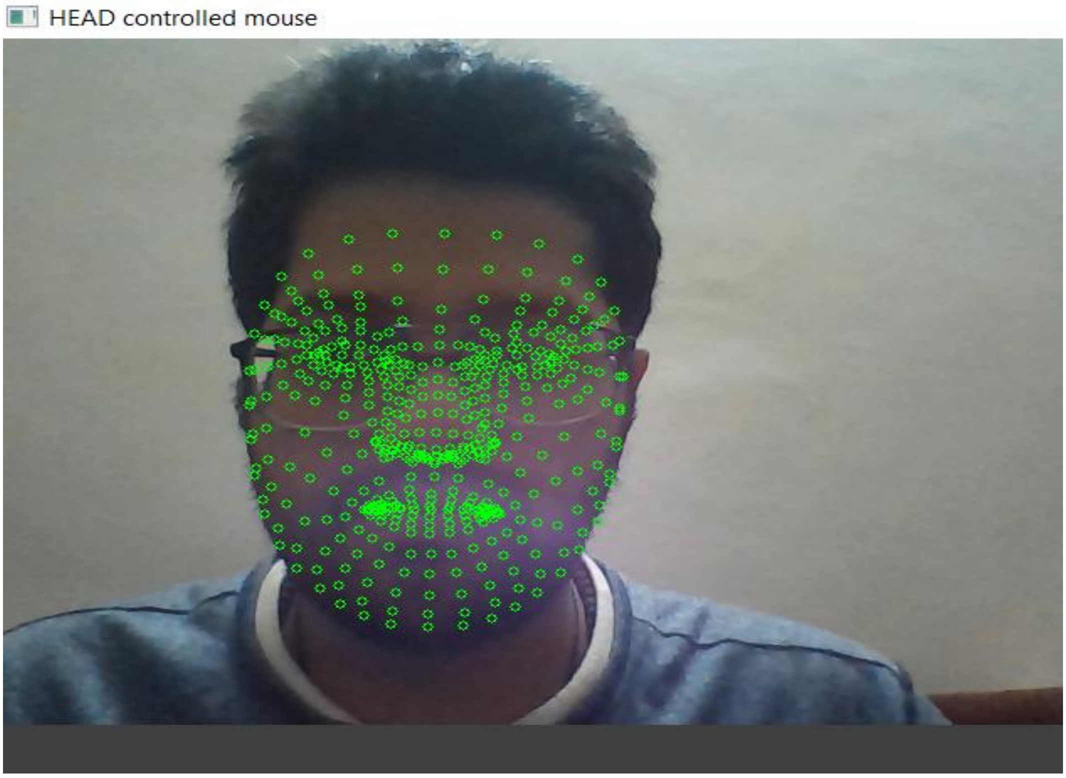
8. Results and Discussion

The results and discussion of a natural language processing based virtual mouse depend on the specific implementation and design of the system. However, in general, a virtual mouse controlled by NLP can provide a convenient and intuitive way of interacting with a computer, as users can control the mouse cursor and perform actions by simply speaking commands. The accuracy of the NLP system plays a crucial role in the effectiveness of the virtual mouse. The results of the NLP system's performance in understanding and executing commands, as well as the user's satisfaction and ease of use, can also be important factors in determining the success of the virtual mouse. Additionally, the integration of the virtual mouse with other computer applications and systems can impact the overall usefulness and practicality of the system.

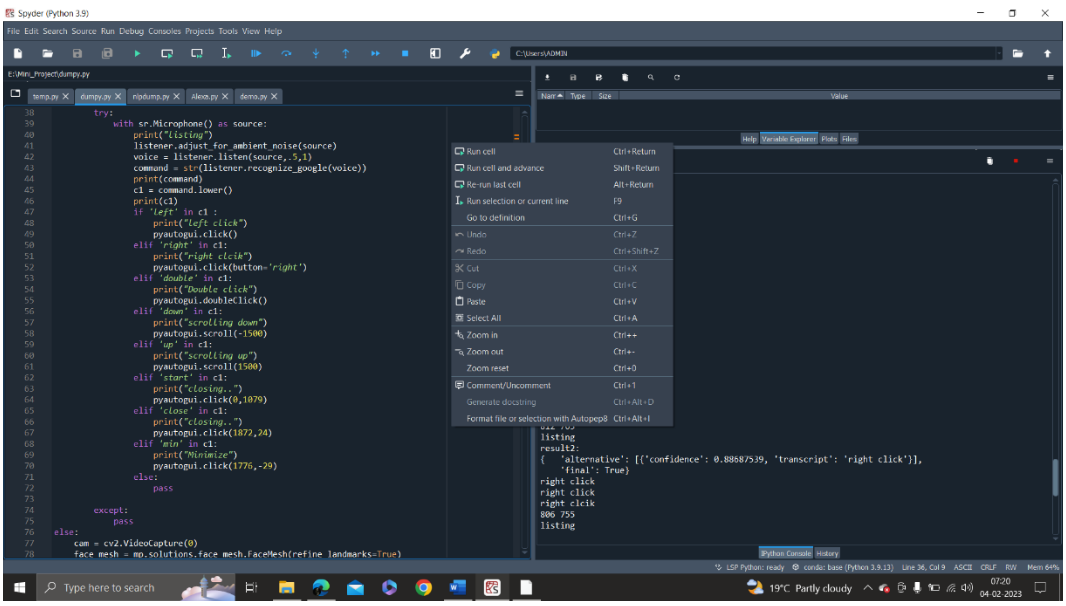
1. Program in execution, displaying the voice commands passed and the current co-ordinates of the mouse on the monitor.



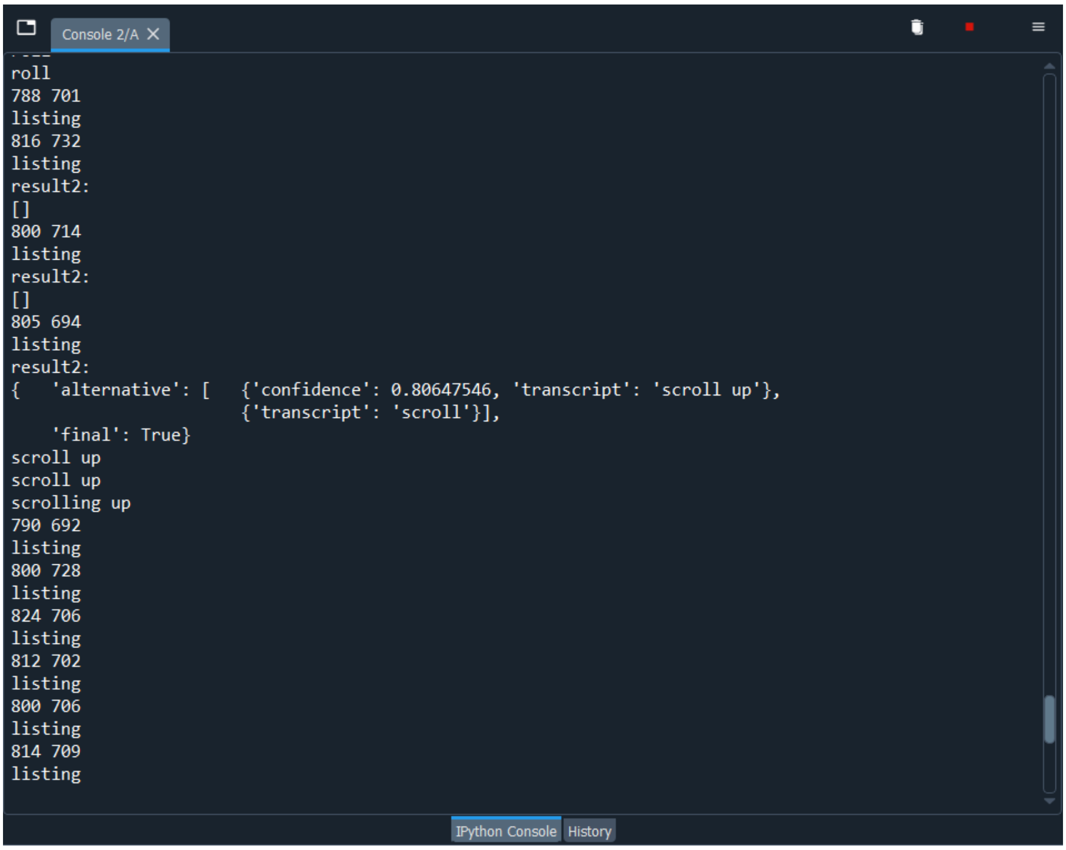
2. Face motion detection and tracking for the movement of mouse cursor on the monitor.



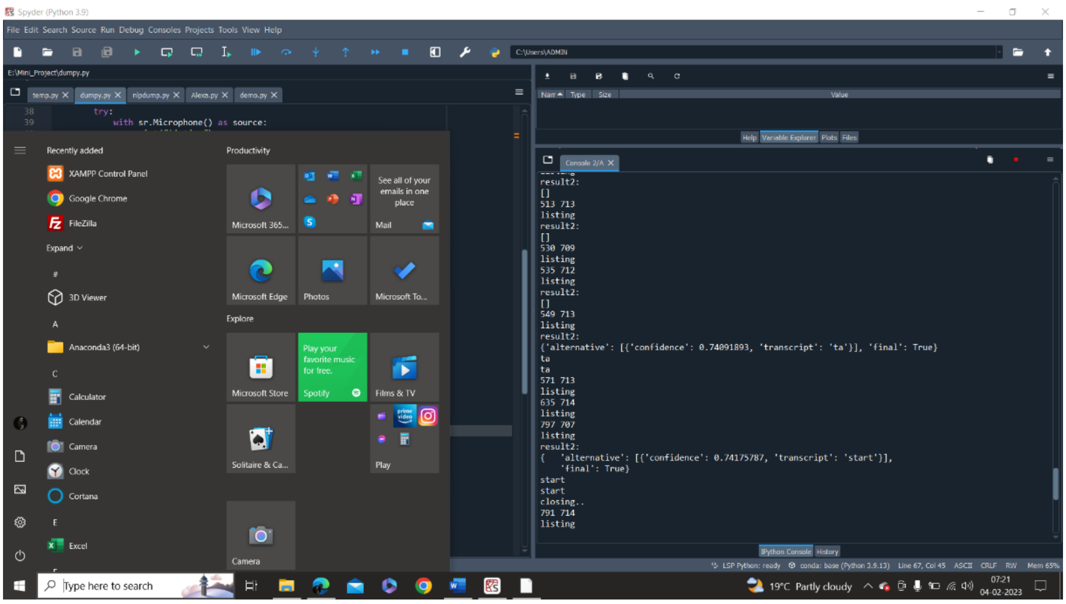
3. Passing voice command 'Right Click' to the application and shows the following execution for the voice command passed.



4. Passing voice command ‘Scroll-Up’ to the application and shows the following execution for the voice command passed.



5. Passing voice command ‘Start’ to the application and shows the following execution for the voice command passed.



9. Conclusion

The main objective of the NLP based virtual mouse system is to control the mouse cursor functions by using the face gestures to move the cursor on the screen and passing voice commands to trigger different types of click and scroll events. The proposed system can be achieved by using web-cam or a built-in camera which detects the face gestures and processes these frames to perform the particular mouse function and external or in-built microphone to pass the commands and trigger the click events in the system. The proposed system can be specially used by handicapped people (who don't have hands) to operate the cursor of the computer system and complete the specified task.

From the results of the model, we can come to the conclusion that the proposed NLP based virtual mouse has performed very well compared to the previous existing model.

10. Scope for Further Enhancement

The proposed model has some limitations such as there is no selecting and dragging and dropping feature. There is some time delay in the movement of the cursor as two functionalities are working at the same time (listening and cursor movement). Hence, we will work next to overcome these limitations by improving the logic and algorithm for face detection and produce more accurate results.

11. Bibliography

1. *Mr.Dhanaraju, Dr.Sreenivas Mekala, A.Harsha Vardhan Rao, CH.Pavan Kumar, R.Lokesh "Human-Eye Controlled Virtual Mouse" IJRASET Journal 2022*
   https://www.ijraset.com/research-paper/human-eye-controlled-virtual-mouse

2. *Ashish Mhetar, B K Sriroop, Kavya AGS, Ramanath Nayak, Ravikumar Javali, Suma K V "Virtual Mouse" ResearchGate Journal 2014*
   https://www.researchgate.net/publication/318852610_Virtual_Mouse

3. *Mr.Gajendra Moroliya, Mr.Sahil Patwekar, Prof.S.P. Gopnarayan "Virtual Mouse Using Hand Gesture" JETIR Journal 2018*
   https://www.jetir.org/papers/JETIR1805666.pdf

4. *Xuebai Zhang, Xiaolong Liu, Shyan-Ming Yuan, Shu-Fan Lin "Eye Tracking Based Control System for Natural Human-Computer Interaction" Hindawi Journal 2017*
   https://www.hindawi.com/journals/cin/2017/5739301/

5. *Xiaozhi Yang, Ian Krajbich "Webcam-based online eye-tracking for behavioral research" Cambridge Core Journal 2023*
   Webcam-based online eye-tracking for behavioral research | Judgment and Decision Making | Cambridge Core

6. *Tata Jagannadha Swamy, M Nandini, Nandini B, Venkata Karthika K, V Laxmi Anvitha, Ch Sunitha "Voice and Gesture based Virtual Desktop Assistant for Physically Challenged People" IEEE Xplore Journal 2022*
   https://ieeexplore.ieee.org/abstract/document/9776746?casa_token=xRWCOD94iHIAAAAA:nmn-42zqF23YiQgXfkHRMP_igYVXOo2eOUa9scfW_Z014g806E72TKMBQL-IaeGHI8xL-SDXxQ

7. *Mohamed Nasor, Mujeeb Rahman K K, Maryam Mohamed Zubair, Haya Ansari, Farida Mohamed "Eye-Controlled Mouse Cursor for Physically Disabled Individual" RearchGate Journal 2018*
   https://www.researchgate.net/publication/325919286_Eye-controlled_mouse_cursor_for_physically_disabled_individual