TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

Kathmandu Engineering College

Department of Computer Engineering

Mid-Term Report

On

COMPUTER VISION BASED VIRTUAL KEYBOARD

[Code No: CT 654]

By

Deepak Thapa	KAT078BCT029
Sudyumna Mishra	KAT078BCT083
Sujan Malakar	KAT078BCT085

Kathmandu, Nepal

2081

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to the Department of Computer Engineering, Kathmandu Engineering College for providing us such a great opportunity to undertake this project.

We would like to express our special thanks of gratitude to **Er. Sudeep Shakya** (Head of Department), **Er. Kunjan Amatya** (Deputy head of Department) of Computer Engineering Department.

We deeply express our sincere thanks to **Er. Krista Byanju** (Project Coordinator), **Er. Ritu Bajracharya, Er. Anju Khanal** for their valuable suggestions, encouragement and recommendation for developing this project. Finally, we would like to extend our thanks to all the teachers and team members who helped with their valuable suggestions and cooperation in every aspect of this project. Any comments and suggestions about the improvement are always appreciated.

ABSTRACT

Computer-vision based virtual keyboard is a virtual keyboard system that uses computer vision technology to track and interpret the movements of user's fingers or hands and allows the user to type on virtual keyboard displayed on a screen or projected surface. The project aims to create a reliable and efficient input method for situations where a physical keyboard is not available or convenient. Computer-vision based virtual keyboard is developed using Python, employing modules of OpenCV. This system uses OpenCV, a computer vision library in Python for hand detection, capturing video from a camera and preprocessing the video stream. Mediapipe is used for accurately interpreting hand movements and gestures through its pre-trained models and components. This project will use Agile Development Model. The program detects users input by opening the webcam and drawing as well as detecting hand landmarks. After that the program detects patterns made by fingertips of user and performs various tasks like Navigation, Input of Key, Volume Control and Brightness Control on the pattern made as well as features like Autocorrect and Word recommendations. With this, computer vision-based keyboard provides a flexible and adaptable input solution for various needs, ranging from accessibility and mobility to education, gaming and healthcare.

Keywords: *Computer Vision, OpenCV, Agile Development Model. Python, Mediapipe*

TABLE OF CONTENTS

ACKNOWLEDGEMENT…….ii

ABSTRACT……...……...iii

LIST OF ABBREVIATIONS…….....vi

LIST OF FIGURES……..…vii

CHAPTER 1: INTRODUCTION……..…….1

1.1 Background Theory……...……..……....1

1.2 Problem Statement……...2

1.3 Objective……..…3

1.4 Scope and Applications……...….3

CHAPTER 2: LITERATURE REVIEW…….……...4

2.1 Existing Systems…….....5

2.2 Limitations of Previous System…….…..6

2.3 Solutions Proposed by Our System...……...6

CHAPTER 3: METHODOLOGY…….……7

3.1 Process Model…….……7

3.2 System Block Diagram…….....9

3.3 Algorithm……..11

3.4 Flowchart……..12

3.5 Use Case Diagram……13

3.6 Sequence Diagram……14

3.7 Tools Used……...…….15

CHAPTER 4: EPILOGUE……16

4.1 Task Completed……....16

4.2 Task Remaining……16

4.3 Gantt Chart……16

REFERENCES…….....17

SCREENSHOTS……..18

LIST OF ABBREVATIONS

AR:	Augmented Reality
CNN:	Computer Neural Network
CV:	Computer Vision
FPS:	Frames Per Second
GUI:	Graphics User Interface
ML:	Machine Learning
OpenCV:	Open-Source Computer Vision Library
VR:	Virtual Reality

LIST OF FIGURES

Figure 1: Agile Model

Figure 2: System Block Diagram

Figure 3: Hand Landmarks defined by Google Mediapipe

Figure 4: Flowchart

Figure 5: Use Case Diagram

Figure 6: Sequence Diagram

CHAPTER 1: INTRODUCTION

1.1. BACKGROUND THEORY

In recent years, there have been many advancements in the input method using keyboard. Virtual keyboards have evolved from traditional physical keyboards due to advancements in technology. These advancements offer more flexible, efficient, versatile and intuitive input method to interact with the PC.

A virtual keyboard is a software component that allows the input of characters without the need of physical keys. Interaction with a virtual keyboard happens mostly via a touch screen interface but can also take place in a different form when in virtual or augmented reality. Virtual keyboards are commonly used on touch screen devices like smartphones and tablets. There are various types of virtual keyboards like touch screen-based keyboard in which user interact to the virtual key through touch which are commonly used in smartphones tablets and touch screen laptops. Projected keyboards like laser projection keyboard projects keyboard layout into the flat surface where user type by tapping the projected keys where sensors detect the input. Gesture based keyboard detect finger and hand movements in the air and translate them into keyboard input.

1.1.1 MACHINE LEARNING

Machine Learning is a field of study in Artificial Intelligence concerned with development and study of statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions. ML is used in various field including natural language processing, speech recognition, email filtering, health sector, etc. One of such field that ML is used is Computer Vision.

1.1.2 COMPUTER VISION

Computer Vision is a field in Artificial Intelligence that uses machine learning and neural networks to teach computer to derive meaningful information from visual inputs such as digital images, videos, etc and perform various actions on basis of those input and programs logic. It is a branch of both AI and ML where the AI can observe, recognize and understand the actions of a body based on its movement, pattern and

special characteristics. Computer Vision has been used in various fields including military, autonomous vehicles, medicine, etc. where Pattern Recognition is used to detect, observe and understand actions in real time or digital images/videos.

1.1.3 VIRTUAL KEYBOARD USING COMPUTER VISION

Computer vision based virtual keyboard is an advanced input method which utilizes camera and image processing algorithms and interpret the hand and finger movements of the user in air as keystrokes and commands. Unlike the traditional virtual keyboard which rely on touch screens or keyboard projections, this technology uses computer vision to analyze the hand movement of the user in real time. The process begins with a camera which have a clear and unobstructed view of the user's hand and continuously capture the video frames of user's hand movement creating a stream of video data. Background subtraction and skin color detection techniques isolate hand from background and edge detection help to distinguish hand outlines. Contour analysis is used to extract shape and boundaries of the fingers. The system uses features extracted from these contours to recognize gestures which are mapped to corresponding keyboard commands.

1.1.4 COMPUTER FUNCTIONALITIES USING COMPUTER VISION

There are various basic computer functionality including but not limited to Volume Control, Brightness Control, Minimization/Closing of apps. All of these functionalities require some degree of interaction with hardware components like Keyboard and Mouse. However, due to Computer Vision, all of these functionalities are performed with finger pattern recognition. The user can perform multitude of tasks with the control of finger (or any other body parts) movement and without the need of any hardware component except for a Web Camera.

1.2. PROBLEM STATEMENT

Despite the tools and framework required to tie in Computer Vision and Basic Computer Functionality like typing, navigating, etc. exist, Computer Vision related programs have only been recently catching up to mainstream market. Because of this there are not many Computer Vision related accessibility programs despite the potential for it. At the same time, CV Based Virtual Keyboard and many other CV Based

Programs cannot be found standalone but rather just integrated into hardware. Because of this, CV Based Virtual Keyboard are not available to many users who are not able to afford the hardware.

1.3. OBJECTIVE

- To make a virtual keyboard with all basic functionalities and with additional features like autocorrection and word suggestion.

- To allow user to perform various tasks like Volume Control, Brightness Control and Navigation using gesture recognition.

1.4. SCOPE AND APPLICATION

The scope and applications of Computer Vision based Virtual Keyboard are as follows:

- **Virtual Reality (VR) and Augmented Reality (AR):** Computer vision-based virtual keyboard provide a way to input texts or command without the need of physical keyboard.
- **Accessibility:** Computer vision-based virtual keyboards provide alternatives for input to people with certain physical disabilities who find difficult to use physical keyboards.
- **Gaming:** Computer vision-based virtual keyboard can enhance gaming experiences by providing more immersive way to control characters.
- **Wearable Devices:** Computer vision-based virtual keyboard can allow text input and control on wearable devices like smart glasses, smart watches and different medical wearables.

CHAPTER 2: LITERATURE REVIEW

Virtual Keyboards have emerged as promising solution to typing texts without a need of physical keyboard. It has grown immensely popular with almost all smartphones using virtual keyboards and various apps being built for Windows/Linux. By applying Computer Vision Algorithms, these keyboards allow users to input text by detecting and interpreting hand gestures/movements captured by camera. This literature review provides a comprehensive overview of the research and developments in the field of computer vision-based virtual keyboards, highlighting key methodologies, applications, challenges, and future directions.

Google developed Mediapipe as a opensource for reading and deploying body patterns for machine learning. Mediapipe allowed programmers to create perception pipelines called Graphs. This allowed feed a stream of images to be taken as input which comes out with hand landmarks rendered on the images. Since 2012, Google started using Mediapipe internally in several products and services. It was initially developed for real-time analysis of video and audio on YouTube. Gradually it got integrated into many more products and is a standard library for almost all Computer Vision based programs today.

In 2006, Nintendo Game Company created console called Wii which allowed programs to read input based on players body movement and create augmented reality based on them on-screen. This feature came back with much needed improvements in Nintendo Switch released at 2017, which allowed video-games to detect hand movement without hindering the fps of games.

Some individual research regarding virtual keyboard using computer vision include:

Avirmed Enkhbat from National Central University worked on computer vision and image understanding has paved the way for advancements in virtual keyboard technology. Citation: Avirmed Enkhbat "Efficient recognition using virtual keyboards." 2020 5th International Conference on Information Technology (InCIT) [1].

Dr. Fei-Fei Li was a Professor of Computer Science at Stanford University and co-director of the Stanford Vision and Learning Lab. Her research focuses on computer

vision, machine learning, and cognitive neuroscience, all of which are fundamental to virtual keyboard development[2].

Y. Zhou, G. Jiang, Y. Lin made a novel finger and hand pose estimation technique for real-time hand gesture recognition, pattern Recognition contributing in field of Computer Vision and CNN[3].

Shumin Zhai has conducted research in HCI, particularly focusing on text input methods and interaction techniques. His work has influenced the design and evaluation of virtual keyboards and other input modalities[4].

Md. Atiqur Rahman Ahad, T. Jie and H. Kim, S. Ishikawa researched in variants and applications of Machine Vision and Applications paving road to various CNN projects[5].

These researchers have made valuable contributions to the advancement of computer vision-based virtual keyboards through their expertise in areas such as gesture recognition, human pose estimation, and human-computer interaction. Their research has paved the way for innovative approaches to text input using computer vision techniques.

2.1. EXISTING SYSTEMS

There are some existing systems which have aimed to change the way of interaction between user and device by changing the need of physical keyboard. Some systems of virtual keyboards include:

- New AI based hardware models like Apple Vision Pro have been using CV based keyboards in their products.
- Wearable device like AirType monitors finger movements in air, effectively replacing physical keyboards.
- Microsoft Holo Lens integrates virtual keyboard which enables typing through hand gestures.

2.2. LIMITATIONS OF EXISTING SYSTEMS

1. Computer vision systems rely on accurate recognition of hand gestures or finger movements to interpret user input. However, they may encounter difficulties in accurately detecting and interpreting gestures, especially in noisy or complex environments.

2. Many computer vision-based virtual keyboards have a limited set of gestures or symbols that users can input, which may restrict their applicability for tasks requiring extensive text input.

3. User friendly functionalities like Auto correction, Word Suggestion and shortcuts are lacking on existing models**.**

4. Real-time processing of hand movements for accurate gesture recognition requires efficient algorithms and hardware capabilities to minimize latency and ensure responsive interaction.

2.3. SOLUTIONS PROPOSED BY OUR SYSTEM

1. When camera detects hand then it will draw a landmark over the hand indicating that data is being read providing better communication between application and user

2. Hand samples of different sizes, skin colors and general shape will be used to train the program for better detection. Additionally, our system is accurate due to background subtraction and proper calibration

3. Along with keyboard various other functionality like Volume control, Brightness control, Word Suggestion and Auto Correction will be implemented.

4. Computer Vision based Keyboard can be taxing for user as they will have to make various patterns with their fingers. To mitigate this issue, simple and non-stressful patterns will be chosen to ensure user comfort.

CHAPTER 3: METHODOLOGY

3.1 PROCESS MODEL

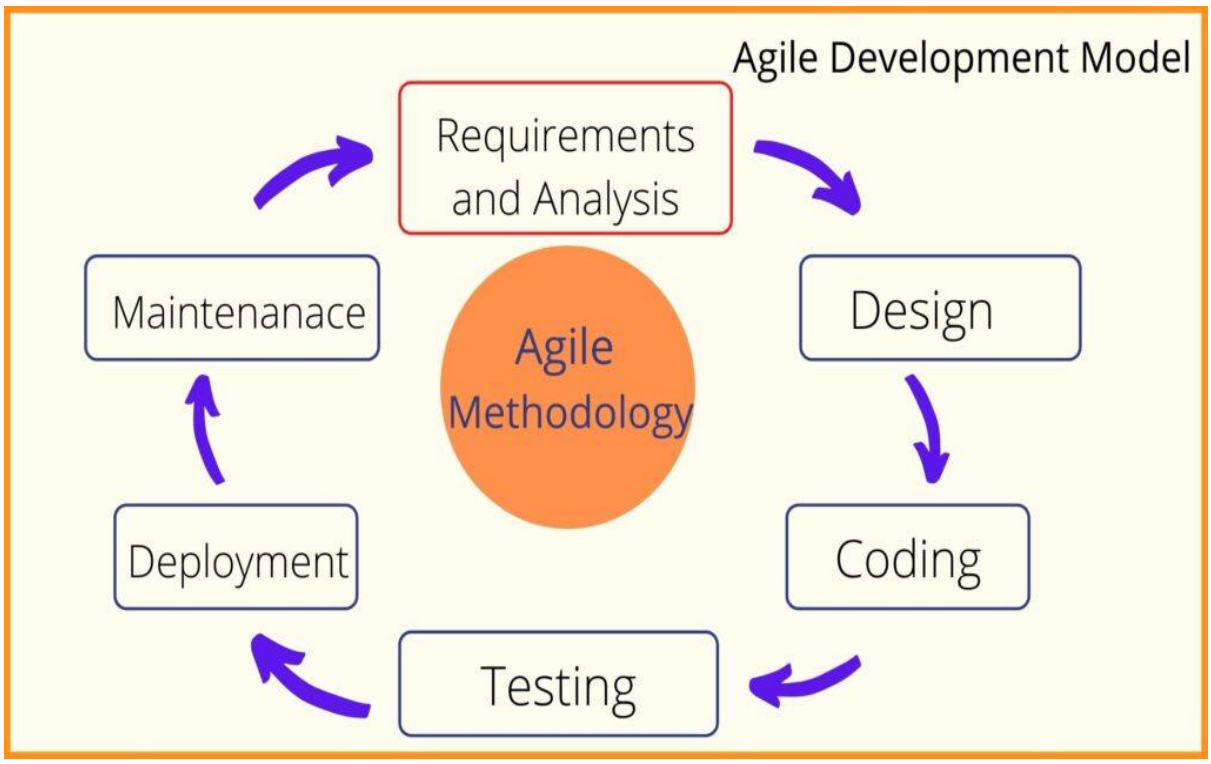

Figure 1: Agile Model

Agile model is a dynamic and iterative approach that focuses on delivering functional software in short development cycles known as sprints. It places a strong emphasis on collaboration, adaptability and ongoing improvement throughout the entire development process.

Agile Process Model is used because with this model work can be divided into groups easily. Additionally, with agile model it is easier to add more functionalities more easily than compared to other models.

Different phases of agile model are:

1. **Requirement analysis:** In this phase, we identify our project requirements and carefully analyze them. A detailed plan is then formulated for the development process, which includes defining the project's scope, goals and timelines.

2. **System Design:** We begin the system design phase by defining the high-level architecture of the system. This includes identifying main components, their interactions and control within the system.
3. **Development:** In this phase, coding and implementation of software takes place based on the design specifications and requirements. This phase includes writing and integrating code modules, implementing planned features and ensuring the software meets the desired functionality.
4. **Testing:** In this phase, examination of functionality, performance and quality of the software are done thoroughly. The goal of testing is to identify and resolve any defects or issues, solve them and ensure the software meet the requirements.
5. **Deployment:** After comprehensive testing of software, it is deployed to the production environment. This phase involves tasks such as configuring servers and installing the software.
6. **Review and Feedback:** Following the deployment, a review and feedback phase takes place. This involves gathering feedback from end-users, conduction of post-deployment assessments and addressing any concerns. Purpose of this phase is to evaluate software's performance, gather insights for future improvements and ensure its effectiveness in meeting user needs.

3.2 SYSTEM BLOCK DIAGRAM

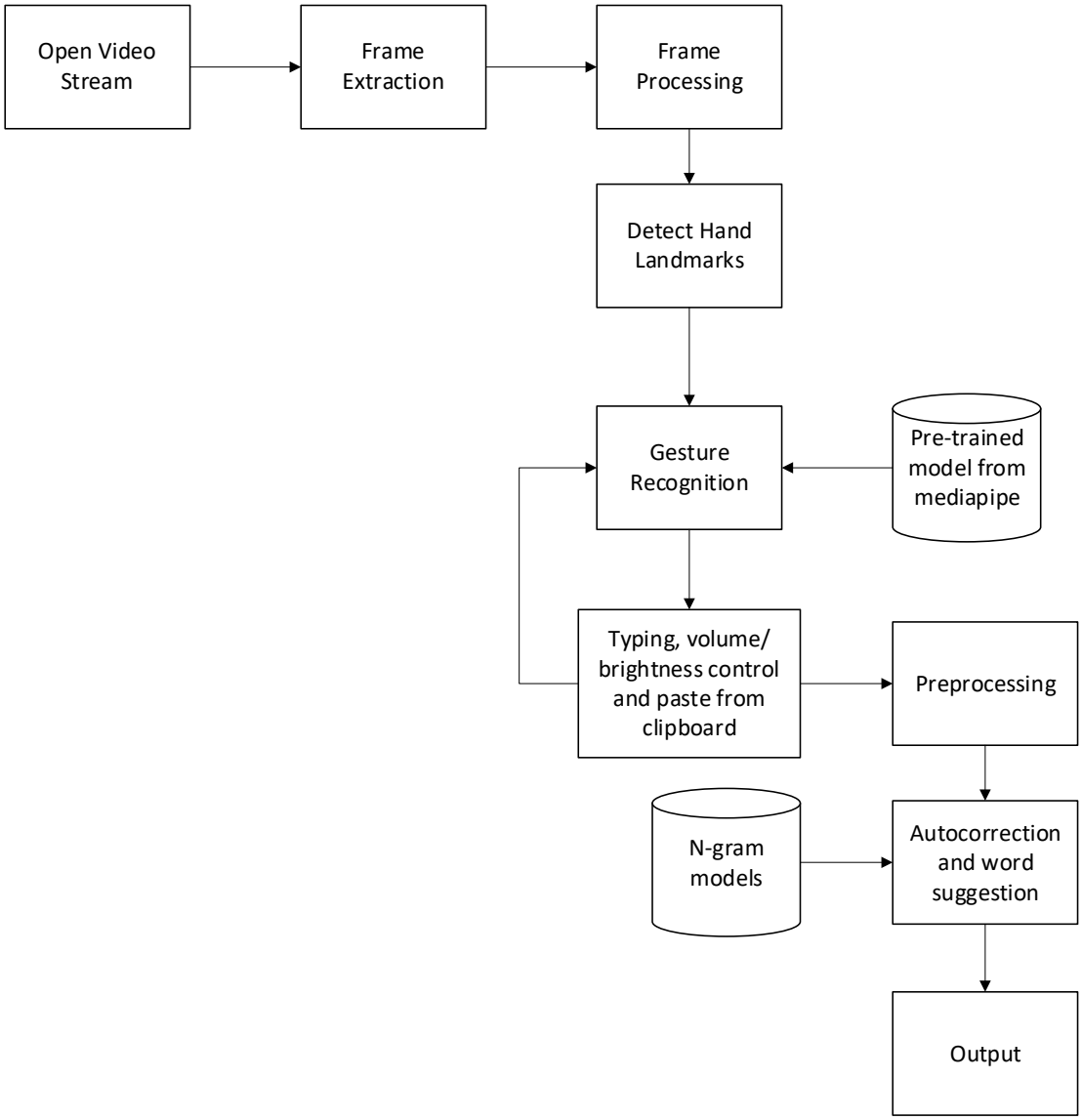

Figure 2: System Block Diagram

Figure 2 showcases the System Block Diagram of Computer Vision Based Virtual Keyboard. The gesture recognition process first involves acquisition of the video feed. Each frame extracted from this stream undergoes detailed analysis to extract relevant features crucial for identifying gestures. This analysis includes leveraging a pre-trained MediaPipe model specialized for detection of hand landmarks within each frame. These landmarks serve as the foundation for recognition of gesture.

After the system identifies the hand landmarks, it proceeds to interpret the specific gesture to be performed. This involves movement of landmarks to predefined gesture

patterns enabling system to distinguish different gestures such as index finger up, index and middle finger up and others. Based on these gestures, system performs corresponding action like typing on keyboard, pasting text from clipboard and control of volume and brightness. This process is repeated continuously.

For the autocorrection and word suggestion features, the input data is transformed to suitable format for AI model through tokenization, lowercasing and removing of punctuations and stop words. Then, by n-gram model, the probability of word on the previous n-1 words in sequence is calculated. The model compares the current input with n-gram patterns and adjust the most likely word or correction based on historical data.

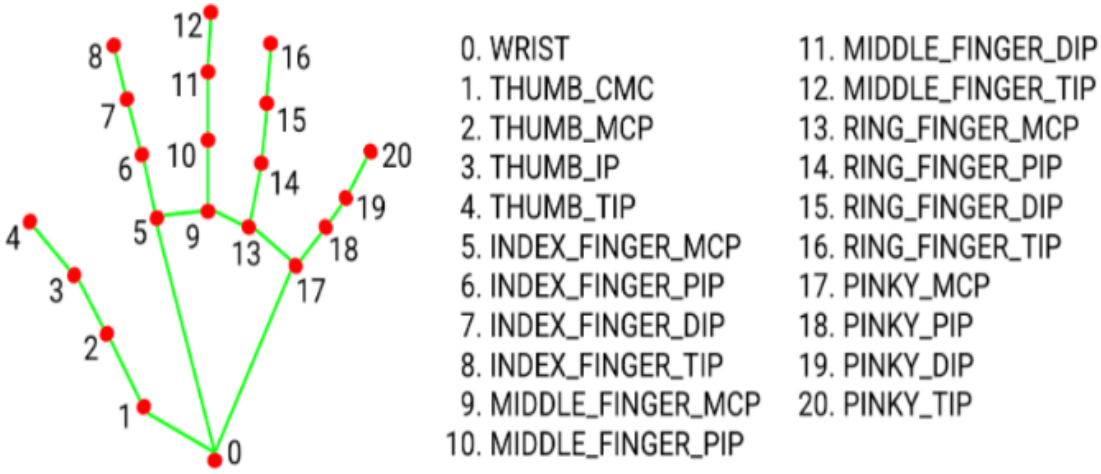

Figure 3: Hand Landmarks defined by Google Mediapipe

Hand landmarks detected by mediapipe is shown in figure above. The 21 knuckle points are marked on user's hand and different gestures are recognized on the basis of fingers raised. For this, y-coordinates of each fingertips are compared and checked whether it is lower than its adjacent landmarks. Through this, gestures are recognized for typing, navigation and pasting text from clipboard. Also, distance between two fingertips is calculated and on the basis of the calculated distance, volume and brightness are increased or decreased.

3.3 ALGORITHM

1. Start
2. Open video-stream
3. If virtual keyboard displayed
    a. Extract frames
    b. Detect hand and fingertips
    c. Draw hand landmarks
    d. If index, middle and ring finger up
        - Paste from clipboard
    e. If index finger up
        - Navigation
    f. If index and middle finger locate area of character
        i. Output of character
        ii. If error is detected
            - Autocorrection
4. Else
    a. Extract frames
    b. Detect hand and fingertips
    c. Draw hand landmarks
    d. If index and thumb moved closer
        - Decrease volume
    e. If index and thumb moved farther
        - Increase volume
    f. If index and pinky finger moved closer
        - Decrease brightness
    g. If index and pinky finger moved farther
        - Increase brightness
5. End

3.4 FLOWCHART

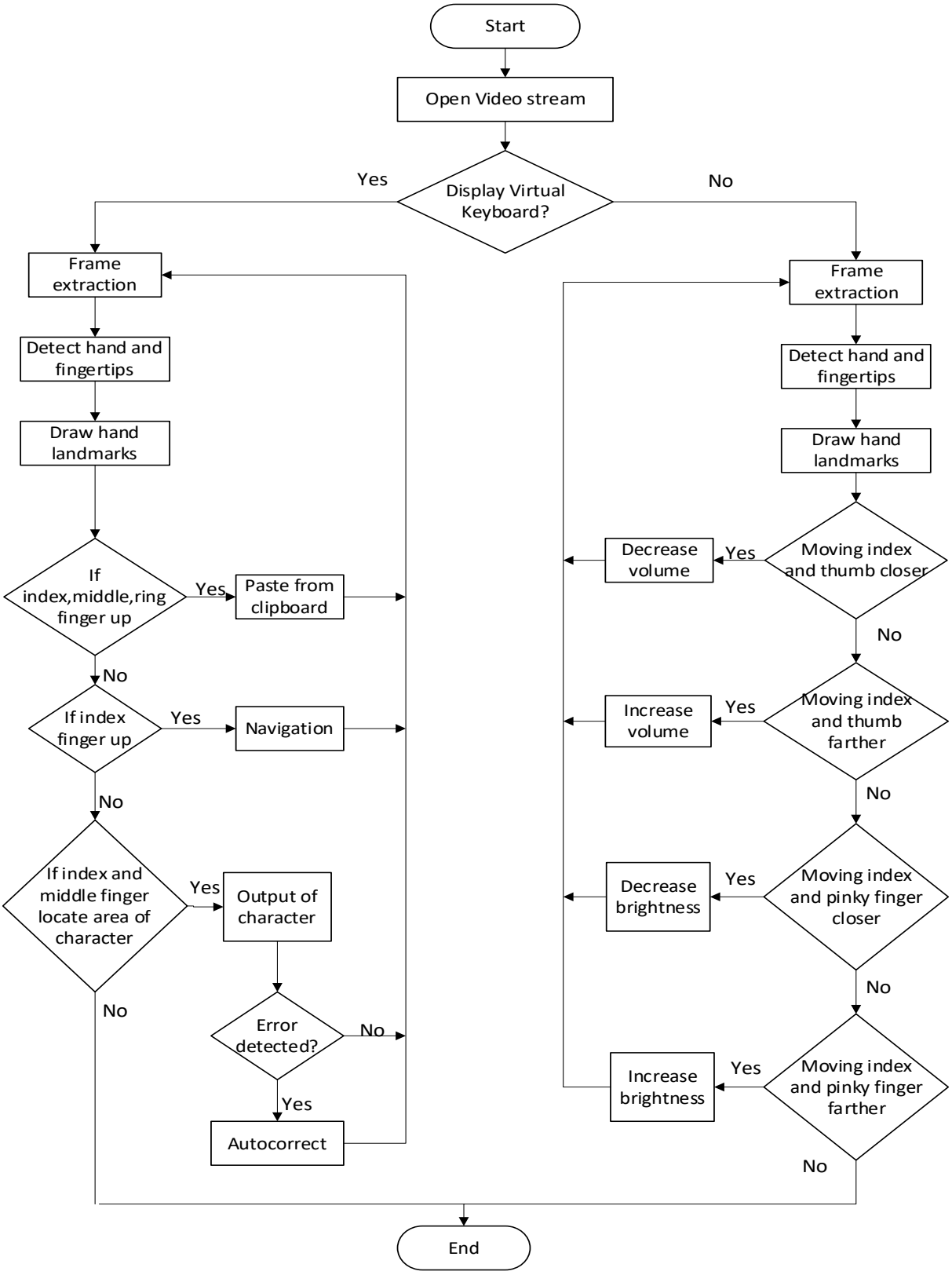

Figure 4: Flowchart

3.5 USE CASE DIAGRAM

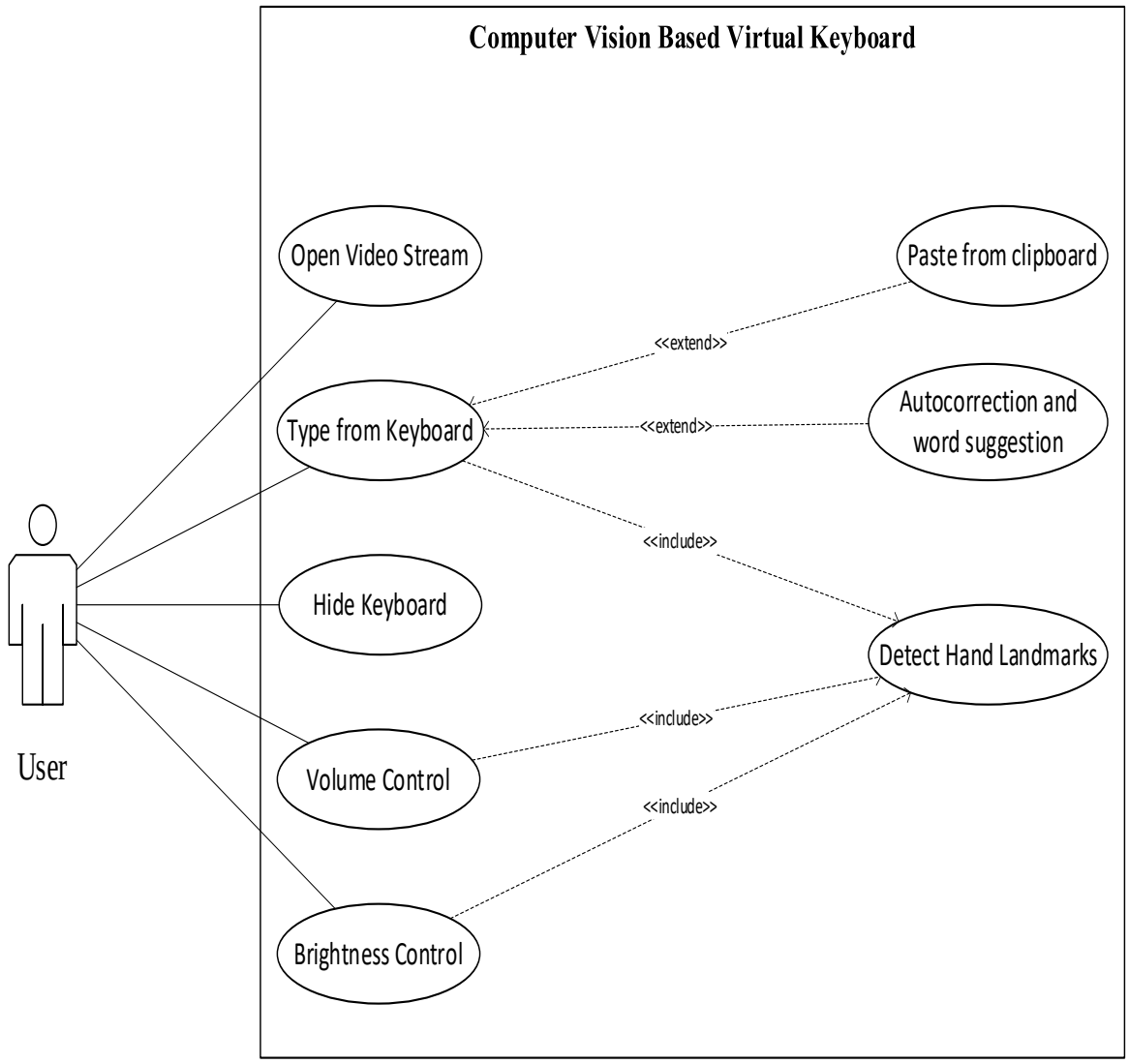

Figure 5: Use Case Diagram

This is the use case diagram for "Computer Vision Based Virtual Keyboard". The diagram shows the interaction of user with the application. User can perform several actions which include opening video stream, typing in the virtual keyboard, hiding the keyboard and volume control and brightness control. Typing and volume and brightness control includes detection of hand landmarks. User can also paste text from clipboard and use autocorrection and word suggestion feature.

3.6 SEQUENCE DIAGRAM

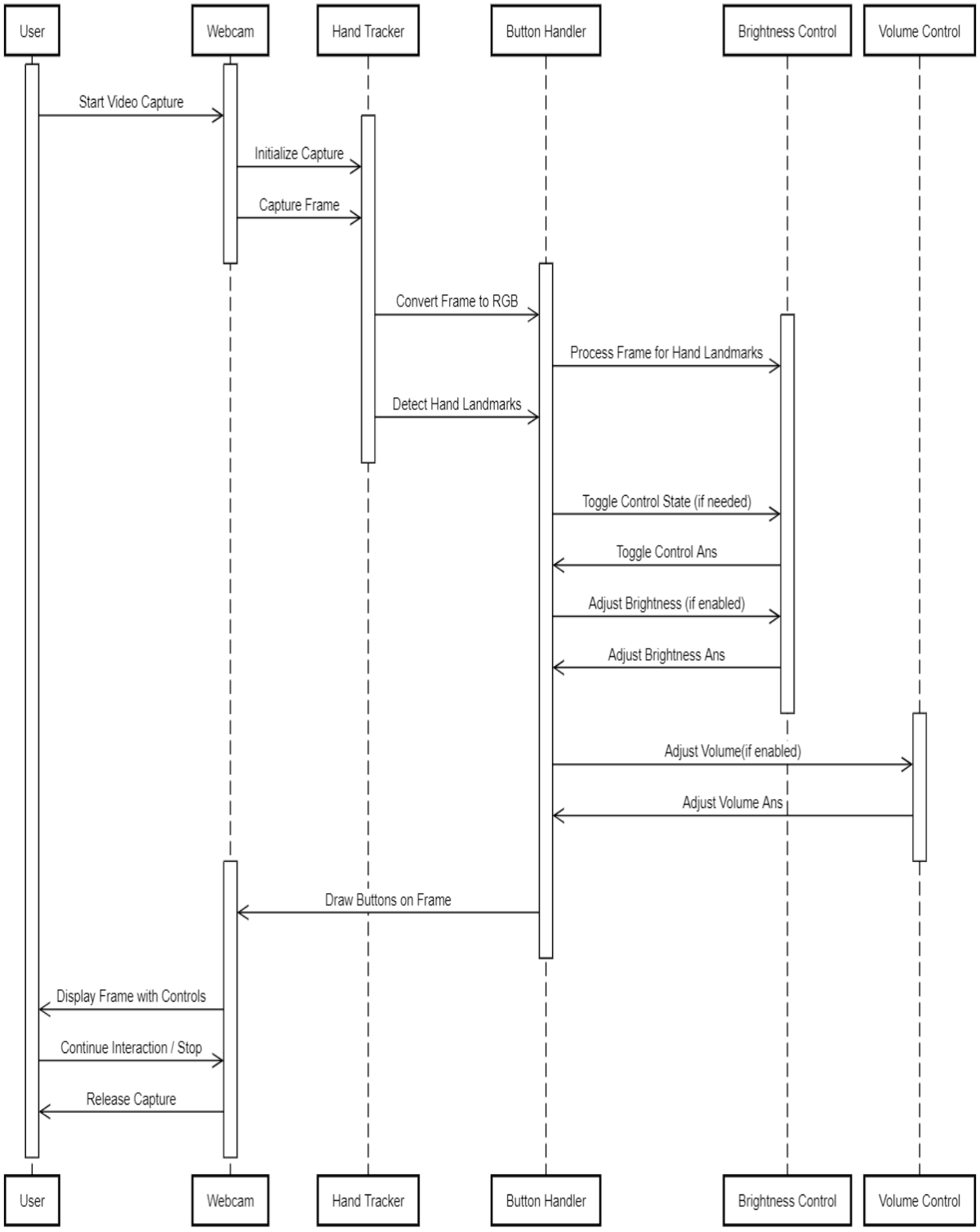

Figure 6: Sequence Diagram

3.7 TOOLS USED

Various system tools used to develop the front end and back end of the project are being discussed in this chapter:

1. **OpenCV:** OpenCV (Open-Source Computer Vision Library) is a free cross-platform computer vision library for real time image processing. It provides an easy-to-use computer vision and comprehensive image processing capabilities like image stitching, video stream processing, camera calibration and diverse image pre-processing tasks.
2. **Mediapipe:** Mediapipe is an open source and flexible framework for building multimodal ML pipelines that allow developers to create complex processing graphs for audio, image and sensor data. It provides a set of pre-built components called “Graphs” that are easily combined to create end-to-end ML pipelines.
3. **Python:** Python is a computer programming language often used to build websites and software, automate tasks and analyze data. It supports multiple programming paradigms, including object-oriented and functional programming.
4. **Pycharm:** Pycharm is an IDE used for programming in Python. It offers different features like code analysis, graphical debugger, an integrated unit tester and integration with version control systems.
5. **Pycaw:** Python Core Audio Windows Library is a python library used for system volume control including mute, change volume, max volume, etc.
6. **Python NLTK:** Python NLTK is a python library used for Natural Language Processing. It is used to access strings provided by user and manipulate them based on programmers use.
7. **Pyperclip:** Pyperclip is a cross-platform Python module for copy and paste clipboard functions. It is used to access user’s clipboard.
8. **NumPy:** NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

CHAPTER 4: EPILOGUE

4.1 TASK COMPLETED

- Implementation of hand tracking technology.
- UI of keyboard.
- Detection of fingertips for volume control.
- Detection of fingertips for brightness control.

4.2 TASK REMAINING

- Addition of a button to hide the keyboard.
- Real Time Visual feedback of key press
- Addition of copy/paste feature.
- Word suggestion and autocorrection.

4.3 GANTT CHART

S.N	Tasks	Fifth Semester						Sixth Semester				
		MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR
1	Documentation											
2	Discussion											
3	Study and Research											
4	Requirement Analysis											
5	System Design											
6	Coding											
7	Testing											
8	Debugging											

REFERENCES

[1] Avirmed Enkhbat "Efficient recognition using virtual keyboards." 2020 5th International Conference on Information Technology (InCIT).

[2] Fei Fei Li,:CVPR 2010: IEEE Conference on Computer Vision and Pattern Recognition". Retrieved December 27, 2018.

[3] Y. Zhou, G. Jiang, Y. Lin, A novel finger and hand pose estimation technique for real-time hand gesture recognition, Pattern Recognition, January 2016 .

[4] Shumin Zhai: Optimised Virtual Keyboards with and without Alphabetical Ordering - A Novice User Study

[5] Md. Atiqur Rahman Ahad, T. Jie, H. Kim, S. Ishika wa, Motion history image: Its variants and applications, Machine Vision and Applications, 2012, pp. 255–281

SCREENSHOTS