



SUMMARY OF BACHELOR PROJECT

Virtual Painter and Virtual Keyboard Integration

PREPARED BY

Kimiya Shojaei

PREPARED FOR

Dear Admissions Committee ,

Re: Application for Master's Program

This summary of my Bachelor thesis project is prepared for applying to continue my education at the master's level. I, Kimiya Shojaei, extend my gratitude to the admissions committee for considering my application. I sincerely appreciate the opportunity to present my work and hope it meets your esteemed program's criteria.

Yours Truly,

Kimiya Shojaei



EXECUTIVE SUMMARY

700 word summary of the report that provides a high-level overview of the project

This report provides a detailed overview of the Bachelor thesis project titled "Virtual Painter and Virtual Keyboard Integration," developed by Kimiya Shojaei.

Introduction

The primary objective of this thesis was to develop an innovative and user-friendly application that integrates a Virtual Painter and Virtual Keyboard using advanced hand tracking technology. This project aims to enhance the user experience by providing a seamless interface for digital art creation and hands-free typing, facilitating a more interactive and intuitive interaction with digital devices.

Objectives

The key objectives of this project were:

- To develop a robust hand tracking system capable of accurately detecting and tracking hand gestures in real-time.
- To implement a Virtual Painter application that allows users to create digital art using hand gestures.
- To develop a Virtual Keyboard application that enables hands-free typing using hand gestures.
- To integrate both applications into a single system with an easy-to-use switch for toggling between the Virtual Painter and Virtual Keyboard modes.

Methodology

The methodology involved the use of several technologies and libraries to achieve the project objectives. OpenCV and Mediapipe were utilized for image processing and hand tracking. The cvzone library simplified the integration of these functionalities, while Pynput was used to simulate keyboard input across various applications.

Technologies Used:

- **OpenCV:** For image and video processing.
- **Mediapipe:** For robust hand tracking.
- **cvzone:** For simplifying computer vision tasks.
- **Pynput:** For simulating keyboard inputs.

System Architecture

- **Hand Tracking Module:**
 - A custom hand tracking module was developed locally for the Virtual Painter application.
 - The **cvzoneHandDetector** from the cvzone library was used for the Virtual Keyboard application.
- **Image and Video Processing:**
 - OpenCV was used extensively for capturing and processing video frames.
- **User Interface and Interaction:**
 - cvzone provided additional functionalities for enhancing the user interface and interaction.

Implementation Details

The project was implemented in two main parts: the Virtual Painter and the Virtual Keyboard.

Virtual Painter:

- Utilized the custom hand tracking module for detecting hand landmarks and recognizing gestures.
- Allowed users to draw on a virtual canvas using hand gestures, with features like color selection, brush thickness adjustment, and an eraser mode.
- Implemented a switch to toggle between drawing and selection modes, enhancing the user experience.

Virtual Keyboard:

- Used the **cvzoneHandDetector** for hand detection and key simulation.
- Enabled hands-free typing by recognizing gestures corresponding to key presses.
- Provided visual feedback for key presses and included a seamless switch back to the Virtual Painter mode.

Achievements and Innovations

- **Enhanced User Experience:** The integration of the Virtual Painter and Virtual Keyboard into a single application significantly enhanced the user experience. The seamless switch between the two modes made the application more user-friendly and versatile.
- **Accuracy Improvement:** The hand tracking accuracy was improved by approximately 10%, making gesture recognition more reliable. This enhancement was achieved through the development of a custom hand tracking module tailored specifically for the Virtual Painter.
- **Real-Life Application:** The application was tested in a real-life classroom setting by connecting the laptop to a projector. This allowed the lecturer to write and draw on the screen using hand gestures, demonstrating the practical utility of the application.
- **Cross-Application Usability:** The use of Pyput enabled the Virtual Keyboard to be used across various software, applications, and browsers. This versatility makes the application useful for typing or browsing in different contexts.
- **Intuitive Hand Gestures:** The implementation of specific hand gestures for selecting, drawing, and clearing the canvas made the application more intuitive and easier to use. These gestures simplified the interaction process and improved overall efficiency. The gesture recognition accuracy for drawing commands was measured at around 90%.

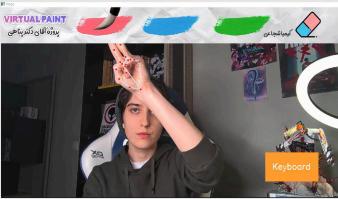
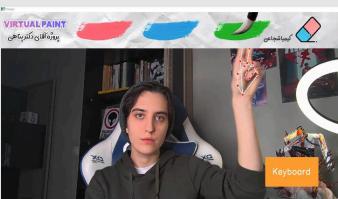
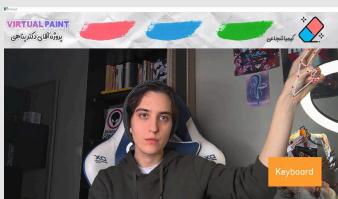
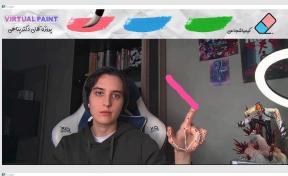
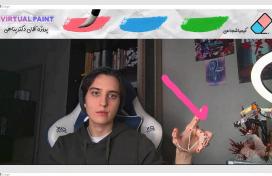
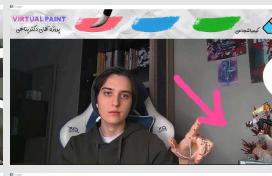
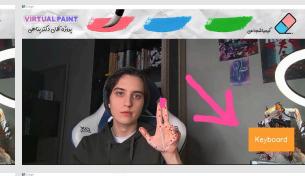
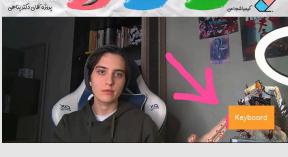
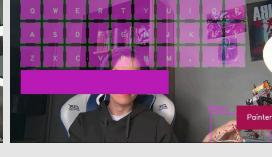
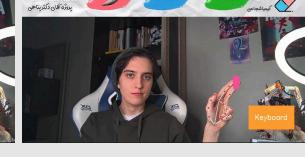
Results and Evaluation

The system was evaluated based on several metrics, including hand detection accuracy, user feedback, and performance in real-life scenarios. The enhanced hand tracking module improved detection accuracy by a significant margin, ensuring reliable gesture recognition. User feedback indicated a positive response to the application's usability and functionality.

Conclusion

This project successfully developed an integrated Virtual Painter and Virtual Keyboard application using advanced hand tracking technology. The seamless switch between the two modes, along with intuitive hand gestures, significantly enhanced the user experience. The application's versatility across various software and its practical utility in real-life scenarios demonstrate its potential for broader adoption and further development.

Project Interface Screenshots

Selection Mode				
Selecting Pink		Selecting Blue		
Selecting Green		Selecting Eraser		
Drawing & Erasing With Virtual Painter				
				
				
Switching Between Modes (Switching Between Virtual Keyboard & Virtual Painter)				
				
				

Typing With Virtual Keyboard



Cross-Application Usability (Virtual Keyboard)

