

Gesture Based Computer Control

Pursuing Versatile and Customizable Gesture Recognition-Based Computer Control Using Open Source, Inexpensive Computer Vision Technology

Glen Warren, Dearborn High School, MI, USA

SOFT036

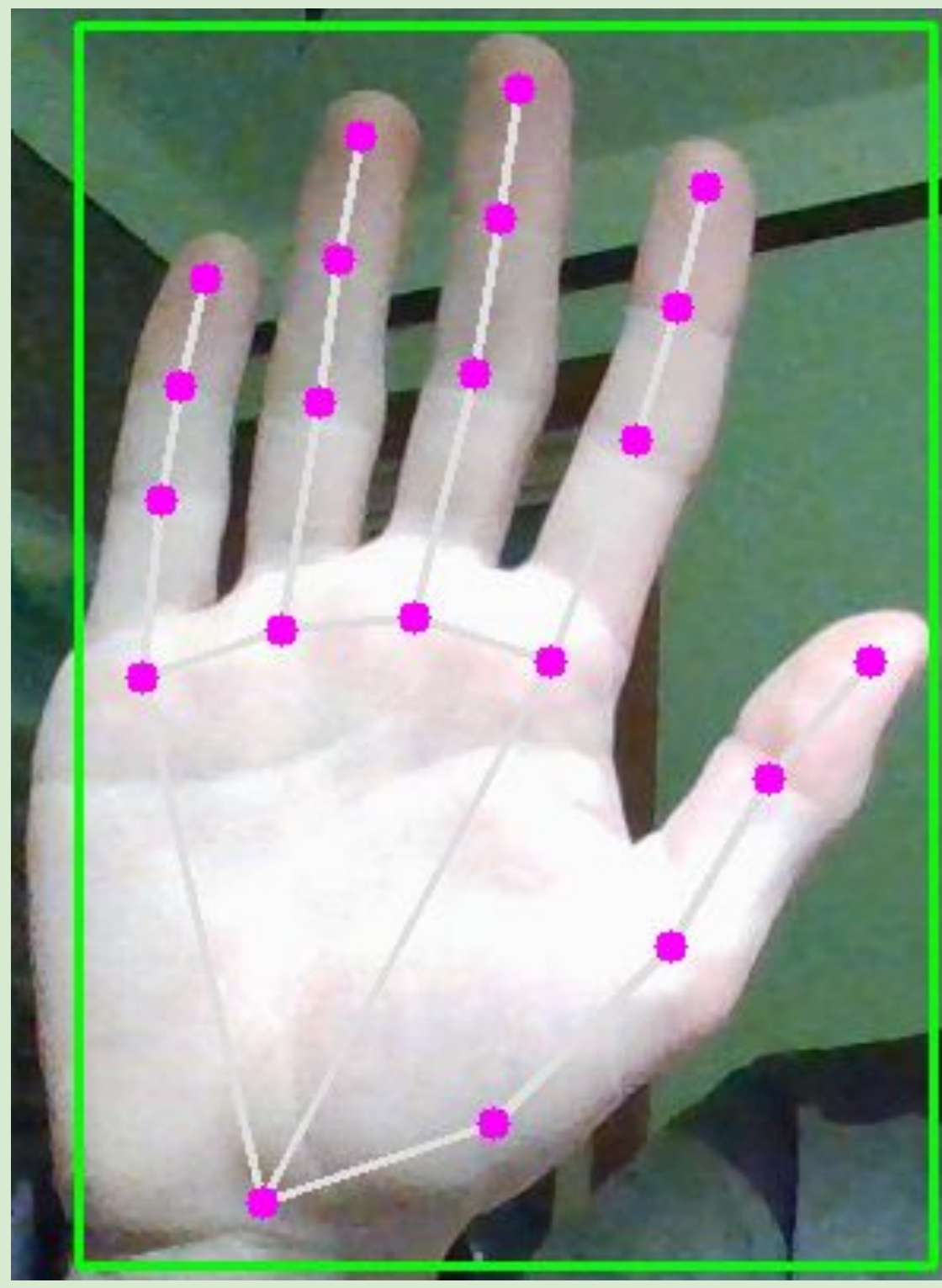
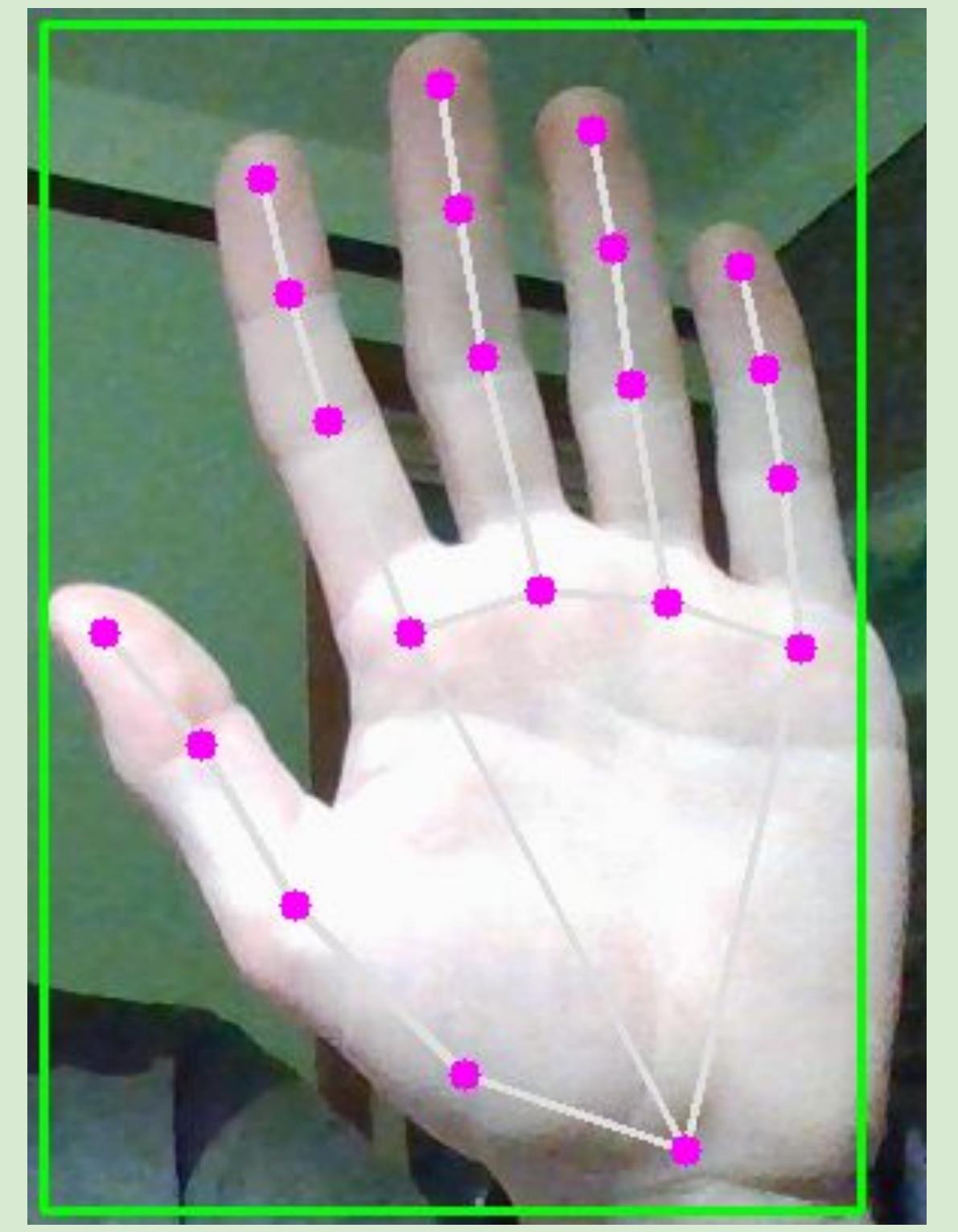


Fig. 1 | System image shows predicted hand landmarks, highlighted, with skeletal prediction.



Introduction

As technology becomes more intertwined with our daily lives, convenient, accurate, and accessible control of that technology is important for sustainability and technological equity. Improving human computer interaction (HCI) is crucial for improving technology's use and integration in our society.

Computer Interaction is Critical. Gesture control can improve:

Accessibility

- Equity in accessibility for those who cannot use traditional control

Convenience

- Customization improves Human Computer Interaction

Sustainability

- Freedom of movement reduces risk for repetitive stress injuries

Engineering Goal

Create a program for natural computer control through multiple integrated control options and customizability

Goals and sub goals:

- Computer control
 - Sufficient Speed and Accuracy
 - Mouse Control
 - Volume and Brightness Slider Control
 - Keybind control
 - Other features
- Customization
 - Graphical User interface
 - Intuitive customization
- Resources
 - Consumer Grade Laptop
 - Built in webcam
 - Open Source Computer Vision Software

Designs & Prototypes

Prototypes were developed over time to reach the final result. Key developments are noted below:

- Relative mouse movement
 - Mouse movement based on frame by frame hand position detection
 - Improves mouse control versatility
 - Screen edge control improvement
- Relative landmark distance comparison
 - Recognizes gesture positions by comparing distances between landmarks
 - Improved performance for any distance or hand orientation, reversed or turned
- Custom Graphical User Interface
 - Designed for system customization
 - Customizes gestures used for control
 - Adjusts speed settings for user
 - Open to future expansion

Result System

The resulting prototype for computer control was capable of real time computer control of mouse, volume, brightness, and other features.

The program recognized hand positions via landmarks and performed computer control actions in real time.

The system successfully recognized gestures based on predicted landmarks. However, due to use of rigid test cases and sometimes inaccurate landmark predictions, some accidental recognition occurred. However, this was rare under normal conditions.

Users also experienced difficulty in performing gestures according to rigid test cases, though they adapted to performing gestures over time. Accidental command repetition challenged user interaction, but allowing customization of time delay between control actions improved usability.

Background

Prior research investigates a variety of methods of gesture recognition, but little research investigates current applications of gesture technology.

Primary methods of HCI include physical mediums of interaction and computer vision techniques. [1,2] Computer vision technologies have shown increasing promise due to the rise of machine learning. Paper [1] also addresses that hands are the most explored and promising method for computer vision HCI.

Past studies have addressed the success of different methods of gesture recognition by utilizing different visual qualities and algorithms for analysis. Studies such as [3-6] present various techniques for recognizing hand gestures, but do not explore potential current day applications for such techniques.

[7] presents a simple mouse control at distance via webcam. This presents a current potential use of gesture recognition, but lacks utility for more complex interaction.

Papers [8] and [9] present use of OpenCV, a modern computer vision library which supports hand recognition. These resources consider more current methods of computer control, but lack extensive exploration of a versatile computer control system.

Method

- Iterative design process
- Rapid Prototyping Technique

To effectively address the research, the result system will be evaluated in areas of:

- Gesture recognition
- Computer control
- Customization of available control functions
- Improvement of natural interaction with computer control.

OpenCV provides real time landmark prediction identification, allowing rapid prototyping.

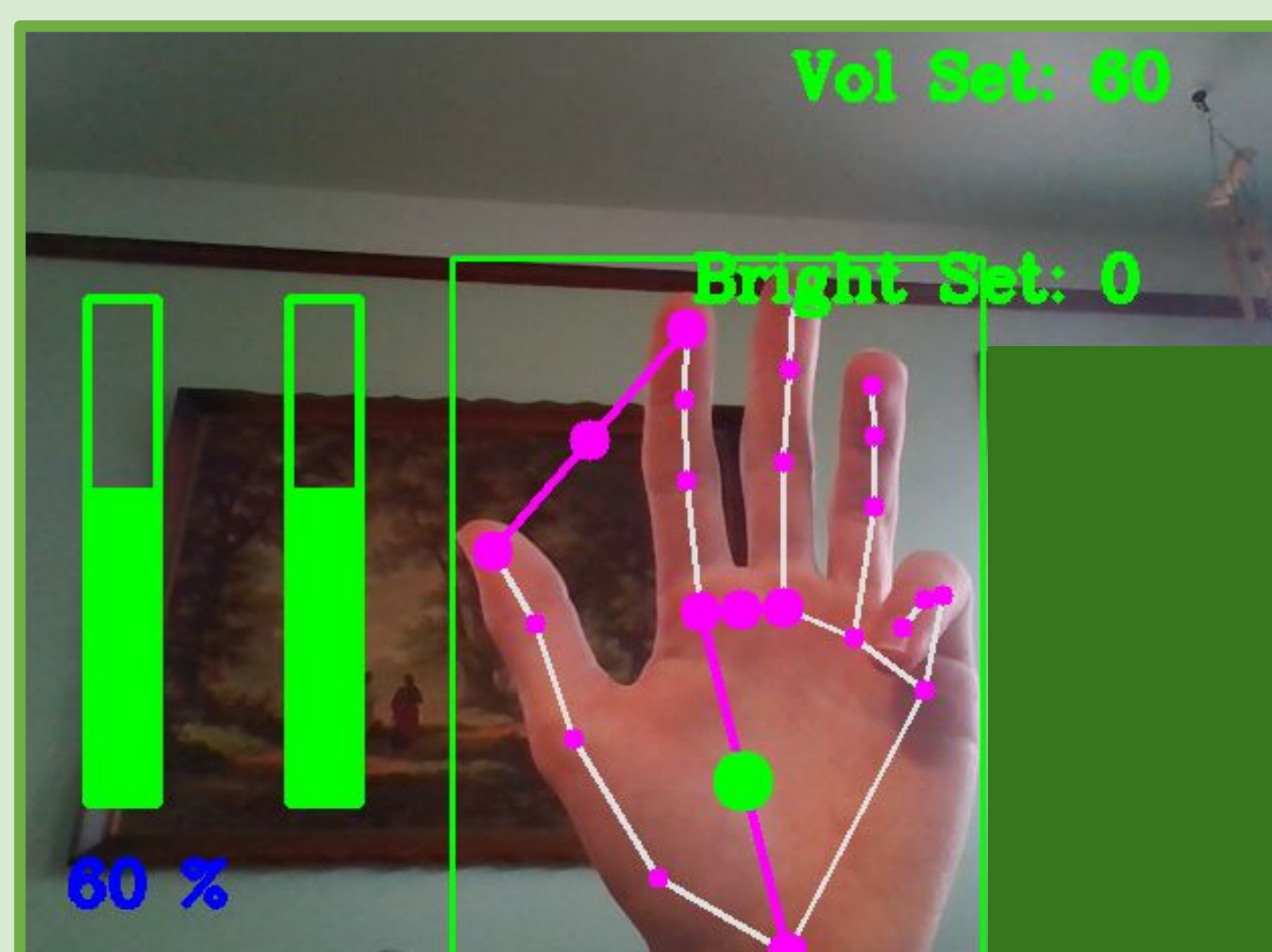


Fig. 3 | Volume and Brightness can be adjusted by a distance slider control and set with a gesture

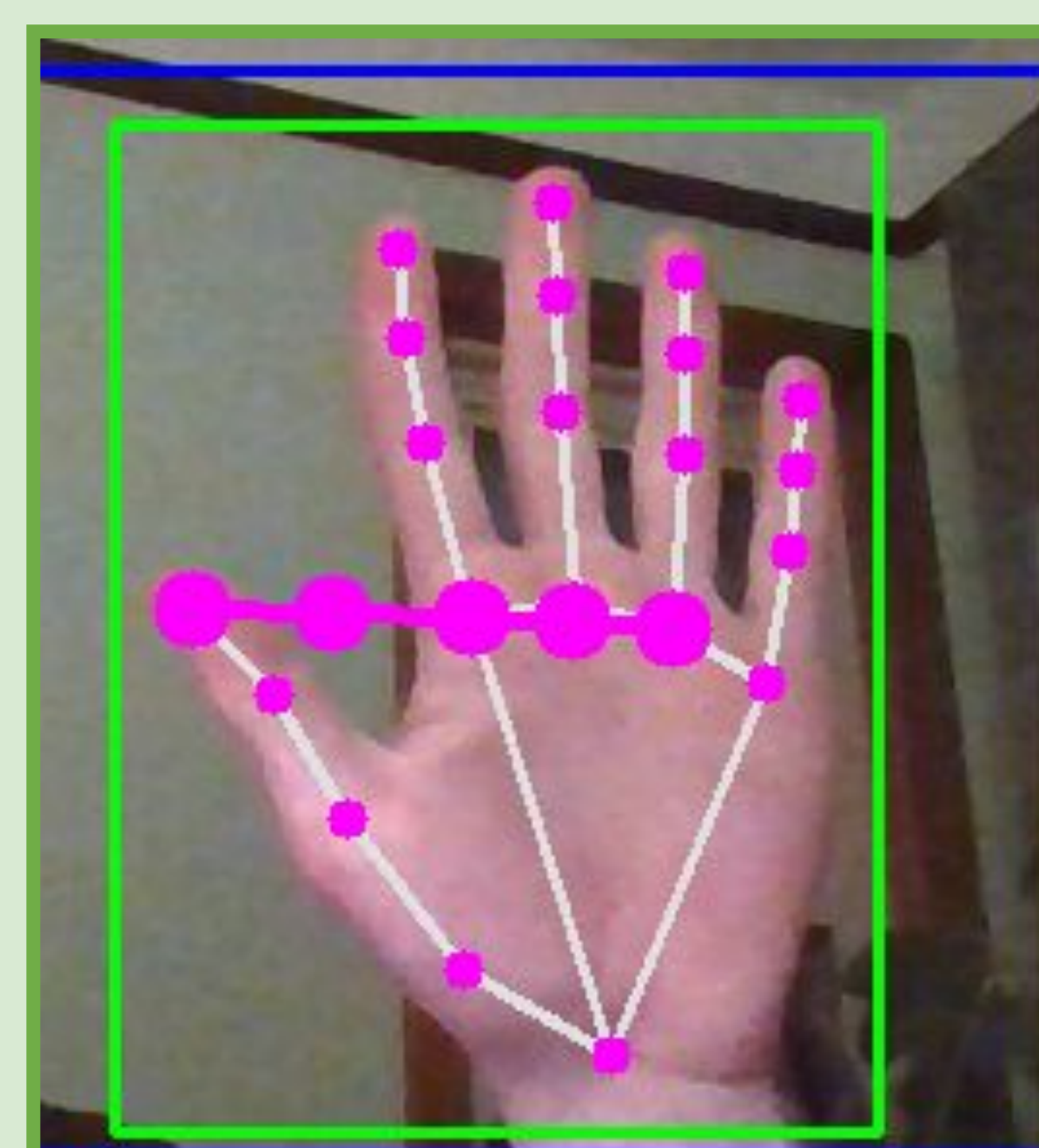


Fig. 2 | Mouse Control operates via relative hand movement frame by frame, clicking action is controlled by distance between thumb and knuckle

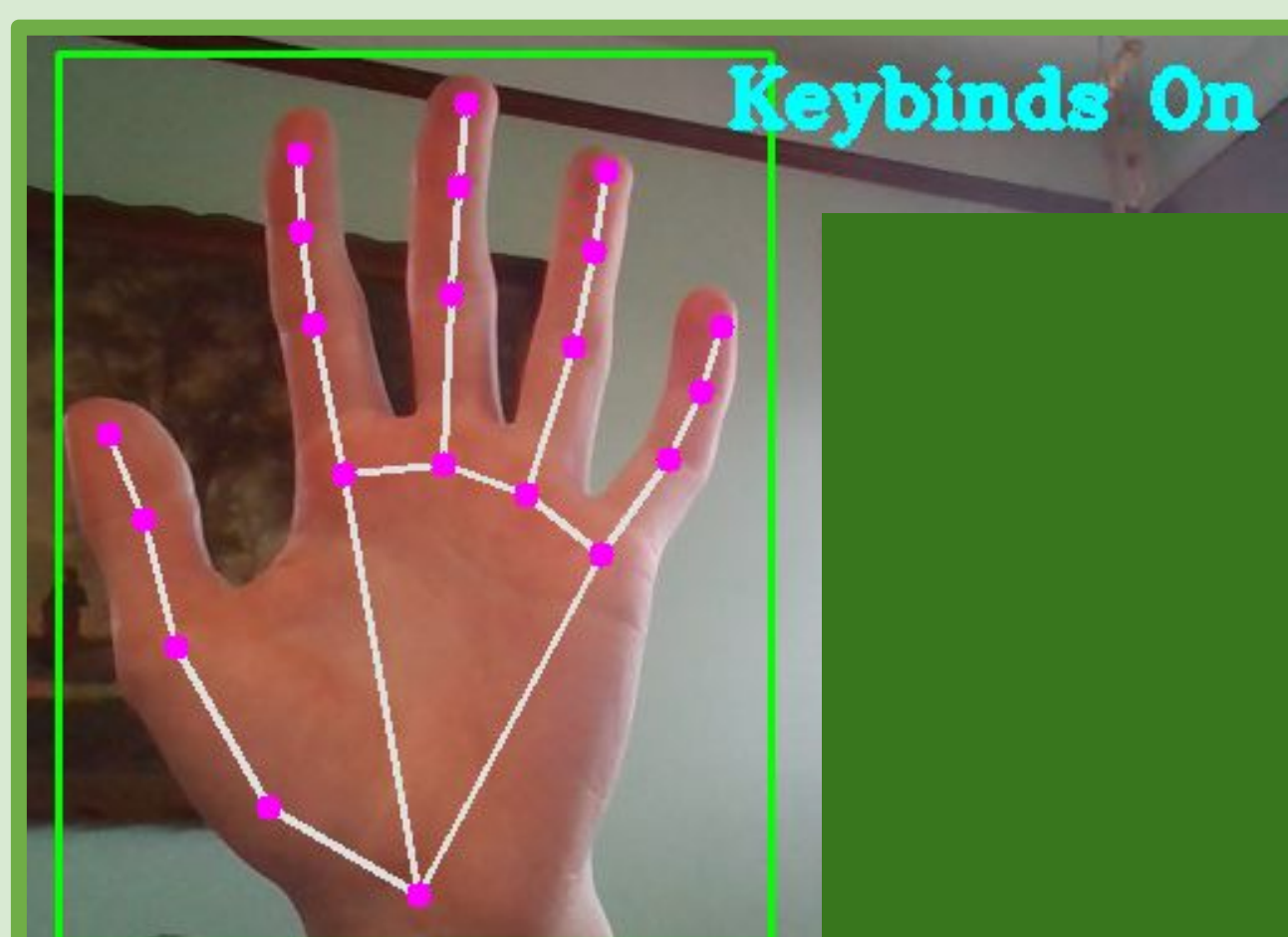


Fig. 4 | Additional features are available through "keybinds" toggle, which allows miscellaneous features to be accessed via gesture control

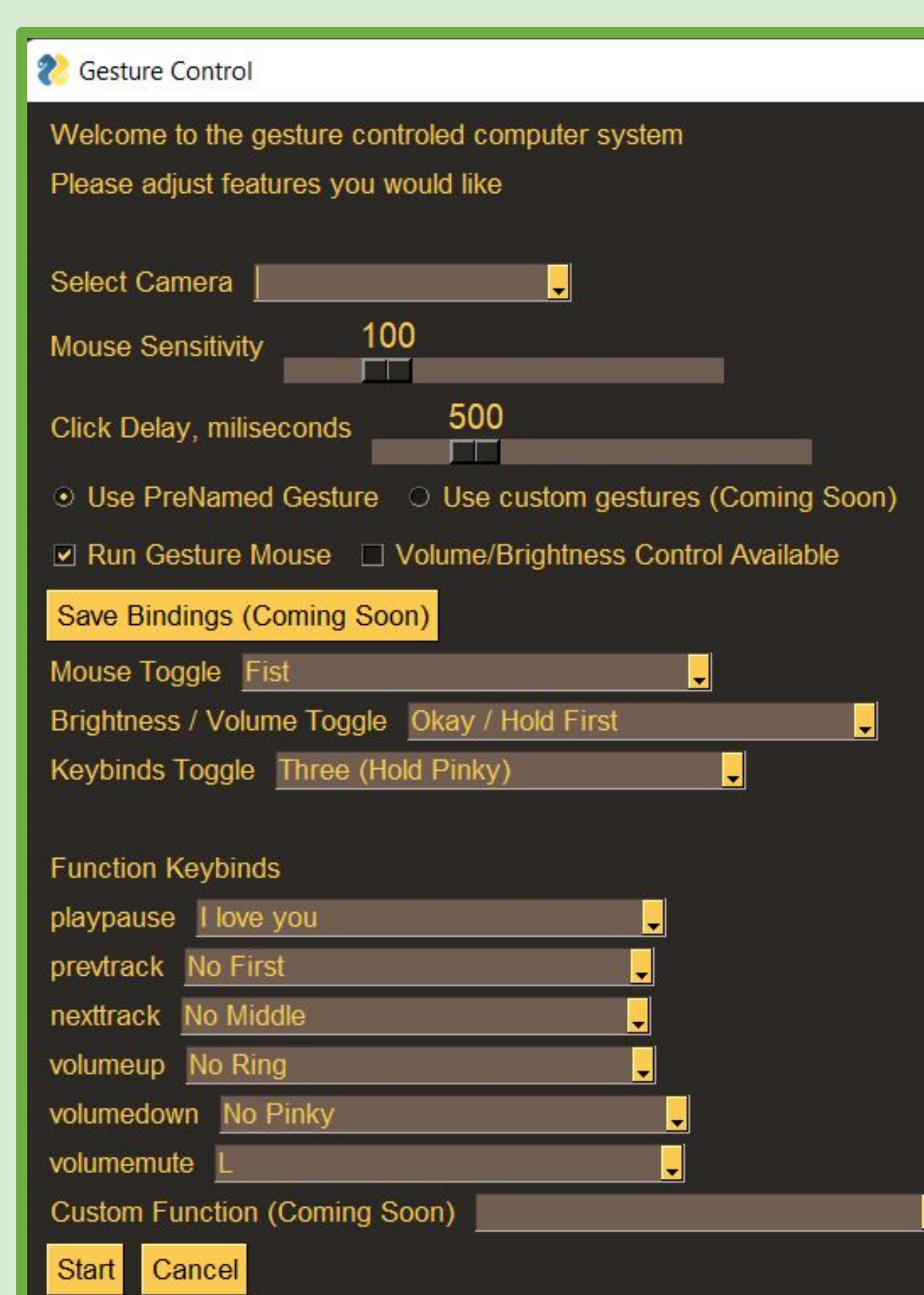


Fig. 5 | Custom Graphical User Interface allows users to customize what gestures are used for each feature. Additional settings allow users to customize their use of the gesture control system

Conclusion and Future Work

Pros:

- Successful Control
- Customization
- Versatility
- Flexibility

Cons:

- Learning Curve
- Accidental Recognition
- Hand Strain
- Limited Precision

The iterative and roughly modular design of the research will allow it to be converted to a library module easily used by other research.

Future Work:

- Convert to a Python Library Module
- Custom Gesture Recognition
- Virtual Reality and Internet of Things
- User Feedback

In order to surpass current mouse and keyboard control methods, certain milestones must be reached in:

- Ergonomics - Less straining and more convenient
- Precision - Close to mouse precision
- Versatility - Wider variety of incorporated features

References

- [1] M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020.
- [2] H. S. Hasan and S. A. Kareem, "Human Computer Interaction for vision based HAND GESTURE RECOGNITION: A survey," 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), 2012.
- [3] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3941-3951, 2016.
- [4] J. Singha, A. Roy, and R. H. Laskar, "Dynamic hand gesture recognition using vision-based approach for human-computer interaction," *Neural Computing and Applications*, vol. 29, no. 4, pp. 1129-1141, 2016.
- [5] Y. S. Tan, K. M. Lim, C. Tee, C. P. Lee, and C. Y. Low, "Convolutional neural network with spatial pyramid pooling for hand gesture recognition," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5339-5351, 2020.
- [6] Jain, Charu, "A Novel Technique for Gesture Recognition System," *International Journal of Recent Research Aspects* ISSN: 2349-7688, Vol. 3, Issue 2, June 2016, pp. 21-24
- [7] Vincze, Marian-Avram, Horațiu-Ștefan Grif, "HAND GESTURES MOUSE CURSOR CONTROL," *Scientific Bulletin of the „Petru Poni” University of Tîrgu Mureș*, Vol. 11 (XXVIII) no. 1, 2014
- [8] Hassan, Murtaza. "AI Virtual Mouse | OpenCV Python | Computer Vision", "Gesture Volume Control", "Finger Counter" YouTube , May 3, 2021. [Video file]. Available: <https://youtu.be/8gPONnGIPgw>, <https://youtu.be/9IEPzbG-xLE>, https://youtu.be/p5Z_GGRC15s. [Accessed: January 21, 2022].
- [9] Mr. Devanshu Singh et al., "Virtual Mouse using OpenCV", *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 9, Issue XII, 2021, <https://doi.org/10.22214/ijraset.2021.38160>