



INTEGRATING COMPUTER VISION IN GAME DESIGN: A MULTI-GAME MENU SYSTEM POWERED BY OPENCV

¹Megha Sharma, ²Sahil Raju Jadhav, ³M Shivashankar, ⁴Vinyas M S, ⁵Sharath M

¹ Assistant Professor, ² Student, ³ Student, ⁴ Student, ⁵ Student

Department of Information Science and Engineering,
Cambridge Institute of Technology, Bangalore, India

Abstract - The rapid developments in machine learning and computer vision have revolutionized many industries, most notably the gaming business. This project makes use of OpenCV in conjunction with Tkinter, a popular Python GUI toolkit, to create an aesthetically pleasing and user-friendly menu system that enables simple and quick game selection. This project includes a menu that enhances interactivity inside four bespoke games (Quiz, Eat the Fruit, Rock-Paper-Scissors, and Number Guessing) by combining OpenCV's image recognition and gesture detection. These games strive to surpass user expectations and showcase the potential of computer vision in gaming by utilizing OpenCV's real-time responsiveness and dynamic interaction features. This invention raises the bar for interactive, user-centered game.

Keywords - Computer Vision, OpenCV, Tkinter, Interactive Gaming, Game Selection Menu, Real-Time Interaction, Gesture Detection, Image Recognition, Gaming Innovation.

I. INTRODUCTION

OpenCV is a potent open-source computer vision framework that is revolutionizing the game business by making it possible to create visually appealing and interactive game interfaces. This project intends to improve user engagement through gesture and command recognition by integrating OpenCV into gaming platforms, offering a smooth and engaging experience. The intention is to create an interface that is easy to use and can adjust to the different tastes of players, thereby becoming the industry leader in accessible and fun gaming experiences. This project takes advantage of OpenCV's functionality through the usage of a Python script that generates a dynamic user interface with a menu for selecting games using the OpenCV and cvzone libraries. This menu offers a selection of interactive games for users to pick from. In the first game, players engage by utilizing hand gestures to select answers in a multiple-choice quiz application. To increase user engagement, users receive both visual and auditory feedback as they progress through the quiz questions. Another imaginative app is "Eat the Fruit." Players utilize facial gestures to grab falling things on screen and discriminate between 'eatable' and 'non-eatable' items thanks to the FaceMesh module, which provides precise facial landmark detection. A game of Rock, Paper, Scissors is also part of the project. In this game, players use hand gestures to play against the computer, which adds a fun element to traditional gaming by repurposing a well-known game in a technologically advanced way. The last game is a number guessing game that asks users to guess a random number by pointing fingers at the webcam. The game continuously compares the user's input with the goal number and provides real-time feedback. Every one of these games showcases how OpenCV's computer vision capabilities can be used creatively and practically to create a rich and immersive gaming experience that raises the bar for interactive gaming environments.

The methods used in the Quiz Game and Eat the Fruit are influenced by Murtaza's Workshop channel, which is well-known for its opencv tutorials. The pygame library is used in the Eat the Fruit code sample of our project to manage audio playing when the player catches an object, improving the game's sensory feedback, which isn't included in the current excerpt. Additionally, the current code snippet cannot handle flipped camera input and missing a few audio indications. Our code also employs more structured global variable management, dynamically sets object placements and speeds, and enhances object reset logic. Our script improves the user experience for The Quiz Game by utilizing Pygame for auditory feedback, running at a higher resolution of 1920x1080, and providing sophisticated text modification possibilities. Along with more useful features including a 'q' key to leave the software and better visual feedback on selections and quiz progression, it also has these features. On the other hand, the first script has a simpler visual design with minimal progress indicators, works at a lesser resolution of 1280x720, and does not have auditory feedback or escape possibilities.

For this interactive gaming project, the installation of Python is essential, along with critical libraries such as OpenCV for computer vision tasks, cvzone for gesture detection, Pygame for audio management, and Tkinter for GUI components. To provide clear video inputs for precise gesture and facial recognition, a high-definition webcam is necessary. Installing the required software and libraries, setting up the webcam, and integrating it with Python scripts to evaluate its functionality are all part of the project setup. Every game script is created with the express purpose of managing real-time video processing and actively involving users with natural actions and expressions on their faces. Additionally, a Tkinter-implemented central game launcher makes it easier to switch between games and improves user experience with interactive menus and advanced background music management. The methods used in the Quiz Game and Eat the Fruit are influenced by Murtaza's Workshop channel, which is well-known for its opencv tutorials. The pygame library is used in the Eat the Fruit code sample of our project to manage audio playing when the player catches an object, improving the game's sensory feedback, which isn't included in the current excerpt. Additionally, the current code snippet cannot handle flipped camera input and missing a few audio indications. Our code also employs more structured global variable management, dynamically sets object placements and speeds, and enhances object reset logic. By using Pygame for auditory feedback, running at a higher resolution of 1920x1080, and providing more sophisticated text customization choices, the second script improves the user experience. This environment supports a number of interactive games, including a number guessing game where players indicate their guesses using finger displays and receive immediate feedback. Other interactive games include a traditional Rock, Paper, Scissors game against the computer using hand gesture recognition, a 'Eat the Fruit' game where players use facial gestures to catch falling objects and distinguish between edible and non-edible items, and a multiple-choice quiz game that uses hand gestures for answers. This all-encompassing configuration produces a versatile and engaging interactive gaming environment.

II. Methodologies

Creating a Multi Game menu system powered by OpenCV goes with the following components:

- **Interface:** The code implements a Tkinter GUI application for launching games stored in a specified directory. It starts by initializing Pygame for background music. Tkinter is used to create the main window with a background image and a listbox displaying available games. Users can select a game from the list and click the "Start Game" button to launch it. The selected game's Python file is executed using `os.system()`. Background music plays throughout, managed by functions `play_background_music()` and `stop_background_music()`. The application leverages Tkinter's event loop (`root.mainloop()`) to continuously handle user interactions and updates on the GUI. Overall, it offers a user-friendly interface for accessing and launching various games with background music support.
- **Quiz Game:** The Quiz Game utilizes computer vision and hand tracking techniques to engage users in answering multiple-choice questions. It begins by capturing video through OpenCV, enabling real-time interaction. The HandDetector module from the cvzone library is employed to track hand movements, allowing users to select answers by hovering over choices. Questions are loaded from a CSV file, enabling easy customization and scalability. As users interact, their progress is visually represented through a progress bar, enhancing engagement and providing feedback. At the end of the quiz, the final score is calculated based on the correctness of answers, giving users immediate feedback on their performance. This game leverages computer vision technology to create an

interactive and engaging learning experience, encouraging users to actively participate and test their knowledge. Through modules like OpenCV and cvzone, it seamlessly integrates video processing and hand tracking functionalities, ensuring smooth user interaction and accurate feedback.

- **Eat the Fruit Game:** The Eat the Fruit Game aims to provide an interactive experience where players catch falling fruits using mouth gestures. The game begins by capturing live video using OpenCV, facilitating real-time interaction. The FaceMeshDetector module from cvzone is then utilized to detect the user's facial landmarks, particularly the mouth region, enabling precise tracking of mouth gestures. Fruits and non-edible objects are loaded as images, appearing as falling objects within the game environment. The player's objective is to "eat" the edible fruits by aligning their mouth with the falling objects, while avoiding non-edible objects. The game dynamically tracks the player's score, incrementing it for each edible fruit consumed. If a non-edible object is mistakenly consumed or if any object reaches the bottom, the game concludes. This implementation provides an engaging and intuitive gaming experience, leveraging computer vision technology to detect and respond to player gestures accurately. Through modules like OpenCV and cvzone, it seamlessly integrates video processing and facial landmark detection functionalities, ensuring smooth gameplay and precise interaction.
- **Rock Paper Scissor Game:** The Rock, Paper, Scissors Game offers players an interactive experience by allowing them to play against the computer using hand gestures. It begins by capturing live video through OpenCV, enabling real-time interaction with the game. Hand gestures are detected using the HandDetector module, which tracks the positions of the player's fingers. The number of fingers extended by the player's hand determines their choice between Rock, Paper, or Scissors. Simultaneously, the computer randomly selects its move from the same set of options. The winner is then determined based on the classic rules of the game. This implementation provides an engaging and intuitive gaming experience, leveraging computer vision technology to detect and respond to player gestures accurately. By utilizing modules like OpenCV and HandDetector, the game seamlessly integrates video processing and hand gesture recognition functionalities, ensuring smooth gameplay and precise interaction.
- **Number Guessing Game:** The Number Guessing Game challenges players to guess a randomly generated number by displaying the corresponding number of fingers. It begins by capturing live video using OpenCV, facilitating real-time interaction with the game. The HandDetector module is then utilized to accurately detect and count the number of fingers shown by the player. Meanwhile, a random number is generated as the target number for the player to guess. The player's total finger count is compared to the target number, providing immediate feedback on whether the guess is higher, lower, or equal to the target number. This feedback guides the player in refining their guesses until they correctly identify the target number. Through the seamless integration of video processing and hand gesture recognition functionalities offered by OpenCV and the HandDetector module, the game delivers an engaging and interactive experience, encouraging players to sharpen their numerical skills while having fun.

System Architecture:

The system architecture gives a high-level overview of the functions and responsibilities of the system. It defines the breakdown of the system into various sub system and the individual roles played by them. The system architecture of the proposed system is shown below in Fig.1.

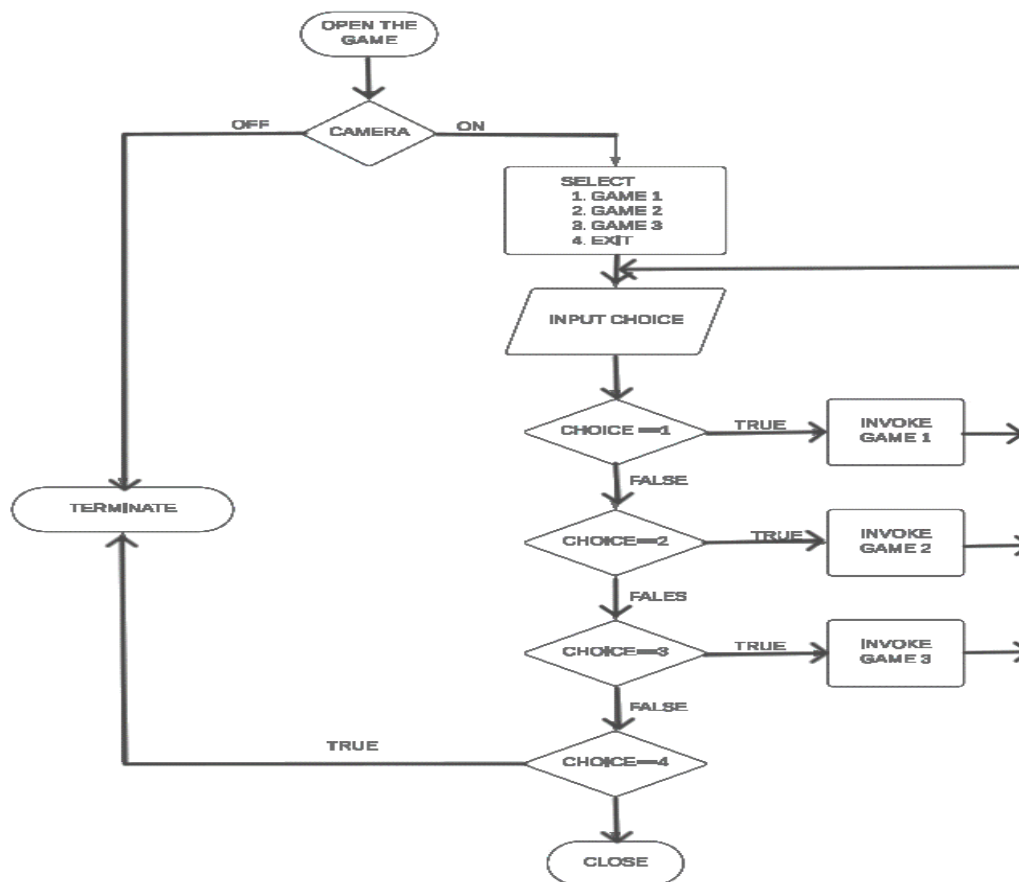


Fig 1: System Architecture

Modules used to build Multi Game Menu System powered by OpenCV:

- **os:** The os module provides a way to interact with the operating system. It allows you to perform operations such as navigating the file system, creating and deleting directories, and executing system commands.
- **tkinter:** Tkinter is a standard GUI (Graphical User Interface) toolkit for Python. It provides a set of tools for creating GUI applications, including windows, buttons, labels, and text boxes. Tkinter is widely used for developing desktop applications with Python's built-in tkinter module.
- **ttk:** The ttk module provides themed widgets that offer a more modern and visually appealing look compared to the standard tkinter widgets. It includes widgets like ttk.Button, ttk.Entry, and ttk.Label.
- **pygame:** Pygame is a set of Python modules designed for writing video games. It provides functionality for creating graphics, playing sounds, and handling user input. Pygame is often used for game development due to its simplicity and ease of use.
- **csv:** The csv module provides functionality for reading and writing CSV (Comma-Separated Values) files. It allows you to easily parse CSV data into Python data structures like lists and dictionaries, and vice versa.

- **cv2:** OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. The cv2 module is the Python interface for OpenCV, providing functions for image and video processing, object detection, and more.
- **cvzone:** cvzone is a Python library built on top of OpenCV and provides additional functionality for computer vision applications. It includes modules for hand tracking, face mesh detection, and more.
- **HandDetector:** The HandDetector class from cvzone's HandTrackingModule is used for detecting hands in images or video streams. It provides methods for finding the positions of the fingertips, detecting gestures, and tracking the movement of the hands.
- **FaceMeshDetector:** The FaceMeshDetector class from cvzone's FaceMeshModule is used for detecting facial landmarks, such as eyes, nose, and mouth, in images or video streams. It can be used for tasks like face tracking, facial expression recognition, and more.

III. IMPLEMENTATION DETAILS

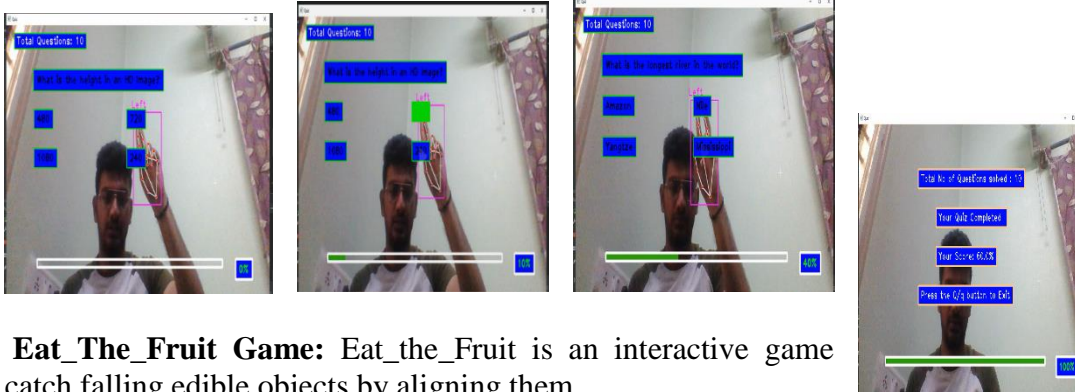
To execute the provided code, first ensure that Python and the necessary modules are installed on your system. Then, run the Python script containing the interface code. Upon execution, a graphical user interface (GUI) window will appear, displaying available game options. This project aims to deliver a seamless gaming experience through the integration of various games such as Quiz, Eat the Fruit Game, Rock Paper Scissors, and Number Guessing Game. Users can select a game from the list by clicking on it. This list is designed with Tkinter which offers a user-friendly environment where players can easily navigate through game options. Additionally, background music is incorporated into the interface to elevate the gaming experience, immersing players further into the world of interactive entertainment. Users can select a game from the list by clicking on it. After selecting a game, clicking the "Start Game" button will initiate the chosen game's execution. The selected game will then start running, while the interface window remains open in the background. Users can interact with the game according to its specific gameplay mechanics. Once the game is complete, users can exit the game window to return to the interface and select another game if desired. Through this project, the aim is not only to entertain users but also to demonstrate the potential of utilizing OpenCV for creating captivating gaming experiences, while simultaneously exploring the functionalities of Python Tkinter for developing intuitive user interfaces.

IV. RESULTS

Interface: The Game Selector GUI provides a user-friendly interface for selecting and playing various games. Presented as a Tkinter application, it offers a list of available games, including Quiz, Eat_the_Fruit, Rock_Paper_Scissors, and NumberGuesser. Users can simply click on the desired game in the Listbox to initiate gameplay. A notable feature of this GUI is the continuous playback of background music, enhancing the overall user experience with an engaging audio ambiance. This combination of intuitive game selection, seamless navigation, and background music creates an immersive environment for users to enjoy their gaming experience. With its simple yet effective design, the Game Selector GUI offers an accessible platform for users to explore and engage with a variety of interactive games at their leisure.



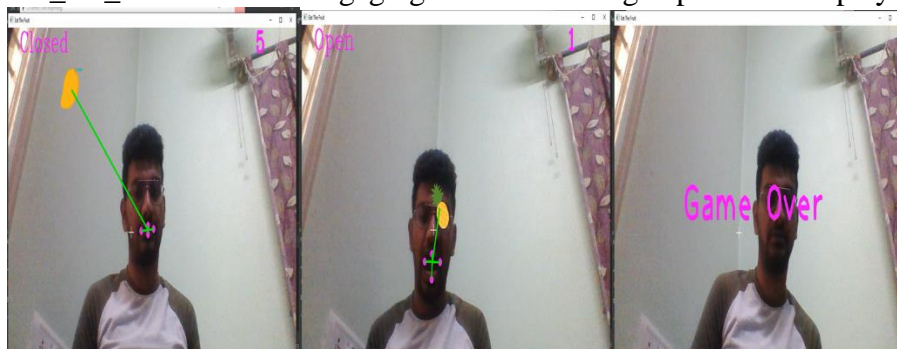
Quiz: The Quiz Game utilizes computer vision technology to engage users in an interactive quiz experience. Questions and answer choices are dynamically loaded from a CSV file, ensuring a diverse range of content for users. Through hand gestures detected by the HandDetector class, users can select answers, adding a unique and intuitive element to the gameplay. The game provides visual feedback with a progress bar, allowing users to track their advancement through the quiz. Upon completion, the user's final score is revealed, providing a sense of achievement and completion. By seamlessly integrating webcam input, dynamic content loading, and interactive feedback mechanisms, the Quiz Game offers an engaging and immersive educational experience for users to enjoy and learn from.



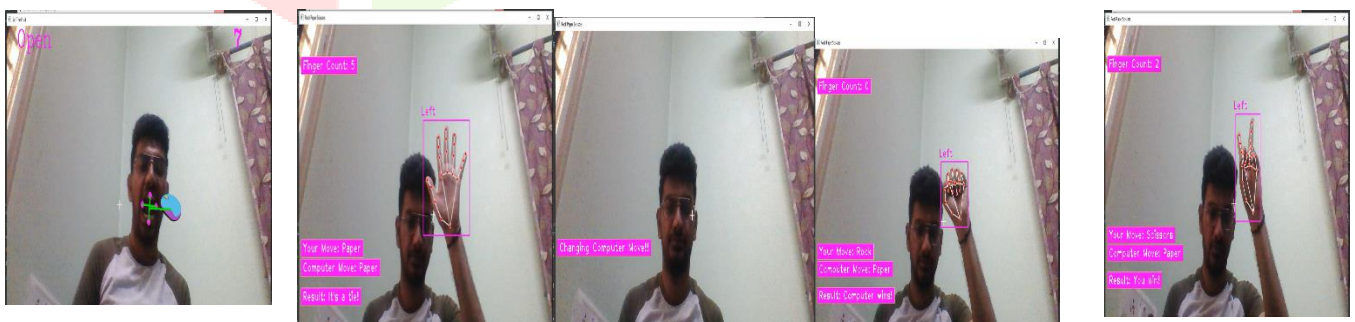
Eat_The_Fruit Game: Eat_the_Fruit is an interactive game where players must catch falling edible objects by aligning them

where players must

mouth with the objects' positions. The FaceMeshDetector class accurately tracks the player's mouth movements, allowing for precise gameplay. Upon successful alignment and "eating" of an edible object, a sound effect enhances the gaming experience, providing immediate feedback to the player. With each successful catch, the player's score increases, adding a competitive element to the game. However, attempting to "eat" non-edible objects or missing too many edible ones results in the game ending, adding challenge and excitement. By combining real-time facial tracking, audio feedback, and strategic gameplay, Eat_the_Fruit offers an engaging and entertaining experience for players of all ages.

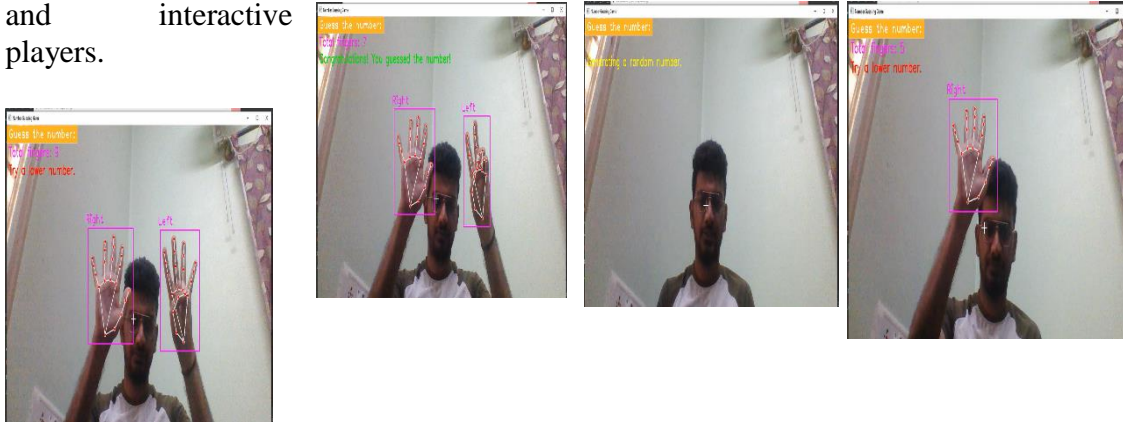


Rock Paper and Scissors Game: Rock_Paper_Scissors is a classic game where players compete against the computer by making hand gestures representing rock, paper, or scissors. The HandDetector class accurately identifies the number of fingers extended by the user, determining their move. Similarly, the computer randomly selects its move. The game then applies the traditional rules to determine the winner of each round. Continuously updating the user's and computer's moves, as well as displaying the result of each round, provides real-time feedback and enhances the gaming experience. With its simple yet engaging gameplay and seamless hand gesture detection, Rock_Paper_Scissors offers hours of fun for players of all skill levels.



Number Guesser Game: The provided code implements an interactive number guessing game using hand gestures captured through a webcam feed. The game interface presents a live video feed titled "Number Guessing Game," where players interact by gesturing with their hands. A HandDetector class identifies hand movements, enabling players to make guesses by extending fingers. The interface includes prompts directing players to guess a randomly generated number between 1 and 10, with feedback displayed based on their guesses. Depending on the accuracy of the guess, messages suggest trying higher or lower numbers. If no hands are detected, the game generates a new random number and provides corresponding instructions. The game continues until the player decides to quit by pressing the 'q' key, ensuring a seamless termination.

process with the release of the webcam feed and closure of all game windows. Overall, it offers an engaging and interactive experience for players.



REFERENCES

- [1] Mengi, Rakshan, Shubam Gupta, Bhawna Sharma, and Sheetal Gandotra. "GAME INTERACTION USING GESTURE RECOGNITION."
- [2] Bradski, Gary. "The openCV library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25, no. 11 (2000): 120-123.
- [3] Rautaray, Siddharth S., and Anupam Agrawal. "Interaction with virtual game through hand gesture recognition." In *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*, pp. 244-247. IEEE, 2011.
- [4] Yoon, Ho-Sub, and Su-Young Chi. "Visual processing of rock, scissors, paper game for human robot interaction." In *2006 SICE-ICASE International Joint Conference*, pp. 326-329. IEEE, 2006.
- [5] Mohd, Tauheed Khan, Anas Akkar, Justin Cassens, Samuel Cregan, and Maame Araba VanderPallen. "Implementing Playing Cards BlackJack Game using OpenCV." (2023).
- [6] Bangole, Narendra Kumar Rao, and S. Ranjana. "Playing Rock-Paper-Scissors Using AI Through OpenCV." In *The Software Principles of Design for Data Modeling*, pp. 41-64. IGI Global, 2023.
- [7] Kumar, Chandan, and Ms Akanksha Kochhar. "Hill Climb Game Play with Webcam Using OpenCV."
- [8] Kim, Seongki, JinHo Ryu, Youngchul Choi, YooSeok Kang, Hongle Li, and Kibum Kim. "Eye-contact game using mixed reality for the treatment of children with attention deficit hyperactivity disorder." *IEEE Access* 8 (2020): 45996-46006.
- [9] Inamdar, Fatima M., Rahul Jambhulkar, Indar Patil, and Akshay Misal. "Snake Game with Motion Control Using Computer Vision." In *International Conference on Soft Computing for Security Applications*, pp. 515-541. Singapore: Springer Nature Singapore, 2023.
- [10] Harini, V., V. Prahelika, I. Sneka, and P. Adlene Ebenezer. "Hand gesture recognition using OpenCv and Python." *New Trends in Computational Vision and Bio-inspired Computing: Selected works presented at the ICCVBIC 2018, Coimbatore, India* (2020): 1711-1719.
- [11] Park, Ji-Young, and June-Ho Yi. "Gesture recognition based interactive boxing game." *International Journal of Information Technology* 12, no. 7 (2006): 36-44.