PAPER NAME

project report team 19-4A.docx

| | |
|---|---|
| WORD COUNT | CHARACTER COUNT |
| 4717 Words | 26141 Characters |
| PAGE COUNT | FILE SIZE |
| 43 Pages | 5.1MB |
| SUBMISSION DATE | REPORT DATE |
| May 11, 2023 9:16 AM GMT+5:30 | May 11, 2023 9:17 AM GMT+5:30 |

## ● 12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 12% Internet database
- Crossref database
- 1% Publications database
- Crossref Posted Content database

## ● Excluded from Similarity Report

- Submitted Works database
- Small Matches (Less then 10 words)
- Bibliographic material

# TABLE TENNIS USING HAND GESTURE-PONGER

## DESIGN PROJECT–1 REPORT

*Submitted by*

| | |
|---|---|
| DANI N | (21113004) |
| DHARSHAN R E | (21113049) |

Under the guidance of

DR.R.KRISHNAVENI

Professor

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

CHENNAI - 603 103

MAY 2023

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

# BONAFIDE CERTIFICATE

Certified that this project report **Table Tennis Using Hand Gesture** is the bonafide work of **DANI N (21113004) & DHARSHAN R E (21113049)** who carried out the project work under my supervision during the academic year 2**022-2023**.

Dr.R.KRISHNAVENI,
**SUPERVISOR**
PROFESSOR
Department of CSE.

**INTERNALEXAMINER**
Name:________________________

Designation:___________________

**EXTERNALEXAMINER**
Name: _________________________

Designation:_____________________

Project Viva - voce conductedon _________________

# ACKNOWLEDGEMENT

First and foremost we would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it.

We wish to express our deep sense of gratitude from the bottom of our heart to our guide **Dr.R.Krishnaveni, Professor**,**Computer Science and Engineering,** for her motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

We would like to extend our heartfelt gratitude to **Dr. J.Thangakumar, Ph.D., Professor & Head, Department of Computer Science and Engineering** for his valuable suggestions and support in successfully completing the project. We are really indebted to our **Dean Computing Sciences Dr.R.Kanthavel,** for his support.

We thank the management of **HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE** for providing us the necessary facilities and support required for the successful completion of the project.

As a final word, we would like to thank each and every individual who have been a source of support and encouragement and helped us to achieve our goal and complete our project work successfully.

# ABSTRACT

Hand motion detection technology is a cutting-edge computer vision application that could revolutionize table tennis. Unlike the traditional inputs such as keyboard & mouse and controller, this technologycaptures live footage of players, tracks their hand movements and landmarks, and uses a deep learning algorithm to identify specific gestures associated with various game functions. By in real-time by providing them a constant efficiency, this recognition technology is attracted players and improves their techniques and their overall gaming experience. In addition, it can make table tennis more accessible to people with mobility places, allowing them to participate in the game using hand movements instead of physical movements. The potential benefits of this technology are not limited to table tennis. However, it is important to note that this technology and game have some limitations such as precise control, the possibility of error detection, and the cost of implementation. But this hand gesture technology can interact with normal people and they can easily understand and can play the game.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| CNN | Convolutional Neural Network |
| RT | Regression Tree |
| FGPA | Field Programmable Gate Array |
| VHDL | Very High-Speed Integrated Circuit |

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Hand Gesture Table Tennis is a game that tracks and recognizes the hand movements of players in a table tennis game. The webcam records video players during gameplay and processes the footage with computer vision technology that recognizes hand movements related to various game functions, e.g. such as hitting the ball and returning the ball. The game offers players a real-time experience of their performance. This technology can be used for more than table tennis and is also accessible to the disabled. Overall, the game of table tennis makes the game more interesting and enjoyable for all skill levels.

## 1.2 The Motivation of the Project

The motivation for creating a table tennis game using hand gestures couldprovide a deep and interactive gaming experience that does not require physical paddles. By using hand gestures, players can have a more natural and easy way of interacting with the game, and it can also promote physical activity and exercise as players move their hands and arms to play this game. This type of game could be a fun and engaging way to challenge players' hand-and-eye coordination and reflexes for hitting the ball.

## 1.3 Problem Definition and Scenarios

In the problem definition, the main point is the lack of a natural and easy way of interacting with the game. Difficulty in executing the shots in our direction andcreating a game where players can control a virtual table tennis game using hand gestures easily, instead of physical paddles. Therefore, the problem is to create a robust system that can accurately track hand gestures and translate them into precise racket movements to provide the best and most engaging table tennis experience.

## 1.4 Organization of the Report

The report consists of 10 chapters. This report begins with a preface to the motives and the project being undertaken. Following that in the second chapter, exemplifications and literature reviews are bandied. We discuss the objects of the proposed affiliated work system and its advantages over the subsisting systems in the third chapter of the report. The system design and flowchart are shown in the report's fourth chapter. The project requirements, which include the hardware and software needed to run this system, are discussed in the fifth chapter of the report. A thorough explanation of the modules utilized in the system can be found in Chapter 6. The perpetration and results deduced from executing the inferred system are covered in Chapters 7 and 8 We have discussed the result's conclusion in Chapter 9. The report consists of the contribution of the team in chapter 10 followed by the references section and appendix.

## 1.5 Summary

The project's primary motivation is to provide an engaging and enjoyable gaming experience that promotes physical activity. The use of hand gestures adds an extra level of immersion to the gameplay, Additionally, the project aims to encourage physical activity by incorporating it into the gameplay.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

The purpose of this chapter is to review the research papers that were taken for this project. The algorithms used for the project will be examined in this chapter. Research papers extracted from reputed journals

## 2.2 Machine Vision-Based Ping Pong Ball Rotation Trajectory

A machine vision-based system can track the rotation trajectory of a ping pong ball. It consists of a high-speed camera, an image processing module, and a computer vision algorithm. The camera captures the movement of the ball, and the image processing module processes the images.

The algorithm used here is CNN. This algorithm uses edge detection and image segmentation techniques to identify the ball in each frame of the video. It then tracks the motion of the ball by calculating its position, velocity, and acceleration over time. By analyzing the rotation of the ball, spin rate, and direction of the ball, which is used to give its trajectory.

Overall, this machine vision-based system can be used in a variety of applications, including sports training,analysis, and review systems in various sports

## 2.3 Machine Learning With The Pong Game

The Pong game uses a CNN to train an AI agent to play against human players. The game is implemented using Python and the Pygame library for graphics and user interface.The CNN is trained using a dataset of pre-recorded gameplay, similar to the project using a neural network. The game screen pixels are pre-processed to extract features, such as the position of the ball and paddles.The CNN is trained using reinforcement learning with a variant of Q-learning called deep Q-learning. The algorithm learns to predict the Q-value of each action in a given state, and the action with the optimal Q-value is chosen.

## 2.4 Pong Game using AI

Pong game is a simple game. It uses AI on both sides to control Paddles. Model reaction time, which waits some period before making decisions, Two-Part strategy: model accuracy, in which the computer knows where the ball will land and addsa random error factor to pretend that the computer is fallible. In this paper, try to balance the game by making the computer skills better if it starts losing.The algorithm used in this paper is Q-learning reinforcement where the agent tries to make its own decision

## 2.5 Design of Pong Game Using VHDL

The Pong game is designed using VHDL to create the necessary hardware components, including the game screen, paddles, and ball. The game logic is implemented using a finite-state machine algorithm that controls the game components based on the user input and the current state of the game.The VHDL design is then synthesized and implemented in a network processing unit. an NPU is specifically designed for accelerating neural network computations and is a more general-purpose programmable chip.

The Pong game implemented in the network processing unit is connected to the computer keyboard. The network processing unit generates the game screen, receives user input, and plays the game based on the user input and the current state of the game.

## 2.6 FPGA Implementation of Pong Game

First, you need to define the rules of the game, such as how the ball and paddles move, how the score is kept, and when the game ends.Next, you create the graphical interface, which is what the player sees on the screen. This includes creating the ball, paddles, and score displays.Then, you [11] use a hardware description language (HDL) such as Verilog or VHDL to create the necessary hardware components for the game logic and graphical interface.After you've written the HDL code, you use a simulator to test and debug any issues.Finally, you use a synthesis tool to convert the HDL code into a configuration file that can be loaded onto the FPGA, which is what implements the Pong game.

In summary, implementing a Pong game in FPGA involves designing the game logic and graphical interface, creating the necessary hardware components in HDL, testing and debugging using a simulator, and then synthesizing the HDL code into a configuration file that can be loaded onto the FPGA.

## 2.7 Summary

Researching several research papers that are related to this work led to the conclusion stated here. Detailed summaries of each research paper have been provided, along with a description of the advantages and disadvantages of each proposed system.

# CHAPTER 3

# PROJECT DESCRIPTION

## 3.1Overview

In this chapter, the objectives and benefits of the project will be discussed. In addition, the existing system will be evaluated and its drawbacks discussed.

## 3.2 Objective of the Project Work

Controls the bats on the screen using hand gestures in multiplayer.Hit the ball back using a computer-controlled AI bat on one side.Compete against computer AI to achieve the highest score.Aim to score points by strategically hitting the ball past the computer’s bat and onto the other side of the screen.The direction of the ball depends on our hand movement

## 3.3Existing System

There are several existing systems for table tennis game are electronic scoreboards, robot training systems, virtual reality ping pong, ping pong ball machines, and ping pong tables with built-in sensors. These systems have different skills, budgets, and preferences and these systems are used for gaming, training, coaching, etc. Scoreboards are used for showing the scores of both sides, virtual reality ping pong is used for training the players or for gaming, ping pong robots are tools to improve player training efficiency, ping pong table with build sensor is used for training the players to improve strategy and techniques.

## 3.4Shortcomings of Existing System

Existing systems include high costs, maintenance requirements and limited availability. These shortcomings will affect the players to fully utilize or access these systems.

### 3.5 Proposed System

In this project, we have created a table tennis game using our hands. we have set up a webcam and accessed it using computer vision technology. using the webcam and importing Mediapipe module(which is from Google) is used to detect hands type and its landmarks

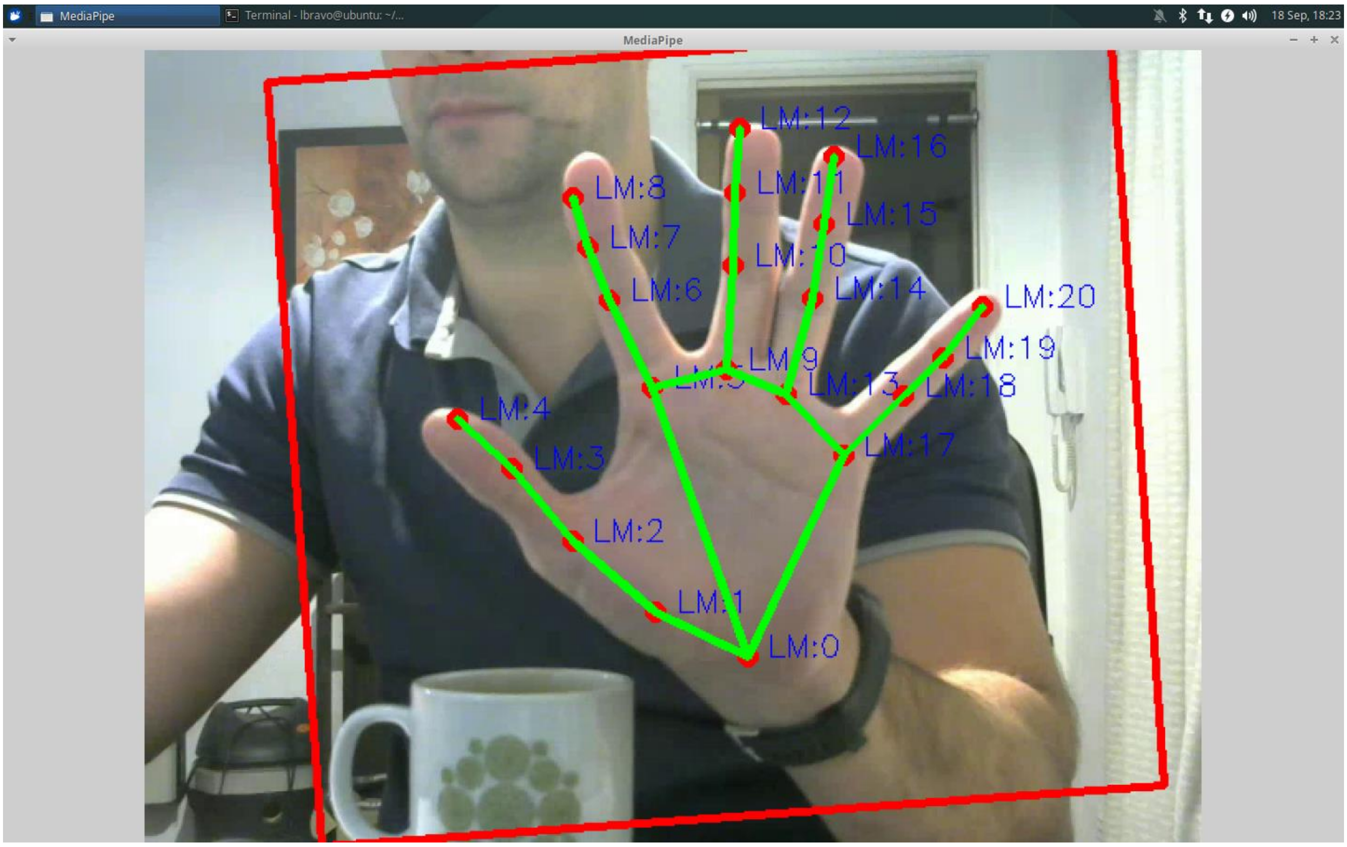


**Figure 3.1 Detecting Hand Landmarks**

The mediapipe model is a machine learning model which is trained using CNN. From Figure 3.1we are detecting the hand landmarks(LM:0, LM:1,…., LM:20). In this game, we have set two modes single player and multiplayer which becomes a fun game for alone players as well as multiplayer. the Game has a total of 11 rounds and displays the score of both players at last

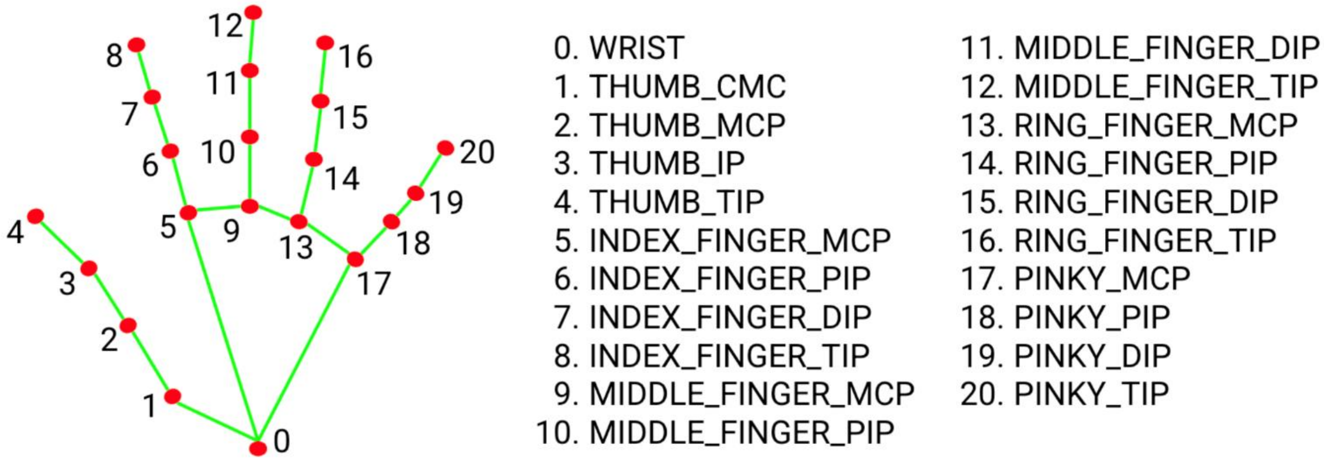


**Figure 3.2 Hand Landmarks detection guide**

Continuously monitor hand gestures and update the game accordingly. Figure 3.2 shows that the landmarks number has each identity. Display the updated game state on the screen.The system would involve machine learning techniques to recognize different hand gestures and map them to specific commands for the game. The system would also require a robust camera or sensor to accurately track hand movements in realtime.

## 3.6 Benefits of the Proposed System

The proposed system can provide the most naturalexperience using hand gestures to control the paddles and can feel easier and more natural than using traditional inputs like a keyboard or joystick. Players must use their arms and hands to move the racquet, which can provide a mild workout.Third, is Accessibility this game can be more accessible for players with disabilities that make traditional inputs difficult to use

## 3.7 Summary

We have summarized this chapter about the Existing system, proposed system, and objective. By evaluating the current system, the chapter justifies the need for this project and sets the stage for the proposed system.

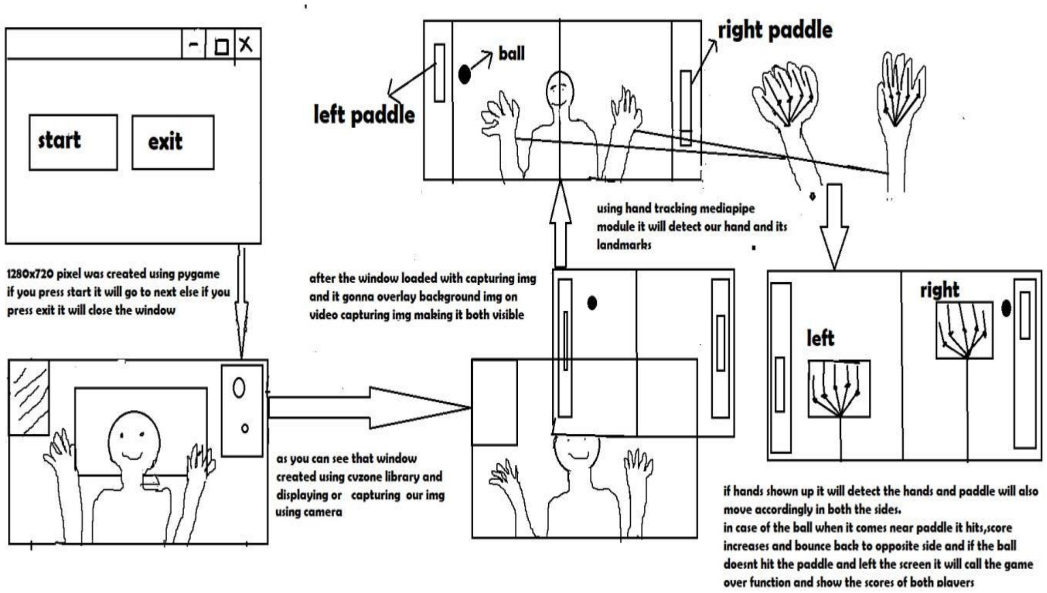
# CHAPTER 4

# SYSTEM DESIGN

## 4.1 Overview

In this chapter,we will provide a clear explanation of the architecture diagram for the proposed system.

## 4.2 Architecture Diagram



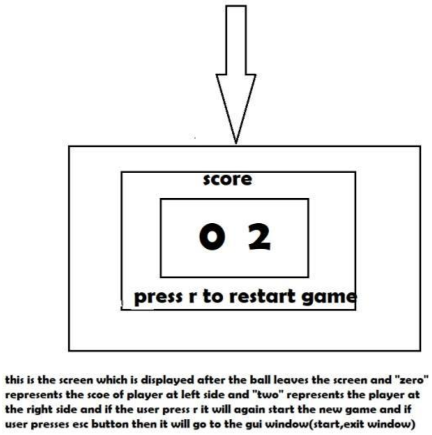
1280x720 pixel was created using pygame
if you press start it will go to next else if you
press exit it will close the window

as you can see that window
created using cvzone library and
displaying or capturing our img
using camera

after the window loaded with capturing img
and it gonna overlay background img on
video capturing img making it both visible

using hand tracking mediapipe
module it will detect our hand and its
landmarks

if hands shown up it will detect the hands and paddle will also
move accordingly in both the sides.
in case of the ball when it comes near paddle it hits,score
increases and bounce back to opposite side and if the ball
doesnt hit the paddle and left the screen it will call the game
over function and show the scores of both players



this is the screen which is displayed after the ball leaves the screen and "zero" represents the scoe of player at left side and "two" represents the player at the right side and if the user press r it will again start the new game and if user presses esc button then it will go to the gui window(start,exit window)

**Figure 4.1 Architecture Diagram**

The architecture Diagram in Figure 4.1 explains how and flow of the game, using the pygame module will create an interface window with 1280x720 resolution window has been created, in the user interface has two buttons start and an exit button if the player presses the exit button it will close all window or else if the player presses the start button it will call the game where the game first call camera to access the live footage in 720p resolution, after capturing the background texture will overlay to make a new interface using OpenCV image processing module. Using the hand-tracking mediapipe module we will detect hands and landmarks, using handmarks player can control the rackets. The ball has its collision to move according to physics, if the player hits the ball they will have the +one score. If the user misses the ball it will go to the restart screen menu by pressing ‘r’ in the restart menu player can play the game again. The player presses the ‘ESC’ key on the keyboard it will go to the user interface.

### 4.3 Summary

The architecture diagram for the proposed system was discussed and the idea of this project was known by this architecture

# CHAPTER 5
# PROJECT REQUIREMENTS

## 5.1 Overview

The hardware and software used in this project will be discussed in this chapter along with its specifications. A brief description of the technology is as follows

## 5.2 Software Specifications

**Framework & Libraries**

- Open cv
- Pygame
- Numpy
- Python 3.10

## 5.3 Technology Used

- **Computer Vision**

  Used for Image Processing and accessing the webcam.

- **Machine Learning**

  Machine learning is a branch of AI and computer science. Which focuses on the use of data and algorithms to imitate the way that human learns.

- **Deep Learning**

  It is a method in AI that teachescomputers to process the data in the way that human learns

### 5.4 Summary

In this chapter, the hardware and software, and technologies used were mentioned.

# CHAPTER 6

# MODULE DESCRIPTION

### 6.1 Overview

The objective of this section is to provide a comprehensive overview of all various modules implemented in this project

### 6.2 Modules

The following modules are used for the implementation of this project

- GUI
- GUI Imports
- Pygame programs
- Image Processing
- Hand Tracking

#### 6.2.1 GUI

GUI in simple terms is an interface for creating video games, Handles graphics, sound, user input, networking,API, and easy to learn

### 6.2.2 GUI Imports

GUI imports are used for importing the images, and audio files, and it is processed to make the interface more colorful.

### 6.2.3 Pygame Programs

It provides constants and events, including key and mouse events, video modes, andcolor values.

### 6.2.4 Image Processing

This is a computer vision library that provides tools for image and video processing and it is used to capture video from the camera and overlay images onto the video feed.

### 6.2.5 Hand Tracking

The handtracking module is a third-party library built on top of OpenCV and Mediapipe that provides tools for hand detection and tracking in real-time video feeds.

## 6.3 Summary

This section described the description of every module which were used in our code.

# CHAPTER 7
# IMPLEMENTATION

## 7.1 Overview

This chapter provides an in-depth description of the implementation process. Additionally, the chapter describes the Pygame interface.

## 7.2 Implementation

Implementation of table tennis using hand gestures is done using pygame(GUI),pygame locals,cv2, and mediapipe modules.First, pygame has been used for the interface of our game.Second, pygame.locals have been used for importing audio files and images.Third, cv2 has been used for accessing the webcam to get a video of a player and processing images for a ball,racket, and background images using resize from imutils to resize the image and change the color using cvtcolor.Fourth, a media pipe has been used for tracking hands and identifying the hands which helps to move the racket based on movement of hands.

## 7.3 Summary

The purpose of this chapter is to provide a detailed description of the proposed system and its implementation. In addition to using Pygame for game interfaces.

# CHAPTER 8
# RESULT ANALYSIS

## 8.1 Overview

Implementation of the whole project is well explained in this chapter and the effectiveness of the proposed system is discussed.

## 8.2Result Analysis

**Table 8.1 Performance Evaluation in Terms ofAccuracy**

| Research Papers | AI Bot | Multiplayer | Hand Tracking |
|---|---|---|---|
| Yilie Wang et al [1] | - | 40.6% | - |
| Doina et al [2] | 44.6% | - | - |
| Akash et al [3] | 64.7% | 70.34% | - |
| Sudhanshu et al [4] | - | 50.4% | - |
| Shith et al [5] | - | 50.2% | - |
| Proposed System- Ponger | 75.43% | 80.76% | 86.83% |

From Table 8.1 Performance evaluation is done in terms of accuracy.

For AI Bot Doina et al [2] got 44.6%, Akash et al [3] got 64.7% comparing this existing system(AI Bot) accuracy we have got 75.43%.

For Multiplayer Yilie Wang et al [1] got 40.6%, Akash et al [3] got 60.34%, Sudhansu et al [4] got 50.4%, Shith et al [5] got 50.2% comparing this existing system(Multiplayer) accuracy we have got 80.76%.

For Hand Tracking none of the Authors used Hand Tracking, we got 86.83% accuracy.

### 8.3 Summary

Thus the implementation of the whole project is well explained in this chapter 8 and the proposed system has also shown how effective it is.

# CHAPTER 9
# CONCLUSION AND FUTURE WORK

### 9.1 Overview

The overview of this chapter is to discuss the result of this project, and this discussion is followed by the conclusion and future work of this project's possibilities

### 9.2 Conclusion

The project provides a personalized gameplay experience by allowing players to control the game using intuitive hand gestures and adapting to individual player styles.

It uses deep learning, machine learning for AI bots, and CNN.

Overall, the project contributes to the advancement of interactive technologies and inspires future research and development in this area

### 9.3 Future work

We will continue to future work on this project by implementing this game as one of the online website games, and apps for the Microsoft Store, play store, and app store

Making more efficient and implementing many modes, difficulties, artworks, skill-based bats

### 9.4 Summary

The conclusion and Future work of this chapter are discussed well, detecting the hand landmarks controlling the bats is successful.

# CHAPTER 10
# INDIVIDUAL TEAM MEMBER's REPORT

### 10.1 Individual Objective

**Dani N:**

Researching the different modes for the game, and journal to fit our idea of this project, exploring the algorithms, and comparing them between these algorithms to which one gives more accuracy

**Dharshan R E:**

Exploring the different interfaces, various research papers, and modules for hand tracking and comparing which module library will give more accuracy, paperwork, designing the game interface, and code for efficiency.

### 10.2 Role of the Team Members

**Dani N:**

Reviewing some research papers and collected algorithms, preparing the PPT, and Some parts of "def codes".

**Dharshan R E:**

Reviewing some research papers and modules to track hands, making efficient interface and some parts “def codes” like AI bot, game over part.

## 10.3 Contribution of Team Members

**Dani N(21113004)**

- Gathering Information such as Research papers, Images, and Youtube videos for reference
- Learning Python Opencv, pygame integration
- Making ppt slides such as Intro, Objective, Architecture Diagram, System requirements, Future work and conclusion, Literature Survey such as Machine Vision Based Ping Pong, Machine Learning with Pong Game, conclusion, and Future work
- Making the pygame Interface and testing it
- Accessing Camera module

**Dharshan R E(21113049)**

- Gathering Information such as Research papers, Camera access, and Youtube videos for reference
- Learning Python Mediapipe, pygame button module integration with Wav
- Making ppt slides such as Abstract, Goals and Motivation, Module and Modular description, Problem definition, Literature Survey such as Pong game using Ai,

VHDL, FGPA, Proposed System/Work, Result and discussion

- Making The pygame button modules and testing it
- Detecting the Hands and its landmarks

# REFERENCES

[1] Yilie Wang and Ling Wang(2021):Machine Vision-Based Ping Pong Ball Rotation Trajectory. Journal of Computational Intelligence and Neuroscience, Volume 22, pp:12,

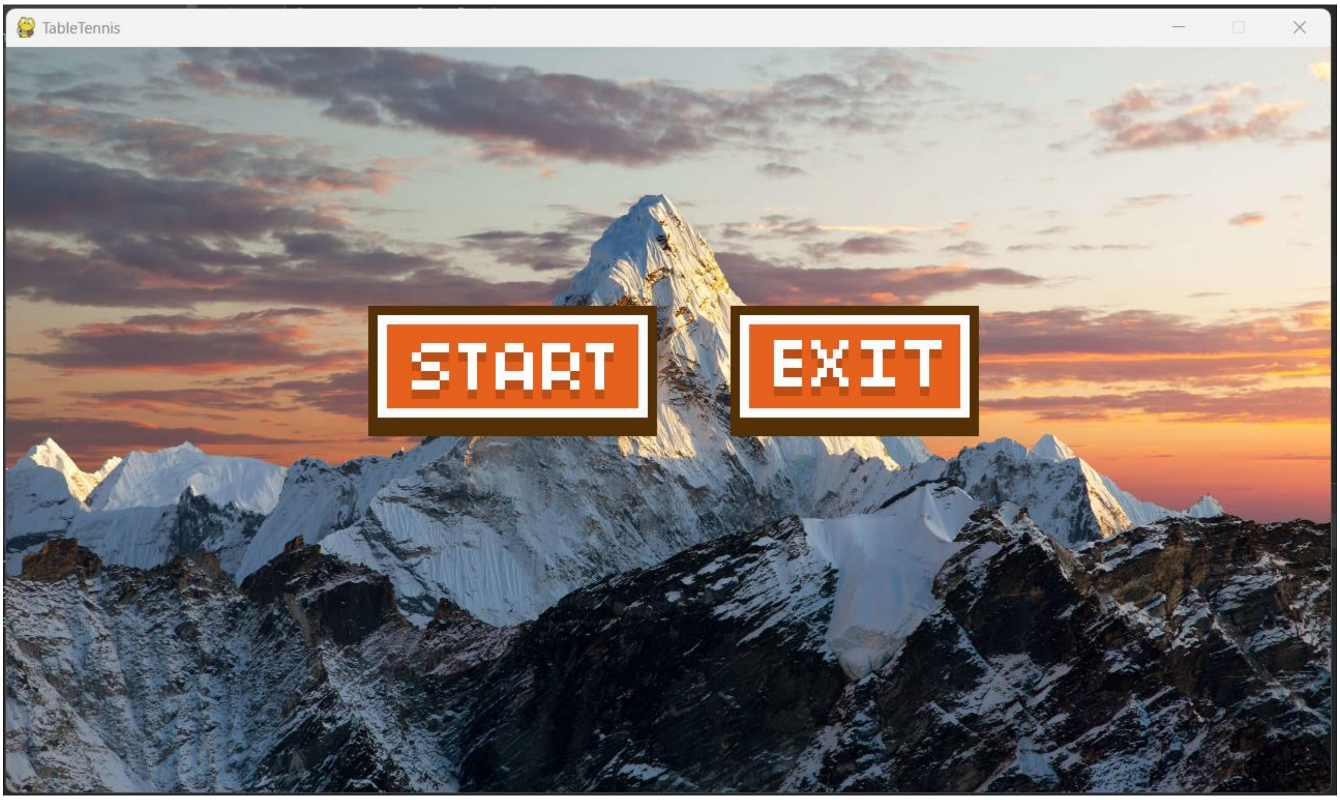
[2] Doina and Logofătu(2022):Machine Learning With The Pong Game. 21st International Conference on Computational Science and Applications(ICCSA), Volume 21, pp:8-15,

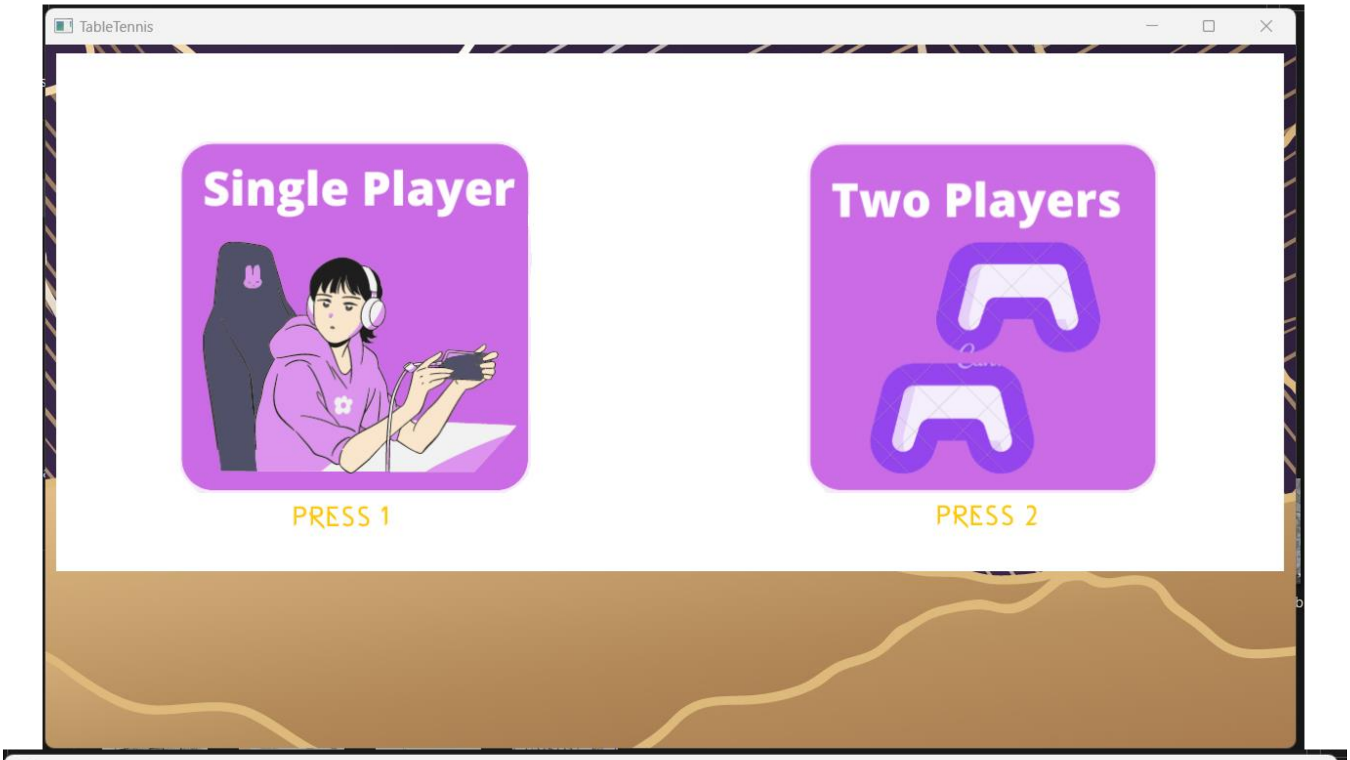
[3] Akash, Ruchika, and Ashish(2022):Pong Game using AI. International Journal of Scientific and Research Publications,Volume 13, pp:93

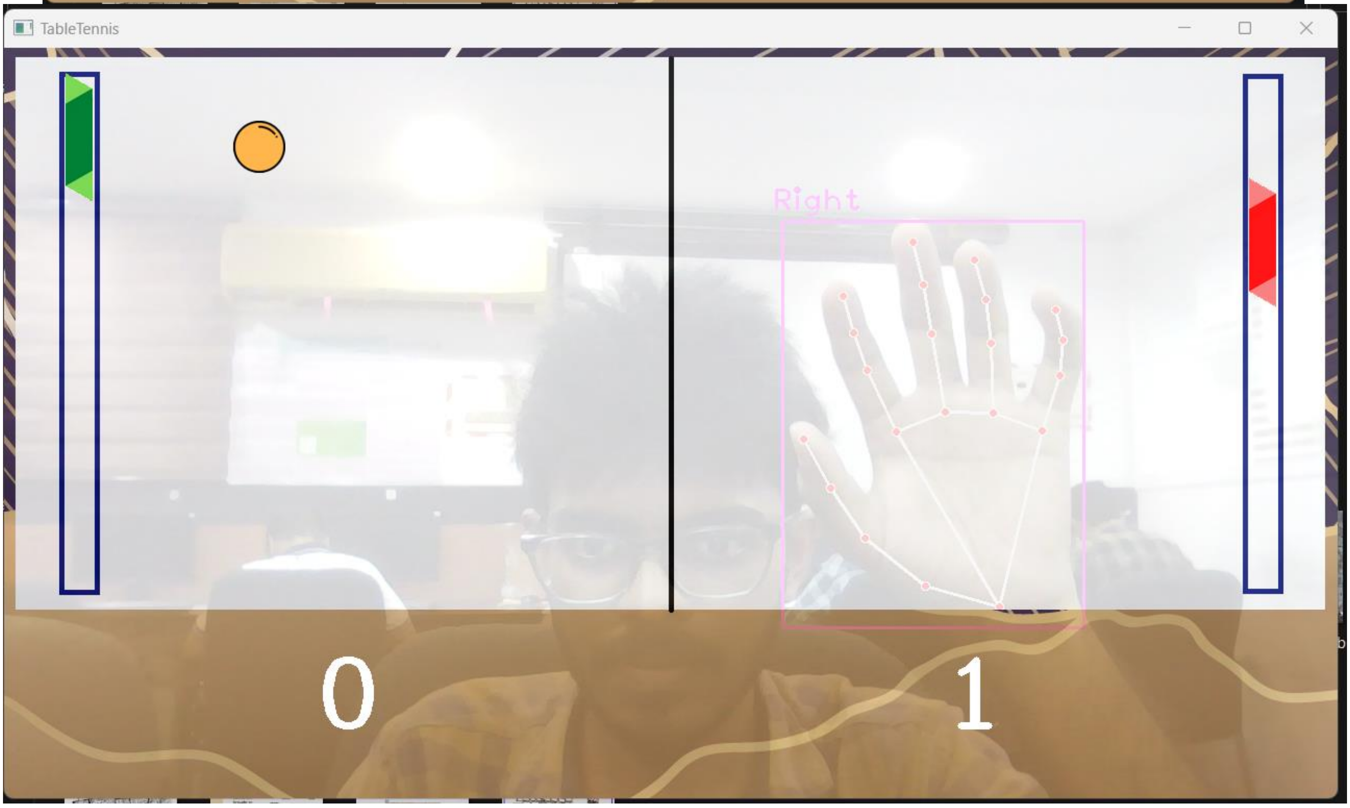
[4] Sudhanshu and Nakul(2022):Design of Game Pong using VHDL.International Journal of Engineering Research & Technology (IJERT),Volume 17,pp:200

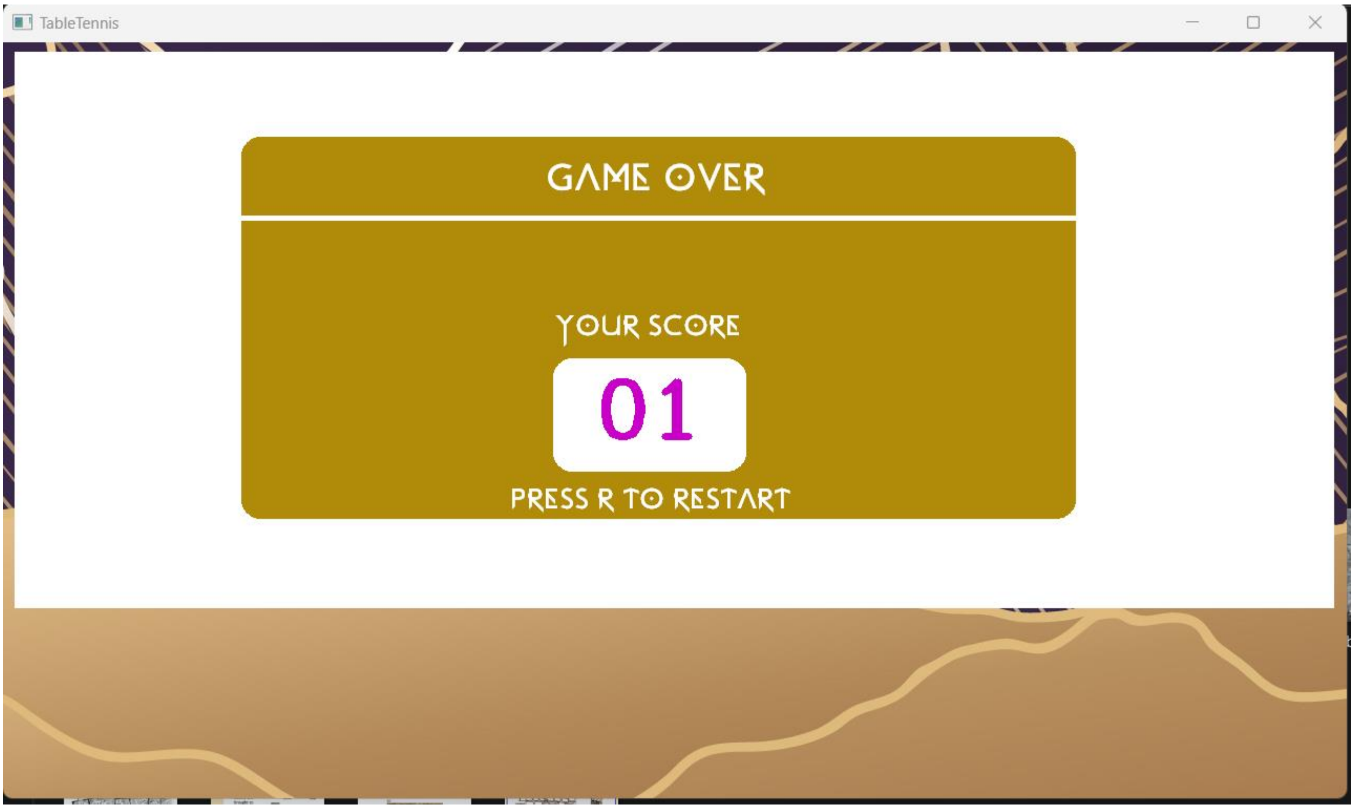
[5] Shith, Huang, and Tsai(2022):FGPA Implementation of Pong Game.Federated Conference on Computer Science and Information Systems,Volume 31,pp:45

# APPENDIX A
# SAMPLE SCREEN




TableTennis
Single Player
PRESS 1
Two Players
PRESS 2


TableTennis
Right
0
1

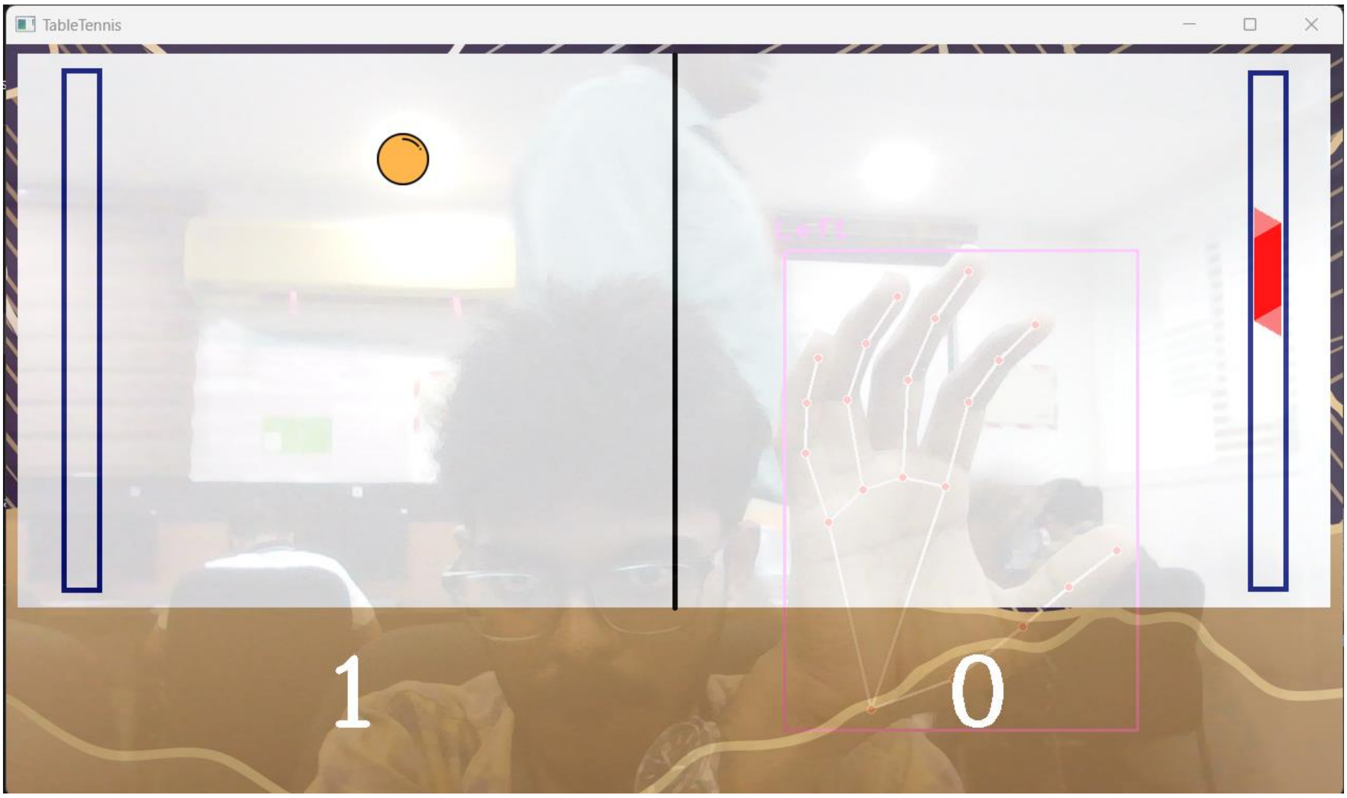


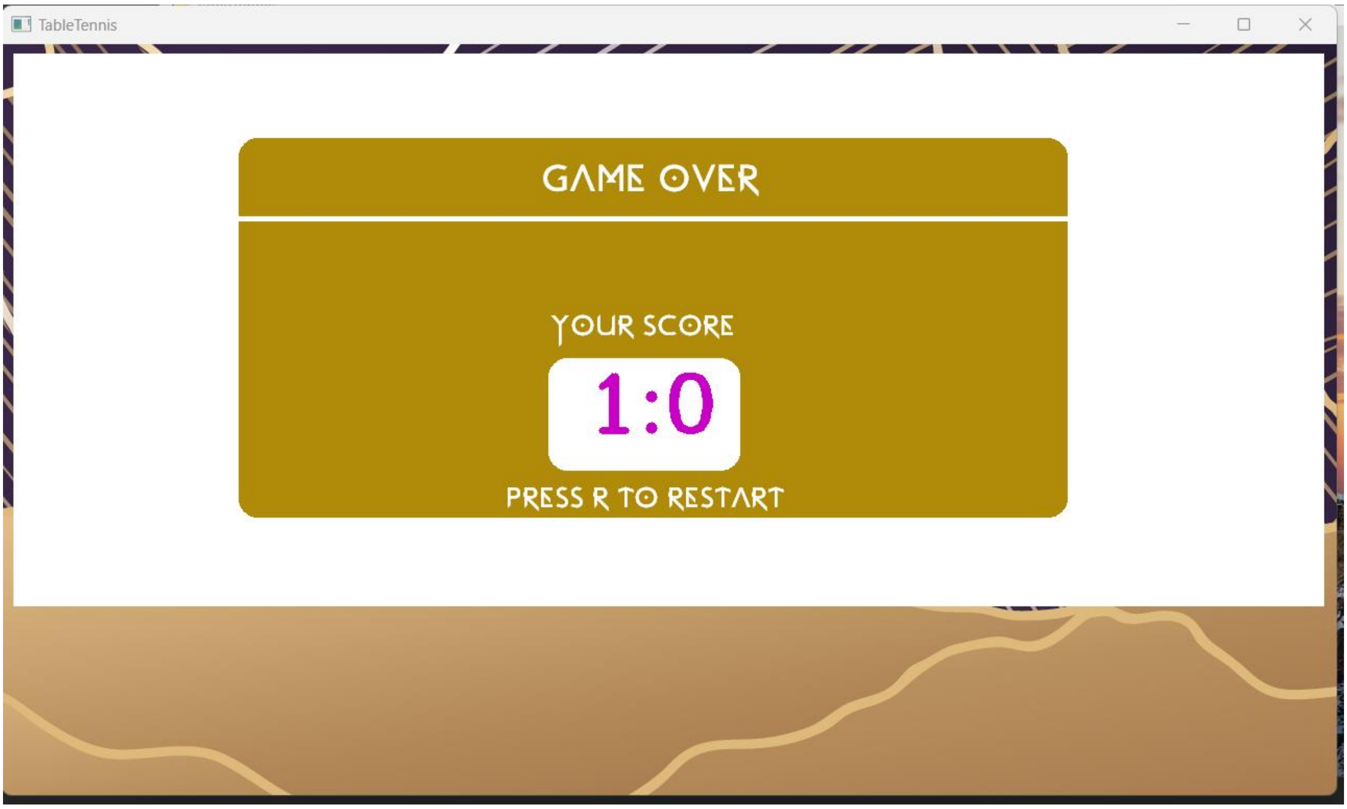
GAME OVER

YOUR SCORE

01

PRESS R TO RESTART





GAME OVER

YOUR SCORE

1:0

PRESS R TO RESTART

## APPENDIX B
## SAMPLE CODE

```
import pygame
from pygame.locals import *
import cvzone
import cv2
from cvzone.HandTrackingModule import HandDetector
import numpy as np
import pygame
pygame.mixer.init()
pygame.init()
# Load button images
play_again_img = pygame.image.load("start_btn.png")
```

```
play_again_img_hover = pygame.image.load("start_hover_btn.png")
play_again_img_clicked = pygame.image.load("start_clicked_btn.png")

quit_img = pygame.image.load("exit_btn.png")
quit_img_hover = pygame.image.load("exit_hover_btn.png")
quit_img_clicked = pygame.image.load("exit_clicked_btn.png")

# Load background image and resize it to screen size
bg_img = pygame.image.load("background.jpg")
bg_img = pygame.transform.scale(bg_img, (1280, 720))

# Load background music
pygame.mixer.music.load("bg_music.wav")
startSound = pygame.mixer.Sound("sfx.wav")
exit_sound = pygame.mixer.Sound("click.wav")

screen_width = 1280
screen_height = 720

screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption('TableTennis')

#define global variable
clicked = False
counter = 0

class button():
```

```
def __init__(self, x, y, normal_img, hover_img, clicked_img):
    self.x = x
    self.y = y
    self.normal_img = normal_img
    self.hover_img = hover_img
    self.clicked_img = clicked_img

def draw_button(self):

    global clicked
    action = False

    #get mouse position
    pos = pygame.mouse.get_pos()

    #create pygame Rect object for the button
    button_rect = self.normal_img.get_rect(topleft = (self.x, self.y))

    #check mouseover and clicked conditions
    if button_rect.collidepoint(pos):
        if pygame.mouse.get_pressed()[0] == 1:
            clicked = True
            screen.blit(self.clicked_img, button_rect)
        elif pygame.mouse.get_pressed()[0] == 0 and clicked == True:
            clicked = False
            action = True
        else:
            screen.blit(self.hover_img, button_rect)
        else:
            screen.blit(self.normal_img, button_rect)

        return action

play_again = button(350, 250, play_again_img, play_again_img_hover,
play_again_img_clicked)
quit = button(700, 250, quit_img, quit_img_hover, quit_img_clicked)
pygame.mixer.music.play(-1)

def table_tennis_game():
    cap = cv2.VideoCapture(0)
    cap.set(3, 1280)
    cap.set(4, 720)

    # Importing all images
    imgBackground = cv2.imread('2DIMAGES/Background.png')
    imgGameOver = cv2.imread('2DIMAGES/gameOver.png')
    imgGameStart = cv2.imread('2DIMAGES/gameStart.png')
    imgBall = cv2.imread('2DIMAGES/ball.png', cv2.IMREAD_UNCHANGED)
    imgBat1 = cv2.imread('2DIMAGES/racket1.png', cv2.IMREAD_UNCHANGED)
    imgBat2 = cv2.imread('2DIMAGES/racket2.png', cv2.IMREAD_UNCHANGED)
    bounce_sound = pygame.mixer.Sound("2DIMAGES/bounce_sound.wav")
    end_sound = pygame.mixer.Sound("click.wav")
    restart_sound = pygame.mixer.Sound("restart.wav")

    # Hand Detector
    detector = HandDetector(detectionCon=0.8, maxHands=2)
```

```
# Variables
ballPos = [100, 100]
speedX = 15
speedY = 15
gameOver = False
score = [0, 0]

gameStart = True
solo = True

while True:
    _, img = cap.read()

    img = cv2.flip(img, 1)

    # Find the hand and its landmarks
    hands, img = detector.findHands(img, flipType=False)  # with draw

    # Overlaying the background image
    img = cv2.addWeighted(img, 0.2, imgBackground, 0.8, 0)

    # Check for hands
    if hands:
        for hand in hands:
            x, y, w, h = hand["bbox"]
            h1, w1, _ = imgBat1.shape
            y1 = y - h1 / 2
```

```
y1 = np.clip(y1, 20, 395)

if solo:
    # Bot controlling the left bat
    bot_y = ballPos[1] - imgBat1.shape[0] // 2
    bot_y = np.clip(bot_y, 20, 415)
    img = cvzone.overlayPNG(img, imgBat1, (59, int(bot_y)))
    if 59 < ballPos[0] < 59 + imgBat1.shape[1] and bot_y < ballPos[1] < bot_y + imgBat1.shape[0]:
        speedX = -speedX
        ballPos[0] += 30
        score[0] += 1
        bounce_sound.play()

    if hand['type'] == "Right":
        img = cvzone.overlayPNG(img, imgBat2, (1195, int(y1)))
        if 1195 - 50 < ballPos[0] < 1195 and y1 < ballPos[1] < y1 + h1:
            speedX = -speedX
            ballPos[0] -= 30
            score[1] += 1
            bounce_sound.play()
else:
    if hand['type'] == "Right":
        img = cvzone.overlayPNG(img, imgBat1, (59, int(y1)))
        if 59 < ballPos[0] < 59 + w1 and y1 < ballPos[1] < y1 + h1:
            speedX = -speedX
            ballPos[0] += 30
            bounce_sound.play()
```

```
                    if hand['type'] == "Left":
                        img = cvzone.overlayPNG(img, imgBat2, (1195, int(y1)))
                        if 1195 - 50 < ballPos[0] < 1195 and y1 < ballPos[1] < y1 + h1:
                            speedX = -speedX
                            ballPos[0] -= 30
                            bounce_sound.play()
        # Game Over
        if solo:
            if ballPos[0] < 40 or ballPos[0] > 1200:
                gameOver = True
        else:
            if ballPos[0] < 40:
                score[1] += 1
            elif ballPos[0] > 1200:
                score[0] += 1
            if ballPos[0] < 40 or ballPos[0] > 1200:
                ballPos = [100, 100]
                speedX = 15
                speedY = 15
            if max(score) == 1:
                gameOver = True

        if gameOver and solo:
            img = imgGameOver
            cv2.putText(img, str(score[0] + score[1]).zfill(2), (565, 375),
cv2.FONT_HERSHEY_COMPLEX,
                        2.5, (200, 0, 200), 5)
        elif gameOver:
            img = imgGameOver
            cv2.putText(img, str(score[0]) + ":" + str(score[1]), (560, 370),
cv2.FONT_HERSHEY_COMPLEX,
                        2.5, (200, 0, 200), 5)

        elif gameStart:
            img = imgGameStart

        # if game not over move the ball
        else:

            # Move the Ball
            if ballPos[1] >= 500 or ballPos[1] <= 10:
                speedY = -speedY

            ballPos[0] += speedX
            ballPos[1] += speedY

            # Draw the ball
            img = cvzone.overlayPNG(img, imgBall, ballPos)

            cv2.putText(img, str(score[0]), (300, 650), cv2.FONT_HERSHEY_COMPLEX,
3, (255, 255, 255), 5)
            cv2.putText(img, str(score[1]), (900, 650), cv2.FONT_HERSHEY_COMPLEX,
3, (255, 255, 255), 5)

        cv2.imshow("TableTennis", img)
        key = cv2.waitKey(1)
        if key == ord('r') or key == ord('R'):
            restart_sound.play()
            ballPos = [100, 100]
            speedX = 15
            speedY = 15
            gameOver = False
            score = [0, 0]
            imgGameOver = cv2.imread('2DIMAGES/gameOver.png')
            gameStart = True
        elif key == ord('1') and gameStart:
            solo = True
            gameStart = False
        elif key == ord('2') and gameStart:
            solo = False
            gameStart = False
        elif key == 27:
            cv2.destroyAllWindows()
            end_sound.play()
            break

run = True
while run:
    screen.blit(bg_img, (0, 0)) # adding background image
    if play_again.draw_button():
        startSound.play()
        table_tennis_game()
        counter = 0
    if quit.draw_button():
        run = False
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
    pygame.display.update()
pygame.quit()
```

## APPENDIX D
## TEAM DETAILS

| S.NO | ROLL NUMBER | TEAM MEMBERS | EMAIL ID | CONTACT NUMBER |
|---|---|---|---|---|
| 1 | 21113004 | Dani N | 21113004@student.hindustanuniv.ac.in | 9791197021 |
| 2 | 21113049 | Dharshan R E | 21113049@student.hindustanuniv.ac.in | 9884204022 |

## 12% Overall Similarity

Top sources found in the following databases:

- 12% Internet database
- Crossref database
- 1% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | |
|---|---|---|
| 1 | coursehero.com<br>Internet | 4% |
| 2 | stackoverflow.com<br>Internet | 2% |
| 3 | researchgate.net<br>Internet | 1% |
| 4 | github.com<br>Internet | <1% |
| 5 | myfik.unisza.edu.my<br>Internet | <1% |
| 6 | umpir.ump.edu.my<br>Internet | <1% |
| 7 | google.co.uk<br>Internet | <1% |
| 8 | codestudyblog.com<br>Internet | <1% |
| 9 | docu.tips<br>Internet | <1% |

| | | |
|---|---|---|
| 10 | sersc.org<br>Internet | <1% |
| 11 | freepatentsonline.com<br>Internet | <1% |
| 12 | lib.dr.iastate.edu<br>Internet | <1% |
| 13 | Heting Liu, Guohong Cao. "Deep Reinforcement Learning-Based Server...<br>Crossref | <1% |
| 14 | docshare.tips<br>Internet | <1% |
| 15 | scholar.uwindsor.ca<br>Internet | <1% |