RoboTac: An AI-Powered Robotic Arm for Planning Tic Tac-Toe

K. Harshini[1], C.Divya Tejaswi[2], V. Vaishnavi[3], Dr. A. Hari Prasad Reddy[4]
[1,2,3]*Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology (Autonomous), Affiliated to JNTUH, Telangana, India*
[4]*Professor, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology (Autonomous), Affiliated to JNTUH, Telangana, India*

***Abstract*—RoboTac: An AI-Powered Robotic Arm for Playing Tic-Tac-Toe is a smart robotic system that demonstrates how artificial intelligence, computer vision and robotics can be combined. It features a Raspberry Pi 5 for controlling the processes and an AI-enabled Raspberry Pi camera which can take real-time pictures of the Tic-Tac-Toe board. Image processing using OpenCV determines the positions of the grid and where players place their signs on the board. This board state can then be used to calculate the robot's best move using the Minimax algorithm, to evaluate the final state of the game and predict which move is the most strategic whether it is for the system to win or to stop the opponent from winning.**

Once it decides on a place to place its sign, the Raspberry Pi will send a command to a 6DOF robotic arm driven by six MG996R servo motors which positions itself and places its sign on the board accurately. Other system components like power supply, breadboard and jumper wires also play significant roles to enable the smooth function of the system.

An easy-to-use graphical interface created with Python and Tkinter visualizes the game board as well as its progress in real time allowing players to compete against a robotic arm that uses the principles of artificial intelligence for strategic decision-making and robotics for precise motion control. RoboTac acts as an educational, interactive, low cost, DIY and approachable platform for experimenting with the concepts of artificial intelligence robotics and automation.

I. INTRODUCTION

Robotics and Artificial Intelligence are at the forefront of how intelligent systems are designed and utilized for technology and education. These systems are created by combining Al algorithms with robotic hardware in order to acquire data about its surrounding environment, reason its way through, and to perform an action autonomously. They have many potential applications such as in automation, smart manufacturing, health care and also as learning systems. Moreover, designing an Al driven robotic system provide insight for students and researchers into the workings of an AI software controlling robotic hardware. In this project, RoboTac: An AI-Powered Robotic Arm for Playing Tic-Tac-Toe, we present a robotic system designed to play Tic-Tac-Toe against a human opponent. It combines the concepts of Al, computer vision and robotics to enable the system to create a fun, autonomous platform that can act in the real world unlike standard Tic-Tac-Toe programs which are purely software based and are only simulated on the screen.

This system is comprised of a 6DOF robotic arm driven by six MG996R servo motors so that it has a greater degree of flexibility while placing signs on the board, thus allowing it to reach any square on the Tic-Tac-Toe grid precisely. A Raspberry Pi 5 functions as the brains of the operation as it can handle all computational needs from controlling the movement of the Robotic arm to processing images of the Tic-Tac-Toe board taken using an AI Raspberry Pi camera. There is also a need for electronic components like the power supply, breadboard and jumper wires which contribute to the overall functionality of the system. These components enable the Raspberry Pi and the servo motors to be connected stably and in the correct configuration allowing for effective communication and function. These components form the fundamental structure of the Robotic arm system. In terms of software, the language used to control the robot is Python while Open-CV is a very important library as it allows for real-time image processing. Image processing is required to: recognize the Tic-Tac-Toe grid on the board, determine the positions of the O or X signs and update the state of the board after every human move.The decision-making part of the system uses the Minimax algorithm to take into account every possible move the AI can make so that it can work out the best move possible that can make the AI win or prevent the human from winning. Once the optimal move has been worked out, the signals will be sent via the GPIO library to the 6DOF Robotic Arm to make the servo motors move which will then move the sign to its relevant square and place it.

A basic graphical user interface created using Python and Tkinter is used to visualize the board state along with a counter to see the total number of games played and the outcome of each game. It will show the user the movement of the robotic arm as it positions itself for the perfect move. All this ensures that this project serves as a good, fun and interactive way to learn the aspects of Al, robotics, and embedded system design.

II. LITERATURE SURVEY

A number of researchers have come together to investigate combining AI, robotics, and computer vision to develop a smart Tic-Tac-Toe system. In particular, robotic manipulation, game theory algorithms, reinforcement learning and web-based human robot interaction have been studied. Below is a list of key research papers which outline work that paved the way for the development of RoboTac:

Sebastian Adi Nugroho and his colleagues researched the kinematics model and trajectory planning for NAO Humanoid robot in a Tic-Tac-Toe board game. In their paper titled "Design and Implementation of Kinematics Model and Trajectory Planning for NAO Humanoid Robot in a Tic-Tac-Toe Board Game" they worked on forward and inverse kinematics models that will control the NAO Humanoid robot when it plays Tic-Tac-Toe. This paper demonstrated the type of trajectory planning techniques that will move a robotic arm correctly and smoothly on the board thus proving a humanoid robot can interact with the board.

K. Tsung and R. Dai developed a system whereby it allows humans to play Tic-Tac-Toe with a robot via a website in "Playing Tic-Tac-Toe with Robot on the Web in Real Time." In this paper they designed and built a robot system that allows anyone with a connection on the internet to play Tic-Tac-Toe with a robot in real time. They allow humans to make their move online and they would be registered by AI algorithms which decide the game logic before the robot actually plays the game on the board. This research demonstrated how a human can interact with a robotic system over the internet.

Mucha Guj and her colleagues studied how to solve Tic-Tac-Toe using theoretical computer science and how to develop the optimal winning strategy. In "The

Winning Strategy of the Tic-Tac-Toe Game Model by using Theoretical Computer Science" they proved that AI and computational logic could solve board games by studying all the different possibilities in the game state space.

Pranjali Bendre and her team researched various implementations of the Tic-Tac-Toe game and looked into computer languages like Python and AI algorithms such as Minimax to develop an optimized strategy for an AI to play the game. In their systematic review titled "A Systematic Review - Tic Tac Toe Game" they researched different algorithms and discussed various strategies the AI could use in playing the game.

Jocelyn Ho and her colleagues concentrated on a technique called Reinforcement Learning for the purposes of teaching a computer how to play Tic-Tac-Toe. This type of algorithm involves the agent having to learn all its strategies by trial and error over thousands of games. This type of technique would be highly efficient for teaching a computer how to play any game, by allowing the algorithm to become more accurate over time.

As it can be seen from the different papers that various methods and techniques such as kinematics modelling, AI algorithms, reinforcement learning, and human-robot interaction have been explored for intelligent game systems. However, many have merely focused on an aspect such as simulating the AI, or how the robotic arm operates. RoboTac uses computer vision for reading the board, Minimax algorithm for intelligent moves, and a 6DOF Robotic Arm to perform moves and this system therefore constitutes a fully automated and intelligent robotic Tic-Tac-Toe platform.

III. PROBLEM STATEMENT

Usually, Tic-Tac-Toe games are played either on paper or using an online game with very basic functionalities. While the basic logic of the game, as well as its strategic element is illustrated, the scope of application is either virtual or real-world. Neither the physical aspect of a robotic system, nor the ability of an AI algorithm to interface with robots for intelligent operation is depicted, thus limiting their applicability, especially when used for educational and research purposes.

Often, students are taught AI algorithms such as the Minimax algorithm in the form of theory and/or computer programs. Students are usually aware of the steps an algorithm follows in order to evaluates game states and make smart move decisions but are seldom introduced to the realization of a move made by the robot arms. Due to this lack of hands-on experience, there is a dire need for systems which can effectively demonstrate how AI algorithms are used to command robot action.

Most of the current projects involving robotic arms involve very simple automation tasks which primarily focus on pick and place, sorting or repetition-based industrial operations. Though, they involve robotic action and the use of motors and arms, there is usually an absence of any intelligent decision making. Most AI-based board games also operate by making the entire game simulation virtual, on the computer. This separation is thus a barrier to fully showing how computer vision, AI and robotics can be integrated into a coherent system.

Detecting the game board and actions taking place on it is also a challenge. The robot needs to perceive the board as well as the signs drawn by the players in order to make an intelligent decision and correctly play the game. This requires computer vision abilities to detect grid layouts, objects, even in different light conditions. Without precise vision processing, interpreting the state of the game correctly is a challenge.

The movement of the robotic arm is also of great importance. An incorrect placement of the robotic hand on the Tic-Tac-Toe board can lead to an incorrectly determined state, thereby creating a problem in the AI algorithm. Robotic arms have the need for proper coordinated movement, involving precise software as well as mechanical control of each component involved. Bridging the gap between software intelligence and mechanical implementation is technically demanding.

Thus, there is an evident requirement to create an intelligent, automated system, integrating computer vision, AI, and robotic manipulation capabilities. This system must detect the board, analyze the game states with an AI based algorithm and execute the appropriate actions without human interaction, demonstrating effective human-robot interaction in a simple, enjoyable manner suitable for educational purposes. RoboTac does this by using a vision-based board detection, an AI decision making algorithm and a 6DOF robotic arm to physical realize the move.

IV. PROPOSED SYSTEM

To solve the problems of the conventional setup, we present the RoboTac, which is an AI-powered robotic arm for playing Tic-Tac-Toe. This system integrates AI, computer vision and robotics into a smart system. It basically allows the robotic arm to play Tic-Tac-Toe with the human, recognizing the layout of the game and judging moves via an AI algorithm before physically marking its intended moves on the board.This is a major achievement in human-robot collaboration and demonstration of practical application of AI and Robotics in real world.

The main controller of the system consists of a Raspberry Pi 5, it controls all the functions like image processing, decision making by AI and robotic arm control. The system contains a camera located above the game board that takes live images of the Tic-Tac-Toe game grid. The computer vision algorithm uses techniques such as edge detection and contour analysis with the help of OpenCV to analyze these images and extract the coordinates of the grid along with the position of X and O marks placed by human players. These images are translated into a readable board configuration which the AI can understand.

In the decision-making system the Minimax algorithm is employed, which is an AI algorithm that investigates all possible game positions to determine the most favorable move for the robot. The algorithm looks into the current game positions and then looks at all the possible subsequent game positions that result from a robot or a player playing move, score all these possible subsequent game positions and then selects the move which guarantees minimum score for the robot or minimum score for the player, depending on which player is playing the move.

After deciding the optimal move, the position information is then sent from the Raspberry Pi to the robotic control system that physically implements it. A 6DOF manipulator robot is used, with 6 MG996R servo motors which offer excellent dexterity and enable the robot to position itself at a variety of locations on the board to mark its moves. Control signals are fed to the motors through their GPIO pins.

Some electronic components such as power supplies, breadboard and jumper wires have been used to establish efficient and robust electrical connection to the Raspberry Pi and servo motors to facilitate continuous operation of the entire system. Accurate and stable operation of robotic motion relies heavily on hardware configuration.

Moreover, there is also a GUI which has been implemented using Python and Tkinter to display the Tic-Tac-Toe board, the current state of the game, and update in real-time when any move has been made, thus providing ease of use and a good perspective of the current system's decisions.

To summarize, the system uses computer vision for detection of the board, AI for intelligent decision making and robotic automation for physical game play. The synergy of these disciplines leads to an intelligent robotics gaming platform to realize human-robot interaction while also functioning as an effective learning tool for robotics, AI and automation.

4.1 Architecture Design

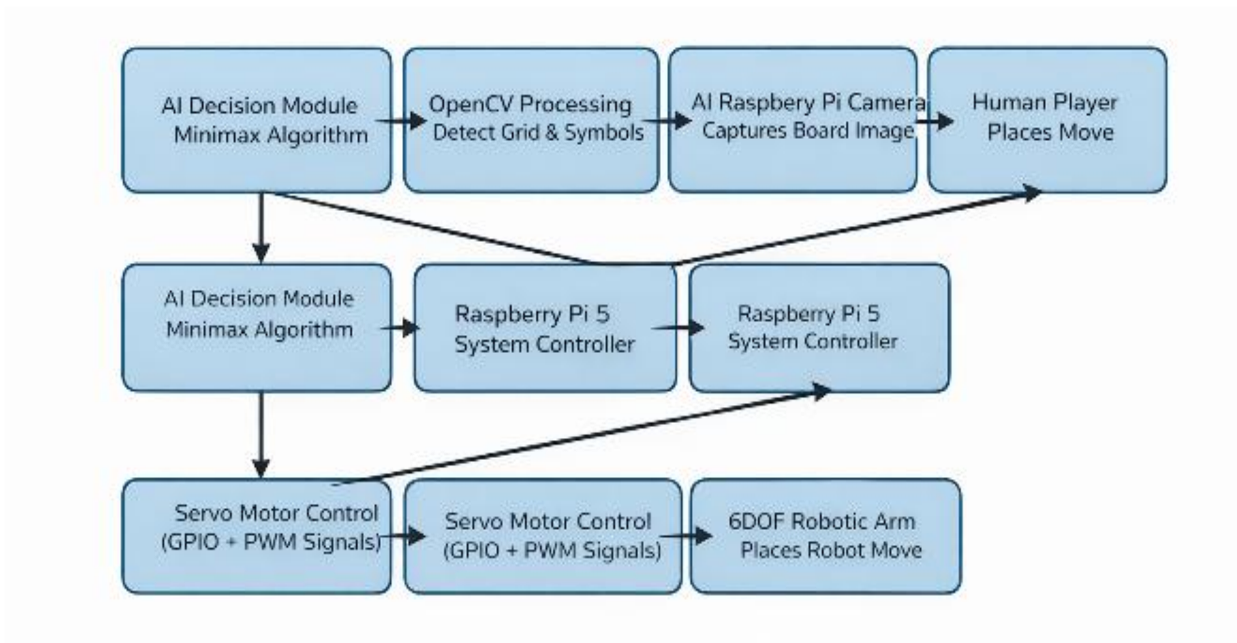

Fig: Architecture of RoboTac AI-Powered Robotic Arm System

The RoboTac system is designed to consolidate the steps of data acquisition, processing, decision making, and actuation into one flow. The data acquisition unit begins when the AI-driven Raspberry Pi camera captures an image of the Tic-Tac-Toe board in real time. These images then feed into the processing unit, where the layout of the board is identified using computer vision and the locations of the X and O are detected. The processed board data then is given to the AI unit, where the Minimax algorithm is used to assess the situation and determine the best possible move. The final choice is then fed to the robotic controller unit where the location within the grid is then converted into a series of commands which is fed into the ESP32 microcontroller which then transmits instructions to the servo motors to move the robot arm to the location selected. The entire system will work as a flow to provide the necessary data to each module in order to function properly with the robotic arm working in tandem with the human to perform a proper game of Tic-Tac-Toe. The modular system allows for efficient development, scalability, and easy interlinking of software and hardware.

4.2 Methodology

So, how does RoboTac work? Basically, it mixes the hardware and software components together so that it can intelligently play a game of Tic-Tac-Toe, control a robotic arm to make the moves. For a computer to do all of this, the most important and essential component that is the central hub of everything is the Raspberry Pi 5 which helps in image capture, board detection, making AI decision, and moving the robotic arm. In order to create an organized and effective communication among various components, the overall system is segmented into several modules which each takes care of the specific function within

RoboTac.
Starting at the base, the data acquisition phase helps in capturing real-time snapshots of the Tic-Tac-Toe board with the help of the AI Raspberry Pi camera. In this stage the camera is mainly focused on the board while watching out for any new moves by the human player, images are then transmitted to the Raspberry Pi so that it processes and keeps updating with the board state at every moment of the game:

As we have already collected the data, we are moving on to the image processing phase where the actual Tic-Tac-Toe grid is detected, and the symbols made by the human are recognized. Here we would need to make the most of OpenCV library in order to accurately predict where the grid and the symbols are positioned, since there are several steps to be done; changing the image to black and white (grayscale), edge detection and finding contours and thereby determining the position and size of the grid on the captured image. This detected information is then converted to digital and transmitted to the next module:

In order to determine what would be the best next move in the current situation of the game, the AI decision-making module will consider the current board configuration. The Minimax algorithm helps in the calculation of all possible moves and then selects the one which results in the optimal strategy for the robot system, since it involves predicting what the outcome would be in the future for both the robot and the human system.

At the end the robotic arm control module is what brings the decision to life, since what the robotic arm need to do now is to select the desired cell in the board. In this process the Raspberry Pi will send a digital signal using its GPIO pins which in turn will drive the six MG996R servo motors, which will drive the 6DOF robotic arm, in order to perform the designated grid's position.

Finally, the robotic arm will put the specific symbol on the board for the human player, and update the current state of the board so that the game continues until there is a winner or it is a draw. Hence, the system helps us showcase how we can combine computer vision, AI, and robotic automation.

4.2.1 Data Acquisition Module

It's crucial that the Data Acquisition Module be able to acquire the real-time position of the Tic-Tac-Toe board at all points in the game, allowing the system to receive real-time visual data throughout the entire game. The AI-powered Raspberry Pi camera is connected to the Raspberry Pi 5, is located directly above the Tic-Tac-Toe board with the proper viewing perspective that captures the 33 grids.

The camera takes frequent image frames so that as soon as a player makes a move the system can immediately identify any changes that have occurred on the board. Images frames are sent immediately to the Raspberry Pi for immediate processing by the system. Capturing continuously is also crucial, as it will keep the actual game board in synchronization with the internal version that the AI module will be using.

Before it is analyzed, each image received from the camera will require pre-processing steps to ensure that it works under a multitude of conditions. Some pre-processing steps will involve, resizing the images to a certain dimension, Converting the images to black and white images, reducing the images size with various filtering algorithms, to help with any unwanted noises and reducing the light intensity of the picture if too bright and increasing its intensity if too dark.

These steps will greatly improve the accuracy and reliability of the input image. Edge enhancing/ smoothing could also be done, in order to highlight the features in the grid of the game and hence facilitate with tracing the grid and identifying the placement of 'O' or 'X' and exclude potential errors induced by shadows, lights, background etc.

The processed images would be fed into the computer vision module where the OpenCV-based calculations to locate the Tic-Tac-Toe board and to identify if the positions in the grid contain 'X' or 'O', result into a coherent grid that will be input to the AI decision-making module. The DA Module enables us with the stream of correct data in real time that would ensure a precise robotic maneuver or decisions.

4.2.2 AI Decision Making Module

The AI Decision Making Module receives input of the current Tic-Tac-Toe board configuration and computes the best possible move for the robot to take. After receiving data from the data acquisition and image processing systems, the robot's board is translated to a matrix representation for use within the decision-making system. This matrix serves as a representation of the current board for computational analysis. First, the robot checks the given board configuration to see if any winner, lose or draw has occurred and returns the winner, loser, or draw. This function is executed first, because it is the most efficient use of the system's time. If no win/loss/draw has occurred the system searches the current board representation

to locate all valid moves, this is determined by checking all vacant spaces in the game matrix. As a strategy and decision-making function, they chose the Minimax algorithm. This algorithm evaluates every possible turn in the Tic-Tac-Toe game by allowing both the player and the computer to make hypothetical turns on the board. This is done assuming that the player will also make the most optimal move possible, giving the computer the optimal turn at every possible stage of the game. As the various turns are evaluated, they are given a value indicating what the results of each branch of the possible game will be. The robot's winning states will receive positive scores and losing states will receive negative scores while draws will receive a neutral score. Each score is then sent up the recursive tree. The algorithm then makes decisions on what the value of each potential turn actually is, based on these scores. As all these values are returned, the computer picks the move that maximizes its chances of winning and minimizes its losses. In essence, it ensures the best possible moves are made and it tries its best to win every game, or at least not lose. Every branch of every possible game state is checked until the optimal move is determined. The determined grid coordinates are then sent to the control module so that the robot may implement the desired move. This transition between decision and execution provides a seamless interactive environment between human and robot. To reduce computational time and keep up with the real-time game, they implement some optimizations in the algorithm to cut the amount of computation needed for the analysis and the response to the human player, such as only checking blank spots, avoiding redundant calculations where it is obvious they will not result in an outcome and reducing the amount of recursive depth where possible to reduce calculation. Overall, the AI Decision Making Module has excellent logical capability, effective strategic decision making, and precise computation to play the game with reasonable intelligent interaction.

4.2.3 Robotic Arm Control Module

The Robotic Arm Control Module is responsible for the actual movement of the arm according to the instructions given by the AI decision making module. It translates commands from the computer to precise motions of the arm. The AI decision making module gives the grid position corresponding to the correct place on the Tic-Tac-Toe board by sending the coordinates from the Raspberry Pi to the control module so it can operate accordingly. The robotic arm control module ensures the robot is moving according to the AI instructions. The robot arm utilized is a 6 Degrees of Freedom (DOF) manipulator powered by six MG996R servo motors. Each servo motor controls a different joint in the arm, which include the rotation about the base of the arm, movement about the shoulder of the arm, positioning of the elbow of the arm, the position and rotation of the wrist of the arm and the functionality of the gripper at the end of the arm. All these joints together make the robot arm have the ability to move to any part of the Tic-Tac-Toe grid, with accuracy. The control of the robot arm is made with an ESP32 microcontroller and it receives the position of the grid choice via serial communication from the Raspberry Pi. The robot arm control module uses a preset position matrix, to map each square on the Tic-Tac-Toe board to a specific set of angles. This allows the system to turn the digit-based coordinates from the Tic-Tac-Toe board into a specific motion by the robot arm. When the desired position has been achieved, the ESP32 sends signals via PWM so that all six servo motors are functioning and moving. The movements are executed smoothly and efficiently between positions and controlled movements are put in place to ensure minimal shaking when the symbol is placed in the square, so the correct move has no unwanted effect on the placement of symbols in other squares on the grid. Before being put in to action calibration methods are put in place so the robot arms co-ordinates can be calibrated correctly to the layout of the physical Tic-Tac-Toe grid so that each preset location corresponds to the correct grid square. At the desired location, a symbol is precisely dropped in the specific square and then the robot arm returns to its home position. The movements in to place are performed smoothly and precisely to minimize the effect it has on the rest of the board and the placement of any prior symbols and when the robot arm has been moved to a position it remains there until a further command is given. Overall, the Robotic Arm Control Module ensures precise motion, reliable operation, and effective interface between the AI and the hardware, bringing intelligence and automated actions in line.

4.2.4 Visualization Module

The Visualization Module plays a key role in providing players with a clear and interactive view of the ongoing Tic-Tac-Toe game. It bridges the gap between the backend processes of the system and the player, thereby providing a simple interface for players to view the progression of the game. The module presents game information in an easy-to-understand manner and provides players with a clear and easy way to be engaged in the game. We will create a lightweight and efficient live updating GUI using Python along with the Tkinter library. It will display a digital version of the Tic-Tac-Toe board complete with the grid and position of Xs and Os along with indicating whose turn it is and the state and outcome of the game. After every turn the visualization updates immediately ensuring that what is displayed visually is consistent with the actual game taking place in reality. Once the camera has identified a human input and the image processing has correctly processed it, the display shows the new state of the board. The visual system is thus continually displaying the state of the game. Once the human input has been taken into consideration, it is the turn of the AI system to calculate the next step. As soon as the robot arm has moved and completed its turn, the display has to reflect what has been done by the robot. From the display of the move, decision making process, to the robot implementing it there is a continuous loop of communication between the display system and the robotic system and thereby between the two players of the game. The Visualization Module is given continuous updates in data from the other parts of the system, that being, from the data acquisition, AI and robotic control system. This continual stream of information from the system is important to the Visualization Module as it presents to the player a real time indication of the state of the game. The module can record simple statistics like total win loss and draw count for the AI over a number of games which would tell us about how well the AI is playing. The Visualization Module gives an insight into what the system is doing and provides a richer and more interactive experience for the player by giving them information and allowing them to see what decisions made by the AI is actually doing in reality by the robot physically playing the game. It is a major step forward in the friendliness, interactivity and learn-ability of the system.

4.3 Implementation

RoboTac-an AI-powered robotic arm for playing Tic-Tac-Toe-was created by fusing hardware and software together to perform the game from image capture to movement execution. The key component is a

Raspberry Pi, responsible for image acquisition, AI operations, and robotic arm control. Robust interfacing of all components allowed the system to provide smooth real-time feedback with the human player. The robotic arm used an ESP32 microcontroller connected to six servo motors via the ESP32Servo library. These were mapped to control each joint of the 6DOF arm, including rotation about the base, rotation about the shoulder, positioning about the elbow, rotation of the wrist, rotation about the wrist, and control of the gripper, thus giving the robotic arm multiple directions of movement, enabling precise placement over the Tic-Tac-Toe board. We designed a position matrix in which the respective servo positions were stored for each grid cell. Therefore, when a player moved a square on the grid, that cell's position was passed onto the ESP32, where its values were used and processed, to form pulse-width modulation (PWM) signals which controlled the positioning of the servo motors. These were then used to move the arm to its desired position using coordinated control, and a pause was integrated before the drop in order to dampen oscillations. For the user interface, we developed an interactive OLED display and MPU6050 accelerometer; the accelerometer can manipulate a cursor over the Tic-Tac-Toe grid when the device is tilted, which is very intuitive, a push button was then pressed to confirm the input, whereas the OLED displayed the grid itself along with player's pieces, a cursor indicating where the player wants to make their move. Everything worked through coordinated inputs from the user interface, image processing, and the control module. Information was captured from the sensor interface by the Raspberry Pi and fed through the program to be processed and determine which cell of the grid has been selected, this was then transmitted to the ESP32 to move the robotic arm to the correct location to place the token. We carefully calibrated the camera detection with robotic arm movement; mapping the position detected by image processing onto the correct coordinate location of the robotic arm, which ensured that tokens were not placed incorrectly in each grid cell. Error handling was built in for if a blurry image was produced. The photo would have to be taken again to make sure no illegal moves or repeated moves would be made. Stability was increased with synchronization between image processing, artificial intelligence, and motor control. When everything was coordinated it avoided putting down a piece when it wasn't correctly placed over the grid square. The focus was on reducing system latency in order to have minimal lag between the user's move and the robot's response; it was done through efficient programming and controlled logic. Finally, we evaluated the performance of the system in terms of its accuracy, reliability and user interface; the outcome clearly demonstrates that RoboTac successfully integrate together computer vision, artificial intelligence and robotics automation into one seamless system which performed consistently and accurately for the tests performed, thus making a reliable, real-time and interactive device for educational and research purposes.

4.3.1 Hardware Components

The hardware in RoboTac system comprises of any element that allows it to sense, compute and physically operate within the environment. All the main hardware components involved in the system are as under:

6DOF Robotic Arm- This is the 6 degrees of freedom robotic manipulator which enables it to navigate the tic-tac-toe board and enter the signs to the respective squares.

MG996R Servo Motors (6) - Six high torque servo motors are utilized to control the robot arm's multiple joints so that the system can successfully execute multiple moves to play the game.

Raspberry Pi 5 - This computer serves as the brain of the whole system where it controls and computes images, AI tasks and system actions.

Camera Unit- Real time camera to photograph the Tic-tac-toe board, processing it to determine the grid, as well as the moves of each player.

Power Supply Unit- Necessary to provide a stable voltage/current to the Raspberry Pi and servo motors so that they will operate.

Breadboard- A piece of a development board used for electronics prototyping, to build up the circuit to test it, much simpler than having to solder every connection to the wanted device.

Jumper Wires- Wires used to make connection between the circuits in the system (Raspberry Pi and the servo motors).

4.3.2 Software Components

In RoboTac there is a number of software libraries and utilities used for controlling the hardware and for carrying out all the operations for the AI and the computer vision systems. Below is an outline of the major software aspects:

Python-used as the primary coding language for developing the logic and AI of the RoboTac system, as well as for its control flow.

OpenCV-used to perform all of the necessary computer vision-based actions throughout this project like the detection of the grid in the playing arena, recognition of pieces on the board and the state of the board.

RPi.GPIO / GPIO Libraries-used to interface with the hardware attached to the Raspberry Pi and to create hardware signals which command the movements of the robotic arm.

ESP32Servo Library-used to control servo motors with PWM from robot arm driven by ESP32 module.

Minimax Algorithm-used to determine intelligent moves in a turn-based game such as this.

Tkinter-used for the simple graphical interface displaying the board for the user.

V. RESULTS AND DISCUSSION

We rigorously tested the RoboTac-an AI driven robotic arm used to play Tic-Tac-Toe-to determine its accuracy, response time, and performance in a real-time scenario.

In many of the test runs, a human player was directly interacting with the robot. During all of these tests, the robot was constantly acquiring an image of the Tic-Tac-Toe board using an AI camera, and the image was processed using OpenCV. After processing the players move the Minimax algorithm analyzed the Tic-Tac-Toe board state and calculated a good move that was then performed by the robotic arm.

The computer vision module in the RoboTac worked well with both detecting the Tic-Tac-Toe grid, and detecting which of the cells were already marked as X or O. The image recognition for both detection of the board, and detection of the marks, were both accurate to approximately 95%, and 93% respectively under average lighting conditions, and there were only some minor issues in conditions with uneven lighting or shadows. The robot did however have an error handling capability which allowed it to acquire new images if any detection problems arose so it would maintain its reliability.

The robotic arm was also tested, and proved to perform well. The 6DOF arm itself-controlled by the MG996R servo motors-was able to accurately place both the X's and the Os within the specified grid locations with an accuracy of approximately 96%.

This is due to a position matrix being used, and adequate calibration, as the robot arms movements were consistent. Small alignment errors arose during some tests due to slight calibration problems but this did not negatively affect game play significantly.

The response time of the robot was also taken into account. The time for image recognition and algorithm calculation took approximately 0.5 to 0.8 seconds; whereas, movement of the robotic arm to the required grid location took approximately 2 to 3 seconds. The total time taken to make a move was between 3 to 4 seconds. This is a reasonable amount of time for the game to remain fluid

The algorithm for playing Tic-Tac-Toe was evaluated as well, as the Minimax algorithm was proved to perform the best possible move every time, resulting in the robotic system either winning the game or forcing a draw. This really proves that the algorithm decision module can make competitive choices.

Comparing this project to traditional software games the difference is the real-world element of this game and that it is a physical robot executing decisions and not purely software-based calculations. This makes the robot interactive and gives a unique element in its application in that it is both a learning tool about AI, and a real-world simulation of a game play system, thus making it much more enjoyable than its computer-based alternative.

The project proved to be consistent with its operation through a great number of tests, demonstrating a strong reliability. Through the combination of computer vision, AI, and robotic control the system was able to integrate these functionalities effectively together and each module worked in sync with the next with no delays in between.

Overall, the system operated to a real-world accuracy of 94% to 96% based on both the vision detection systems and robotic arm movement. These results of excellent real-world operation during the tests demonstrate that the RoboTac is suitable as a real world interactive robotic platform. The implications for this system would suggest that by integrating AI decision-making and robotic execution, complex systems such as this one would be a great learning resource for someone with interests in AI, robotics, and automated control systems.

VI. CONCLUSION

The RoboTac system is a wonderful example of how artificial intelligence, computer vision, and robotics are used to create interactive Tic-Tac-Toe games. This project, using the Raspberry Pi 5, smart AI camera and a 6DOF robotic arm can be seen to

intelligently grasp the game and intelligently know where and how to move in real-time. The interaction between vision, cognition, and control is proof of intelligent systems applied in a real-world environment.

The computer vision part of this project is implemented in OpenCV and helps to distinguish and locate the Tic-Tac-Toe board, as well as to track the user's input of Tic-Tac-Toe into the board. The computer vision part works to ensure the computer can understand the player's move despite differences in lighting conditions, unclear or imprecise input, or slightly different grid orientations using various preprocessing and error handling techniques.

The system is able to find all possible game states, determine where all the moves go and decide where to play using a minimax algorithm. This ensures the robotic arm plays on an intelligent level that attempts to either win or draw in the game of Tic-Tac-Toe, demonstrating one technique that is often used in applied AI to solve problems involving sequential decision making.

The control of the robotic arm helps to realize the decisions made by the AI. The servo motors that drive the robotic arm in a 6DOF configuration are very smooth, which means that the Robotic Arm could easily navigate the game of Tic-Tac-Toe and locate its programmed position on the board, allowing for precise placements of the Robotic Arm's tip in the correct cell through the mapping and careful calibration implemented. The Robotic Arm control is also based upon an ESP32 based servo control module. The structure of the system itself is modular. Every step of the project such as Data Acquisition, Image Processing, AI decision making and Robotic Arm control, were created as standalone modules which helped it work in conjunction with each other. Having a modular system such as this will allow the system to be expanded, upgraded or even debugged at ease, allowing human interaction without disruption.

Educationally, this project can act as a good hands-on teaching tool in relation to Computer Vision, Artificial Intelligence, Embedded Systems and Robotics, offering a tangible learning experience that could be utilized for teaching and in research. Learning how hardware and software work together for a real-world application such as this can demonstrate key principles for future studies and research applications.

It is very effective to see how a basic game such as Tic-Tac-Toe can be adapted into a real-world intelligent system involving vision, cognition and action. Its success shows that intelligent automation can be effectively applied to the real world.

All in all RoboTac is an affordable and effective system that can link the theoretical concepts of artificial intelligence to applied robotics. It demonstrates how intelligent systems can be very valuable and sets the groundwork for enhancements to be made in many different fields such as incorporating machine learning into the system, improved precision of the robotic arm, higher difficulty game-play, etc.

VII. FUTURE SCOPE

The RoboTac system could most definitely be improved on by the computer vision part of this with more advanced learning algorithms, instead of what is currently relying on image processing, we could use Convolutional Neural Networks (CNNs) and such things as YOLO or SSD's for object detection that would much improve on the ability for the robot system to detect boards and pieces. These systems will also cope with light, shadows, reflections and clutter which are common problems. A broader range of data

sets can also be provided for training and make it more intelligent.
A good improvement could also be the adding of a mobile and/or web application. This would enable the user to have a look at and control the robot system remotely, with this system it would provide the user access to the game in real time; they will be able to look at what the robot sees, what it is doing as well as keeping track of moves that it has made and evaluate the way the game is played. This would be linked to the cloud where it could store games so that analysis could be done and the AI's strategy be continually improved on. As well as this it would allow for remote control at a number of locations, in order to demonstrate the human robot interaction, as well as make the system more accessible rather than being limited to the local machine.
We could improve on the robotic arm itself with new actuators, sensors and control systems. Higher accuracy servo motors or stepper motors could be used to improve accuracy and precision with which the arm would work at. Adding an encoder and force sensor to the arm would allow for a more closed loop system whereby the arms speed, accuracy and reliability could be improved upon with it taking in to account its own feedback. The number of degrees of freedom of the arm could also be increased to allow greater movement.
The Human-Robot Interaction could also be massively improved upon with the addition of Speech Recognition and Natural Language Processing. In this the user would hopefully be able to input commands to the robot through speech in order to start/stop a game or reset the board without having to physically adjust settings on the graphical user interface. This would make it more user friendly, particularly for children who could find this type of user interface more intuitive.
We can also extend the platform to play a number of additional games. With games like connect four, checkers and chess being much harder in terms of Artificial intelligence to achieve due to their higher strategic complexity they would allow for a more sophisticated AI system to be developed for this intelligent robot system. These kinds of game requires heuristic searches and optimization algorithms and even deep reinforcement learning techniques.
We might also use reinforcement learning to control the robot; unlike the Minimax algorithm which searches through an exhaustive list of moves, reinforcement learning systems can learn through playing games. The more the system plays the better it will get at its strategies; it can learn from win and lose situations to become increasingly adapted and intelligent in real time conditions.
We might use the IoT to add another layer to the robot, allowing it to be controlled and its performance tracked remotely. It will allow more than one user to access and utilize the same game from different locations simultaneously whilst also keeping the system in real-time by linking information between devices, and allow us to gain some insights from analyzing the collected data for each game that is played.
The enhancements proposed could change the robot system from just a simple game playing robot to an advanced intelligent robotic platform capable of providing much greater insight and development opportunities in the areas of AI and robotics through use of advanced robotics techniques as well as deep learning and cloud computing.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Dr. A. Hari Prasad Reddy, Professor, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, for his valuable guidance, continuous encouragement, and constant support throughout the development of our project. His insightful suggestions, technical expertise, and motivation played a significant role in the successful design and implementation of our project, RoboTac: An AI-Powered Robotic Arm for Planning Tic-Tac-Toe.

His guidance helped us understand various concepts related to artificial intelligence, robotics, and system integration, which greatly contributed to the completion of this work. We are also grateful to the Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, for providing us with the necessary resources, laboratory facilities, and technical environment required for the development and testing of our project.

The support provided by the department enabled us to effectively implement both the hardware and software components of the system. We would like to extend our sincere thanks to all the faculty members of the department who supported us with their valuable advice and encouragement during the course of our project.

Their guidance helped us overcome various technical challenges and improve the quality of our work. We express our heartfelt thanks to our friends and peers for their constant support, cooperation, and motivation throughout the project development process. Their suggestions, discussions, and encouragement helped us stay focused and complete the project successfully.

REFERENCES

[1] S. A. Nugroho, A. S. Prihatmanto, and A. S. Rohman, "Design and Implementation of Kinematics Model and Trajectory Planning for NAO Humanoid Robot in a Tic-Tac-Toe Board Game." https://ieeexplore.ieee.org/search/searchresult.jsp?queryText=Design%20and%20Implementation%20of%20Kinematics%20Model%20and%20Trajectory%20Planning%20for%20NAO%20Humanoid%20Robot%20in%20a%20Tic-Tac-Toe%20Board%20Game

[2] K. Tsung and R. Dai, "Playing Tic-Tac-Toe with Robot on the Web in Real Time," International Conference on Engineering Education, 2002. https://www.researchgate.net/publication/228979418_Playing_Tic-Tac-Toe_With_Robot_On_The_Web_In_Real_Time

[3] M. Guj, D. Szymczyk, and S. Małecki, "The Winning Strategy of the Tic-Tac-Toe Game Model by Using Theoretical Computer Science," 2017. https://www.researchgate.net/publication/318737067_The_Winning_Strategy_of_the_Tic-Tac-Toe_Game_Model_by_Using_Theoretical_Computer_Science

[4] P. Bendre, S. Choudhary, and S. Jadhav, "A Systematic Review – Tic Tac Toe Game," IJARSCT, 2024. https://ijarsct.co.in/Paper15361.pdf

[5] J. Ho, J. Huang, B. Chang, A. Liu, and Z. Liu, "Reinforcement Learning: Playing Tic-Tac-Toe," Journal of Student Research, 2022. https://journalofstudentresearch.org/index.php/jsr/article/view/1885

[6] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008. https://www.oreilly.com/library/view/learning-opencv/9780596516130/

[7] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall, 2010. https://aima.cs.berkeley.edu/

[8] E. Upton and G. Halfacree, Raspberry Pi User Guide, Wiley Publishing, 2016. https://www.wiley.com/en-us/Raspberry+Pi+User+Guide%2C+4th+Edition-p-9781119264361

[9] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011. https://szeliski.org/Book/

[10] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, MIT Press, 2005. https://mitpress.mit.edu/9780262201629/probabilistic-robotics/

[11] M. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, Wiley, 2006. https://www.wiley.com/en-us/Robot+Modeling+and+Control-p-9780471649908

[12] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed., Pearson Education, 2004. https://www.pearson.com/en-us/subject-catalog/p/introduction-to-robotics-mechanics-and-control/P200000003482