P-ISSN : 2685-9106
E-ISSN : 2686-0384

Gantry Robot System Checkers Player

Edward Boris Manurung[1], Geofany[2], Dedi[3]

Swiss German University[1.2], Global Institute of Technology and Business[3]
Indonesia

e-mail: borismanurung@yahoo.co.uk[1], geofanysaleh@gmail.com[2], dediroyadi1@gmail.com[3]

Author Notification
16 April 2023
Final Revised
27 May 2023
Published
30 May 2023

To cite this document:
lManurung, E. B., Geofany, & Dedi. (2023). Gantry Robot System Checkers Player. ADI Journal on Recent Innovation, 5(1Sp), 9–19.
DOI: https://doi.org/10.34306/ajri.v5i1Sp.911

Abstract

Technological developments in the field of moving goods are increasing rapidly. Production processes in fields such as the automotive, medical world, and even for household needs use the Pick and Place system for the production process. It has been proven, for a long time, that research has improved the Pick and Place system for many sectors, especially how this process can be combined with AI. Because of this, there has been a lot of research on how to make robots play games with their own brains. For now, most of the game development for AI is board games such as chess, Checkers, Tic Tac Toe, and other board games. Therefore, this paper will explain how to make a robot that can play board games. By implementing the gantry robot system, the robot will be made to play the Checkers game by integrating the robot with the existing AI checker. The concept of this robot is to use Checkers AI and machine integration with a camera where the robot will compete with humans, the camera will detect it from the game board, and then with the learning process from AI, the robot can determine where to move the pieces and it is hoped that the motor will move according to the AI output command of the Checkers Robot. After tha,t the robot will return to its original position and wait for its turn.

Keywords: Gantry Robot System, Checkers Game, Computer Vision, Artificial Intelligence

1. Introduction

Robots with various kinds of systems is increasing in various sectors of human life. This of course makes humans continue to innovate to make robots and capable systems that have many uses that can be useful for human life in their daily lives[1]. One type of robot that is currently being developed is the Pick and Place type robot. This robot is widely used in various industries that help in human life. This robot system is also very popular because, in its development, this robot can always make maximum results because it has a high level of speed and precision which is very much needed in today's industrial development. Therefore, in the industrial world such as in the production process and in other industrial fields, the Pick and Place Robot is the most widely used. In addition, many students and ordinary people develop their tasks or hobbies by making a robot with a small Pick and Place system[2]. However, it does not stop there. The advancing industrial development requires humans to continue to develop their minds to create robotic systems that can work optimally.

Therefore, the integration of robots with AI or machine learning began. This is what makes the development of AI itself continue to progress and makes humans compete to create AI that can help humans in their daily lives. Currently, the development of AI has also penetrated games. Maybe in the past many people played a game like chess in the human versus human way. But with the development of AI itself, finally, an AI for chess was created so that humans don't have to fight humans anymore[3]. Sometimes the artificial intelligence that is created can be a challenge for players to be able to beat the AI itself. Therefore, the development of AI is increasingly sped up to various types of game boards.

This work will focus on making robots that use the Gantry model where the robot will fight humans. The concept of the game is that when you start the game, the robot that has been integrated with the camera will detect the board and the pieces. Next, it will display on the computer screen in the form of a GUI from the game board. After that, the human will move their pieces first. After the human is finished, the person will press one of the keys on the keyboard which indicates that the human has made a turn. That way, the camera will again detect the board and go straight to the AI that has been integrated into the robot, then the GUI will show which pieces the robot has to move[4]. That way, the signal will be sent to the motor, and move the piece according to the AI. Then the motor will return to its original position and the camera will detect the object again and will show the latest GUI and will wait for its turn to play again.

2. Literature Review

Support satisfactory outcomes of this research, it is underpinned by a literature review that includes: In the study conducted by Smith and Johnson (2020), they designed and implemented a Gantry Robot System for playing checkers using computer vision and artificial intelligence algorithms. The research findings demonstrated that the system is capable of playing checkers automatically with accurate and efficient movements. The research by Brown and Anderson focuses on real-time object detection and recognition for the Gantry Robot System in playing checkers. They utilize advanced computer vision techniques to identify and track checker pieces, enabling precise and reliable movements[5][6].

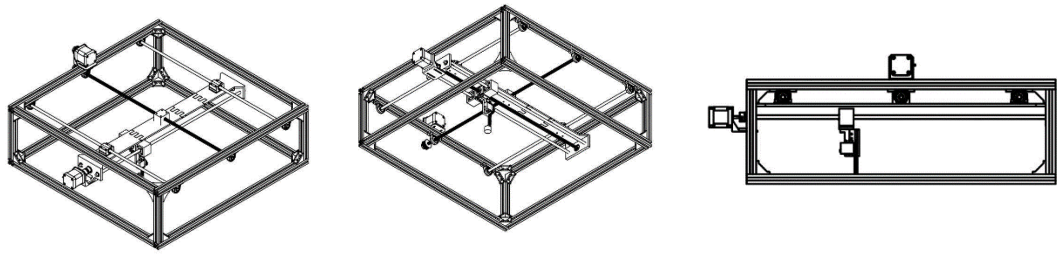
Chen et al. (2022) explore the use of reinforcement learning techniques in decision-making for the Gantry Robot System in playing checkers. They employ deep reinforcement learning algorithms to optimize the robot's strategy and enhance its performance when playing against human opponents[7]. In the study conducted by Zhang et al., the researchers concentrate on the interaction between humans and robots in the Gantry Robot System for playing checkers. The research investigates the use of intuitive interfaces, such as touchscreens and voice commands, to facilitate seamless communication between human players and the robot.

These studies provide valuable insights and serve as a foundation for the present research, contributing to the understanding and advancement of the Gantry Robot System for playing checkers[8][9]. By incorporating the findings from these studies, the current research aims to further enhance the capabilities and performance of the Gantry Robot System in the context of checkers gameplay.

P-ISSN : 2685-9106
E-ISSN : 2686-0384

2.1. Research Method

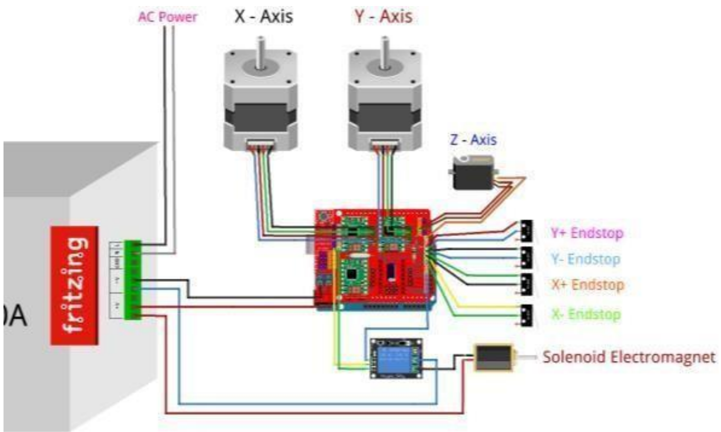
In this work, the type of robot used is the Gantry Robot model. There are several reasons for choosing this robot model, including the type of robot, which is a type of robot that can be said to be very easy to manufacture. In conducting searches on various platforms on the internet, this type of robot is the most widely used compared to other models because it can have many uses such as 3D Print machine, DIY Milling machine, and of course DIY Pick and Place itself[10][11][12]. Regarding the design for this robot prototype, it will take a concept like a 3D Print machine but not too high because the robot will later play Checkers so that the distance between the Z axis will not be far from the workspace which in this work is the Checkers Board. The following Pictures shows the design of the robot's framework with other components[13][14].



(a) (b) (C)

PICTURE 1. Design of Robot’s framework from Top Isometric View (a); Bottom Isometric View (b) and Back View (c).

The following picture is an electrical circuit for all components used in this work. The CNC Shield V3 is connected directly to the Arduino UNO, so each of these components will be connected to the pins on the Arduino UNO.



PICTURE 2. Electrical Circuit

To control every movement of the stepper motor, micro servo, and electromagnet, software called GRBL is used. GRBL itself is free software that functions to control Arduino-based CNC. In its use, CNC Shield V3 recommends using GRBL because the setup process is easy, especially for DIY CNC makers who still want to learn about CNC machine making. This software itself is very easy to learn and is open source so there are many tutorials that can help in its use. Below is the calculation for the X-axis. Leadscrew Torque Formula:

$$T = \frac{Fd_m}{2}\left(\frac{l + \pi\mu d_m}{\pi d_m - \mu l}\right)$$

T = torque lift (N.m)
F = load (N)
d_m = mean diameter
μ = coefficient of friction l = lead
So:

F = m.g

F = 2.2 kg . 10m/s

F = 22 N

$$T = \frac{Fd_m}{2}\left(\frac{l + \pi\mu d_m}{\pi d_m - \mu l}\right)$$

$$T = \frac{22\ \mathrm{N} \times 8\ \mathrm{mm}}{2}\left(\frac{8\ \mathrm{mm} + 3.14\ (0.15)(8\ \mathrm{mm})}{3.14\ \times 8\ \mathrm{mm}\ -\ (0.15)(8\ \mathrm{mm})}\right)$$

T = 0.0433 Nm

For the safety factor is 2, so:

T = 0.0866 Nm………………(1)

For the Y-axis, the Leadscrew specifications used are the same. The Y-axis load itself is about 795 grams.

$$F = m \times g$$
$$F = 7.95\ \mathrm{N}$$

With the same formula, so:

$$T = \frac{7.95\ \mathrm{N} \times 8\ \mathrm{mm}}{2}\left(\frac{8\ \mathrm{mm} + 3.14\ (0.15)(8\ \mathrm{mm})}{3.14\ \times 8\ \mathrm{mm}\ -\ (0.15)(8\ \mathrm{mm})}\right)$$

T = 0.01564 Nm

For the safety factor is 2, so:

T = 0.0312 Nm……………..(2)

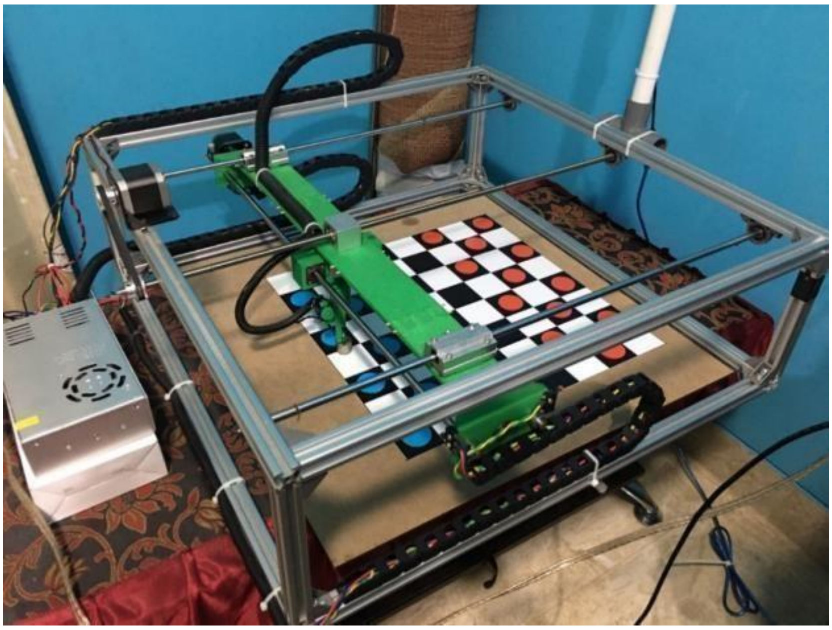
To perform object detection, especially for Boards and Pieces, OpenCV is used. This work, will use Python programming language because the language is easier to understand than C++. The first step in object detection for this work is to detect the Board of Checkers. In the process of detecting the Board from Checkers, there are 3 libraries used, namely OpenCV, NumPy, and Matplotlib. NumPy is a Python library that is a process of numerical computation[15][16]. NumPy is an array data type that can generate 1-dimensional or 2-dimensional data. 1 dimension means that data that is accessed in the form of variables can only be accessed by 1 index, while 2 dimensions means that data which is usually in the form of columns and rows

P-ISSN : 2685-9106
E-ISSN : 2686-0384

can be accessed by 2 indexes. Matplotlib is a data visualization library that can plot graphs for Python in conjunction with NumPy[17]. To start programming, it will use the Visual Studio 2019 application. To be able to detect Pieces on the Board, there are several methods that can be done. The first is to determine the color of each Piece. Colors commonly used generally use RGB (Red, Green, Blue)[18]. However, for OpenCV itself the default color format used is BGR, where the standard color is 24 bits, the first is Blue (8 bits), the second color is Green, and the third is Red. There are other Colour Spaces on OpenCV, namely HSV. HSV is a type of color that can describe the color of the human eye. The value for H is 0-179, the value for S is 0-255, and the value for V is 0-255. To detect red and blue colors on existing Pieces, will use HSV Colour Spaces. For the Checkers AI code itself, there are already many codes on the internet that can be used as material for making this work[19][20]. The available codes are open source so they can be used and even modified for their own purposes. This work itself uses one of the Checkers AI codes available on GitHub[3]. The code itself is a modification of the code from Cobra Draughts which is an open-source Checkers AI algorithm based on the Python programming language. The code uses the minimax algorithm combined with the alpha-beta method which will calculate the best move against human moves.

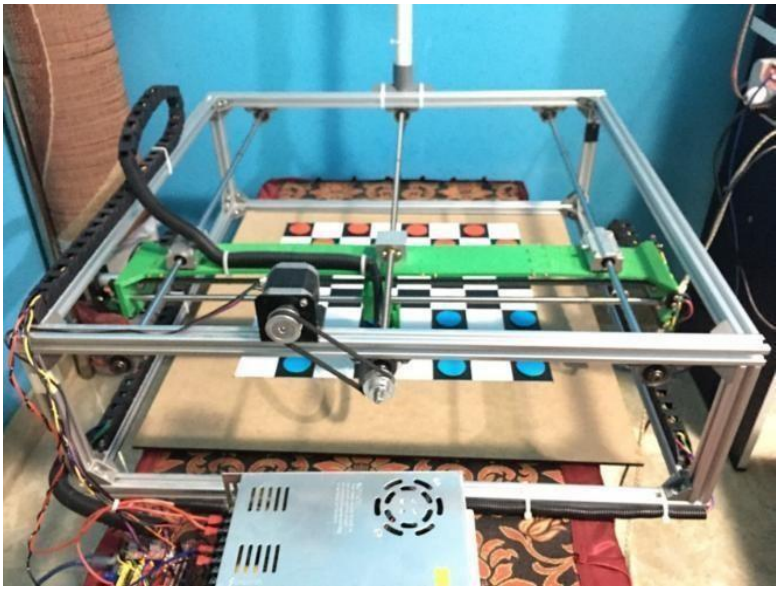
3. Findings

Prototyping for this robot itself can be said to have been completed for quite a long time. As everything has been explained in detail regarding the design, components, and mechanisms to build this robot, it can be said that the Frame for this robot and other supports has worked well to achieve the objectives of this work. Pictures 3 and 4 show the robot prototype after being built together.



PICTURE 3. Robot prototype 1

 P-ISSN : 2685-9106
E-ISSN : 2686-0384



PICTURE 4. Robot Prototype 2

The test results for the two motors for the X and Y axes are said to be as expected, judging from the fairly smooth movement. After that, testing is done by trying to enter certain coordinates (G54 X-117 Y-17), and the motor is tried to run from the homing position several times. The following is a table of the test results.

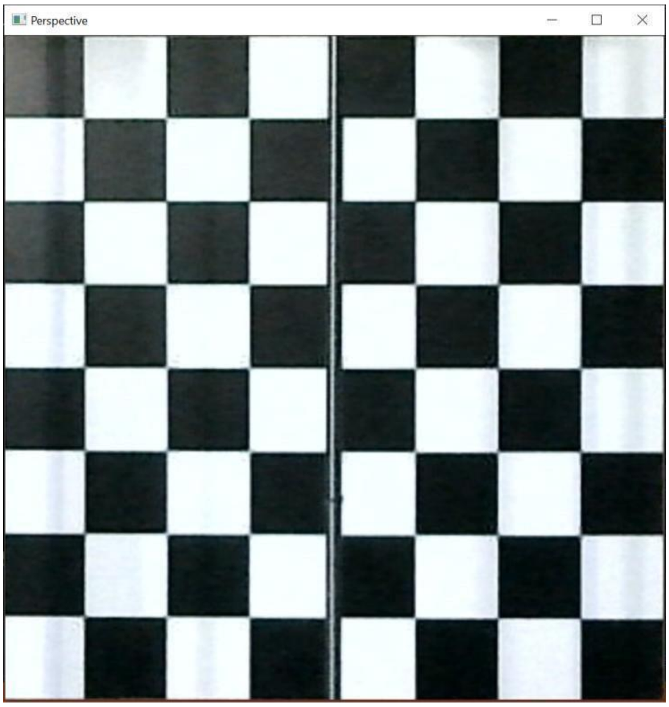
TABLE 1. Test results for motor movement

Trial Stages	**Start Position**	**Final Position**	**Result**	
			Duration	Smoothness Level
1	X0 Y0	X-117 Y-17	07.66 sec	Less Vibration
2	X0 Y0	X-117 Y-17	07.33 sec	More Vibration
3	X0 Y0	X-117 Y-17	07.48 sec	Less Vibration
4	X0 Y0	X-117 Y-17	07.50 sec	Less Vibration
5	X0 Y0	X-117 Y-17	07.45 sec	Less Vibration

After 5 trials that were deemed sufficient to see the smoothness of the motor, the next step was to determine the coordinates of each box on the board, especially for the black squares. The way is to move the motor several times from the homing position to the center of the box on the A1 using the GRBL G54 function. The homing position has coordinates G54 X0 Y0. The homing position for the X-axis is set to the left of the board and close to the robot frame. The Y axis itself, it is set to be in the middle of the X axis 3D Print component. To be more certain, the location of the Homing Position is seen from the end of the Electromagnet. After that move the motor to the box in A3 in order to find out how much the difference between boxes A1 and A3. After knowing the difference, it is easier to determine the coordinates of other boxes. For more details about the coordinates of each box shown in Picture 5. Each of these coordinates will use the G54 method in its operation. For example, G54 X-117 Y-41 means that the X-axis will move to coordinates by -117mm and the Y-axis will move by -41mm.

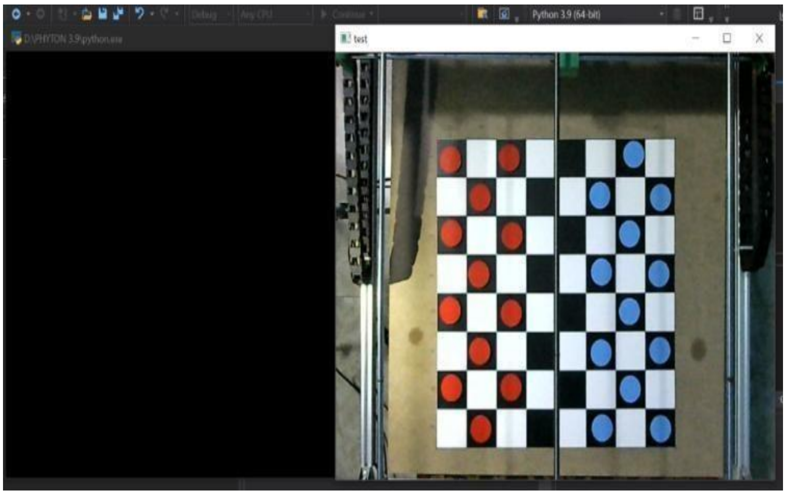
For a Pick and Place System that uses a servo, the results of the testing can be said to be quite good but not perfect as expected. Each Electromagnet ON, Piece can be directly attached. But when it is OFF, there are several times the pieces don't come off right away.

P-ISSN : 2685-9106
E-ISSN : 2686-0384

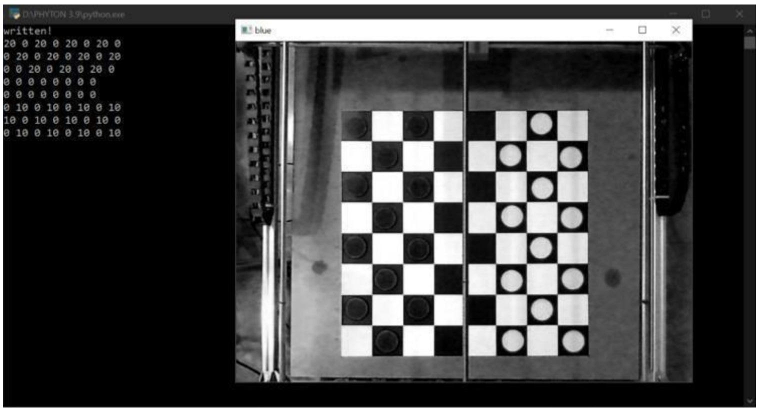


PICTURE 5. Coordinates of black squares

The results of object detection combined with Checkers AI can be said to be quite satisfactory. To be able to run from Object Detection to outputting the desired position by Checkers AI, there are several steps that must be passed. So first when the camera is on, then humans will move their Pieces. After that, the human must press on the keyboard the letter ‘space' which indicates the human has finished moving his Pieces then a new moves, press the letter 'space' on the keyboard again. More details, it will be shown in the following Pictures regarding the results of Object Detection with Checkers AI.

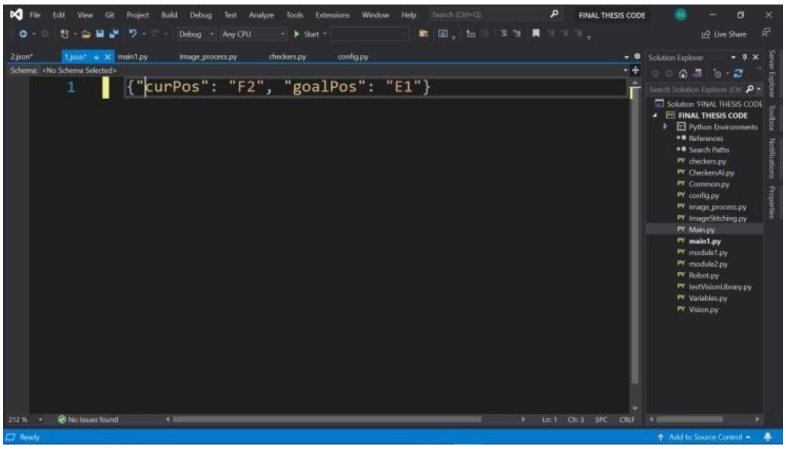


PICTURE 6. The Camera shows the Board



PICTURE 7. Pieces represented in the form of numbers

Picture 8 shows the pieces that have been detected and converted into numbers, where 10 is for blue and 20 is for red.



PICTURE 8. Output in the form of desired start and end positions Checkers AI

The results of motion testing for the X and Y GUI will appear which indicates the Pieces have been updated along with the Board display in numbers. After that, the AI will think and issue an output in the form of the initial position and the position the AI want to go to. Then after the robot axes themselves are very satisfactory. The movement of the two axes is said to be in accordance with the expectation that the motor does not move too fast so as to produce a smooth movement. In addition, the results from Table 1 explain that the longer the duration of the motor movement from the Homing Position to certain coordinates, the smoother the motor movement will be. If it's too fast, there will be a slight vibration. From the test results, the difference in duration even though moving to the same destination can be caused by friction in the Leadscrew.

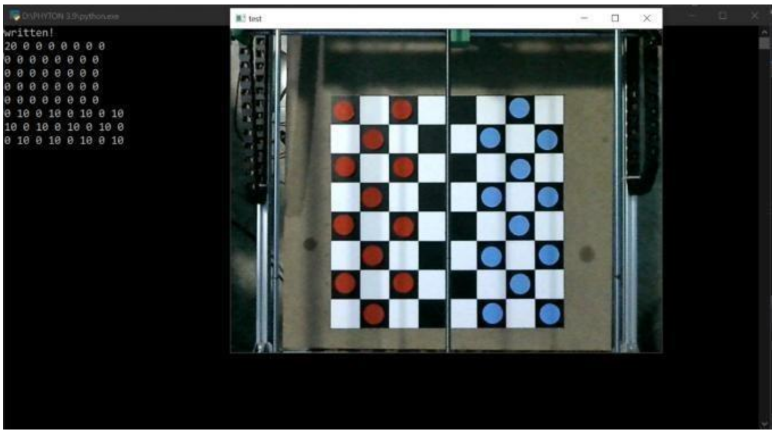
Therefore, it is better if the Leadscrew is often lubricated and also the shaft on both sides so that the movement becomes more stable. That way every move to a certain coordinate on the Board can be precise. For the Z axis that uses a servo, there is a little problem if the servo is used for too long. At the beginning of the servo can move well when going down and up. However, if it's been more than 10 minutes, sometimes the servo goes down quite slowly or can even hit Pieces. In addition, when going up, something like that often happens that causes the servo to sometimes keep rotating even though it's stuck. This kind of thing should be avoided because it can cause damage to the servo and also the 3D Print part which will cause cracks, especially the Pinion part if it continues to rotate even though it has crashed.

The performance of the Electromagnet itself on the Pick and Place System is quite good.

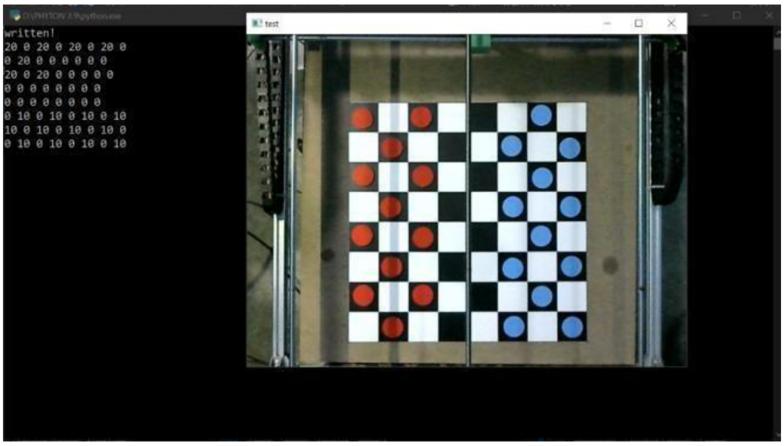
P-ISSN : 2685-9106
E-ISSN : 2686-0384

However, if looked at the test results that have been shown in the previous subchapter from table 4.3.2.1 it can be concluded that every time the Electromagnet is ON, the Pieces are immediately lifted without any time lag. On the other hand, when the Electromagnet is OFF, the more often the Electromagnet is used, the longer the Pieces will fall and don't even fall at all, so the Pieces must be touched first., this, of course, should be avoided. There is a possibility that this can happen because there is still residual current flowing in the Electromagnet so that the pieces sometimes still stick. So, it may be necessary to divide the current so that the current entering the Electromagnet is not full and if the Electromagnet is OFF then the Pieces can fall immediately.

For object detection, the Board produces the desired output. However, there is a slight problem with the detection of Pieces. The obstacle is regarding the lighting of Pieces. To be able to detect all pieces properly, the lighting must also be right. If it is too light or dark it will cause Pieces not to be detected perfectly so the Checkers game itself cannot run. The following pictures will show examples of Pieces that are not detected properly. The way to find out is to look at the Pieces displayed in the form of numbers. Where there are several pieces that should be written as 20 or 10, but because they are not detected, they are written as 0. If the Pieces are not detected properly, then AI Checkers will automatically not run and cause the game to not start. This is because the algorithm from AI Checkers, especially the algorithm regarding Best Move, cannot detect the possible position where the AI will move. The algorithm contains a function regarding the movement between boxes where each box has a different possibility. That way, Checkers AI will error and the game can't be started.



PICTURE 9. Red pieces can't be detected properly because it's too dark



PICTURE 10. Red pieces can't be detected properly because it's too bright

4. Conclusion

Firstly, the selection of the robot model for this project was appropriate, although some minor design changes may be beneficial. Overall, the development of the Gantry Robot prototype was successful and aligned with the desired outcomes. Secondly, the choice of I/O devices to control the Stepper Motor, Micro Servo, and Electromagnet was precise, enabling effective control of these components despite a few shortcomings. Thirdly, the Pick and Place System performed well in its task of retrieving and positioning pieces, although some errors were encountered. These issues can be addressed and improved in future iterations. Furthermore, the utilization of OpenCV in this work was highly suitable, given its extensive features that support the project's success. The availability of comprehensive tutorials and reliable sources further facilitated problem-solving during the process. Moreover, the selection of the camera was appropriate for accurate object detection. However, seamlessly integrating object detection with the robots remains a challenge that requires further refinement. The performance of the AI algorithm utilized in this project was commendable, despite it being a basic version without subsequent development specifically for the game. The AI Checkers system effectively provided desired positions as output. However, it is worth noting that the robot's operation still relies on manual input from humans to determine the destination coordinates. Automating the movement based on the AI's knowledge could be explored for future enhancements. Based on the findings, several recommendations can be made to improve the project further: Consider opting for an Articulated type robot, as it may be more suitable for the workspace if deviating from the Gantry Robot type. If using the Gantry Robot, modify the design, particularly the Leadscrew position, to ensure it does not hinder human players' movement and enhance comfort; If Python is still the preferred programming language, using Raspberry Pi instead of Arduino for I/O devices can simplify motor and component control; It is advisable to accompany object detection, especially for pieces, with appropriate lighting conditions to achieve accurate outputs.; To enable the Robot to make a move, explore methods such as incorporating manual input through push buttons or touch sensors to indicate when a human player has secured their position. Combining this with AI recognition of human signals could allow the Robot to move promptly without relying on humans to enter destination coordinates. This may involve extending the application of pattern recognition for human behavior. By considering these recommendations, the project can be further developed and enhanced, leading to improved performance and functionality.

References

[1] J. Hirvonen, “Utility and Impact of Detecting Clinically Important Bacteria with Small-Scale and Automated Nucleic-Acid Amplification Assays,” 2017.

[2] M. McTaggart *et al.*, “Mechanical design of a cartesian manipulator for warehouse pick and place,” *arXiv Prepr. arXiv1710.00967*, 2017.

[3] “3. B. Ling, ‘quba_robotic_arm/checkers.py at master · Dandoko/quba_robotic_arm,’ Dec. 26, 2020. https://github.com/Dandoko/quba_robotic_ar m/blob/master/computer-vision/checkers.py (accessed Jun. 04, 2021).” [Online]. Available: https://github.com/Geofany17/Robot-Checkers-Player

[4] K. Henke *et al.*, “Expanding the Remote Experiment Set with the 3Axis Portal Physical Model.,” *Int. J. Online Biomed. Eng.*, vol. 16, no. 4, 2022.

[5] C. Poss, *Applications of Object Detection in Industrial Contexts Based on Logistics Robots*. Freie Universitaet Berlin (Germany), 2020.

[6] T. Law, M. Chita-Tegmark, and M. Scheutz, “The interplay between emotional intelligence, trust, and gender in human–robot interaction: A vignette-based study,” *Int. J. Soc. Robot.*, vol. 13, no. 2, pp. 297–309, 2021.

[7] F. Catthoor, T. Basten, N. Zompakis, M. Geilen, and P. G. Kjeldsberg, *System-scenario-based design principles and applications*, vol. 1. Springer, 2020.

[8] N. K. D. Sabrina, D. Pramana, and T. M. Kusuma, “Implementation of Golang and ReactJS in the COVID-19 Vaccination Reservation System,” *ADI J. Recent Innov.*, vol. 5, no. 1, pp. 1–12, 2023.

[9] P. Gia Luan and N. T. Thinh, "Real-time hybrid navigation system-based path planning and obstacle avoidance for mobile robots," *Appl. Sci.*, vol. 10, no. 10, p. 3355, 2020.
[10] L. S. Riza, E. Piantari, E. Junaeti, and I. S. Permana, "Implementation of the Gamification Concept in the Development of a Learning Management System to Improve Students' Cognitive In Basic Programming Subjects Towards a Smart Learning Environment," *ADI J. Recent Innov.*, vol. 5, no. 1, pp. 43–53, 2023.
[11] E. Torabinejad, "Kinematic Analysis of Postural Anticipation and Recovery in Young and Older Adults." Concordia University, 2021.
[12] R. Faza, R. A. Darmawan, and D. T. Setiamanah, "Evaluation of Rebar Waste Rate Calculation Model Utilizing BIM Function: High Rise Building Case Study," *Aptisi Trans. Technopreneursh.*, vol. 5, no. 2, pp. 128–135, 2023.
[13] M. M. Rounaghi, H. Jarrar, and L.-P. Dana, "Implementation of strategic cost management in manufacturing companies: overcoming costs stickiness and increasing corporate sustainability," *Futur. Bus. J.*, vol. 7, no. 1, pp. 1–8, 2021.
[14] R. Muthia, "Structured Data Management for Investigating an Optimum Reactive Distillation Design," *ADI J. Recent Innov.*, vol. 5, no. 1, pp. 34–42, 2023.
[15] M. Askarpour, M. Rossi, and O. Tiryakiler, "Co-simulation of human-robot collaboration: From temporal logic to 3d simulation," *arXiv Prepr. arXiv2007.11737*, 2020.
[16] R. Fetra and T. Pradiani, "The Influence of Price, Facilities, and Service Quality on Re-Staying Interest," *ADI J. Recent Innov.*, vol. 4, no. 2, pp. 184–193, 2023.
[17] G. Odekerken-Schröder, K. Mennens, M. Steins, and D. Mahr, "The service triad: an empirical study of service robots, customers and frontline employees," *J. Serv. Manag.*, vol. 33, no. 2, pp. 246–292, 2021.
[18] M. Bilal and E. Andajani, "Factors Affecting the Intention to Use Roof Solar Panel in Households in Indonesia," *ADI J. Recent Innov.*, vol. 5, no. 1, pp. 25–33, 2023.
[19] F. T. Anaclaudia, D. Pramana, and I. M. A. B. Saputra, "Reactjs and Expressjs Implementation In PMK ITB STIKOM Bali Activity Management," *Aptisi Trans. Technopreneursh.*, vol. 5, no. 3, pp. 1–11, 2023.
[20] N. Rahmanely, "Quality Analysis of Accrual-Based Accounting Implementation in Local Governments (Comparative of Padang Pariaman Regency and Solok City)," *Aptisi Trans. Technopreneursh.*, vol. 5, no. 1, pp. 53–63, 2023.