Real-Time Domino Recognition and Motion Planning for Vision-Guided Robotic Arm Manipulation

Aryaduta Putra Perkasa[1], Joel Napitupulu[2], Moses Paliyama[3]

Computer Engineering Departement,Faculty of Intelligent Electrical and Informatics Technology, Sepuluh Nopember Institute Of Technology
Email: [1]5024201077@mhs.its.ac.id, [2]5024201017@mhs.its.ac.id, [3]5024201048@mhs.its.ac.id

Abstract

A robotic arm is a type of mechanical arm with similar functions to a human arm; it's typically programmable and used for a variety of industrial applications, most commonly for 'pick and place' tasks. These machines are vital for repetitive, precise tasks, and their use has expanded beyond industrial settings into realms like entertainment and education. However, when engaging in a complex task such as playing domino games, robotic arms encounter unique challenges. They must be able to recognize and interpret the cards, understand their positions and the game state, and respond in real-time. Traditional robotic systems struggle with these tasks as they lack the necessary sensory input and processing capabilities to perform such dynamic and nuanced activities. Previous research has shown that integrating computer vision with robotic arms can overcome these limitations. Computer vision systems enable robots to interpret visual data, allowing them to perform tasks that require a higher level of perception and interaction with their environment, such as identifying objects, their locations, and their movements. This study employs the YOLO (You Only Look Once) computer vision algorithm to endow a robotic arm with the ability to play dominoes. YOLOv8, the latest iteration, uses a convolutional neural network (CNN) to detect objects within an image by predicting bounding boxes and class probabilities. By integrating YOLOv8 into the robotic system, we address the challenges of card recognition, position determination, and real-time response. The system divides the visual field into a grid, with each cell responsible for predicting the presence and class of objects within it. This allows the Dobot Magician robotic arm to accurately identify domino cards, their orientation, and position, enabling it to interact appropriately during the game. The methodology encompasses calibrating the Dobot arm to align with the visual input from cameras, interpreting this input through YOLOv8 to determine the precise location and orientation of the domino cards, and executing the necessary movements to engage in the game effectively. This research not only demonstrates the practical application of YOLO-enhanced computer vision in robotic arms for entertainment and educational purposes but also sets a precedent for future advancements in human-machine interaction

Key Words: Dobot, Algorithm, Robotic, YoloV8, Technology

1. INTRODUCTION

In the modern era of technology, robotics has made remarkable strides, transcending the confines of traditional applications and exploring new frontiers in entertainment and education. The development of the Dobot Magician stands as a testament to this progress. It is a multifaceted robotic arm that has been engineered to execute a plethora of tasks with exceptional precision across three axes. With the facility to interchange its end effectors, such as grippers or pens, the Dobot Magician is adept at performing a variety of functions, ranging from object manipulation and robotic programming to 3D printing and automated processing. The ingenuity of the Dobot Magician is further accentuated by its incorporation of robot arm kinematics. The mathematical analysis that enables the robot to position its arm to a specified location and orientation. Kinematics is crucial to robotics engineering, providing the foundational equations and algorithms that direct the robot's movements. It is divided into two primary branches: forward kinematics, which computes the position of the end-effector based on known joint angles, and inverse kinematics, which involves calculating the joint angles to achieve a desired end-effector position.

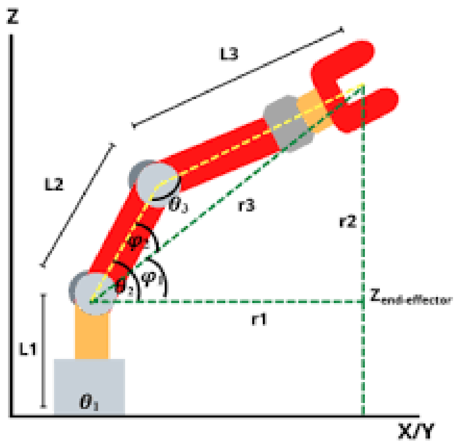

Figure 1.1 Arm Robot Kinematics

In the specific scenario of the Dobot Magician playing domino cards, kinematics is applied to precisely orchestrate the movements required for the robotic arm to interact with the cards. This involves meticulously determining the joint angles and motions to accurately position the end-effector over the table, whether for picking up, placing, or organizing the cards in a particular arrangement. The kinematic algorithms are designed to ensure that the robot operates efficiently and smoothly, translating two-dimensional camera inputs into three-dimensional, precise actions. This technological innovation exemplifies the significant milestone in robotics evolution, where machines like the Dobot are now engaged in activities that demand not only strategic thinking but also precise coordination, far beyond simple mechanical functions. It marks a pivotal moment where robots are increasingly seen as partners in cognitive tasks, signaling a future where robotics intersects with human experiences to enhance and expand the realm of possibilities.

This activity aims to explain how Dobot, with the help of advanced sensors and revolutionary algorithms like YOLOv8, can be set up to participate in domino games. YOLOv8 is an object detection algorithm that uses a convolutional neural network (CNN) to predict bounding boxes and class probabilities for each object within an image. YOLO operates by dividing the image into an SxS grid. Each grid cell is responsible for predicting the objects within it, including bounding boxes and class probabilities. The bounding box is a rectangular box that defines the location and size of the object. The class probability is a value that indicates the likelihood of a grid cell containing an object of a certain class. YOLOv8's role in the project is to identify and locate domino cards within images. It does this by analyzing the visual data, recognizing the distinct patterns on the dominoes, and delineating them with bounding boxes. The algorithm assigns a class probability to each detected domino, indicating the confidence level of the detection. This enables the system to accurately count and categorize the domino cards based on their number combinations, which is crucial for applications that require precise identification and sorting of these elements in various gaming or analytical scenarios.

This project demonstrates how the robot can identify and respond to the movements and strategies of opponents, creating an engaging and interactive playing experience. Furthermore, this research explores the potential for similar applications in the contexts of education and entertainment, underscoring how such advanced technological integration can expand the scope of robotics and pave the way for closer human-machine interactions in the future. This introduction will discuss the key aspects of Dobot in the context of playing domino cards, providing a foundation for further understanding of how this technology can change the way we interact with robots in our daily lives.

2. SYSTEM DESIGN AND METHOD

2.1 System Overview

This integrated system is designed to operate in three primary stages: input, processing, and output, ensuring a seamless transition from visual data to robotic action. Initially, the input stage involves capturing visual data through cameras employing two different perspectives. In the processing stage, this visual data is analyzed by the program, which employs sophisticated algorithms to accurately identify and determine the positions of domino cards on a table. The software interprets their orientation and spatial placement, essential for the subsequent robotic interaction.

The output stage is where the Dobot robotic arm comes into play. It is equipped with servos that enable movement along the x, y, and z axes with high accuracy. This movement requires meticulous calibration to ensure the robotic arm's actions are in sync with the camera's field of view. The main code are designed for dynamic recognition, adapting to changes within the camera's visual field in real-time. This ensures that the robotic arm can respond to live updates of the domino cards' positions. As a result, the Dobot arm, directed by these updated coordinates, interacts with the domino cards using an end effector that could be a gripper or other tools, depending on the task at hand. The precision with which the system operates is critical, as the robot coordinates must align with the two-dimensional image data to perform accurate three-dimensional movements. This allows the system not only to pick up and arrange domino cards but also to adapt to a variety of complex tasks across different industries. The system exemplifies the potential of integrating computer vision with robotic automation, paving the way for advanced applications that require real-time responsiveness and high precision.

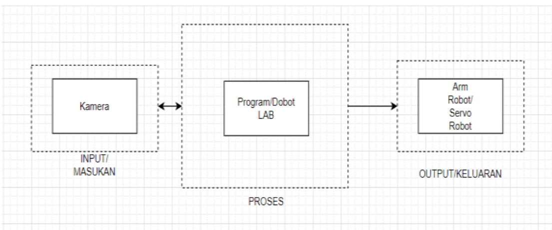

Figure 2 1 System Diagram

2.2 System Planning

The system design begins with determining the precise position so that the perspective of the robot can be easily calibrated with the perspective of the camera. The placement of the cameras is divided into two: one camera to check the available cards, and another to check the cards placed on the table by other players. As seen in Figure 2.2, the position of the camera for available cards is located beside the robot. This is due to the limited reach of the robot's arm, necessitating a placement that provides an effective impact on the activity. The figure also shows that the placement of the second camera is in front of the robot. This is because the reach of the robot and the camera taken into account results in the position as shown in the figure..

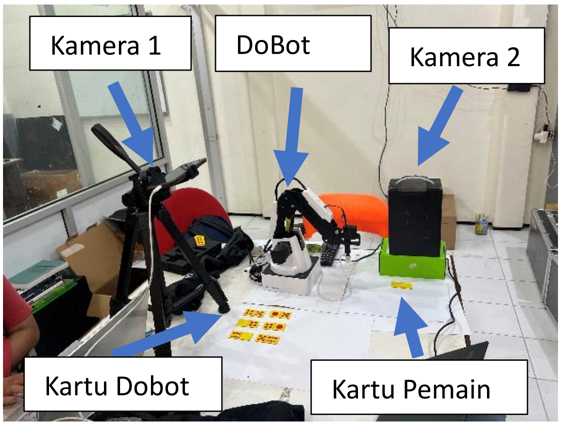

Figure 2 2 Camera and Robot Positioning

2.3 System Workflow

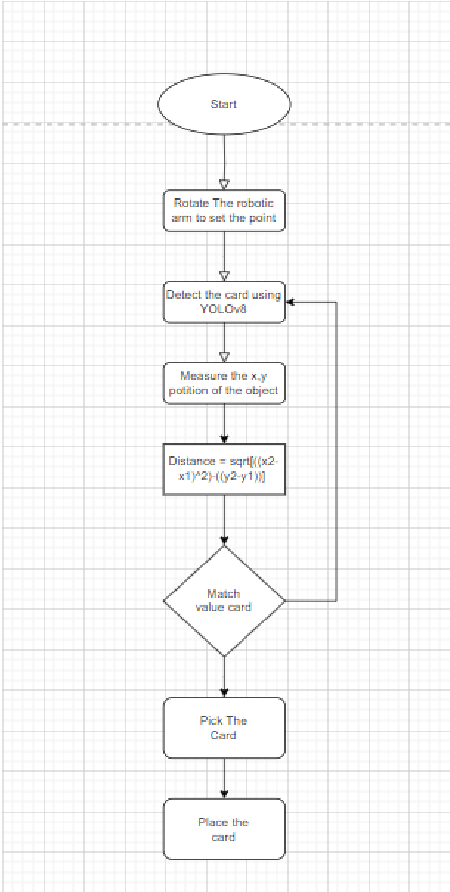

Figure 2.3 System Flow

Following the initial detection and positioning, the workflow incorporates a critical analysis phase where the robotic arm's onboard computing system matches the value of the identified card with the necessary game strategy. This decision-making step is crucial as it determines the robotic arm's next course of action within the context of the domino game's rules and current state. Once a match is established, signifying that the correct card has been identified and its role within the game sequence is understood, the robotic arm proceeds to the 'Pick The Card' stage. This involves the arm engaging its end effector, which use suction apparatus, to securely grasp the domino card. Precision at this stage is paramount to prevent disrupting the layout of the game or damaging the cards.

In this advanced robotic system, following the detection of a domino card using the YOLOv8 algorithm, the robotic arm computes the distance to the object using the distance formula

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

This calculation is critical for several reasons. It enables the arm to accurately gauge its position relative to the card, ensuring positioning accuracy and precise movement planning. Understanding the distance is also key to collision avoidance, as it allows the robot to navigate its environment safely, avoiding any unintended interactions with other cards or objects on the table. Furthermore, this distance measurement informs the robotic arm how to adjust its gripper, both in terms of the opening necessary to approach the card and the force required to grasp the card securely without causing slippage or damage. Efficient movement to the target card is also energy-efficient, reducing unnecessary wear on the robotic arm's servos and joints, which is essential for the longevity and maintenance of the system. Once the robotic arm calculates the distance and confirms the card's match value, it can execute the pick-and-place sequence. This involves the arm moving to the card's location, picking it up, and then placing it in the designated spot on the playing field. This sequence of actions demonstrates the robotic arm's ability to translate two-dimensional image recognition into three-dimensional physical movements, allowing it to interact effectively in the strategic game of dominoes.

The 'Place the card' action is the culmination of the arm's task sequence, requiring the robot to navigate to a specified location and accurately release the card, effectively executing the intended move in the domino game. This placement must be performed with delicate precision to ensure that the card is positioned correctly in relation to the other cards already in play. This integrated workflow not only showcases the robotic arm's capacity to perform complex tasks that require agility and cognitive processing but also exemplifies the advanced level of autonomy that modern robotics can achieve. By leveraging sophisticated algorithms like YOLOv8 for real-time image recognition and precise kinematic calculations for movement, the robotic arm becomes more than a tool—it becomes an active participant capable of engaging in strategic play.

3. IMPLEMENTATION

3.1 Data Training

The initial phase involves training the robot on how to detect cards. In this context, it is essential to develop the capability for the robot to identify cards and extract the coordinates of the midpoint of the bounding box. The example in this repository utilizes the YOLOv8 model with the "detect-model.pt" format. However, if a different model type is preferred, it can be configured within the code. If there is a desire to utilize a detection algorithm other than YOLOv8, it is possible, provided that it can extract detection classes and the coordinates of the bounding box's midpoint.

Essentially, the program needs to be designed to accept detection data in the form of arrays representing classes, x coordinates, and y coordinates. This flexibility allows for the accommodation of various detection algorithms, facilitating the adaptation of the program to different models as long as it can process and utilize the required information.

It is crucial to emphasize that the chosen detection model should be adequately trained with a dataset encompassing sufficient variations of the cards to be detected. This ensures that the robot can recognize and process a diverse range of card types with high accuracy.

```
detected_card = ["2", "1", ...]
x_card = [104.98, 93.61, ...]
y_card= [292.72, 249.02, ...]
```

Figure 3.1 Detection Result Variable

The second part involves calibration, where the goal is for the robot to determine coordinates for card retrieval. It's important to note that the coordinates detected by the camera may not align with the internal coordinates of the robot. For example, if the camera detects coordinates x, y = (10,25), the actual internal coordinates of the robot might be (39,75). Hence, calibration becomes necessary. To calibrate, a collection of points needs to be gathered based on the camera coordinates and the range of the robot's end effector.

In this case, let's consider a scenario where a magician robot is provided with six cards that can be grasped by the end effector. Therefore, data coordinates for these six cards need to be collected. Since each card, in this instance, contains two object classes (assuming one card consists of two distinct objects), there will be a total of twelve data points for the six cards.

The calibration process involves aligning the camera-detected coordinates with the robot's internal coordinates, considering the unique characteristics of the robot's end effector. By collecting and correlating data points for each card, the calibration ensures accurate mapping between the camera's perspective and the robot's internal coordinate system. This step is crucial for precise and reliable card retrieval by the robot's end effector based on the detected coordinates.

3.2 System Calibration

Figure 3. 2 Camera Perspective

Based on the previous explanation, the perspectives of the camera and the robot are indeed different, as each has its unique coordinate system. The camera views an object from a certain angle and distance, while the robot must interact with that object in a real three-dimensional space. This difference presents a unique challenge in robotics systems, where the position of the object detected by the camera must be reinterpreted to align with the robot's coordinate system. To synchronize the camera's perspective with the Dobot's coordinate system, system calibration is required. This calibration process involves determining reference points or 'baselines' within the camera's view, whose positions are known in the real space where the Dobot operates. By using these reference points, the system can calculate the necessary transformation to convert the two-dimensional coordinates seen by the camera into three-dimensional coordinates that can be understood and operated by the robot.

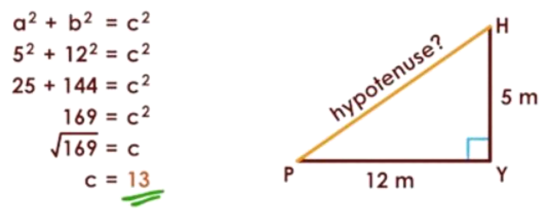

Figure 3. 3 Pythagorean theorem

Let's take an example where one card has two detection classes [1,2] with two detected object coordinates (x,y) = [(214,261), (214, 313)]. In the code above, we take a try with parameters x,y = (215,263). It's important to note that sometimes the coordinates detected by the bounding box during play do not correspond with the calibration coordinates. For instance, if we place a card at position (12,24) based on object detection, in reality, during play, the bounding box coordinates might shift slightly to (13,26). If we use standard IF-ELSE statements, we cannot perform the robot's internal coordinate conversion because the coordinates will not match. Therefore, we take an approximation approach. If there is a calibration coordinate (12,24), then we will approximate which point is closest to the calibration coordinate (12,24). Now, if there are two points (13,26) and (20,29), we can easily determine that the first point (13,26) is the closest to the calibration coordinate (12,24). To do this, we will use the Pythagorean theorem (hypotenuse to find the distance between two points (calibration point and the camera point during play)) and we will store it in an array for each calibration point we have. Add ELSE-IF statements according to the points owned.

4. CONCLUSION

The camera integration in the Dobot arm is a significant step in improving the ability and robotic intelligence to perform card capture tasks with a high level of precision and efficiency. The presence of two cameras used, one to monitor the table and the other attached to the Dobot arm itself, opens up opportunities to achieve a higher level of accuracy. In this context, the use of computer vision technology through cameras not only expands the robot's ability to work automatically, but also provides the basis for a substantial increase in functionality. One of the main benefits of using a camera on the table is its ability to provide visual information to the Dobot arm. This camera can detect the position, orientation, and other attributes of the cards placed on the table. Thus, the Dobot arm can process this visual data to plan the movement and card capture strategy with high accuracy. This not only allows the robot to pick up cards efficiently but also to handle variations in card positions and conditions carefully.

On the other hand, the camera installed on the robot arm opens up opportunities for real-time monitoring during the card capture process. This allows the robot to constantly update information about the card being retrieved, o that it can adjust its movement and response to changing conditions over time. This two-camera integration creates a feedback loop system that allows the Dobot arm to learn and adapt during task execution.

REFERENCE

i. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” computational intelligence and neuroscience, vol. 2018, pp. 1–13, 2018.

ii. A. Sophokleous, P. Christodoulou, L. Doitsidis and S. A. Chatzichristofis, “Computer Vision Meets Educational Robotics,” Electronics, vol. 10, pp. 1–24, 2021.

iii. A. R. A. Tahtawi, M. Agni and T. D. Hendrawati, “Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics,” Journal of Robotics and Control, vol. 2, pp. 469–475, 2021.

iv. C. A. G. Gutiérrez, J. R. Reséndiz, J. D. M. Santibáñez and G. M. Bobadilla, “A model and simulation of a five-degree-of-freedom robotic arm for mechatronic courses,” IEEE Latin America Transactions, vol. 12, pp. 78–86, 2014.

v. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” computational intelligence and neuroscience, vol. 2018, pp. 1–13, 2018.

vi. C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu and T.-Y. Liu, “A hybrid switched reactive-based visual servo control of 5-DOF robot manipulators for pick-and-place tasks,” IEEE Systems Journal, vol. 9, pp. 119–130, 2015.

vii. O. Glaufe, O. Gladstone, E. Ciro, C. A. T. Carvalho, and L. José, “Development of robotic arm control system using computational vision,” IEEE Latin America Transactions, vol. 17, pp. 1259–1267, 2019.