Developing a Mobile Robot to Detect and Collect an Object based on CNN and IMU Sensor

Iffah Syafiqah Abdul Ghani[1], Nyayu Latifah Husni[2], Selamat Muslimin[2], Ekawati Prihatini[2], Grzegorz Królczyk[3], Wahyu Caesarendra[1,3*]

[1] Faculty of Integrated Technologies, Universiti Brunei Darussalam, Bandar Seri Begawan, Brunei
[2]Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Palembang, Indonesia
[3]Faculty of Mechanical Engineering, Opole University of Technology, Opole, Poland
wahyu.caesarendra@ubd.edu.bn

Abstract. Object detection and tracking technologies in robotics have gained prominence in industrial and warehouse processes, offering increased efficiency, safety, and productivity. Robots equipped with these technologies excel in replacing humans in repetitive tasks, allowing individuals to focus on more complex and creative endeavours. However, challenges such as real-time processing, sensor limitations, and high costs hinder widespread adoption. This study addresses these challenges by proposing the use of the Raspberry Pi 3+ as a cost-effective solution for robotics applications, specifically in object detection and tracking. The Raspberry Pi, known for its affordability and adaptability, serves as a low-cost, single-board computer capable of interfacing with various sensors and cameras. In addition, its application in robotics relies on striking a balance between cost efficiency and computational power. This study aims to develop an advanced robotic mobile system employing Convolutional Neural Networks (CNN), OpenCV, and an Inertial Measurement Unit (IMU) sensor. The proposed system leverages CNN for precise object detection and localization, OpenCV for real-time image processing and bounding box generation, and an IMU sensor for enhanced spatial awareness and navigation.

Keywords: CNN, Image classification and localization, IMU sensor, Mobile robot, Raspberry Pi.

1 INTRODUCTION

In past years, the integration of object detection and tracking technologies in robotics is gaining more popularity in various industrial and warehouse processes as these technologies continue to advance [1]. Their applications are expected to expand further, offering even more benefits to businesses and industries. These robots are often used to replace humans in repetitive tasks. This can free up individuals from repetitive work, allowing them to focus on more creative or complex tasks. In essence, developing robots for object collection from one point to other serves to increase efficiency, improve safety, reduce costs, and enhance productivity as well as consistency [2].

N. L. Husni et al. (eds.), *Proceedings of the 8th FIRST 2024 International Conference on Global Innovations (FIRST-ESCSI 2024)*, Advances in Engineering Research 261,
https://doi.org/10.2991/978-94-6463-678-9_42

Robots can work without the need for breaks, rest, or sleep. This non-stop operation ensures that tasks are performed consistently and without interruptions, leading to higher productivity [3]. In addition, robots excel at repetitive tasks, consistently replicating the same motions and actions without variation. This repeatability is crucial in tasks that require a consistent level of quality [4]. When it comes to the safety and reliability of robots, they can work in hazardous environments such as nuclear operations, reducing the risk to human workers. This ensures a safer working environment and minimizes disruptions due to workplace accidents [5] [6].

As technology continues to advance, robots are being used in an increasingly wide range of applications, from logistics and manufacturing to agriculture and healthcare [7]. However, the integration of object detection and tracking technologies in robotics can be challenging for several reasons such as real-time processing and sensor limitations as well as it is high cost. Many robotic applications require real-time object detection and tracking. Achieving real-time performance when processing data from sensors and cameras with limited computational resources is a formidable challenge. This task necessitates a combination of hardware enhancements, algorithmic optimizations, and strategic engineering decisions to strike a balance between computational demands and operational requirements [8].

At the core of every mobile robot lies its design, which intricately weaves together material selection, motion mechanisms, fabrication methods, and navigation systems. Material selection stands as the cornerstone, dictating the fundamental attributes of the robot's structure. Motion mechanisms play a pivotal role in facilitating agile navigation through varied terrains. Yunardi et al. [9] delve into the realm of omnidirectional wheels and tracks, showcasing their efficacy in enabling precise and efficient movement through cluttered environments. Fabrication methods, such as additive manufacturing and CNC machining, translate design concepts into physical reality, offering rapid prototyping capabilities and uncompromising precision [10]. Finally, navigation systems serve as the cognitive compass of the mobile robot, guiding its movements through complex environments. Sheu et al. [11] explore sensor integration for autonomous navigation, emphasizing the synergistic interplay between Light detection and ranging (lidar), cameras, and ultrasonic sensors to perceive surroundings with clarity and enable obstacle avoidance.

The seamless integration of processor and sensor technologies forms the bedrock upon which autonomy and intelligence are built in mobile robotics. NVIDIA Jetson processors provide a paradigm-shifting solution in real-time computation and decision-making [12]. Sensor integration plays a linchpin role in enabling comprehensive environmental perception, facilitating the robot's understanding of its surroundings with clarity [13]. Lidar technology, in particular, offers three-dimensional mapping capabilities with unparalleled accuracy and range [14].

The integration of deep learning, particularly CNN, heralds a new era in mobile robot capabilities, endowing them with unprecedented perceptual acuity and cognitive prowess [15]. OpenCV libraries emerge as a cornerstone in enhancing robot perceptual and interaction capabilities [16]. Moreover, advancements in deep learning architectures, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, further expand robot perception and adaptation capabilities [17].

This study utilize the Raspberry Pi 3+ as a cost-effective solution for robotics applications, particularly in object detection and tracking. Known for its affordability and adaptability, the Raspberry Pi serves as a low-cost, single-board computer capable of interfacing seamlessly with various sensors and cameras. This makes it an appealing choice for budget-constrained projects or educational initiatives, where its general-purpose computing capabilities and extensive community support facilitate effective teaching of robotics and programming. While the Raspberry Pi is widely used in educational and prototyping contexts, its application in robotics hinges on balancing cost efficiency and computational prowess. In scenarios characterized by small-scale deployments and moderate computational demands, the Raspberry Pi is a pragmatic choice for tasks such as object detection and tracking [18].

Motivated by these factors, this paper aims to develop a mobile robot capable of detecting and collecting objects through the integration of Convolutional Neural Networks (CNNs), OpenCV, and various sensors, including an Inertial Measurement Unit (IMU) and cameras. The proposed system leverages the feature learning capabilities of CNNs for accurate object detection and localization. OpenCV is utilized for real-time image processing, preprocessing tasks, and generating precise bounding boxes around detected objects [19]. Additionally, an IMU sensor and cameras are incorporated to enhance the robot's spatial awareness and navigation capabilities. By fusing information from CNNs, OpenCV, IMU sensors, and cameras, the robotic system aims to achieve efficient object retrieval, demonstrating a comprehensive approach that combines deep learning, computer vision, and sensor technologies for enhanced robotic perception and interaction in real-world environments

2 CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNN), also referred to as ConvNets, represent a category of deep neural networks extensively utilized in the realm of image processing and analysis. Most contemporary algorithms for object identification heavily rely on CNN. In effectively extracting information from images, CNN leverages two intrinsic characteristics present in two-dimensional patterns: local connection, also known as spatial locality, and translation invariance [20]. For instance, when considering a cat pattern, recognition is based on a grouping of adjacent pixels rather than individual ones scattered throughout the image, illustrating the concept of local connectivity. The principle of translation invariance asserts that identifying a cat entity is contingent on its spatial arrangement, irrespective of whether it is positioned at the periphery or the central region of the image [21].

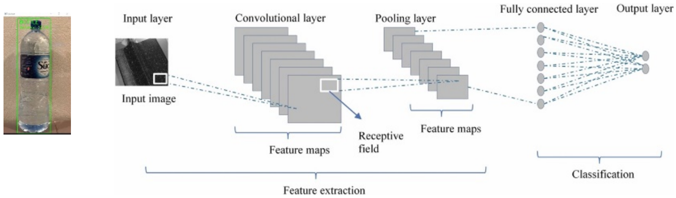


Fig. 1 A general configuration of a Convolutional Neural Network. [22]

2.1. Convolutional Layer

The pivotal role of the neural network is underscored by the convolutional layer, as its name suggests. This involves employing filters, also termed kernels or feature detectors, to extract features in a convolutional manner, as elaborated in the subsequent section. The procedure entails calculating the product of each element within a given set, followed by summing the resulting values to generate a new value. While traversing the image, the filter produces a fresh matrix known as the 'convolved feature,' 'Feature Map,' or 'Activation Map.' In scenarios with multiple channels, such as RGB photos comprising three channels, there exists a corresponding kernel for each channel.

The filter layer encompasses several relevant characteristics, with the principal parameter being padding or zero-padding. In the described scenario, when a filter takes a matrix as input and produces a single output value through a singular computation, the resulting matrix undergoes a reduction in its dimensions. Consequently, values positioned at the outer edges may encounter a resolution decrease contingent on the size of the utilized kernel. Padding becomes crucial in this context as it entails incorporating zero values to the image to augment the size of the resultant feature map.

Rather than making incremental adjustments to the filters, an alternative is to skip intermediate phases and simultaneously modify two or more steps. The stride parameter 's' in CNN is an additional factor that dictates the number of pixels the filter is displaced during each iteration. Increasing the stride parameter results in a decrease in the degree of overlap between adjacent regions, leading to the creation of smaller feature maps. Consequently, this can efficiently conserve computational resources. The depth of the convolutional layer is typically determined by the number of filters, which is often equivalent to the number of input channels. These parameters are distributed throughout the network and are commonly referred to as "parameter sharing" or "weight sharing," significantly reducing the overall quantity of parameters [23].

2.2. ReLU Layer

ReLU stands out as a frequently employed activation function in deep learning models. As depicted in Figure 12, its output is determined by selecting the larger value between 0 and the input value. Therefore, if the input is negative, the output becomes zero, while for positive inputs, the output remains unaltered and equals the input value.

The principal aim of ReLU is to introduce non-linearity into the neural network architecture, thereby enhancing its capability to capture intricate relationships. Moreover, ReLU offers the advantage of superior computational efficiency when compared to alternative activation functions.

2.3. Pooling Layer

A pooling layer is utilized to mitigate overfitting and streamline the processing of incoming data. Similar to the convolutional layer, the pooling layer conducts a scanning operation over a localized area, computes a value, and repeats this process by shifting the window both horizontally and vertically. The focus here is on an individual named Max. Pooling, a commonly employed technique, involves selecting the maximum value within a localized area, leading to a reduction in the size of the pooled feature map. In convolutional neural networks, the pooling layer integrates two crucial parameters: stride and window size [24].

2.4. Fully Connected Layer

Following convolutional/ReLU layers and pooling layers, the final flattened pooled feature map is subjected to processing in a fully connected layer, positioned just before the CNN's classification output. This fully connected layer shares similarities with traditional artificial neural networks, establishing connections between each neuron in the preceding layer and every neuron in the subsequent one. In classification models, the SoftMax activation function is frequently applied in the output layer [25].

3 SYSTEM DESIGN AND IMPLEMENTATION

This study seeks to achieve independent object retrieval through the application of advanced machine learning techniques. In the initial phase, the focus is on precisely locating the target object in the environment, utilizing computer vision methods. Subsequently, the identified object undergoes a thorough examination through the integration of OpenCV and CNNs, specialized in image recognition tasks. The model is crafted to identify the unique features of the target, facilitating accurate recognition. Once the object is identified, the autonomous robot is programmed to navigate towards it using path planning algorithms, ensuring both efficiency and safety.

The subsequent phase involves executing the retrieval process. With an equipped arm mechanism, the robot can accurately seize the identified object, enabling tactile interaction with the surroundings. After a successful seizure, the robot is programmed to place the object into a designated container, such as a basket, illustrating a comprehensive approach to autonomous object retrieval. The overarching goal of this research is to bridge the gap between artificial intelligence and robotics, highlighting the potential of machine learning to empower machines for independent navigation, identification, and manipulation of objects in real-world scenarios.

The implementation of autonomous object retrieval described in the workflow can be broken down into several major key stages. These stages represent the major components and steps involved in the overall process. A detailed illustration of this sequential decision-making process is presented in Figure 2, and a comprehensive explanation will be provided in the subsequent sections.

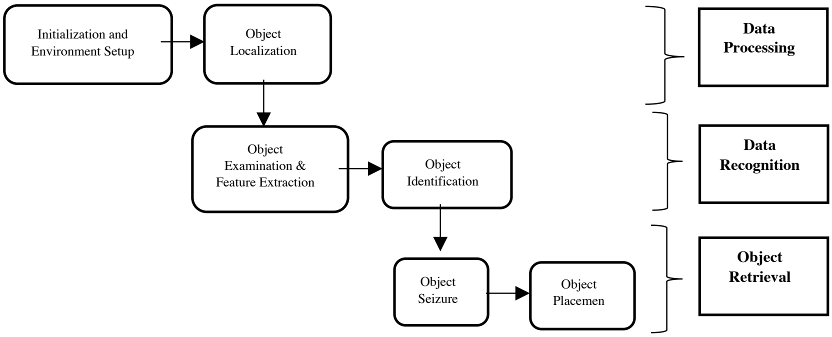


Fig 2 Workflow of the system.

The autonomous object retrieval system begins with the Initialization and Environment Setup, ensuring the operational readiness of systems and sensors. Subsequently, the Object Localization phase employs computer vision techniques to precisely locate the target object, followed by Object Examination and Feature Extraction using OpenCV and CNNs to extract distinctive features. Machine learning techniques are then applied in Object Identification, recognizing the object based on its features. Path Planning and Navigation algorithms determine the optimal route for the autonomous robot to navigate towards the identified object. The robot utilizes a Robotic Arm for precise object seizure, and tactile interaction with the environment demonstrates physical engagement. After seizing the object, it is placed into a designated container, showcasing a holistic approach. The system undergoes System Evaluation and Optimization to enhance efficiency and accuracy, concluding with the End of Process, signifying the successful completion of the autonomous object retrieval task.

In the context of this investigation, the chosen focal point is a water bottle, which acts as the exemplar for the study as it is a standard object in most areas. The primary objective is to implement object detection methodologies, and this is accomplished through the deployment of a webcam as a sensing tool.

3.1. Object Detection Methodology

The intricacies of this detection process hinge upon the symbiotic integration of OpenCV, a powerful framework for image processing, and a pre-trained CNN model specifically engineered for object detection tasks. By employing this combination of sophisticated technologies, the study aims to ascertain a comprehensive understanding of object recognition and localization within the given experimental framework.

This approach underscores the significance of merging computer vision techniques with advanced machine learning models to enhance the accuracy and efficiency of object detection. The choice of a water bottle as the subject of analysis provides a tangible and controlled environment for evaluating the proposed methodology's effectiveness. The subsequent paragraphs delve into the detailed methodologies and outcomes of this integration, shedding light on the intricacies involved in achieving robust object detection outcomes in the specified research context.

3.2. OpenCV for Webcam Input

In the initial phase, the primary step entails the utilization of OpenCV to seize video frames originating from the webcam. This procedural undertaking encompasses the initiation of a connection to the webcam, the systematic capture of individual frames within an iterative loop, and the subsequent presentation of the captured frames. This fundamental stage serves as the foundational process for acquiring a sequence of visual data from the webcam, forming the basis for subsequent analytical procedures.

Subsequently, the procedural continuum involves the systematic utilization of OpenCV to initialize a connection with the webcam, the iterative extraction of individual video frames, and the subsequent rendering and display of these frames. This foundational step underscores the essential role of OpenCV in facilitating the seamless interfacing with the webcam, thereby laying the groundwork for subsequent analytical processes reliant on the acquired visual data.

3.3. Distance Estimation

The general approach for gauging distance through object detection used in this study is presented in Figure 3. Figure 3 illustrates the perception of an image through a refracted lens. The variable d_0 represents the theoretical distance from the lens, while d_i serves as an estimate of the expected image appearance. By employing a triangle on the left side, a corresponding triangle is constructed on the right with the new refracted image and base d_0. This new triangle shares the same base of d_0 and theoretically identical perpendicular distance.

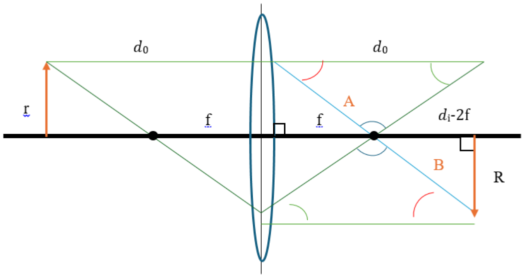


Fig. 3 Focal length and object image relationship.

where:

d_0 = distance of object from lens

d_i = distance of refracted image from Convex lens

f = focal length

Figure 3 shows the theoretical distance estimation based on refraction where the similar triangles and focal length relationship. Let r (on the left) be the object height and d_o is the distance of the object to the lens. This relationship is to estimate the distance of the object from the lens using the relationship between focal length and object image.

Upon examining the two triangles on the right, it becomes apparent that d_0 and d_i are parallel. The angles formed on each side of these triangles are opposite to each other, indicating that both triangles on the right are similar. Consequently, the ratio between them is expressed in Equation (1). By comparing the two similar triangles on the right, both featuring a right angle, A and B emerge as the hypotenuses of these similar triangles, each having a right angle. From reference to [65], this comparison leads to Equation (2) and ultimately results in Equation (4). Equation (3) is derived from the relationship between Object Detection and Focal Distance.

$$\frac{d_0}{d_i} = \frac{A}{B} \tag{1}$$

$$\frac{d_0}{d_i} = \frac{A}{B} = \frac{f}{d_i - f} \tag{2}$$

$$\frac{1}{f} = \frac{1}{d_0} = \frac{1}{d_i} \tag{3}$$

$$f = \frac{2\,\pi \times 180}{360} \tag{4}$$

$$d = f + \frac{R}{r} \tag{5}$$

Equation (5) is proposed by plugging Equation (4) into Equation (5) to estimate the distance of an object using webcam's object detection in centimetres (cm). Figure 4 shows when there is a clear vision of the object and the system determines the distance d.

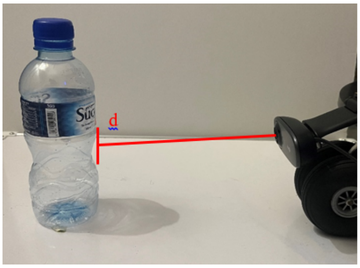


Fig. 4. Line of vision of the webcam and object.

When object detection is implemented, both classification and localization occur simultaneously, as illustrated in Figure 5.

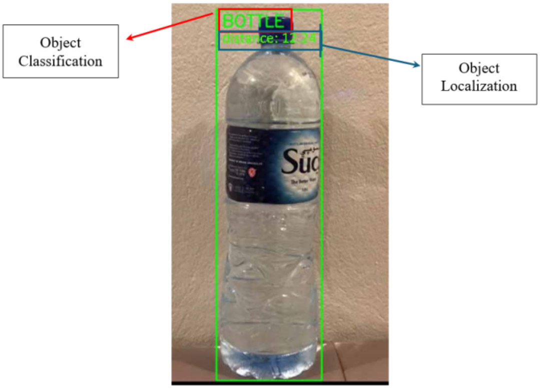


Fig. 5 Object detection on a water bottle.

4 MOBILE ROBOT DESIGN

The utilization of the robot in this autonomous object retrieval system is facilitated through the Eddie Robot by Parallax, which has been modified in a specific manner to align with the requirements of the described workflow. The Eddie Robot serves as the physical platform for implementing the sophisticated integration of computer vision, machine learning, and robotics. This specific modification likely includes the incorporation of the necessary hardware and software adaptations to enable the robot to execute the tasks outlined in the autonomous object retrieval system.

The modifications to the Eddie Robot could encompass enhancements such as the integration of sensors, actuators, and the gripper mechanism necessary for object detection, movement coordination, and loading functionalities. Additionally, adjustments to the robot's control algorithms and programming might be implemented to ensure seamless communication between the various components of the system. The specific modifications tailored to the Eddie Robot contribute to its ability to navigate autonomously, identify target objects, and execute precise movements with the gripper.

This utilization of the Eddie Robot showcases the adaptability and versatility of robotic platforms, as it can be modified to meet the specific needs of a given application. The synergy between the Eddie Robot and the outlined modifications exemplifies how a commercially available robotic platform can serve as a foundation for implementing advanced autonomous systems, fostering innovation in the intersection of robotics and artificial intelligence. This can be shown in the Figures 6 and 7.

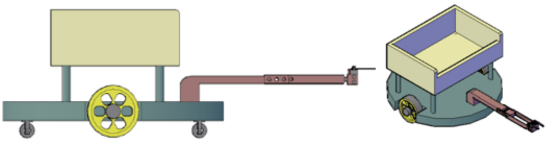


Fig. 6. 3D design of the robot.

Fig. 7. Actual robot design.

4.1. Deriving Specification

The integration of object detection and tracking technologies in robotics has emerged as a prominent trend within industrial and warehouse processes, offering a multitude of benefits including increased efficiency, safety, and productivity. By equipping robots with these advanced technologies, organizations can effectively replace humans in repetitive tasks, thereby allowing individuals to redirect their focus towards more complex and creative endeavors. However, despite the potential advantages, widespread adoption faces several challenges, including real-time processing constraints, limitations imposed by available sensors, and the associated high costs.

In response to these challenges, this thesis proposes a solution centered around the utilization of the Raspberry Pi 3+ as a cost-effective computing platform specifically tailored for robotics applications, particularly in the domain of object detection and tracking. Renowned for its affordability and adaptability, the Raspberry Pi serves as a low-cost, single-board computer capable of interfacing seamlessly with a variety of sensors and cameras. While the Raspberry Pi is commonly employed in educational settings and prototyping scenarios, its application within robotics necessitates striking a delicate balance between cost efficiency and computational power.

Motivated by these considerations, the primary objective of this thesis is to develop an advanced robotic mobile system that leverages CNN, OpenCV, and an IMU sensor. The proposed system architecture integrates these technologies synergistically, with CNN serving as the cornerstone for precise object detection and localization, while OpenCV facilitates real-time image processing and bounding box generation. Furthermore, the inclusion of an IMU sensor enhances the system's spatial awareness and navigation capabilities.

The overarching goal of this research endeavor is to realize an efficient and effective robotic mobile system capable of proficient object retrieval in real-world environments. By amalgamating insights from deep learning, computer vision, and sensor technologies, the proposed system aims to significantly enhance robotic perception and interaction within dynamic settings. Through the delineation of deriving specifications, this thesis endeavors to elucidate the problem statement, articulate the proposed solution approach, and delineate the anticipated outcomes of the research effort.

4.2. Robot Movement

In guiding the robot's movement towards a spotted water bottle, the strategy relies on the central coordinates of a bounding box. This box outlines the detected object, like a water bottle, identified by combining OpenCV and CNN. The critical data here is the center of the box, specifically its x and y coordinates, which become the instructions for the robot's path. These coordinates help the robot understand where the water bottle is in the video frame, steering its movement accordingly.

The process unfolds as the algorithm calculates the central points of both the bounding box and the frame, determining how the detected object fits into the overall view. The robot's movement kicks in when these coordinates are checked against each other, figuring out if the water bottle is close to the center of the frame. This decision point sets off the robot's motion, directing it toward the targeted object. By adding a proximity threshold that can be adjusted, the system becomes flexible, balancing accuracy and responsiveness. This not only showcases a smart coordination strategy but also makes sure the robot can handle different situations, where objects might vary in size or location, and the environment might change.

4.3. Loading Mechanism

The loading mechanism introduces an additional layer to the autonomous system, showcasing the mobile robot's capability not only in object detection and movement but also in the physical interaction and manipulation of objects. The integration of a gripper attached to the front of the mobile robot represents a sophisticated extension of its functionalities, allowing for precise grasping and manipulation of the detected water bottle. The gripper serves as a mechanized hand, enabling the mobile robot to execute a controlled and dexterous approach towards the identified object. This component further bridges the gap between artificial intelligence and physical action, highlighting the system's versatility in both perceiving and interacting with its environment.

The placement of the basket on top of the mobile robot as the loading destination adds an additional layer of complexity to the autonomous retrieval process. Once the water bottle is successfully detected, approached, and grasped by the gripper, the mobile robot is programmed to lift and transport the object to the designated basket location on its top. This spatial arrangement emphasizes the robot's ability not only to navigate in the horizontal plane but also to manipulate objects in the vertical dimension. The loading mechanism showcases a holistic approach to autonomous object retrieval, illustrating how the system seamlessly integrates computer vision, machine learning, and robotics to achieve a comprehensive level of autonomy. Overall, the loading mechanism demonstrates the system's capability to handle both perception and physical interaction, underscoring the potential of such integrated technologies in real-world scenarios.

4.4. Gripper Design

The gripper design, equipped with three degrees of freedom (3-DOF), plays a pivotal role in the loading mechanism of the autonomous robot. The three degrees of freedom provide the gripper with the ability to move in three different directions, enhancing its versatility in grasping and manipulating objects. This design facilitates a sophisticated level of control and precision, allowing the gripper to adapt to various object shapes and sizes during the loading process.

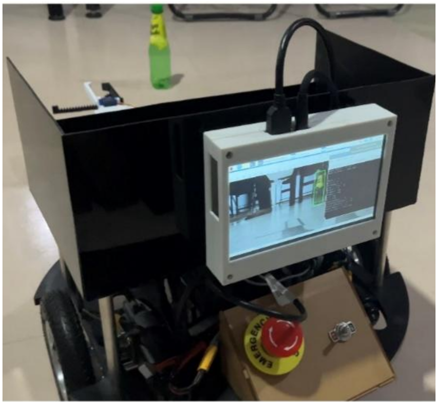
The first degree of freedom typically involves the gripper's ability to open and close, enabling it to securely hold onto the detected water bottle. This pivotal motion is fundamental for the gripping mechanism, ensuring a firm and stable grasp on the object. The remaining two degrees of freedom contribute to the gripper's flexibility in orienting the water bottle within its grasp. These additional movements allow the gripper to adjust the object's position as needed, aligning it with the desired orientation for placement into the designated basket.

The gripper's 3-DOF design reflects a careful consideration of the functional requirements for the autonomous loading mechanism. The ability to open and close, combined with versatile movements for object orientation, ensures that the gripper can effectively handle the complexities of the loading task. This design choice emphasizes the importance of a balanced gripper configuration, where sufficient degrees of freedom are provided to enhance adaptability and precision in the robot's physical interactions with objects in its environment.

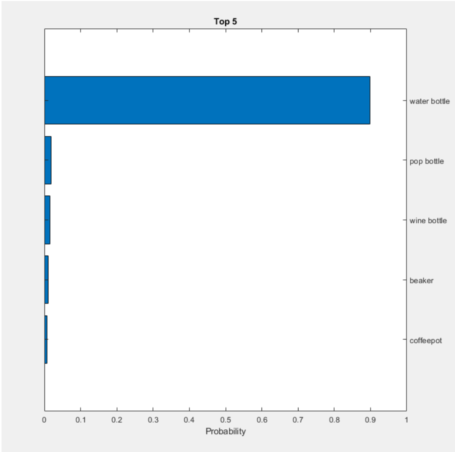
5 RESULTS AND DISCUSSION

5.1. Object Detection Discussion

Authors exploration of bottle detection provides valuable insights into the intricate dynamics of proximity and lighting conditions. Notably, the model exhibits an impressive 90% accuracy when the bottle is in close proximity, showcasing its adeptness at capturing detailed features at short ranges—a vital attribute for applications requiring precision in object detection at close distances shown in Figure 8. However, our findings underscore the model's sensitivity to lighting conditions, with optimal brightness significantly influencing its predictive capabilities. The high accuracy observed under favourable lighting conditions suggests a reliance on well-illuminated scenes for feature extraction, emphasizing the consideration of environmental lighting as a crucial factor affecting model reliability.



(a)

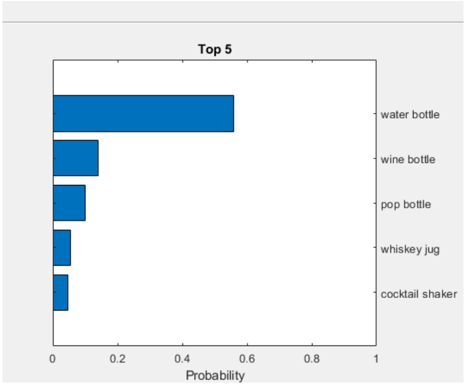


(b)

Fig. 8 Live feed of the object detection with bounding box: (a) monitor view; (b) probability.

Despite these promising results, it is essential to acknowledge the challenges and limitations identified in our study. Specifically, we observed a notable drop in performance when lighting conditions were suboptimal, resulting in a reduced accuracy of 56% shown in Figure 9. This emphasizes the model's vulnerability to challenging environmental factors, necessitating further refinement for robust performance under diverse conditions. Real-world scenarios often present varying distances and lighting conditions, demanding a more adaptable model. Future efforts should be directed towards addressing these limitations, exploring techniques such as dataset augmentation and advanced model architectures to enhance object detection performance across a broader spectrum of environmental conditions. This study lays the groundwork for future improvements, highlighting the potential of CNN-based models in object detection while recognizing the imperative of addressing challenges for real-world deployment.

(a)



(b)

Fig. 9 Object detection of a water bottle with reduced light: (a) monitor view; (b) probability.

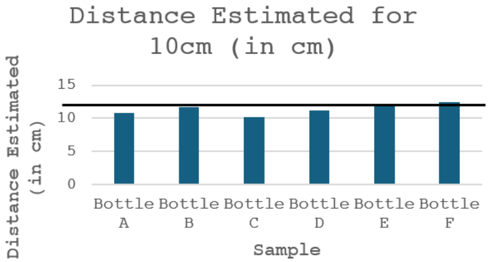
5.2. Distance Estimation

Tabel 1. The accuracy of the distance estimated

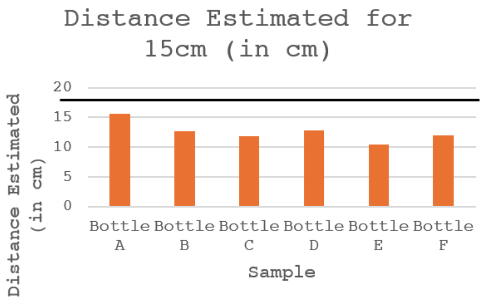
Sample	Bottle A	Bottle B	Shortcuts
Distance estimated for 10 cm	Alt+A	Head 2	Ctrl + 2
Distance estimated for 15 cm	Alt+N	Head 3	Ctrl+3
Distance estimated for 20 cm	Alt+R		

To ascertain the precision of an object, we employ equation (6) to calculate distance using a solitary camera. When the objects are positioned directly within the camera's field of view, the system will make distance estimations for them. In the Table 1 shows the accuracy of the distance estimated compared to the actual distance.

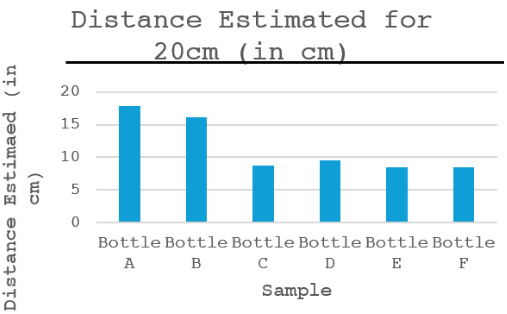
This experimental arrangement involves configuring the arm to a fixed position, aligning the camera with the object on the same plane. The objects are positioned directly in the camera's line of sight at distances of 10cm, 15cm, and 20cm as presented in Figure 10.



(a)



(b)



(c)

Fig. 10 Distance estimated and actual distance difference: (a) 10 cm; (b) 15 cm; (c) 20 cm.

As illustrated in Figure 10, when bottles are positioned farther from the camera, the detected distance becomes shorter. This suggests that Bottle A and Bottle B are slightly larger than Bottle C, Bottle D, Bottle E, and Bottle F. Consequently, larger bounding boxes are generated, evident in the results. Conversely, for Bottle C, Bottle D, Bottle E, and Bottle F, being smaller in size, smaller bounding boxes are produced. The size of the bounding box is influenced by the proximity of the object—closer objects result in larger bounding boxes, while those at a greater distance yield smaller one.

CONCLUSIONS

The conclusion of the experiment yielded valuable data that was collected and analysed. Several conclusions have been drawn concerning the current state of the system, and there is a significant amount of future work that needs exploration. This section will delve into each aspect of the system and discuss the necessary steps for enhancing its performance.

It is shown that the robot relies on computer vision for tasks like object recognition and tracking, low FPS can hinder the performance of these algorithms. The reduced frame rate negatively impacts the accuracy of identifying and monitoring objects, diminishing the robot's ability to interact intelligently with its surroundings. Furthermore, the low FPS may destabilize the control system, resulting in jerky or unstable movements, making it challenging to control the robot effectively.

In terms of the robotic arm performance, the current design was not sturdy and versatile enough to grasp various bottles which makes evaluations to be limited. The arm and gripper experienced a great number of issues such as width limitation, unsteady arm, limited movement that resulted in the system not being able to retrieve objects.

To address low FPS, potential solutions include hardware upgrades, code optimization, and a careful balance in sensor usage to ensure efficient processing and optimal performance. Regular monitoring and testing are essential to identify and rectify any issues promptly, ensuring the autonomous robot operates seamlessly in various environments.

REFERENCES

[1] A. Dhaliwal, “The Rise of Automation and Robotics in Warehouse Management,” 2020, pp. 63–72. DOI: 10.1201/9781003032410-5

[2] H. Klaus and E. Blessing, “The impact of robotics on society and civilization,” *Robotics Engineering*, Feb. 2024.

[3] S. Pfeiffer, “Robots, industry 4.0 and humans, or why assembly work is more than routine work,” *Societies*, vol. 6, p. 16, May 2016. DOI: 10.3390/soc6020016

[4] H. Hardy, M. Walsum, L. Livermore, and S. Walton, “Research and development in robotics with potential to automate handling of biological collections,” *Res. Ideas Outcomes*, vol. 6, Nov. 2020. DOI: 10.3897/rio.6.e61366

[5] J. Trevelyan, W. Hamel, and S.-C. Kang, “Robotics in hazardous applications,” pp. 1521–1548, Jan. 2016. DOI: 10.1007/978-3-319-32552-1_58

[6] C. Wong, E. Yang, X.-T. Yan, and D. Gu, *An overview of robotics and autonomous systems for harsh environments*. 2017, p. 6. DOI: 10.23919/IConAC.2017.8082020

[7] A. H. M. M. Alahi and K. Potter, “The future of robotics: advancements and implications,” *Comput. J.*, Jan. 2024

[8] H. Naeem, J. Ahmad, and M. Tayyab, *Real-time object detection and tracking*. 2013, p. 153. DOI: 10.1109/INMIC.2013.6731341

[9] R. Yunardi, D. Arifianto, F. Bachtiar, and J. Prananingrum, “Holonomic implementation of three wheels omnidirectional mobile robot using DC motors,” *J. Robot. Control JRC*, vol. 2, Mar. 2021. DOI: https://doi.org/10.18196/jrc.2254

[10] M. N. Osman Z, N. Mat Yahya, M. Mohd Ali, and K. Johan, "Review on CNC-rapid prototyping," Mater. Sci. Eng. Conf. Ser., vol. 36, p. 2032, Sep. 2012. DOI: 10.1088/1757-899X/36/1/012032

[11] J.-P. Sheu, C.-C. Chang, K.-W. Lo, and C.-W. Deng, "Design and implementation of a navigation system for autonomous mobile robots," *IJAHUC*, vol. 6, pp. 129–139, Aug. 2010. DOI: 10.1504/IJAHUC.2010.034966

[12] M. Soori, R. Dastres, B. Arezoo, and F. Karimi Ghaleh Jough, "Intelligent robotic systems in industry 4.0, A review," *J. Adv. Manuf. Sci. Technol.*, p. 2023, Jan. 2024. DOI: 10.51393/j.jamst.2024007

[13] J. Wu, J. Gao, J. Yi, P. Liu, and C. Xu, *Environment Perception Technology for Intelligent Robots in Complex Environments: A Review*. 2022, p. 485. DOI: 10.1109/CCISP55629.2022.9974277

[14] B. Schwarz, "Lidar: Mapping the world in 3D," *Nat Photonics*, vol. 4, Jul. 2010. DOI: 10.1038/nphoton.2010.148

[15] R. Castanyer, S. Martínez-Fernández, and X. Franch, *Integration of Convolutional Neural Networks in Mobile Applications*. 2021, p. 34. DOI: 10.1109/WAIN52551.2021.00010

[16] I. Farkas, T. Malík, and K. Malinovská, "Grounding the meanings in sensorimotor behavior using reinforcement learning," *Front. Neurorobotics*, vol. 6, p. 1, Feb. 2012. DOI: 10.3389/fnbot.2012.00001

[17] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, Dec. 2020. DOI: 10.1007/s10462-020-09838-1

[18] N. Tj, "Fast and economical object tracking using raspberry Pi 3.0," *Int. J. Eng. Adv. Technol.*, vol. 8, pp. 336–338, Sep. 2019. DOI: 10.35940/ijeat.F1070.0886S19

[19] C. Bezerra, F. Vieira, and D. Carneiro, "Autonomous robotic navigation approach using deep Q-network late fusion and people detection-based collision avoidance," *Appl. Sci.*, vol. 13, p. 12350, Nov. 2023. DOI: 10.3390/app132212350

[20] L. Chang *et al.*, "Convolutional neural networks in image understanding," vol. 42, pp. 1300–1312, Sep. 2016. DOI: 10.16383/j.aas.2016.c150800

[21] Z. Gu, "Recognition of feline images based on deep learning," *Appl. Comput. Eng.*, vol. 5, pp. 345–352, Jun. 2023. DOI: 10.54254/2755-2721/5/20230594

[22] S. B. Jha and R. F. Babiceanu, "Deep CNN-based visual defect detection: Survey of current literature," *Comput. Ind.*, vol. 148, p. 103911, Jun. 2023. DOI: 10.1016/j.compind.2023.103911

[23] M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, p. 52, Mar. 2023. DOI: 10.3390/computation11030052

[24] A. Zafar *et al.*, "A Comparison of pooling methods for convolutional neural networks," *Appl. Sci.*, vol. 12, no. 17, p. 8643, Aug. 2022. DOI: 10.3390/app12178643

[25] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *ArXiv E-Prints*, Nov. 2015.