Received Date: 20th December, 2024.
Revision Date: 6th January, 2025.
Accepted Date:6th March, 2025.

RoboSort: Automated Object Sorting Robotic Arm

Kripesh Paudel[1*], Suresh Gurung[2], Urusha Luitel[3], Gaurav Gautam[4], Smriti Nakarmi[5]
[1]Department of Electronics, Communication and Information Engineering, Kathmandu Engineering College, Nepal.
E-mail: kripeshpaudel22@gmail.com
[2]Department of Electronics, Communication and Information Engineering, Kathmandu Engineering College, Nepal.
E-mail: gurungsuresh28@gmail.com
[3]Department of Electronics, Communication and Information Engineering, Kathmandu Engineering College, Nepal.
E-mail: luitelurusha@gmail.com
[4]Assoc. Professor, Department of Electronics, Communication and Information Engineering, Kathmandu Engineering College, Nepal.
E-mail: gaurav.gautam@kecktm.edu.np
[5]Assoc. Professor, Department of Electronics, Communication and Information Engineering, Kathmandu Engineering College, Nepal.
E-mail: smriti.nakarmi@kecktm.edu.np

Abstract - This paper discloses the development and functionalities of an automated robotic arm system that eases the need for heightened precision and efficiency when sorting different objects. This is accomplished utilizing a 5-degree-of-freedom robotic arm, which reduces unnecessary handwork while increasing sorting accuracy and speed. It utilizes advanced computer vision and image processing methods based on the OpenCV library and Python programming to implement analysis of an object's live video feed and object image capture. These images were implemented in the real-time webcam and processed through YOLOv5 for effective object detection, subsequent classification and sorting based on objects' color and shapes using robotic arm. This method facilitates faster operations with closer to zero errors than humans, especially when critical precision is a requirement.

Keywords - 5 DOF, OpenCV, Inverse Kinematics, Python, Robotic Arm, YOLOv5, Object Detection

Introduction

Automated object sorting is quickly becoming a game changer in a variety of industries, increasing efficiency and streamlining procedures. The use of robotic arms for this purpose represents a significant leap in industrial automation, since it allows for the speedy and precise classification of objects based on certain criteria. These robotic arms use advanced technology, such as machine learning algorithms and computer vision, which accurately recognize, grasp and place objects. This combination of servo motors, cameras, microcontrollers, and software enables real-time decision-making during the sorting process. The use of technologies such as OpenCV, Python, and YOLOv5 illustrates the complexity of these systems, which provide greater effectiveness and reliability in industries.

** Corresponding Author*

Robotic arms play an important part in automation, enabling great precision and efficiency in a wide range of industrial and non-industrial applications. The importance of robotic arms emerges from their ability to do repetitive tasks with consistent accuracy, minimizing human error and increasing efficiency. In the context of the project, the creation of a 5 DOF robotic arm represents a significant advancement in automation. The initiative's goal is to automate the object sorting process, reducing the requirement for manual labor while improving sorting efficiency and precision. This robotics application demonstrates not only robotic arms' adaptability, but also their critical role in transforming traditional sorting processes.

The research paper covers key points to enhance sorting procedures in robotic arms. The main goal is to uncover ways to improve the efficiency of object sorting by integrating the OpenCV, the Python programming language, and a 5-degree of freedom robot arm. It aims to increase sorting speed and precision by combining real-time video analysis with YOLOv5 for accurate object detection and grouping based on color and form.

Literature Review

Robotic arms are increasingly critical in various industries, focusing on enhancements in object detection, shape/color classification, and autonomous sorting. This research contributes to the expanding field of robotic arm applications by improving object recognition and autonomous sorting tasks.

Chen, Tsai, Zhang, and Wang proposed a 3D object detection and grasping system for the Dofbot robotic arm with the help of YOLOv5 and ROS MoveIt! The whole system performed effective edge-device operations while maintaining an IEEE standard, providing vital insights into the automated grasping technologies of today [1].

Harshit S. Badiger and co-authors demonstrated the opportunity to enhance product quality and provide reduced expenses by using machine vision, non-contact inspection, and the kinematics of robots. This research on the cost-effective integration of robotics deals with object detection through OpenCV-Python and arm programming using Python 3.5 for its practical application on a programmable robot arm [2].

Lennon Fernandes and Shivakumar B.R. have developed an industrial sorting robotic system where HSV value manipulation has been put in addition to contour detection for object recognition. Their servo motor-enabled arm sorts objects based on shape and color, holding high prospects for further improvement of the system using more colors and shapes [3].

Additionally, the work of Vishal Kumar and colleagues is in a multifunctional robot arm for operations that demand fine control and 3D vision using a Raspberry Pi 3 and Arduino Mega 2560. Such tasks need highly accurate arm movements that they have already realized, with their arm being enabled with wireless protocols and voice recognition for effective object detection and placement—a feature of modern manufacturing owing to AI [4].

Methodology

Data Collection

For our task, we require data consisting of images of different objects, all of which are in different colors and shapes. According to our requirements, we could not find objects with such features on the internet. So, we manually collected the dataset by capturing the images of the 3D-printed objects. The shape of the object is a cube, cylinder, or prism. We used data augmentation techniques such as rotation, brightness changes, shifting, and noise reduction to increase the total number of images to approximately 750. We used LabelImg, a graphical tool for image annotation, to label and annotated the images by drawing bounding boxes around the objects and classifying them into certain predefined categories. The XML file contains the image's width, height, and depth, as well as the bounding box coordinates and label to which it belongs.

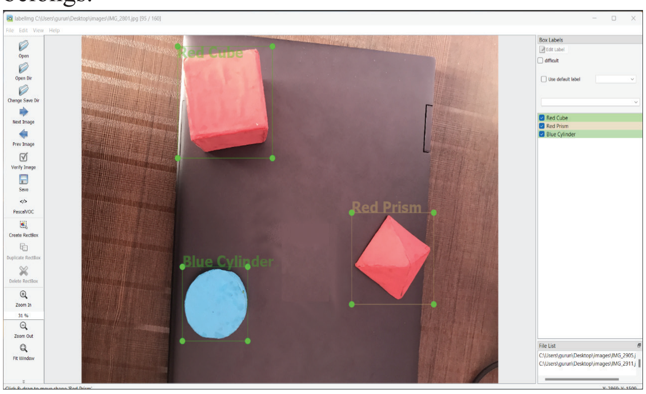


Fig. 1 LabelImg Interface

Object Detection using YOLOv5

Object detection is a computer vision method that aims to precisely identify and locate a specific object inside an image or video [5]. Numerous studies have been conducted on object detection in recent years, and writers have published their papers referring to it as a "state-of-the-art" model. Some of the well-known models are R-CNNN, Fast R-CNN, YOLO, etc.

YOLOv5 is the fifth iteration of the YOLO series, further refining and enhancing the capabilities of its predecessors. Developed by Ultralytics, YOLOv5 introduces improvements in model architecture, training strategies, and performance. Notable features include increased accuracy, versatility, and adaptability for various domains. YOLOv5 has gained popularity for its real-time object detection capabilities and ease of use, making it a preferred choice for researchers and developers. With advancements in both speed and accuracy, YOLOv5 has been successfully applied in applications such as robotics, where rapid and precise object detection is crucial for navigation, interaction, and decision making.

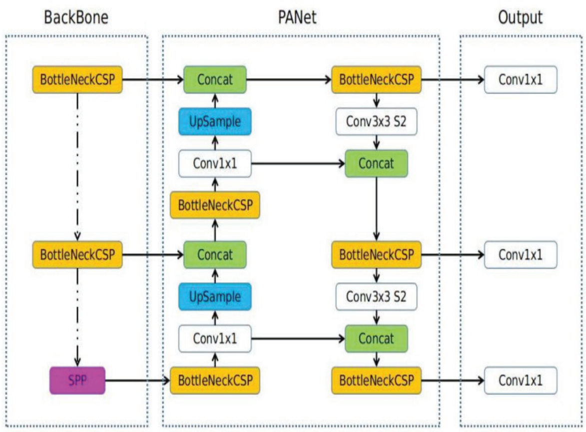


Fig. 2 Overview of YOLOv5[6]

System Working

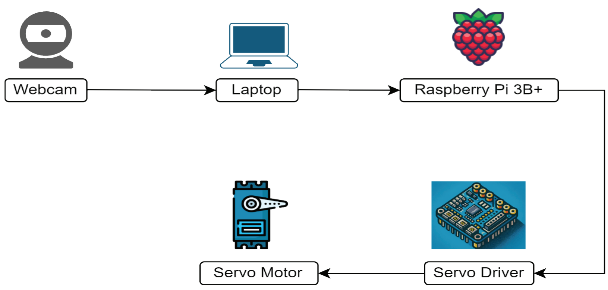


Fig. 3 Working of the System

The above figure shows the working of the system, which includes a Raspberry Pi, webcam, servo motor, and servo motor driver. The system's first step is to capture images from the webcam. Next, the laptop processes the captured image to extract all the necessary information. Next, the system classifies the three defined colors (red, green, and blue) and the three basic shapes (cylinder, prism, and cube). We compute the coordinates of these objects' centroids. This information is transferred to the Raspberry Pi. Then the required pulse for each motor is generated through the servo driver, initiating the robotic arm to pick and place the detected object.

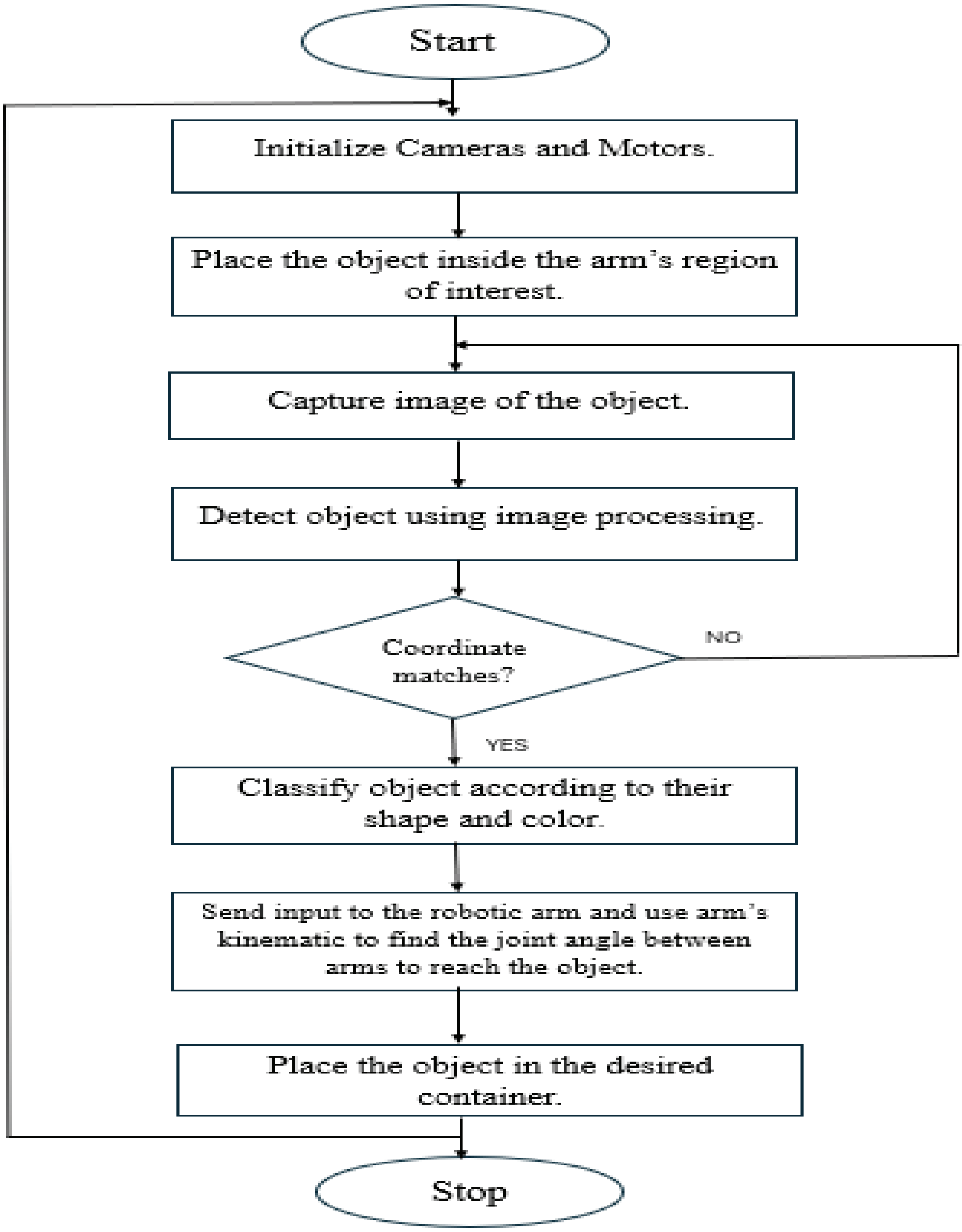
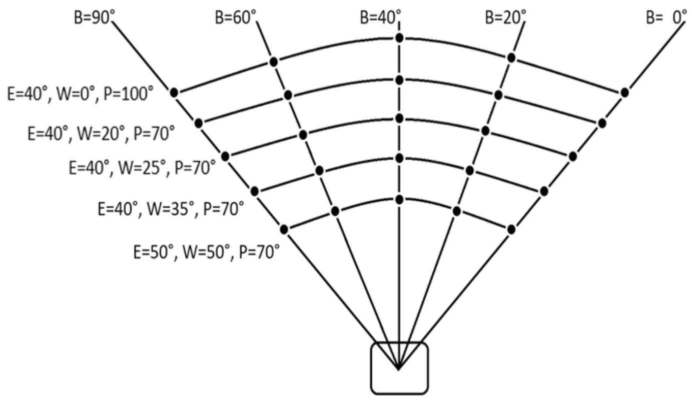


Fig. 4 Flowchart of the System

Inverse Kinematics

The following diagram details the mapping of the reachable workspace of the robotic arm and how it can be used for object manipulation; the area where inverse kinematics can be used. Initially, the workspace is accurately mapped by sending incremental pulses to the servo motors and noting the position of the end effector. An object is placed within the workspace after the system is powered up. The position of the object is captured using the webcam, the coordinates of the centroid is calculated and accurate pulse commands are sent to the servo motors to effectively move the arm to where the object is placed. Finally, by taking the shape and color information about the object into account, the object is moved to the designated container.



Glossary:
1.B:-base
2.E:-Elbow
3.W:-Wrist
4.P:-Pivot

Fig. 5 Implementing Inverse Kinematics

Results and Analysis

The goal is to make a robotic arm system that can correctly sort objects on the basis of its shape and color, and place it in the designated location.

After providing the input images to the trained model, the following results were obtained:

Fig. 6 Testing of Objects Using Trained Model

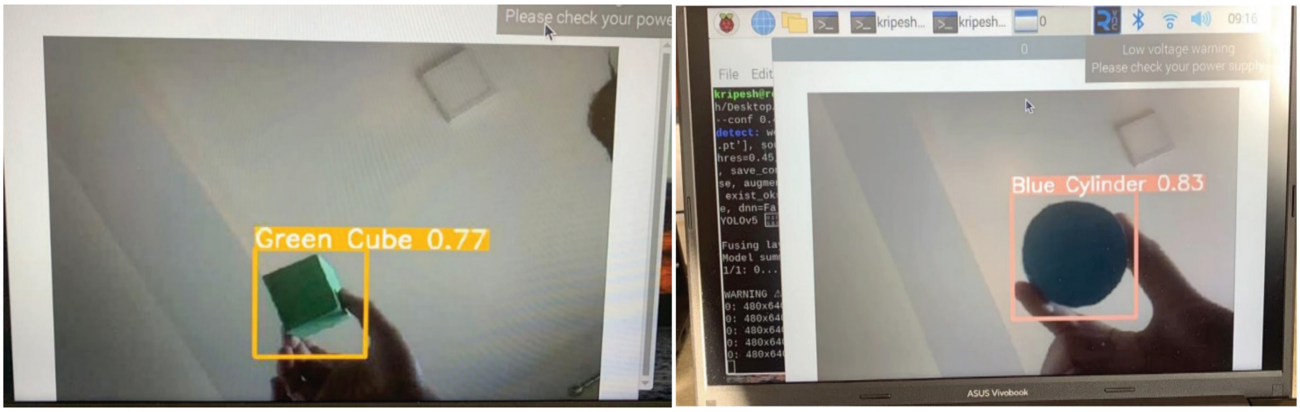


Fig. 7 Output Sample for Object Detection

A confusion matrix is the matrix that summarizes the predicted results and actual results. The confusion matrix for the model we developed is shown below. This matrix visually displays the performance of our algorithm, with the predicted labels on the y-axis and the true labels on the x-axis. The diagonal cells from top-left to bottom-right show the number of correct predictions for each class, which are the true positives for our model. The non-diagonal cells indicate misclassifications, where the predicted label does not match the true label.

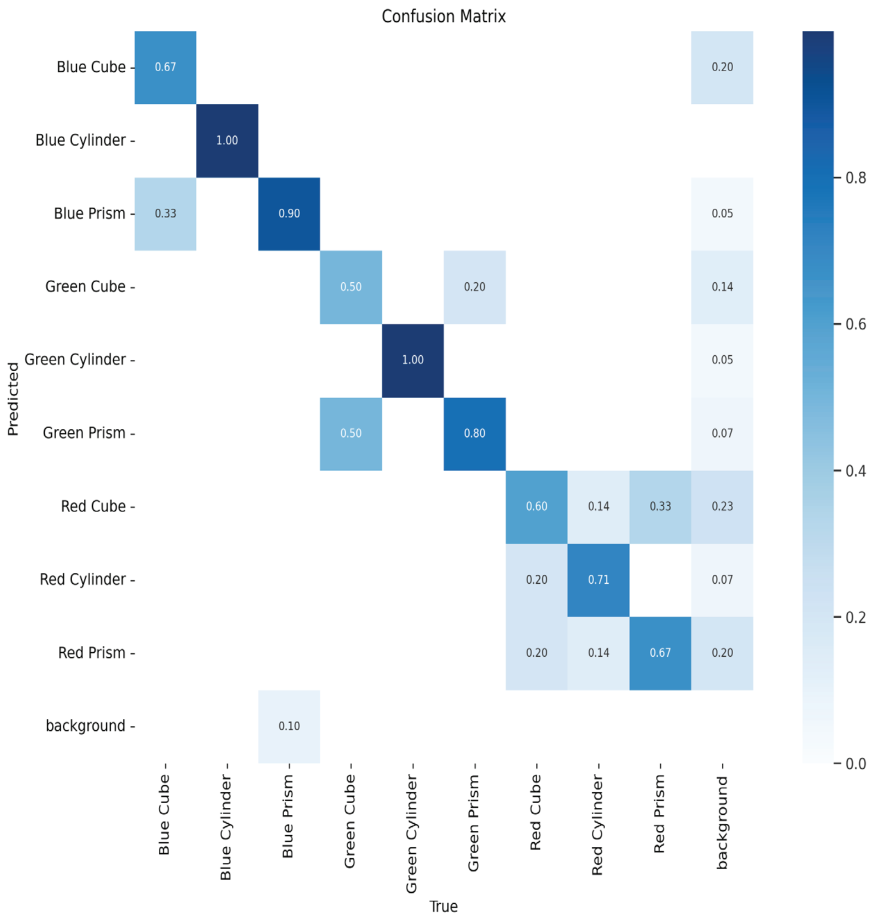


Fig. 8 Confusion Matrix of the Model

Various metrics can be calculated from a confusion matrix and are used to assess how well the model performed.

Accuracy: This is the most basic metric and simply measures the proportion of correctly classified objects. In this case, the model has an accuracy of 94.23%, which means it correctly classified 94.23% of the objects in the image. The accuracy of the model is calculated using the formula [7]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{(i)}$$

Precision: It tells us how many of the objects that the model classified as a particular class actually belonged to that class. It is calculated using the formula:

$$Precision = \frac{TP}{TP+FP} \quad \text{(ii)}$$

Recall: It tells us how many of the actual objects in a particular class were correctly identified by the model. It is calculated using the formula:

$$Recall = \frac{TP}{TP+FN} \quad \text{(iii)}$$

F1-score: This metric is a harmonic mean of precision and recall, and it is a way of combining these two metrics into a single score. An F1-score of 1 means that the model has perfect precision and recall, while a score of 0 means that the model is always wrong. It is calculated using the formula:

$$F1-Score = 2 \times \frac{Precision\ X\ Recall}{Precision+Recall} \quad \text{(iv)}$$

The precision, recall, and F1 score for each individual object has been calculated and is shown in the table below:

Table 1

Evaluation Metrics

Object	Precision	Recall	F1 Score
Blue Cube	0.9571	0.7701	0.8535
Blue Cylinder	1	1	1
Blue Prism	0.9	0.7031	0.7895
Green Cube	0.5	0.5952	0.5435
Green Cylinder	1	0.9524	0.9756
Green Prism	0.7143	0.4673	0.565
Red Cube	0.6667	0.5333	0.5926
Red Cylinder	0.7172	0.7245	0.7208
Red Prism	0.67	0.5537	0.6063

The performance of the model can be interpreted based on the values of the metrics. For example, let us look at the values of the "Blue Cube". The precision for "Blue Cube" is 0.9571, which means that out of all the objects the model classified as "Blue Cube", 95.71% were actually blue cubes. The recall for "Blue Cube" is 0.7701, which means that out of all the actual blue cubes in the image, the model correctly identified 77.01%. And the F1-score for "Blue Cube" is 0.8535, which is a good score.

The hardware component of the project achieved sophisticated manual control over the robotic arm's servo motors, which is crucial for the precise manipulation required in dynamic object handling tasks. We achieved this by integrating the servo motors with a PCA9685 PWM servo motor driver, which enables precise modulation of servo angles.

The robotic arm worked efficiently in detecting objects and placing them at designated location, where the success rates acquired were at about 70%. Still, there were also some reasons for not getting a 100% success. Inconsistencies in lighting or minor changes in the position of the object may affect the ability of the camera to determine the coordinates of the centroid accurately, which may show fluctuations in object identification. It is also probable that performance errors relate to other mechanical limitations, whether concerning preciseness regarding servo motor control or minor inaccuracies regarding how the arm mechanically responds to pulse commands. These are further possibilities concerning the calibration inaccuracies that may be introduced into a system of how the theoretical model for the movements of an arm is related to actual physical execution. These will be overcome by refining the vision system, providing better consistency under varying lighting conditions, enhancing motor control precision, and having continuous calibration that can bridge the gap between theoretical and practical implementation.

Conclusion

Our study introduces a system that utilizes a 5-DOF robotic arm and modern computer vision technologies such as OpenCV and YOLOv5 to effectively sort things. This system effectively categorizes items according to their color and shape, achieving high levels of precision and minimizing the requirement for user involvement. Deep learning is a rapidly developing area of research. The algorithms currently in use may become obsolete as new methodologies are consistently being developed to enhance models. These advancements can be incorporated into system, which will enhance the system's efficiency.

Future enhancement

The paper focuses on optimizing the program to reduce resource consumption and enhance performance by refining the algorithm, eliminating unnecessary computations, and minimizing memory use. This would ensure the management of resource-intensive tasks to prevent Raspberry Pi overloads. Additionally, upgrading to more sophisticated object detection algorithms will improve the ability to identify objects with complex textures or irregular shapes, thereby enhancing sorting accuracy. Integrating a pressure sensor into the gripper mechanism will create a closed-loop system, allowing real-time feedback on the gripping force for precise object handling without causing damage. This adjustment will enable the robotic system to dynamically adapt its grip based on the object's characteristics, further automating the process by providing immediate feedback on the success of each gripping action.

References

[1] C.-W. Chen, A.-C. Tsai and Y.-H. Zhang, "3D object detection combined with inverse kinematics to achieve robotic arm grasping," in 10th International Conference on Orange Technology (ICOT), Sanghai, 2022.

[2] H. S. Badiger, H. Kattimani, A. S. Hegde and A. Kumar, "Object Detection for Pick and Place Industrial Robots Using OpenCV-Python," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 6, no. 4, pp. 578-586, 2019.

[3] L. Fernandes and B. Shivakumar, "Identification and Sorting of Objects based on Shape and Colour using robotic arm," in 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2020.

[4] V. Kumar, Q. Wang, W. Minghua, S. Rizwan, S. M. Shaikh and X. Liu, "Computer vision based object grasping 6DoF robotic arm using picamera," in 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, 2018.

[5] "Object Detection Guide - Everything You Need to Know," Fritz ai, Jul. 14, 2023. https://fritz.ai/object-detection.

[6] I. Hogan, D. Qiao, R. Luo, M. Moattari, A. Carthy, F. H. Zulkernine, F. Rivest and M. Breton, "FireWarn: Fire Hazards Detection Using Deep Learning Models," in The Third IEEE International Conference on Cognitive Machine, 2021.

[7] Hossain, G. , Rashid, M. , Islam, M. , Sarker, A. and Yasmin, M. (2022) Towards Mining Public Opinion: An Attention-Based Long Short Term Memory Network Using Transfer Learning. Journal of Computer and Communications, 10, 112-131.