2D Object detection with monocular depth estimation

Abrar Naim Shahiruddin Bin Shahbudin*, Che Wan Ar-Rayyan Bin Che Wan Shamsiruddin†,
Muhammad Ammar Bin Mohd Hazlan‡, Muhammad Tareq Adam Bin Ellias§,
Muhammad Zahirul Isyraf Bin Mohamed Aidi Shahriz¶

***Abstract*—Computer vision techniques, such as monocular distance estimation combined with object detection algorithms like YOLOv8, have emerged as powerful tools for robotic arm pick-and-place tasks. Monocular distance estimation provides a cost-effective solution for spatial localization by leveraging a single camera, making it ideal for low-cost robotic systems. YOLOv8, with its real-time detection capabilities and robust performance, enables precise identification and localization of objects in cluttered environments.**

However, integrating these techniques into low-cost robotic systems poses unique challenges due to computational constraints and hardware limitations. This paper focuses on the application of monocular distance estimation and YOLOv8 in guiding robotic arms for pick-and-place operations. While the core emphasis is on optimizing the vision pipeline, we briefly address the importance of tailoring inverse kinematics models to the robotic arm's design to achieve smooth joint motions. The proposed approach demonstrates how accessible computer vision technologies can enhance automation capabilities in low-cost robotics, paving the way for more versatile and efficient robotic systems.

***Index Terms*—2D Object Detection, Monocular Depth Estimation, YOLOv8, Robotics, Computer Vision**

I. Introduction

In this section, you will introduce the topic of your paper, providing background information on the problem you're addressing. You should also state the primary objectives of your research and the significance of your work.

Briefly describe the structure of the paper, outlining the contents of each section. The introduction should capture the reader's attention and explain why the research is important.

II. Related Work

In robotic systems, object detection and depth estimation are critical as they enable robotic arms to perform pick and place tasks. Accurate perception of the object's location and its distance are essential for efficient and precise manipulation. Recent studies have focused on integrating these two tasks to streamline robotic workflows and improve operational accuracy. Below, we explore key approaches in these domains, emphasizing their strengths, weaknesses and relevance to our work.

A. Object Detection

For the robotic arm to be able to identify and localize objects within a scene, Object Detection must be a vital component. Traditionally, researchers use classical computer vision to do object detection but over time, the object detection methods have evolved to advanced deep learning-based models.

III. Methodology

The methodology for the 2D object detection and monocular depth estimation system leverages the integration of YOLOv8 for object detection, a robotic arm for manipulation, and monocular depth estimation using a pinhole camera model. The approach is designed to detect an object from a monocular camera feed, estimate its depth, and position a robotic arm to interact with the object. The main components of the system are described below.

A. ROS Setup and YOLOv8 Model Integration

The system is developed within the Robot Operating System (ROS) framework, which handles the communication between the various system components, including the camera, object detection model, and robotic arm. A ROS node is initialized to facilitate the interaction between the software modules. The YOLOv8 Nano model is used for object detection, where a fine-tuned model file is loaded for real-time object detection. YOLOv8, known for its efficiency and accuracy in detecting objects in images, is employed due to its ability to process high-resolution images with low computational cost, making it suitable for robotic vision tasks [1].

B. ROS Subscriptions and Publications

The system subscribes to the `/camera/color/image_raw` topic for real-time image input from the camera, which captures RGB images of the environment. The object detection module processes these images to identify objects. The system also publishes joint commands to the robotic arm through the `/armX_joint/command` topics, which are used to move the arm joints and control the gripper for object manipulation.

C. Camera and Arm Parameters

A set of known parameters is defined to relate the camera's measurements to the robot's workspace where the camera's intrinsic parameters, including focal length and camera height, are defined for the depth estimation process. The arm parameters, such as segment lengths and tilt angles, are used to transform the camera measurements into the robot's coordinate system. These parameters are crucial for precise arm movement and ensuring the correct positioning of the gripper to the detected object.

D. Initial Positioning of the Robotic Arm

Before the object detection and manipulation process, the robotic arm and gripper are moved to a *ready* position which ensures that the arm begins each task from a consistent and safe starting point. The arm is set to a default position using joint commands, which minimizes the risk of collisions and ensures smoother task execution.

E. YOLO Object Detection

The YOLOv8 model detects objects in the camera feed by processing the raw image frames. When an object of interest is detected, the model outputs the bounding box dimensions, which represent the position and size of the object in the image frame. YOLOv8's real-time detection capabilities make it well-suited for dynamic environments where object positions may change rapidly.

F. Distance Estimation

To estimate the object's distance from the camera, the system applies the pinhole camera model, which uses the object's size in pixels to calculate its real-world distance. The formula for distance estimation relies on the known width of the object and the focal length of the camera:

$$D = \frac{W_f}{W_p}, \tag{1}$$

where D is the distance to the object, W_f is the real-world object width, and W_p is the object's width in pixels. This distance estimation approach is commonly used in monocular depth estimation tasks [2].

G. Coordinate Transformation

Once the object is detected, the system transforms the 2D image coordinates into the 3D robot workspace coordinates. The camera's offset and tilt angle are taken into account to adjust for any misalignment between the camera and the robot's base frame. The horizontal and vertical offsets are calculated based on the camera's intrinsic parameters and the object's position in the image. These adjustments are necessary to align the detected object's position with the robot's coordinate system.

The horizontal offset, Δx, is given by:

$$\Delta x = (x_{\text{obj}} - x_{\text{center}}) \cdot k, \tag{2}$$

where x_{obj} is the object's horizontal position in the image, x_{center} is the image's center, and k is a scale factor derived from the camera's parameters.

H. Arm Positioning and Angle Calculation

To move the robotic arm to the object, the system calculates the required joint angles based on the object's position and the robot's arm kinematics. Using the law of cosines, the system computes the angles needed to position the gripper at the object's location in 3D space:

$$\theta = \cos^{-1}\left(\frac{m^2 + n^2 - d^2}{2mn}\right), \tag{3}$$

where m and n represent the transformed coordinates of the object, and d is the distance to the object. These calculations allow the robotic arm to adjust its joints and position the gripper accurately at the object's center.

I. Pickup and Manipulation

Once the arm is positioned correctly, the gripper is commanded to open, move to the object's location, and close around the object. The system then returns the arm to its ready position, completing the object manipulation task. These operations are controlled through ROS joint commands, ensuring precise and safe arm movement.

J. Safety and Calibration

Throughout the system, safety protocols are incorporated to ensure safe interaction with objects and the environment. The system's joint angles are clamped to a safe range to avoid damaging the robotic arm. Additionally, the camera calibration and depth estimation parameters are periodically validated to maintain system accuracy.

IV. Experiments and Results

This section should include the experimental setup, evaluation metrics, and results obtained from running your experiments. Use tables, figures, and charts to present the results. You should also provide an analysis of the results, comparing them to other approaches or baselines if applicable.

V. Limitation

A. Dependency on Image Quality

The accuracy of both object detection and depth estimation is very sensitive to the quality of input images. The performance of the system can significantly be degraded by low-resolution images, poor lighting conditions, motion blur and occlusions. For example, YOLOv8's object detection and depth estimation model will struggle to extract precise depth details in images with overexposed or underexposed lighting, hence dropping the accuracy. This limitation emphasizes the need for robust pre-processing techniques and the capability of a model to adapt to challenging visual environments.

B. Computational Resource Requirements

Powerful computational resources are essential for a combination of frameworks like object detection and depth estimation especially in real-time application. Though YOLOv8 is readily optimized for speed, having depth estimation incorporated alongside it amplifies the computational burden. This limitation is vital for deployment on devices that are limited from resources such as drones, mobile platforms, or embedded systems. However, it can be overcome by implementing efficient hardware acceleration such as high-end GPUs or specialized AI processors

VI. Conclusion

The conclusion summarizes the main findings of the paper, discusses their implications, and suggests potential future work. You may briefly mention any limitations of your study and how these could be addressed in future research.

References

[1] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016. [Online]. Available: https://arxiv.org/abs/1612.08242

[2] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," 2017. [Online]. Available: https://arxiv.org/abs/1609.03677