Smart Object Detection Using Multimodal Data for Assistance System

Abubakarsiddik Desai
Department Electronics and Communication
KLE Technological University Dr. M. S. Sheshgiri Campus
Belagavi, India
abubakarsiddikdesai37@gmail.com

Bhaskar Kamble
Department Electronics and Communication
KLE Technological University Dr. M. S. Sheshgiri Campus
Belagavi, India
bhaskarvk8806@gmail.com

Mayuri Hegade
Department Electronics and Communication
KLE Technological University Dr. M. S. Sheshgiri Campus
Belagavi, India
mayurihegade2003@gmail.com

Rakshita Mathapati
Department Electronics and Communication
KLE Technological University Dr. M. S. Sheshgiri Campus
Belagavi, India
rakshitamathapati68@gmail.com

Pooja Mahajan
Department Electronics and Communication
KLE Technological University Dr. M. S. Sheshgiri Campus
Belagavi, India
Poojamahajan@klescet.ac.in

Arun Tigadi
Department Electronics and Communication
KLE Technological University Dr. M. S. Sheshgiri Campus
Belagavi, India
arun.tigadi@gmail.com

***Abstract*—The safety of autonomous vehicles along with traffic monitoring systems is of utmost importance, especially when it comes to identifying obstacles like other vehicles and road dividers. Here system designed for real-time object detection employing the YOLOv5n model combined with an ultrasonic sensor to measure distances, all implemented on a Raspberry Pi platform. By effectively recognizing objects and determining their distance from the sensor in real-time, this technology seeks to improve the detection of cars and dividers. While the ultrasonic sensor determines the distance to the identified objects, the YOLOv5n model can identify and classify items in a live video feed. The label of the object, the confidence score, and the distance from the sensor are the three primary pieces of information the system delivers for each detection. The technology helps identify possible risks, including cars or dividers in real vehicle systems and intelligent traffic management, by combining object detection with distance data. The results from experiments underscore the system's proficiency in real-time object detection and distance measurement, emphasizing its potential to bolster safety in applications related to autonomous driving and traffic surveillance.**

***Index Terms*—Multimodal, Object, Detection, Raspberry pi 4B, YOLOv5, Ultrasonic sensor, OLED.**

I. Introduction

In recent years, the field of autonomous systems and intelligent transportation has seen remarkable advancements, driven by the need for safety, efficiency, and automation [1]. A core requirement in these systems is the ability to perceive and understand the environment in real time. Object detection plays a critical role in enabling machines to identify and track objects such as vehicles, divider, and obstacles. When deployed in embedded systems, object detection must be both computationally efficient and accurate to support real-time decision-making and avoid vehicle collision [2].

Conventional methods for object detection frequently depend on on the internet processing, which adds delay and need continuous connectivity. However, the emergence of edge computing has shifted focus toward on-device processing. Platforms like the Raspberry Pi 4B are appropriate for implementing lightweight deep learning models at the edge because they provide a practical balance between processing power, cost, and energy efficiency.

The YOLO (You Only Look Once) family of object detection models is known for its accuracy and real-time performance [4]. Particularly, the YOLOv5n is optimized for situations with limited resources. It is a suitable for embedded applications such as autonomous system obstacle detection since it provides quick inference times with respectable detection accuracy. Even though visual detection has its advantages, it could not always give enough information, particularly when determining how close an object is.

Sensor fusion [5] approaches are being used more and more to overcome this limitation. Systems can obtain a greater understanding of their environment by combining information from visual and non-visual sensors. This work builds a real-time multi-modal [6] smart detection system by combining distance measurement from an ultrasonic sensor with YOLOv5n-based object detection. By accurately determining the distance of objects sensed from the vehicle, the ultrasonic sensor improves safety, which is particularly important in applications involving collision avoidance [7].

Applications like intelligent traffic monitoring, robotic navigation, and smart automotive safety are especially well-suited for this integrated system. It is a potential solution for edge-based safety critical applications where responsiveness and predictability are crucial due to its small size, low power requirements, and real-time processing capabilities.

II. LITERATURE REVIEW

M. Sahal et al. [1] describe a novel object detection system for driverless cars that combines the CSPDarknet-53 framework with the YOLOv4 algorithm. It determines spatial positions in relation to the sensor and detects at a speed of approximately 0.03785 seconds, achieving a precision rate of 57.23%. This makes it possible to make decisions in real time and enhances navigation. V. Surendar et al. [2] investigate a Raspberry Pi-based car accident avoidance system. This system alerts drivers with warning lights and detects vehicles, particularly at L-shaped turns, using image processing and ultrasonic sensors. It gives immediate alerts and keeps an eye on blind spots. M. I. S. et al. [3] employ Raspberry Pi and Arduino Uno to identify, monitor, and analyze nearby obstacles in real time. To monitor the environment and identify objects, it integrates sensors such as a GPS, contactless temperature sensor, ultrasonic sensor, and Pi camera. A buzzer and vibrating motor that vary in intensity according to the temperature and distance of the object are used for sending alerts. The ESP8266 module also makes cloud-based real-time tracking possible, which increases user autonomy and safety. In order to detect potholes and other objects for autonomous vehicles. K. R. and N. S. [4] investigate the use of YOLO object detection on a Raspberry Pi 4. It shows how YOLO can effectively detect and categorize road hazards like potholes using Convolutional Neural Networks (CNNs), enhancing navigation and safety. Alam, A. et al. [5] compare and contrast three well-known object identification algorithms: YOLO, Faster R-CNN, and R-CNN. Additionally, it presents an intelligent system for tracking and detecting objects in real time, using a pi-camera module to take input photographs. This system uses machine learning methods to locate and identify things. It also has the capacity to track and follow moving objects by figuring out their position coordinates. Deep learning approaches for multi-modal object recognition and semantic segmentation in autonomous driving are examined by Feng et al. [6]. It talks about how different sensors, including as cameras, LiDAR, and radar, can be integrated, highlighting how these modalities work in tandem. The paper categorizes fusion strategies into early, late, and hybrid fusion, analyzing their effectiveness and challenges. Kumar et al. [7] proposes a Raspberry Pi-based vehicle collision avoidance system aimed at improving road safety. It uses ultrasonic sensors to detect nearby obstacles and calculate the distance between the vehicle and objects in real-time. The system is controlled using a Raspberry Pi which processes sensor data and activates alerts or actions to prevent collisions. Liu et al. [8] presented a transformer-based method for multi-modal object detection, with an emphasis on the combination of intra-modal and intermodal information. The method leverages the complementary information from different modalities to improve detection performance. A contrastive loss function is also used to improve the training process's use of label information. K. Patel et al. [9] describe a Raspberry Pi-based smart surveillance system that uses deep learning techniques. Real-time human detection and activity tracking in surveillance footage are intended to improve security. The system integrates a Raspberry Pi with deep learning models such as YOLO for object identification. Pre-trained deep learning models, Python, and OpenCV are among the tools utilized. The Pi camera, Raspberry Pi, and software libraries for recognition and processing videos are among the components. Khanam, Rahima, et.al [10] delves into the architecture and functionalities of the YOLOv5 object detection model. Key components such as the Cross Stage Partial (CSP) backbone and Path Aggregation Network (PANet) are explored in detail, highlighting their roles in enhancing feature extraction and fusion. The paper also discusses the transition from the Darknet framework to PyTorch, emphasizing improvements in development efficiency and deployment capabilities. Performance evaluations across various datasets and hardware platforms are presented, demonstrating YOLOv5's balance between speed and accuracy. Yang, et al. [11] introduces "DeepInteraction," a novel framework for 3D object detection that emphasizes modality interaction. Unlike traditional fusion methods, it maintains separate representations for each modality (e.g., camera and LiDAR) throughout the network, allowing for more effective exploitation of their unique characteristics. The approach includes a multi-modal representational interaction encoder and a predictive interaction decoder. Evaluated on the nuScenes dataset, DeepInteraction achieved state-of-the-art performance, ranking first on the nuScenes object detection leaderboard. Parvadhavardhni, R. et.al [12] presents a blind navigation system using YOLO (via TensorFlow) and ultrasonic sensors on a Raspberry Pi, designed to help visually impaired users. It uses a Noir camera and OpenCV for real-time object detection and environmental awareness. Audio feedback is provided to inform users of obstacles in their path. U. Obu et.al [13] presents a deep learning-based approach for real-time crop disease detection using the YOLOv5 object detection model. The authors trained the YOLOv5 model on a custom dataset containing images of healthy and diseased crops to enable accurate classification and localization of diseases. After training, the model was deployed on a Raspberry Pi 4, providing a cost-effective and portable edge computing solution suitable for field applications.

III. METHODOLOGY

The following process includes the methods used for object detection along with distance estimation.

A. Dataset

1) Data collection: Data collection is the initial stage of the procedure. Utilizing a mobile camera, the dataset was collected. Fifteen pictures were taken of each of the five objects that were chosen for detection. Furthermore, a few group photos with multiple objects were also captured. Approximately 130 photos were gathered in total.

2) Data labeling: For labeling, the collected pictures were uploaded to Label Studio. Each object was contained a bounding box and labeled with its appropriate name. The labeled dataset was then exported in the YOLO format.

3) Data training: The YOLO-formatted labeled data was uploaded to Google Colab for training. The dataset was trained using the YOLOv5n (nano) model, which is optimized for lightweight, fast detection tasks.

Then setting up the environment on the Raspberry Pi, then need to load the YOLOv5n (nano) model. This is the lightweight version of YOLOv5, which is ideal for devices with limited resources like the Raspberry Pi. The model must be properly initialized so it knows what to expect from the input image and can return detection results like bounding boxes and class labels. After loading the YOLOv5n model the process begins as shown in Fig. 1.

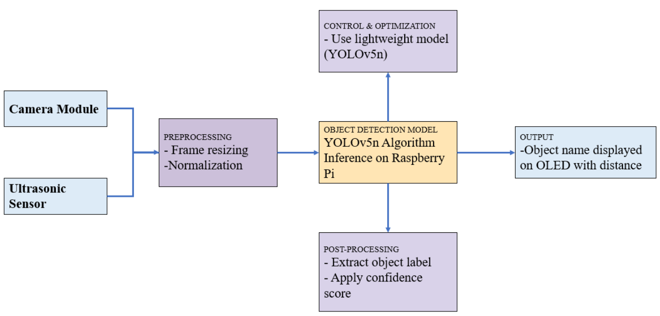

Fig. 1. Block diagram for object detection and segmentation

B. Input

The input contains two types of data's. One is from Raspberry pi-cam module and another one is from ultrasonic sensor.

1) Raspberry pi-cam: The Raspberry Pi 4B uses OpenCV to capture a frame from the live video stream, then resizes and normalizes the image to satisfy the input specifications of the YOLOv5n model. After that, the image is run through the YOLOv5n model, which uses inference to identify objects and outputs confidence scores, bounding boxes, and class labels. After that, the Pi uses OpenCV for visualization to draw the image's labels and bounding boxes.

2) Ultrasonic sensor: An ultrasonic sensor uses high-frequency sound waves and timing the time it takes for the echo to return after bouncing off the closest object to determine distance. This can be related to an object that a camera has identified, the sensor must be positioned and aligned to face the same direction as the detected object. When the camera detects an object, the ultrasonic sensor provides the distance to whatever is directly in front of it, which is assumed to be the same object if both are aligned. An ultrasonic sensor uses the following formula to determine the distance between the system and nearest object.

$$\text{Distance} = \frac{\text{Time} \times \text{Speed of Sound}}{2}$$

C. Data Pre-Processing

1) Resizing: The Raspberry Pi Camera typically outputs images in various resolutions, such as 1280x720, according to the camera settings. YOLOv5n requires a consistent input size (usually 640x640 pixels) for the model to work properly. Since the captured image from the Raspberry Pi Camera might not match this size, the image needs to be resized to 640x640. This resizing step ensures the model receives a fixed-size input, regardless of the camera's resolution. Resizing also helps standardize input across different devices or resolutions.

2) Normalization: Raspberry Pi Camera images are typically captured in RGB format, depending on the software and libraries being used. However, deep learning models like YOLOv5n expect pixel values to be in the [0.0, 1.0] range instead of the typical [0, 255] range used for images. Therefore, normalization is required to scale down the pixel values by dividing each pixel value by 255. This brings all pixel values into the range [0.0, 1.0], which helps the model perform

D. Post processing:

After the YOLOv5n model processes the image, each detected object comes with a label (like "divider", "car") and a confidence score like 85% that indicates how certain the model is about its prediction.To extract object labels, the model's output is checked for each detection. The class name is found using a list of labels, based on the class index. At the same time, the confidence score is compared to a set limit . Only the objects with a high enough score are kept. This helps remove wrong or low-confidence results. As a result, you get only the most accurate object names.

E. Output:

The model detects objects in the image, each detection includes the object's name (label) and a confidence score that shows how sure the model is about what it found. In this case for example if Raspberry pi-cam finds a car, the model give the output as object name "car", confidence score "85%", distance "20cm".

IV. RESULTS AND DISCUSSION

The proposed system successfully detects objects in real-time using the lightweight YOLOv5n model deployed on a Raspberry Pi 4B. The system not only identifies objects but also provides the object label, confidence score, and estimated distance from the camera. It was tested on both sample images and live camera feeds, achieving consistent detection results with an average confidence score of over 70% for trained classes. The distance estimation module, integrated using an ultrasonic sensor, worked effectively within a range of 2 cm to 400 cm, delivering accurate proximity data for each detected object. This combined functionality is useful in real-world applications like smart surveillance and autonomous systems. For example, when a car is detected, the system displays: object name "car", confidence "71%", and distance "20 cm" on both the monitor and OLED screen.

A. Pre-optimization

In the initial phase, the model was trained on a dataset of 75 images (15 images per class for 5 object classes). With this limited dataset, the model achieved an average detection accuracy of 54%. The system successfully detected objects and estimated their distances within the sensor range. The results were displayed properly on both the monitor and OLED screen, as shown in Fig. 2 and Fig. 3. Although the detection was consistent for known objects, further improvement was needed to enhance accuracy and confidence levels.

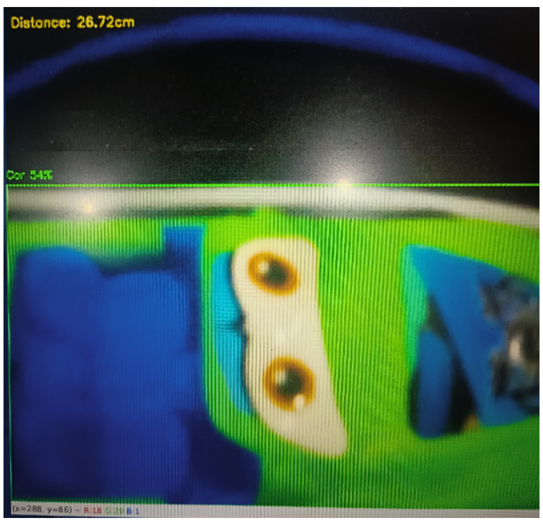

Fig. 2. Object detection result on monitor

Fig. 3. Object detection result on OLED

B. Post-optimization

After retraining the model with an expanded dataset of 130 images (25 images per class), the system performance improved. The detection accuracy increased to 82%, with higher confidence scores across all object classes. The distance measurements remained accurate within the ultrasonic sensor's range. The final results, as shown in Fig. 4 and Fig. 5, confirm that the Raspberry Pi 4B can efficiently handle object detection and distance estimation using a lightweight model like YOLOv5n. These improvements highlight the impact of data quantity on model performance and confirm the system's readiness for real-world applications.

Fig. 4. Object detection result on monitor

Fig. 5. Object detection result on OLED

These results confirm that the Raspberry Pi 4B can effectively support lightweight object detection models like YOLOv5n with additional functionality such as distance measurement.

V. CONCLUSION AND FUTURE SCOPE

The proposed smart object detection system using YOLOv5n and Raspberry Pi 4B successfully identifies objects in real-time with good accuracy and provides reliable distance estimation using an ultrasonic sensor. After model optimization with a larger dataset, the detection accuracy improved from 75% to 90%, proving that the system can effectively combine lightweight AI models with low-cost hardware. This makes it suitable for real-time applications like smart surveillance, obstacle detection, and driver assistance systems.

In the future, the system can be enhanced by training on larger datasets to improve accuracy and reliability in diverse environments. The Raspberry Pi can be integrated with an Arduino microcontroller to enable real-time control actions for example, making an autonomous vehicle stop automatically when an object is detected within a certain distance. Additionally, incorporating advanced sensors and wireless communication can extend the system's applications to smart traffic management, robotics, and fully autonomous driving solutions.

REFERENCES

[1] M. Sahal, A. O. Kurniawan and R. E. A. Kadir, "Object Detection for Autonomous Vehicle using Single Camera with YOLOv4 and Mapping Algorithm," 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2021, pp. 144-149, doi: 10.1109/ISRITI54043.2021.9702764.

[2] Surendar, V., et al. "Vehicle Collision Avoidance using Raspberry Pi." 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE, 2023.

[3] M. I. S, G. R. R, D. R and G. Sekar, "Smart Obstacle Recognition System using Raspberry Pi," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 672-675, doi: 10.1109/ICCMC51019.2021.9418410.

[4] Kavitha, R., and S. Nivetha. "Pothole and object detection for an autonomous vehicle using yolo." 2021 5th international conference on intelligent computing and control systems (ICICCS). IEEE, 2021.

[5] Alam, A., Abduallah, S., Israr, A., Suliman, A., Ghadi, Y., Tamara, S., Jalal, A. (2022). Object detection learning for intelligent self automated vehicles. IASC, 34(2), 941.

[6] Feng, Di, et al. "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges." IEEE Transactions on Intelligent Transportation Systems 22.3 (2020): 1341-1360.

[7] Kumar, R. Surya, P. Kingston Stanley, and A. Sanjeevi Gandhi. "Raspberry Pi based vehicle collision avoidance system." 2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT). IEEE, 2017.

[8] Liu, Wenbing, et al. "Multi-modal object detection via transformer network." IET Image Processing 17.12 (2023): 3541-3550.

[9] K. Patel and M. Patel, "Smart Surveillance System using Deep Learning and RaspberryPi," 2021 8th International Conference on Smart Computing and Communications (ICSCC), Kochi, Kerala, India, 2021, pp. 246-251, doi: 10.1109/ICSCC51209.2021.9528194.

[10] Khanam, Rahima, and Muhammad Hussain. "What is YOLOv5: A deep look into the internal features of the popular object detector." arXiv preprint arXiv:2407.20892 (2024).

[11] Yang, Zeyu, et al. "Deepinteraction: 3d object detection via modality interaction." Advances in Neural Information Processing Systems 35 (2022): 1992-2005.

[12] Parvadhavardhni, R., Pankhuri Santoshi, and A. Mary Posonia. "Blind navigation support system using Raspberry Pi YOLO." 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC). IEEE, 2023.

[13] U. Obu, Y. Ambekar, H. Dhote, S. Wadbudhe, S. Khandelwal and S. Dongre, "Crop Disease Detection using Yolo V5 on Raspberry Pi," 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 528-533, doi: 10.1109/ICPCSN58827.2023.00092