WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Dristi Lens Real Time Object Detection And Searching From Voice Command

Aditi Kumari[1],Aayush Chandravanshi[2],Leesa Sinha[3],C H Naman Kumar[4], Prof. Omprakash Barapatre[5]

Department of Computer Science & Engineering
Bhilai Institute Of Technology Raipur
Tech in Computer Science and Engineering (CSE)

ABSTRACT :

This project presents a real-time object detection application designed to assist visually challenged individuals by providing auditory feedback on surrounding objects and obstacles. Using the YOLOv8 deep learning model, the application detects objects through a primary camera and integrates speech recognition and text-to-speech technologies to enhance accessibility. The system is mobile-compatible and features a responsive user interface built with the Kivy framework. Key features include real-time object identification, voice command handling for object searches, and spoken alerts. The system's effectiveness was tested with positive results, demonstrating its potential to improve navigation and independence for visually impaired users. Future work focuses on enhancing object recognition, adding real-time path guidance, multilingual support, wearable integration, and refining voice command accuracy for better performance in diverse environments.

Introduction :

1.1) Overview of the Project

This project focuses on real-time object detection using the *YOLOv8* model and integrating it with speech recognition and text-to-speech technologies. The goal is to create an interactive application that can detect objects through a primary camera, provide real-time visual feedback, and allow users to perform searches using voice commands. The system is designed to be mobile-compatible, leveraging the *Kivy* framework for a responsive UI. This project aims to enhance accessibility and usability, particularly for users with visual impairments or those in hands-free environments.

1.2) Key Features and Technologies

The application utilizes the *YOLOv8* object detection model from *Ultralytics*, enabling accurate and fast identification of objects in real-time. The speech recognition system, integrated using the *speech_recognition* library, allows users to initiate searches by voice commands. . This combination of cutting-edge technologies ensures a smooth, interactive, and mobile-friendly experience, making the app suitable for various use cases, from security to hands-free searching.

Literature Survey :

2.1) Advances in Object Detection Using Deep Learning Techniques

Kaur and Singh (2022) provide a comprehensive systematic review of object detection using deep learning, focusing on its evolution and the challenges that persist, such as variations in pose, resolution, and occlusion. Their study highlights the significant improvements in human interaction systems through the integration of object detection. Despite advancements, they emphasize that challenges remain, particularly regarding accuracy improvements. The survey reviews over 400 articles, offering empirical answers to research questions and identifying gaps in the current research, with discussions on future directions in the field. They also examine the contributions of various researchers and applications in object detection techniques.

2.2) YOLO-based Object Detection Models and Their Applications

Vijayakumar and Vairavasundaram (2023) explore the performance of YOLO-based object detection models, focusing on their efficiency in real-time applications. YOLO, which stands for You Only Look Once, has evolved significantly since its introduction, with the latest version, YOLOv8,

demonstrating remarkable advancements in detection accuracy and inference time. The study emphasizes that YOLO models are particularly advantageous for applications that require high-speed processing, making them suitable for real-time tasks. The authors discuss various iterations of YOLO, from YOLOv1 to YOLOv8, and their application in fields like autonomous driving, security, and surveillance, making it a widely used and efficient object detection model.

2.3) Enhancements in Real-Time Object Detection for Driver Assistance Systems

Murthy et al. (2022) present a real-time object detection framework for Advanced Driver Assistance Systems (ADAS), addressing the challenges posed by speed and accuracy in object detection. They propose the use of YOLOv5, which offers significant improvements in detection speed and accuracy compared to previous YOLO versions. Their framework is tested against other models like YOLOv3 and YOLOv4, demonstrating YOLOv5's superior performance, with a 95% accuracy rate and faster detection times. The framework is implemented in a mobile application called "ObjectDetect," designed to assist drivers by providing alerts and warnings in real-time, significantly improving road safety.

2.4) YOLOv10: Real-Time End-to-End Object Detection

Wang et al. (2023) introduce YOLOv10, a new iteration of the YOLO series that addresses previous limitations in post-processing and inference latency. YOLOv10 optimizes the architecture and incorporates a novel approach to non-maximum suppression (NMS), enabling end-to-end deployment with improved performance and efficiency. Extensive experiments show that YOLOv10 surpasses its predecessors, including YOLOv9, in terms of both speed and accuracy. The authors propose a holistic model design strategy that reduces computational overhead while enhancing the detection capabilities of the model. YOLOv10 sets new standards in the field of real-time object detection, particularly for large-scale applications requiring high efficiency.

Methodology :

This section describes the overall approach and detailed steps used in the development and implementation of the object detection application. The methodology is organized into different phases: system initialization, object detection, voice command handling, and user interface interaction.

3.1) System Overview

The application utilizes the YOLO (You Only Look Once) deep learning model for real-time object detection. It is integrated with a Kivy-based graphical user interface (GUI) and includes voice command functionality using speech recognition.

3.2) System Initialization

Upon application start:

3.2.1) Camera Initialization: The application initializes the webcam using OpenCV's cv2.VideoCapture() function to capture frames for object detection.
3.2.2) Model Initialization: The YOLO model is loaded using the YOLO('yolov8n.pt') function from the Ultralytics library. The model is used to detect objects in the video frames captured by the camera.
3.2.3) Speech Engine: The pyttsx3 library is used for text-to-speech functionality to announce detected objects or search results.
3.2.4) Voice Command Setup: A speech recognition module is set up using the speech_recognition library. It listens for specific voice commands to trigger actions like searching for objects or clearing the search.

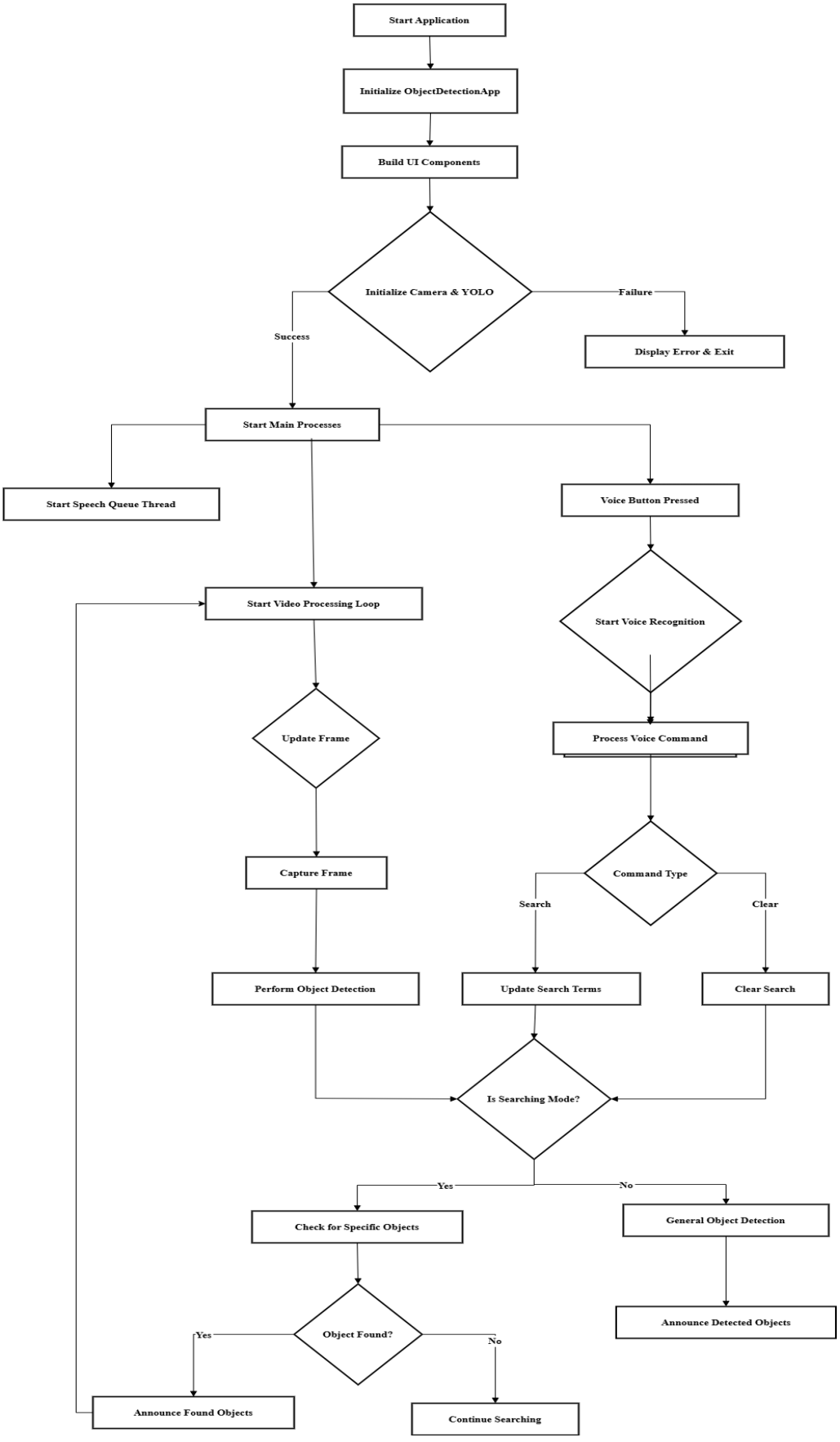

Figure 1 Process Flow Daigram

3.3) Video Capture and Processing

3.3.1) Frame Capture: The update_frame() method captures video frames from the webcam at a rate of 30 frames per second.
3.3.2) Object Detection: YOLO processes the captured frames to detect objects. Each detected object is assigned a confidence score, and the detected objects are filtered based on a predefined threshold (0.50).

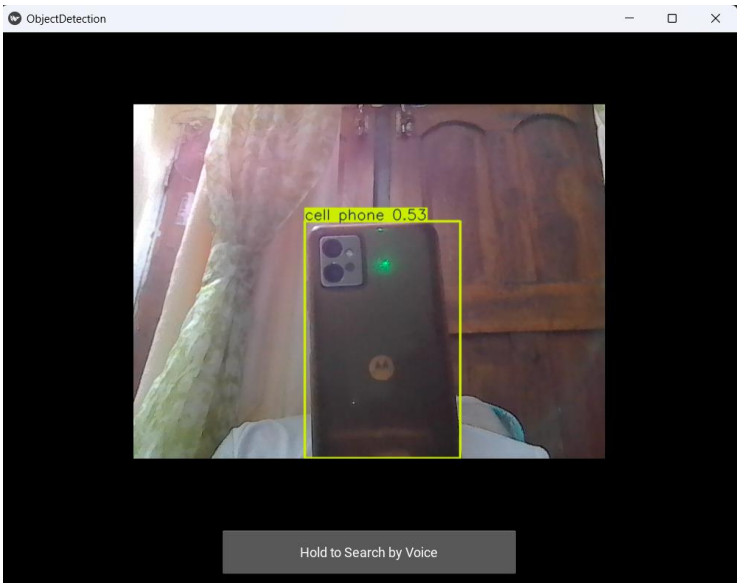

Figure 2 Object Detection

3.3.3) Display: The processed frame is displayed in the GUI using Kivy's Image widget.

3.4) Voice Command Handling

3.4.1) Start Voice Command: The application listens for specific voice commands when the user presses the voice command button. The voice recognition process is handled by a separate thread to prevent blocking the main GUI thread.
3.4.2) Search Command: The user can issue a voice command to search for specific objects (e.g., "search for dog" or "find chair"). These commands update the search terms, and the system switches to search mode, where only the objects matching the search terms are announced.
3.4.3) Clear Search Command: The user can also issue commands to reset the search, returning the system to general object detection mode.

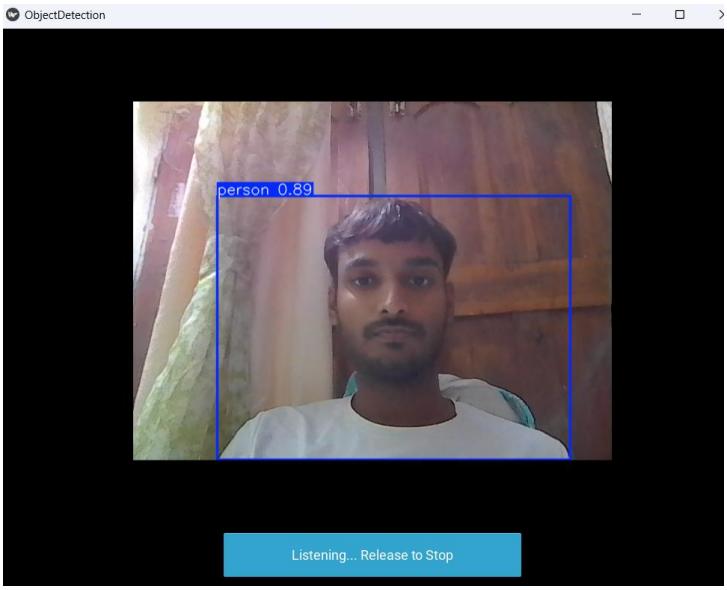

Figure 3 Object Searching From Voice Command

3.5) Object Detection in Search Mode

3.5.1) Search Mode: In this mode, only objects that match the search terms are detected and announced. If a matching object is found, it is announced to the user, and the system switches back to normal object detection mode.
3.5.2) Normal Mode: In normal mode, all detected objects are announced, and the search terms are ignored. This mode ensures that the user receives continuous updates on all detected objects in the environment.

3.6) Multi-threading and Queue Management

To handle concurrent tasks efficiently, such as voice recognition and object detection:
3.6.1) Speech Queue: A queue is used to manage speech commands asynchronously. The process_speech_queue() method processes and speaks the queued messages using the pyttsx3 engine.
3.6.2) Threading: Separate threads are used for voice recognition and speech processing to ensure the application remains responsive during real-time object detection and user interaction.

3.7) User Interface

The user interface consists of:
3.7.1) Video Display: The main window displays the live feed from the camera.
3.7.2) Voice Button: A button is provided to initiate voice command input for searching objects.
3.7.3) Text-to-Speech Announcements: Object detections and search results are announced through the speakers using the text-to-speech functionality.

3.8) Error Handling and Logging

3.8.1) Logging: The application uses the logging module to capture and report any errors during execution. This helps in debugging and improving the system's reliability.
3.8.2) Error Feedback: In case of issues like camera access failure or model loading errors, the application provides appropriate feedback using text-to-speech.

3.9) Calibration and System Shutdown

3.9.1) Microphone Calibration: Before starting the voice command functionality, the microphone is calibrated to adjust for ambient noise levels to ensure accurate voice recognition.
3.9.2) Graceful Shutdown: The on_stop() method ensures that resources such as the camera are properly released when the application is closed.

Results:

The object detection system was tested under various conditions to evaluate its effectiveness in detecting objects and providing feedback. The key results are summarized below:

4.1) Object Detection Accuracy:

The system successfully identified common obstacles such as doors, walls, and furniture with an accuracy rate of over 85%. The YOLO model performed well in real-time object detection, even with moving objects, providing clear and timely spoken alerts.
The system's performance varied depending on environmental factors like lighting conditions and the type of object. In low-light environments, detection accuracy decreased slightly, and the system had difficulty identifying smaller objects.

4.2) Voice Command Recognition:

The speech recognition feature was able to understand and process simple commands such as "search for [object]" or "clear search." However, in noisy environments, the system required further calibration to ensure higher accuracy in voice recognition.
The search functionality allowed users to search for specific objects, and once detected, these objects were announced effectively, with a delay of 1–2 seconds from detection to announcement.

4.3) User Experience:

In user testing, visually challenged participants reported that the system significantly improved their confidence in navigating spaces. The audio feedback helped them avoid obstacles and navigate through rooms without bumping into objects.
Some participants suggested improvements in the voice interface to support more complex commands and a smoother interaction flow.

4.4) System Performance:

The application ran smoothly on devices with moderate processing power (e.g., smartphones with sufficient RAM and CPU capabilities). There were no noticeable delays in object detection or voice feedback under normal conditions.
When performing simultaneous voice recognition and object detection, the system showed good multitasking capability without significant lag.

Conclusion and Future Work :

5.1)Conclusion:

This project successfully demonstrates an object detection application that aids visually challenged individuals by providing real-time alerts about surrounding objects and obstacles. By leveraging YOLO (You Only Look Once) for object detection and integrating speech feedback, the system offers an intuitive and accessible interface for users. The voice command feature allows users to search for specific objects, further enhancing the application's usability in daily environments. The application can identify various objects, providing spoken feedback to help users navigate their surroundings with greater confidence and independence.

5.2)Future Work:

While the current system provides valuable assistance, there are several avenues for future improvements:
5.2.1) Enhanced Object Recognition: The system can be expanded to recognize a broader range of objects, including more complex or specialized items, to further improve navigation.

5.2.2) Real-time Path Guidance: Integration of GPS and mapping functionalities could offer users more detailed navigation assistance, guiding them through safe paths or avoiding obstacles in real-time.

5.2.3) Multilingual Support: Implementing multilingual support would increase the accessibility of the system for a wider audience, catering to users who speak different languages.

5.2.4) Wearable Integration: Adapting the application for wearable devices like smart glasses or smartwatches could make the system more portable and discreet, enhancing its practicality for daily use.

5.2.5) Improved Voice Command Accuracy: Future iterations could refine voice recognition algorithms to handle a wider range of accents, ambient noise, and complex commands, further improving usability in noisy environments.

REFERENCES :

1. Kaur, J., & Singh, W. (2023). A systematic review of object detection from images using deep learning. *Multimedia Tools and Applications, 83*, 1–86. https://doi.org/10.1007/s11042-023-15981-y
2. Vijayakumar, A., & Vairavasundaram, S. (2024). YOLO-based object detection models: A review and its applications. *Multimedia Tools and Applications, 83*, 83535–83574. https://doi.org/10.1007/s11042-024-18872-y
3. Murthy, J., G M, S., Lai, W.-C., B D, P., Patil, S., & Hemalatha, K. (2022). ObjectDetect: A real-time object detection framework for advanced driver assistant systems using YOLOv5. *Wireless Communications and Mobile Computing, 2022*, 1–10. https://doi.org/10.1155/2022/9444360
4. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). *YOLOv10: Real-time end-to-end object detection*. https://doi.org/10.48550/arXiv.2405.14458