High Speed Real-Time Object Detection Using YOLO-Based Deep Learning Models

Sirela Meena[1], Umamaheswararao Mogili[2], K. J. Pravalika[3],
P. Ravi Kiran[4], N. Naveen[5], Y. Chetan[6], Mongolo Goudo[7]
[1,2]*Assistant Professor, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India.*
[3,4,5,6,7]*B. Tech, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India.*

***Abstract*—Real-time object detection is an important research area in the field of Computer Vision that focuses on identifying and locating objects in images and video streams with minimal delay. With the rapid advancement of Artificial Intelligence and deep learning technologies, object detection systems have become increasingly accurate and efficient. These systems are widely used in various applications such as surveillance systems, autonomous vehicles, robotics, healthcare monitoring, smart city infrastructure, and assistive technologies for visually impaired individuals. The ability to detect objects in real time enables intelligent systems to respond quickly to changes in their environment. In recent years, deep learning-based detection algorithms have significantly improved the performance of object detection systems. One of the most popular algorithms is the You Only Look Once (YOLO) model, which performs object detection using a single neural network. Unlike traditional object detection techniques that require multiple processing stages, YOLO predicts bounding boxes and object class probabilities simultaneously in a single pass through the network. This approach greatly reduces computation time and enables real-time processing. The proposed system in this research utilizes the YOLO-based deep learning model to detect objects from live video streams. The system captures video input using a camera and processes the captured frames continuously. Each frame is analyzed by the trained YOLO model, which identifies objects present in the scene and draws bounding boxes around them. The detected objects are then labeled with their corresponding class names, such as person, car, bottle, chair, and other commonly recognized objects. The system is implemented using the Python programming language along with the OpenCV computer vision library. Python provides a flexible programming environment for integrating machine learning models, while OpenCV is used for capturing video frames, processing images, and displaying the detection results in real time. The combination of Python and OpenCV allows the system to process visual data efficiently and supports the implementation of advanced computer vision techniques.**

***Index Terms*—Object Detection, YOLO, Deep Learning, Real-Time Systems, Computer Vision, Text-to-Speech.**

I. INTRODUCTION

Object detection is a fundamental and challenging task in computer vision that involves identifying objects in images or videos and determining their spatial locations. Its applications are diverse, ranging from intelligent surveillance and autonomous navigation to medical image analysis and assistive technologies. Traditionally, object detection relied on handcrafted features and classical machine learning algorithms, which required extensive manual feature engineering and domain expertise. However, these methods often struggled in complex environments with background clutter, illumination variations, and occlusions. The emergence of deep learning, particularly Convolutional Neural Networks (CNNs), revolutionized object detection by enabling automatic feature learning from raw image data. CNN-based detectors showed remarkable improvements in accuracy and robustness, but early models prioritized accuracy over speed, making them unsuitable for real-time applications. YOLO (You Only Look Once) introduced a novel single-stage object detection paradigm that treats detection as a regression problem, achieving real-time performance while maintaining competitive accuracy. A proposed YOLO-based

system integrates text-to-speech technology to provide both visual and auditory feedback, improving system usability and accessibility.

II. LITERATURE REVIEW

Real-time object detection has become an important research area in computer vision with the rapid advancement of deep learning techniques. Modern object detection systems are widely used in applications such as autonomous driving, surveillance systems, traffic monitoring, robotics, and smart cities. Researchers have proposed various deep learning models to improve detection speed and accuracy. Among these models, the YOLO (You Only Look Once) architecture has gained significant attention due to its ability to perform object detection in real-time with high efficiency. Joseph Redmon and Ali Farhadi introduced the YOLO object detection framework, which treats object detection as a regression problem and predicts bounding boxes and class probabilities directly from images. Their research demonstrated that YOLO can process images at high speed while maintaining competitive detection accuracy compared to traditional region-based detection methods [1]. Similarly, Wei Liu et al. proposed the Single Shot MultiBox Detector (SSD), a deep learning-based object detection model designed for real-time applications. Their study compared SSD with YOLO and concluded that both approaches provide efficient detection performance for real-time systems [2]. Tsung-Yi Lin et al. developed the COCO dataset and evaluation metrics that are widely used to benchmark object detection models including YOLO. Their work significantly contributed to improving training datasets and evaluation standards for object detection research [3]. In another study, Kaiming He et al. introduced Residual Networks (ResNet), which improved feature extraction in deep learning models. Their architecture has been widely used as a backbone network in modern object detection frameworks including advanced YOLO versions [4]. Ross Girshick proposed the Region-based Convolutional Neural Network (R-CNN) approach for object detection. Although highly accurate, the method required multiple stages for detection, which made it slower compared to YOLO-based models [5].

Anil K. Jain conducted extensive research in pattern recognition and machine learning, contributing significantly to image processing and object recognition systems that form the foundation of modern detection techniques [6]. Similarly, Rama Chellappa worked on computer vision and deep learning models for image understanding and recognition tasks, influencing modern object detection [7]. C. V. Jawahar explored deep learning approaches for scene visual recognition. His work highlighted the importance of large datasets and efficient neural networks for improving detection accuracy [8]. P. J. Narayanan conducted research on large-scale visual computing and deep learning-based image analysis systems that support real-time object detection applications [9]. In another research work, Mitesh M. Khapra investigated neural network architectures for computer vision applications, focusing on improving deep learning efficiency and scalability [10]. Partha Pratim Roy proposed several machine learning approaches for image processing and pattern recognition, which contribute to the development of modern object detection techniques [11]. Additionally, Subhasis Chaudhuri worked on image processing and machine learning algorithms for visual recognition systems, highlighting their applications in surveillance and intelligent systems [12]. Peiyuan Jiang et al. presented a comprehensive survey of YOLO algorithms and their improvements, including YOLOv3, YOLOv4, and YOLOv5. Their study concluded that YOLO-based models provide a good balance between detection speed and accuracy [13]. Another significant contribution was made by Xiaohan Cong et al., who analyzed the development and applications of YOLO algorithms in real-time detection systems. Their research highlighted the effectiveness of YOLO models in various practical applications such as surveillance and autonomous vehicles [14-16]. Finally, Yuhan Yan and Lin Zhang investigated the application of YOLO models for detecting objects in complex environments such as aerial imagery and traffic monitoring systems. Their study concluded that improved YOLO architectures significantly enhance detection performance in real-time scenarios [17, 18]. Some of the artificial intelligence, machine learning and deep learning models are described in details [19-23].

III. METHODOLOGY

The proposed system is designed to perform high-speed real-time object detection and provide audio feedback by announcing the names of detected objects. The system integrates deep learning-based object detection with speech generation techniques to create an interactive and intelligent detection system. The overall methodology consists of several stages including video capture, image preprocessing, object detection, object classification, and speech output generation. Initially, the system captures live video input through a camera connected to the computer system. The camera continuously records the surrounding environment and converts the visual information into digital frames. These frames are extracted sequentially from the video stream so that they can be processed individually. Continuous frame capturing allows the system to monitor the environment in real time and detect objects appearing within the camera view.After capturing the video frames, the system performs preprocessing operations to prepare the frames for object detection. In this stage, the captured frames are resized according to the input size required by the detection model. Image normalization techniques are also applied to adjust brightness, contrast, and pixel values. These preprocessing steps help improve the detection accuracy and ensure that the images are compatible with the deep learning model.

Once the preprocessing stage is completed, the processed frames are passed to the object detection model. The proposed system uses a YOLO-based deep learning model for detecting objects within the video frames. The YOLO algorithm is a single-stage detection model that predicts bounding boxes and object class probabilities in a single pass through the neural network. This design significantly reduces computational complexity and enables faster detection speed compared to traditional multi-stage detection methods. The YOLO model divides the input image into a grid structure. Each grid cell is responsible for detecting objects whose center falls within that cell. The model predicts multiple bounding boxes for each grid cell along with confidence scores indicating the probability that an object is present in that region. Additionally, the model predicts the class probabilities corresponding to different object categories. During the detection process, the model analyzes the visual features present in each frame and compares them with patterns learned during the training phase. The trained model has already learned to recognize different object categories such as person, car, bicycle, bus, cow, and several other objects commonly found in real-world environments. Based on this knowledge, the model can accurately identify objects appearing in the captured frames. Once the objects are detected, the system generates bounding boxes around the detected objects. These bounding boxes visually highlight the location of objects within the frame. Each bounding box is associated with a class label representing the name of the detected object and a confidence score indicating the reliability of the detection result. Only objects with confidence scores above a predefined threshold are considered valid detections.

The detected object labels are then extracted from the detection results and passed to the speech generation module. The speech generation module uses a text-to-speech mechanism to convert the object labels into spoken words. This module transforms the textual object names into audio signals that can be played through the system speaker. For example, if the detection model identifies a bus within the camera frame, the system extracts the label “bus” and sends it to the speech synthesis module. The speech module then generates an audio output announcing the word “bus.” This process enables the system to provide real-time audio feedback to the user. In cases where multiple objects are detected in a single frame, the system processes each detected object sequentially. The labels of detected objects are passed one by one to the speech module so that the system can clearly announce each detected object. This ensures that users receive accurate information about all objects present in the scene. The detection and speech generation processes operate continuously in a loop. As the camera captures new frames, the system repeats the detection process and updates the detection results dynamically. This continuous loop enables the system to detect newly appearing objects instantly and provide immediate audio feedback. To ensure smooth real-time performance, several optimization techniques are applied within the system. Frame resizing and efficient memory usage are used to reduce computational overhead. The system also uses optimized deep learning models that provide faster inference speed while maintaining acceptable detection accuracy.

Another important component of the system is the integration of the object detection model with the computer vision framework. The system is implemented using Python programming language along with the OpenCV library. OpenCV provides functions for capturing video frames, processing images, and displaying detection results on the screen. Using OpenCV, the system draws bounding boxes around detected objects and displays the object labels directly on the video frames. This visual output allows users to see the detected objects while simultaneously hearing the spoken object names through the audio output system. The proposed methodology is particularly useful for applications where real-time object awareness is required. For example, the system can assist visually impaired individuals by informing them about objects present in their surroundings through audio output. Similarly, the system can be used in surveillance systems to detect and monitor objects appearing in security camera footage. In addition to assistive applications, the system can also be applied in robotics and automation environments where machines need to recognize objects and interact with their surroundings. By combining object detection with speech feedback, the system provides an intuitive and interactive interface between machines and users. Overall, the proposed methodology integrates real-time object detection with speech generation to create an efficient and interactive system. The use of the YOLO deep learning model ensures high detection speed and accuracy, while the speech module enhances user interaction by providing audible feedback. This combination makes the system suitable for various real-world applications that require fast and reliable object detection.

IV. RESULTS AND DISCUSSION

The proposed system successfully detects multiple objects in real time and displays bounding boxes with class labels. The system maintains stable frame rates under normal lighting conditions. Audio feed back significantly enhances accessibility and user interaction, especially for visually impaired users. Detection accuracy decreases slightly for very small or heavily occluded objects, but overall performance demonstrates a good balance between speed and accuracy. Fig. 1 shows the detection result for a bus. The system successfully identified the bus in the input frame and displayed a bounding box along with the corresponding label. This demonstrates that the proposed detection model is capable of recognizing large objects accurately in real-time environments. Fig. 2 illustrates the detection of a bicycle using the YOLO-based object detection model.

Fig 1: Bus detected using the proposed real-time object detection system.

The system correctly detected the bicycle and labeled it with high confidence. The bounding box clearly indicates the location of the object within the frame, confirming the efficiency of the detection algorithm. Fig. 3 presents the detection of multiple objects in a single frame. In this example, the system successfully detected persons along with a cell phone. The ability to detect multiple objects simultaneously shows that the proposed system can handle complex scenes effectively. Fig. 4 shows the detection of a cow using the proposed system. The model accurately identified the cow and displayed the corresponding label with a bounding box. This result demonstrates that the system can detect animals and other objects present in the environment. From these results, it can be observed that the proposed system is capable of detecting different types of objects in real time with reasonable accuracy. The integration of the object detection model with the text-to-speech module further enhances the usability of the system by providing audio feedback for the detected objects.

Fig 2: Bicycle detected using the YOLO-based detection model.

Fig 3: Persons and a cell phone detected in the input frame.

Fig 4: Cow detected by the proposed system.

V. CONCLUSION

The project successfully implements a real-time object detection system using a YOLO-based deep learning model. The system captures live video input from a camera and processes each frame for object detection. The YOLO model identifies multiple objects in real time with high speed and accuracy. Detected objects are highlighted with bounding boxes and labels on the screen. In addition to visual output, the system converts object names into audio feedback using a text-to-speech module. This allows users to hear the detected object names without constantly looking at the display. The system performs detection continuously, ensuring smooth and low-latency operation. The model is capable of detecting several common objects present in the environment. The implementation is simple, efficient, and user-friendly. Therefore, the proposed system can be useful in assistive technologies, smart monitoring systems, and real-time computer vision applications. The system is implemented using the Python programming language along with the OpenCV computer vision library. Python provides a flexible programming environment for integrating machine learning models, while OpenCV is used for capturing video frames, processing images, and displaying the detection results in real time. The combination of Python and OpenCV allows the system to process visual data efficiently and supports the implementation of advanced computer vision techniques.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[5] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[6] W. Liu *et al.*, "SSD: Single shot MultiBox detector," in *Proc. European Conf. Computer Vision (ECCV)*, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. European Conf. Computer Vision (ECCV)*, 2014.

[9] X. Long, K. Deng, G. Wang *et al.*, "PP-YOLO: An effective and efficient implementation of object detector," *arXiv preprint arXiv:2007.12099*, 2020.

[10] P. Hurtik, V. Molek, J. Hula *et al.*, "Poly-YOLO: Higher speed and more precise detection for YOLOv3," *arXiv preprint arXiv:2005.13243*, 2020.

[11] P. Ganesh, Y. Chen, Y. Yang *et al.*, "YOLO-ReT: Towards high accuracy real-time object detection on edge GPUs," *arXiv preprint arXiv:2110.13713*, 2021.

[12] D. Yang *et al.*, "A streamlined approach for intelligent ship object detection using EL-YOLO algorithm," *Scientific Reports*, 2024.

[13] Diwakar and D. Raj, "Recent object detection techniques: A survey," *International Journal of Image, Graphics and Signal Processing*, vol. 14, no. 2, pp. 47–60, 2022.

[14] J. Khoramdel *et al.*, "YOLO-Former: YOLO shakes hand with vision transformers," *arXiv preprint*, 2024.

[15] R. Esram, B. B. V. L. Deepak, U. R. Mogili, and P. S. Sundar, "Agribots concepts and operations—a review," in *Applications of Computational Methods in Manufacturing and Product Design*, 2022, pp. 31–40.

[16] P. S. Sundar, B. B. V. L. Deepak, R. Esram, and U. R. Mogili, "Overview of sensors for measuring soil parameters supporting agricultural practices," in *Applications of Computational Methods in Manufacturing and Product Design*, Singapore: Springer, 2022, pp. 41–48.

[17] P. Jana, A. Biswas, and Mohana, "YOLO-based detection and classification of objects in video records," in *Proc. IEEE Int. Conf. Recent Trends in Electronics Information Communication Technology*, 2019.

[18] S. S. D. K. M. Lakshmi *et al.*, "Online dynamic outpatient queue system for automated token generation in hospitals," *Science, Technology and Development Journal*, vol. 12, no. 7, pp. 71–78, 2023, doi: 23.18001.STD. 2023.V12I07.23.37707.

[19] U. Mogili, K. V. Ampolu, B. Rajasekharam, and M. J. Timothy, "AI-driven interaction in AR environments," *Journal of Digital Economy*, vol. 3, no. 1, pp. 228–234, 2024.

[20] M. J. Timothy, B. Rajasekharam, K. V. Ampolu, and U. Mogili, "Threat detection using AI in cybersecurity systems," *International Journal of Intelligent Systems (IJIS)*, vol. 7, no. 1, pp. 1–7, 2023.

[21] K. V. Ampolu, U. Mogili, M. J. Timothy, and B. Rajasekharam, "Machine learning models for predictive maintenance," *International Journal of Intelligent Systems (IJIS)*, vol. 6, no. 4, pp. 1–7, 2022.

[22] B. Rajasekharam, M. J. Timothy, U. Mogili, and K. V. Ampolu, "Machine learning models for predictive maintenance," *Journal of Digital Economy*, vol. 2, no. 2, pp. 95–101, 2023.

[23] B. Soujania, K. V. Ampolu, M. J. Timothy, and U. Mogili, "Classifying disease information forums through semantic similarity-based machine learning," *Science, Technology and Development Journal*, vol. 14, no. 2, pp. 67–75, 2025.