International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

YOLOV4 Based Blind Assistant System for Real-Time Object Detection

FRANCIS SHAMILI S[*1], PRATHYUSH N P[*2], KENES K ROBY[*3] PRADEEP S[*4]

*[*1]Assistant professor, Department of CSE, Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur, Tamil Nadu, India.*

*[*2,3,4]UG Students, Department of CSE, Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur, Tamil Nadu, India.*

---***---

Abstract – This project focuses on object detection using the YOLOv4-tiny model, a lightweight version of the YOLO (You Only Look Once) algorithm designed for real-time object detection. The model is loaded with pre-trained weights and a configuration file, making it capable of detecting various objects from a webcam feed. Once an object is detected, the system identifies the class, evaluates the confidence of the detection, and calculates the bounding box coordinates to highlight the object in the image. The system applies Non-Maximum Suppression (NMS) to remove overlapping bounding boxes and retain only the most relevant ones. In this implementation, the program captures video frames from a webcam, pre-processes them to a format suitable for the YOLO model, and passes them through the neural network to generate predictions. These predictions are then analyzed to identify the objects in the frame, and relevant details such as the object's label and confidence score are extracted. If an object belongs to a specified class (e.g., "elephant," "bird," "horse," or "zebra"), the system triggers an HTTP request to send this information to the Blynk IoT platform for remote monitoring. The integration with Blynk IoT allows for real-time monitoring and remote alerts. By sending the detected object's label to Blynk, the system facilitates quick action based on the objects being tracked. This setup could be used for various applications, including surveillance, automated tracking systems, and environments where real-time detection and remote reporting are essential. Additionally, the use of the YOLOv4-tiny model ensures that the object detection process is both fast and efficient, making it suitable for applications requiring low-latency responses.

1.INTRODUCTION

Object detection has emerged as a powerful application of computer vision and deep learning in recent years. It involves identifying and locating objects of interest within images or video streams. Real-time object detection is particularly useful in scenarios like security surveillance, autonomous driving, and wildlife monitoring. This project leverages a lightweight, high-performance model known as YOLOv4-tiny to detect objects in live webcam footage.YOLO (You Only Look Once) is an advanced deep learning algorithm that processes the entire image in one go, making it faster and more efficient compared to traditional region-based techniques. YOLOv4-tiny, a smaller version of the full YOLOv4 model, is optimized for devices with lower processing power, while still maintaining reasonable accuracy and speed. This makes it ideal for real-time applications on embedded or low-resource systems.In this system, once an object is detected, it checks if the object belongs to a predefined set of classes such as "elephant", "bird", "horse", or "zebra". If matched, the system sends this information to the Blynk IoT platform using an HTTP request. This integration with IoT allows real-time updates and monitoring of object occurrences, enabling timely alerts and potential action based on the detection. The system is highly versatile and can be deployed in a variety of environments. From tracking endangered species in wildlife reserves to monitoring restricted areas for specific intrusions, it presents a modern, scalable solution. The use of OpenCV, YOLO, and Blynk provides a robust technological foundation that bridges the gap between real-time image processing and IoT-based notifications. With advancements in artificial intelligence and edge computing, the integration of real-time vision systems with IoT platforms has opened new possibilities across industries. From smart cities to conservation efforts, the ability to detect and transmit visual data instantly has become a game-changer. This project stands as a practical implementation of such integration, combining the power of computer vision (YOLOv4-tiny) with cloud-based IoT (Blynk) to deliver a responsive, intelligent system capable of detecting key objects and alerting users in real time. This not only enhances monitoring capabilities but also enables proactive decision-making in dynamic environments.

2.5 Objective :

primary objective of this project is the development of a real-time object detection system that leverages the YOLOv4-tiny model and integrates it with the Blynk IoT platform. This integration aims to facilitate remote monitoring and alerting capabilities. The core function of the system is to accurately detect specific classes of objects from a live webcam feed. Upon detection of a target object, the system is designed to generate and transmit alerts, enabling its application in scenarios that demand immediate surveillance and monitoring. A critical objective is to ensure the system's operational efficiency, even when deployed on devices with limited hardware resources. The selection of the YOLOv4-tiny model is strategic, as it is optimized for speed and low power consumption, making it suitable for implementation in embedded systems like Raspberry Pi or other edge computing devices. This optimization broadens the system's applicability across various cost-effective environments without compromising its core functionality. Furthermore, the project seeks to enhance the user experience by providing an intuitive and accessible method for receiving and monitoring detection results remotely. This is achieved through the Blynk mobile application and web dashboard, which offer users a seamless interface for real-time communication and monitoring. The emphasis on real-time communication is intended to support proactive response mechanisms, thereby contributing to the development of smarter and more responsive systems .In summation, the project's objectives are centered around creating an efficient, versatile, and user-friendly object detection system. By combining the power of the YOLOv4-tiny model with the connectivity of the Blynk IoT platform, the project aims to deliver a solution that can be readily deployed in diverse settings, from wildlife monitoring to security surveillance, providing timely alerts and enabling effective remote oversight. In summation, the project's objectives are centered around creating an efficient, versatile, and user-friendly object detection system. By combining the power of the YOLOv4-tiny model with the connectivity of the Blynk IoT platform, the project aims to deliver a solution that can be readily deployed in diverse settings, from wildlife monitoring to security surveillance, providing timely alerts and enabling effective remote oversight.

3.EXISTING SYSTEM

Existing systems often rely on **manual monitoring** or **basic motion sensors**, lacking intelligent automation. They require constant human supervision, causing **delays in response**.

While some use basic object detection, they typically suffer from **low accuracy**, **limited object classification**, and **no real-time communication** with remote platforms—making them less effective for **IoT or remote monitoring** applications.

Another key limitation is the **lack of real-time connectivity** with **cloud platforms**. Most systems store data locally or use basic alerts like buzzers, without sending information to **mobile apps or external systems**. This reduces **responsiveness** and limits **remote monitoring capabilities**.

DISADVANTAGES:

Internet Dependency for IoT Updates

The Blynk platform relies on an internet connection to send and receive data. In the event of network failure or limited coverage areas (e.g., deep forests or remote zones), the real-time updates won't reach the user, reducing system effectiveness.

Limited Object Classes

The pre-trained YOLOv4-tiny model detects only the 80 classes included in the COCO dataset. If your use case requires detecting unique objects not included (like rare animals or machinery), you would need to retrain the model, which can be time-consuming and requires technical skill.

4.PROPOSED SYSTEM

Utilizes YOLOv4-tiny, a lightweight deep learning model, for real-time object detection. Captures live video from a webcam and detects specific objects like elephant, zebra, bird, and horse. Sends instant updates to users via the Blynk IoT platform using HTTP API calls. Enables automated monitoring without the need for manual supervision. Displays detection data on a remote-accessible dashboard (mobile app or web).Designed for efficient performance even on low-power devices (e.g., Raspberry Pi).Modular and adaptable—easily reconfigured for different objects or applications .Ideal for use in wildlife monitoring, security systems, or other remote sensing tasks.

.

. ADVANTAGES:

Real-Time Object Detection

The system uses YOLOv4-tiny, a real-time object detection model that can detect and classify objects instantly from live video feed. This allows for immediate response and monitoring, which is essential in time-sensitive scenarios such as wildlife detection or security surveillance.

Lightweight and Efficient

YOLOv4-tiny is a compressed version of the full YOLO model, optimized for devices with limited processing power. It provides high-speed processing with minimal latency, making it suitable for deployment on laptops, Raspberry Pi, or embedded systems without requiring a high-end GPU.

4.1 SCOPE OF THE STUDY

This project defines a clear scope centered on the development and implementation of a real-time object detection system, with specific boundaries outlined to maintain focus and achieve tangible outcomes. The primary scope of the system lies in its capacity for **real-time detection**, meaning it is engineered to process video input from a webcam and identify objects within that video stream with minimal latency. This capability is crucial for applications requiring immediate feedback and response, such as surveillance or time-sensitive monitoring. The system's detection capabilities extend to **multiple object support**. It is designed to recognize a variety of objects, primarily those included in the COCO dataset. This encompasses a range of categories, including animals, vehicles, and common household items, demonstrating the system's versatility in handling diverse visual information. However, the project also incorporates **predefined class alerting**, which narrows the focus to specific objects of interest. In the context of the provided documentation, the system is configured to trigger alerts only when it detects objects belonging to the

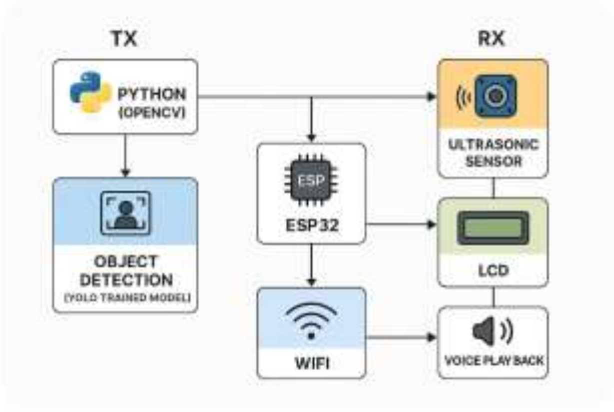
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

classes of "elephant," "bird," "horse," and "zebra." This selective alerting mechanism makes the system particularly well-suited for targeted surveillance applications, where monitoring specific entities is paramount. A significant component of the project's scope is its **IoT integration** through the Blynk platform. This integration enables the system to transmit detection results to the cloud, thereby facilitating remote monitoring. Users can access and monitor the system's output from virtually any location using a mobile or web application, enhancing its accessibility and utility .Furthermore, the project emphasizes **low-resource optimization**. By employing the YOLOv4-tiny model, the system is optimized to operate efficiently on devices with limited computational power. This consideration is essential for deploying the system in cost-effective or resource-constrained environments, such as embedded systems. The scope also includes **scalability**, as the system is designed to accommodate the addition of new object classes. This can be achieved by updating the COCO names file and adjusting the detection criteria, providing a degree of flexibility for adapting the system to different requirements .To ensure user comprehension, the project incorporates **user-friendly output**. OpenCV is utilized to display detected objects with bounding boxes and labels, providing clear visual feedback and facilitating easy interpretation of the system's results

3.3 System Architecture

The system architecture of this embedded project is a well-structured integration of various hardware components, each playing a specific role in delivering a functional, real-time monitoring and alert system. At the core of this architecture is the **ESP32 microcontroller**, a powerful and versatile module with built-in Wi-Fi and Bluetooth capabilities, which acts as the brain of the system. All the inputs and outputs are processed through this microcontroller.

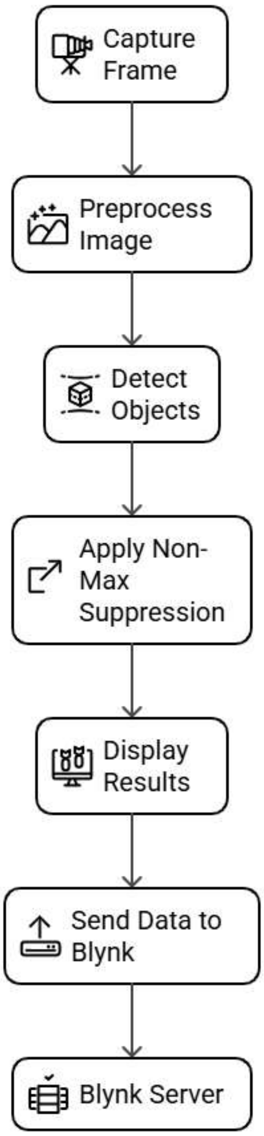
The **power supply** unit, typically from a DC adapter or USB connection, provides the required electrical energy to operate the system. This power is first routed through a **voltage regulator**, which ensures that all components receive a consistent and safe voltage level, thus preventing damage and maintaining system stability.An **ultrasonic sensor (HC-SR04)** is connected to the ESP32 and functions as the primary input device. It measures the distance between the sensor and any object in front of it using sound waves and echoes. This data is sent to the ESP32 for processing. The microcontroller analyzes the data and checks if the distance meets certain pre-defined thresholds. If the object is too close or too far based on the programmed conditions, the ESP32 triggers appropriate output responses .The outputs are managed through two components: a **16x2 LCD display** and a **speaker or buzzer**. The LCD display is used to show live sensor readings, status messages, or warnings in a user-friendly manner. The speaker/buzzer provides an auditory alert if a critical condition is detected, such as an obstacle too close to the sensor.



4.4 Data Flow Diagram (DFD)

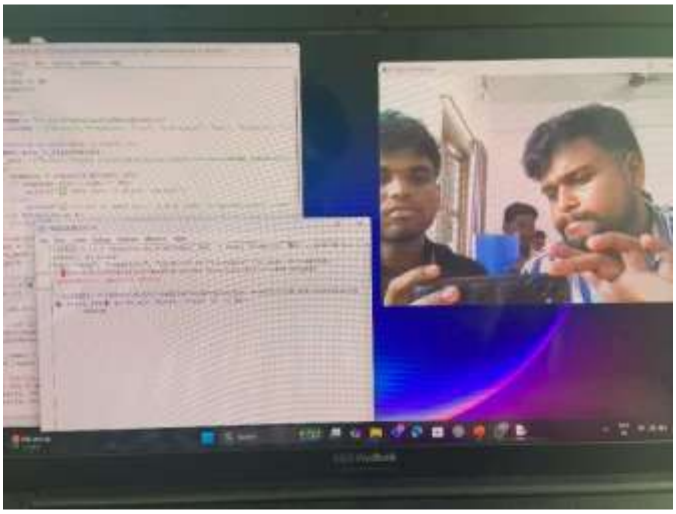
An **ultrasonic sensor (HC-SR04)** is connected to the ESP32 and functions as the primary input device. It measures the distance between the sensor and any object in front of it using sound waves and echoes. This data is sent to the ESP32 for processing. The microcontroller analyses the data and checks if the distance meets certain pre-defined thresholds. If the object is too close or too far based on the programmed conditions, the ESP32 triggers appropriate output responses. The outputs are managed through two components: a **16x2 LCD display** and a **speaker or buzzer**. The LCD display is used to show live sensor readings, status messages, or warnings in a user-friendly manner. The speaker/buzzer provides an auditory alert if a critical condition is detected, such as an obstacle too close to the sensor.

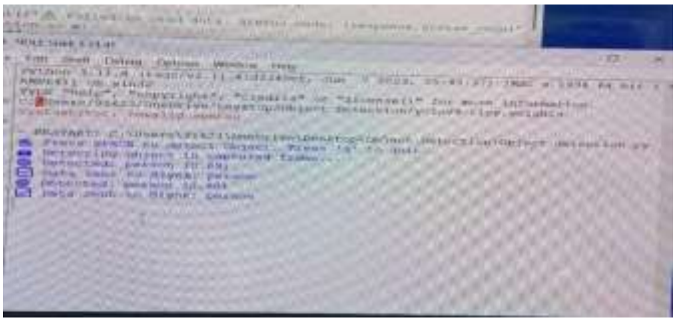
The **power supply** unit, typically from a DC adapter or USB connection, provides the required electrical energy to operate the system. This power is first routed through a **voltage regulator**, which ensures that all components receive a consistent and safe voltage level, thus preventing damage and maintaining system stability



.

IMPLEMENTATION AND RESULT





5.CONCLUSION

This project demonstrates an efficient real-time object detection system using YOLOv4-tiny and the Blynk IoT platform. It combines computer vision and deep learning for accurate, high-speed object detection, with remote monitoring through mobile devices. The system is cost-effective, deployable on low-resource devices,

and adaptable to various applications such as wildlife monitoring, home security, and smart automation.
With its modular design and instant alert system, it reduces the need for human supervision and supports timely decision-making. This scalable solution offers great potential for future enhancements, including multi-camera support and edge AI, bridging the gap between computer vision and IoT for smarter, more responsive monitoring systems.

REFERENCES

[1] Jie Fang; Weitao Zhang; Shasha Zhang; Nan Wang "Positive-Incentive Pseudo Label Optimization for Salient Object Detection", Doi: 10.1109/ACCESS.2024.3476077

[2] Jumabek Alikhanov; Hakil Kim"Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods," Doi: 10.1109/ACCESS.2023.3292539

[3] Shanping Ning,Rui Guo; Pengfei Guo; Lü Xiong; Bangbang Chen "Research on Foreign Object Intrusion Detection for Railway Tracks Utilizing Risk Assessment and YOLO Detection", Doi: 10.1109/ACCESS.2024.3504010

[4] Rakpong Kaewpuang; Minrui Xu; Wei Yang Bryan Lim; Dusit Niyato; Han Yu; Jiawen Kang A Progressive-Assisted Object Detection Method Based on Instance Attention", Doi: 10.1109/ACCESS.2024.3459941

[5] Chansreynich Huot; Kimleang Kea; Tae-Kyung Kim; Youngsun Han "Enhancing Knapsack-Based Financial Portfolio Optimization Using Quantum Approximate Optimization Algorithm", DOI: 10.1109/ACCESS.2024.3506981

[6] Soohyun Park; Hankyul Baek; Joongheon Kim "Spatio-Temporal Multi-Metaverse Dynamic Streaming for Hybrid Quantum-Classical Systems", DOI: 10.1109/TNET.2024.3453067.

[7] Ishmeen Kaur Garewal; Chaitanya V Mahamuni; Shruti Jha "Emerging Applications andChallenges in Quantum Computing: A Literature SurveyDOI: 10.1109/icABCD62167.2024.10645271

[8] Xia Liu; Geng Liu; Hao-Kai Zhang; Jiaxin Huang; Xin Wang, "Mitigating Barren Plateaus of Variational Quantum Eigensolvers," DOI: 10.1109/TQE.2024.3383050

[9] Param Pathak; Vidhi Oad; Aditya Prajapati; Nouhaila Innan, "Resource Allocation Optimization in 5G Networks Using Variational Quantum Regressor,"DOI: 10.1109/QCNC62729.2024.00025

[10] P. Venkata Ramana; T. Suresh Balakrishnan "Noise Removal in IBM Real Time Quantum Computer with Deviation Quantum Algorithm in Comparison with Variational Quantum Eigensolver Algorithm", DOI: 10.1109/ICAIT61638.2024.10690723.

[11] Kang Kai; Zhu Peipei; Zhang Fan, "Research on the Resource Allocation Optimization Model of Automobile Based on Cloud Computing Resource Scheduling Algorithm," DOI: 10.1109/ISPCEM64498.2024.00171.

[12] Jonathan Wurtz; Stefan H. Sack; Sheng-Tao Wang, "Solving Nonnative Combinatorial Optimization Problems Using Hybrid Quantum–Classical Algorithms DOI: 10.1109/TQE.2024.3443660

[13] Mostafa Raeisi-Varzaneh; Omar Dakkak; Adib Habbal; Byung-Seo Kim "Resource Scheduling in Edge Computing: Architecture, Taxonomy, Open Issues and Future Research Directions," DOI: 10.1109/ACCESS.2023.3256522

[14] Rakpong Kaewpuang; Minrui Xu; Dusit Niyato; Han Yu; Zehui Xiong; Jiawen Kang "Stochastic Qubit Resource Allocation for Quantum Cloud Computing DOI: 10.1109/NOMS56928.2023.10154430

[15] Alireza Furutanpey; Johanna Barzen; Marvin Bechtold; Schahram Dustdar; Frank Leymann; Philipp Raith, "Architectural Vision for Quantum Computing in the Edge-Cloud Continuum," DOI: 10.1109/QSW59989.2023.00021

[16] Hoa T. Nguyen; Muhammad Usman; Rajkumar Buyya "iQuantum: A Case for Modeling and Simulation of Quantum Computing Environments," DOI: 10.1109/QSW59989.2023.00013

[17] Lei Fan; Zhu Han., "Hybrid QuantumClassical Computing for Future Network Optimization," DOI: 10.1109/MNET.001.2200150

[18] Lingfeng Lu; Yongmei Sun; Yaoxian Gao; Weiwen Kong., "A Resource Allocation Scheme for IndependentCore Transmission of Quantum Key Distribution," DOI: 10.1109/ICCT56141.2022.10072711

[19] Lingfeng Lu; Yongmei Sun; Yaoxian Gao; Weiwen Kong, "Resource Management for Edge Intelligence (EI)-Assisted IoV Using QuantumInspired Reinforcement Learning," DOI: 10.1109/ICCT56141.2022.10072711