iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72162

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Real-Time Object Detection Using Deep Learning

Shraddha Khonde[1], Pratik Manjare[2], Saniraj Desai[3], Gunjan Narkhede[4], Shivam Rathod[5]
Department of Computer Engineering, MES Wadia COE, SPPU, Pune, 411001

Abstract: ***Combining deep learning with computer vision has enabled significant advancements in real-time object detection. Leveraging the capabilities of the YOLOv8 architecture, the proposed system efficiently detects and tracks objects across various input sources, including static images, live webcam feeds, and video streams. By integrating a Flask-based web application with the YOLOv8 model, the system delivers a secure, responsive, and user-friendly interface for real-time detection. The model is trained on a custom dataset, enhanced through extensive data augmentation techniques to improve generalization across diverse environments. Key performance metrics such as Mean Average Precision (mAP), precision, recall, and frames per second (FPS) were used to evaluate accuracy and speed. Results demonstrated robust performance with up to 75% mAP at IoU 0.5 and real-time processing speeds of 25–30 FPS on GPU-enabled systems. Challenges such as performance degradation under low-light and high-occlusion conditions were addressed through thoughtful dataset preparation and architectural tuning. The system is scalable and adaptable, making it suitable for real-world applications such as surveillance, industrial monitoring, and accessibility tools.***

Keywords: ***YOLOv8, Real-Time Object Detection, Deep Learning, Convolutional Neural Network (CNN), Flask Web Application, Custom Dataset Training, Computer Vision.***

I. INTRODUCTION

The rapid advancement of computer vision and deep learning technologies has revolutionized the field of object detection, enabling real-time applications across various domains. Traditional computer vision approaches, while effective for specific tasks, often struggle with real-time processing and adaptability to diverse scenarios. This study explores the integration of YOLOv8, a state-of-the-art deep learning model, with a web-based application framework to develop a robust real-time object detection system. The system combines the strengths of YOLOv8's detection capabilities with the accessibility of web technologies, providing a powerful solution for practical applications. The research focuses on developing a Flask-based web application that leverages the YOLOv8 model for real-time object detection across multiple input modalities. The system supports real-time webcam feeds, image uploads, and video processing, making it versatile for various use cases. A key aspect of the research is the implementation of user authentication and secure access control, ensuring that the system can be deployed in environments requiring controlled access. The integration of Flask's web framework with YOLOv8's detection capabilities represents a significant advancement in making advanced computer vision tools accessible to non-expert users while maintaining security and performance.

The study addresses several critical aspects of real-time object detection systems, including model training on custom datasets, efficient web integration, and practical deployment considerations. Through comprehensive evaluation, the research aims to demonstrate the system's effectiveness in real-world scenarios while identifying areas for future enhancement. The findings contribute to the broader field of computer vision by showcasing how modern deep learning models can be effectively integrated with web technologies to create practical, accessible solutions for real-time object detection tasks.

II. LITERATURE SURVEY

Deep learning is revolutionizing product discovery by offering elegant models and techniques that balance accuracy and speed. Key features such as anchor points, feature pyramids, and non-maximum features (NMS) play a key role in improving discovery. Popular architectures include Faster R-CNN, which includes a Region Proposal Network (RPN) to generate nested boxes; it is used for many predictions. RetinaNet solves the class ambiguity problem with Focal Loss, while EfficientDet achieves high accuracy with low loss. Training strategies such as data augmentation, changing learning to use weights before training, and weight loss combining iterations and repetitions are essential for effective modeling. Tools like TensorFlow, PyTorch, and APIs like Detectron2 are easy to develop and deploy. Factors like dataset diversity, model complexity, and computational resources affect performance. Cloud computing increases efficiency and effectiveness while prioritizing steps like evaluating and optimizing strategies. YOLO's grid-based approach estimates bounding boxes, confidence scores, and class probabilities, eliminating redundant heuristics in NMS.

©IJRASET: All Rights are Reserved

Techniques like sigmoid processing for multiple datasets to ensure stability enable deep learning that is not required for real-world operations.[1]

The research uses the Microsoft COCO dataset (2014 - 2017) to create a diagnostic model that can identify different problems using YOLO's grid-based algorithm for accurate prediction and checkbox design. This work involves geographic information systems (GIS) systems including land use, elevation, parcels, roads, and areas to develop a virtual map for use in the real world. Additionally, the project includes speech-to-text recognition by converting analog words into text through extraction and decoding. Interaction is enabled using of gesture recognition, image frame detection, hand tracking, and motion recognition. The application was evaluated by 27 participants including blind people, their caregivers, and IT professionals by the ISO 25010 standard. Tests have shown that it is possible as an additional device for visually impaired users to walk independently without the need for traditional aids such as a white cane.[2]

The YOLO architecture enables fast and accurate discovery of real-time objects through simple processes such as preprocessing, exploration, and visualization. Techniques such as background subtraction, segmentation, and background transformation enhance motion detection capabilities, while hybrid methods increase accuracy. Metrics such as accuracy and mAP can measure performance. The research focuses on the development of a YOLO-based system for instant detection and tracking of moving objects, efficient management, and distinction between stationary and moving.[3] The method uses the VGG16 model for feature extraction and is trained on the PASCAL VOC 2012 dataset for vehicle detection, utilizing a dynamic system of adjustable bounding boxes to enhance visibility. The key features include mobility, scalability, and migration rate defined by the Competition Over Unity (IoU) metric with a threshold of 0.5. By combining and optimizing VGG16 with the deep Q-network, the model achieves real-time targeting with high accuracy and consistent performance.[4]

The heritage analysis approach incorporates machine learning to solve interpretation and data challenges from different data sources such as cameras and sensors. It combines machine learning, deep learning, and various integrated methods to develop performance models.

Technologies such as XAI (LIME, SHAP) increase prediction accuracy, while adaptive learning and integration increase efficiency and flexibility. The framework is evaluated using accuracy, recall, and F1 scores and is designed for real-world applications, providing powerful tools for asset management and information management decisions.[5] This work leverages datasets such as Traffic Alerts and KITTI for training and validation, focusing on train signs and object detection.

Among the benchmarks, YOLOv8 stands out with 0.986 mAP at 0.5 threshold and performs well in terms of speed, accuracy, and scalability. Advanced tracking with ByteTrack and DeepSORT, multiple item tracking with Kalman filters, and reliable detection. Linear search using OpenCV and YOLO-P for deep learning for simplicity. Metrics such as MAP and IoU enable product distribution, driving, and maintenance support.[6]

Convolutional neural networks (CNN) have the ability to detect valuable objects; models such as AlexNet, VGGNet, and ResNet have surpassed human performance by reducing errors by 15.3% and 3.6% in 2012 and 2015, respectively. YOLO (You Only Look at Once) revolutionizes object detection by combining fast and accurate neural network connections with regression-based bounding box estimation. YOLOv4 provides better and more accurate results than previous models.

In this study, YOLOv4 is combined with CNN to detect flying objects at 30 FPS, and datasets such as DigitalBlurSet and NaturalBlurSet are used to solve the blur problem. The system achieved 73% detection accuracy, showed good performance in terms of precision (0.669) and recall (0.775), and improved real-time video analysis thanks to the integration between objects and blur.[9] To solve the problem of limited population data in the training model, we created a drone dataset containing 3,698 images and 3,719 different objects.

It improves detection by combining the collected large objects, small data, and images captured by small drones. Mosaic's data enhancement technology is used to integrate the four images to increase the training diversity. The research also investigates replacing the weighting of YOLOv4 with MobileNetv3 to increase the speed without compromising the extraction results. The Mobile-YOLO model is specifically designed for UAV target detection by optimizing the performance of small platforms by integrating MobileNetv3, SPP, PA-Net, and YOLO.[10]

III. METHODOLOGY

The objective is to develop a web-based real-time object detection system utilizing the YOLOv8 deep learning model. The project is designed to handle multiple input types, including live webcam feeds, uploaded images, and video files. The overall methodology involves custom dataset preparation, model training and configuration, web application development, evaluation of system performance, and final integration for real-time deployment.

A. *Dataset Preparation and Augmentation*

To train the object detection model effectively, a custom dataset is created based on the specific detection needs of the application. This dataset includes images and video frames captured under varying lighting, angle, and environmental conditions. Each image is manually annotated with bounding boxes and associated object labels. Data augmentation techniques such as rotation, flipping, scaling, and brightness adjustments are applied to enhance dataset diversity. This process improves the model's robustness in detecting objects under real-world variations.

B. *Model Configuration and Training*

The YOLOv8 model is customized to identify object classes relevant to the application. Initially, the model is loaded with pre-trained weights (e.g., from the COCO dataset) and then fine-tuned on the custom dataset. The training involves tuning key hyperparameters like learning rate, batch size, and number of epochs to achieve optimal performance. During the training process, validation accuracy is monitored to ensure a balance between detection accuracy and inference speed.

C. *Web Application Development*

A Flask-based web application is developed to provide an accessible interface for object detection. The system includes user authentication using Flask-Login to manage secure access. Input handling modules are created to accept webcam streams, image uploads, and video files. Real-time webcam detection is implemented using continuous video streaming via multipart JPEGs. The front-end interface displays detection results with bounding boxes and labels overlaid on the input data.

D. *Experimental Setup*

The model training and real-time testing are carried out on a system with an Intel i7-10700F processor, NVIDIA GTX 1660 SUPER GPU (6GB), and 16GB RAM. The software stack includes Windows 10/11, Python 3.9, the Ultralytics YOLOv8 framework (built on PyTorch), Flask for backend logic, and OpenCV for image and video processing. This setup ensures smooth training and real-time execution of the detection system.

E. *Performance Evaluation*

The trained model is evaluated using key performance metrics. Mean Average Precision (mAP) at 0.5 IoU is used to assess detection accuracy. Precision and Recall are calculated to measure the model's ability to reduce false positives and capture all relevant objects. Frames Per Second (FPS) is used to evaluate the real-time speed of the system. Additionally, memory and GPU/CPU usage are monitored during inference to assess resource efficiency.

F. *Real-Time Integration and Deployment*

The final trained model is integrated into the Flask application for real-time object detection. Webcam input is processed in real-time using OpenCV, and uploaded images and videos are handled through dedicated pipelines with progress indicators. The user interface clearly displays detection outputs. Secure user login and session handling are implemented to restrict unauthorized access. The system is suited for real-world applications like surveillance or industrial monitoring where fast and reliable object detection is required.[14],[15],[16]

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

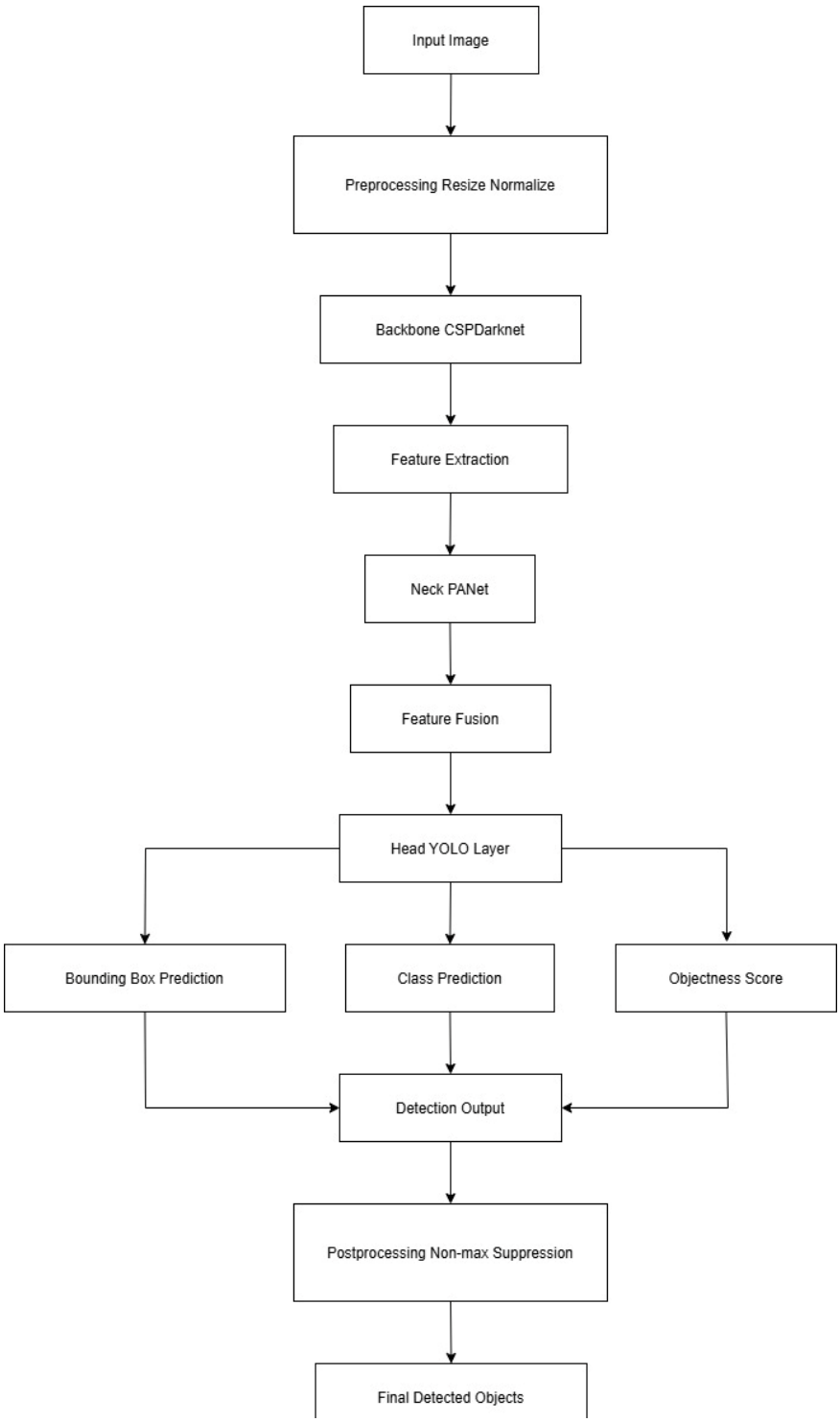


Fig. 1. Architecture of YOLOv8

IV. RESULTS AND DISCUSSION

The evaluation of real-time object detection system uncovered particular qualities and shortcomings over different execution measurements. The custom-trained YOLOv8m demonstrate illustrated solid exactness on the custom dataset, accomplishing a Cruel Normal Exactness (mAP) of 75% at IoU 0.5, with accuracy and review measurements of 80% and 70% separately. The framework performed especially well on inactive pictures (78% mAP) compared to real-time webcam nourishes (70% mAP), highlighting the challenges of preparing live video streams with variables like movement obscure and variable lighting.

The information enlargement methods essentially made strides the model's generalization capability, contributing a 5-7% increment in mAP on the approval set.

In terms of effectiveness, the framework accomplished noteworthy real-time execution when utilizing an NVIDIA GTX 1660 SUPER GPU, keeping up a normal of 25-30 FPS for webcam streams at 720p determination. This execution was steady for transferred recordings of comparable determination, permitting for close real-time investigation. In any case, on CPU-only induction utilizing an Intel i7-10700F, the FPS dropped essentially to 5-7 FPS, underscoring the significance of GPU increasing speed for real-time applications. For inactive picture preparing, the framework illustrated a responsive idleness of 150-200 milliseconds per picture on GPU.

The system's strength was tried over different challenging conditions. Whereas it appeared sensible execution in direct varieties of lighting and fractional impediment, its exactness dropped underneath 50% in extreme low-light conditions or with intensely blocked objects. The Carafe web application kept up steady operation amid delayed webcam spilling sessions (tried for up to 1 hour) and viably dealt with blunder scenarios such as invalid record transfers or camera detachments. In terms of adaptability, the framework overseen outlines with up to 10-15 objects without noteworthy execution debasement, in spite of the fact that preparing time expanded somewhat past this density.

The Jar web application given an natural interface for clients to associated with the question discovery framework over all three input modalities (webcam, picture, video). The client verification module guaranteed secure get to to the application, whereas visual criticism of recognized objects (bounding boxes and names) was clear and convenient. The discretionary text-to-speech include for declaring identified objects was detailed by test clients as a valuable openness improvement. The system's design permitted for simple adjustment to modern question classes through retraining with extended datasets and upheld distinctive YOLOv8 show variations for adjusting speed and exactness requirements.

These discoveries recommend that whereas the framework is practical for viable sending in scenarios requiring real-time protest discovery with controlled get to, there are clear regions for enhancement. The challenges in identifying little or intensely blocked objects and execution debasement in low-light conditions highlight particular zones for show improvement. Future work may center on growing the custom dataset to incorporate more challenging scenarios, investigating progressed information enlargement procedures, and exploring building alterations to move forward location precision in troublesome conditions. The system's current arrangement is appropriate for person client sessions, but adaptability for concurrent clients would require standard web scaling solutions.

The investigate illustrates a fruitful usage of a custom protest location arrangement that combines the qualities of YOLOv8's location capabilities with a secure, user-friendly web interface. Whereas there are confinements in terms of dataset differing qualities and equipment necessities, the framework gives a strong establishment for commonsense applications and iterative improvements. Future work will point to address these impediments and assist improve the system's capabilities, strength, and down-to-earth appropriateness.

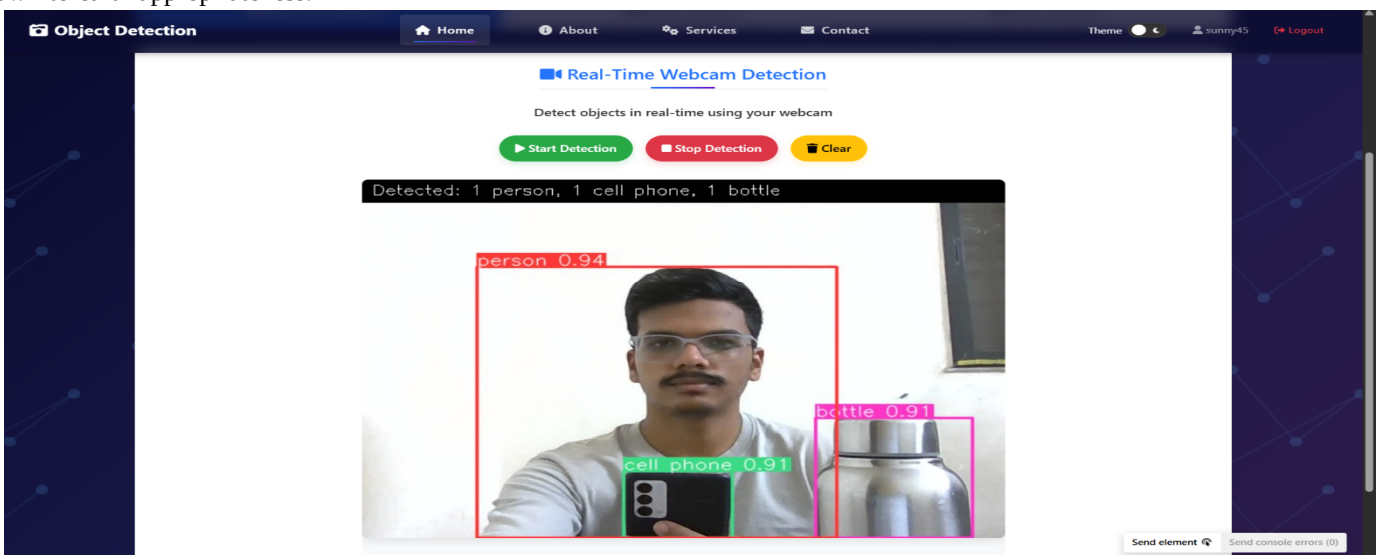


Fig.2 Real-Time Object Detectionsample result 1

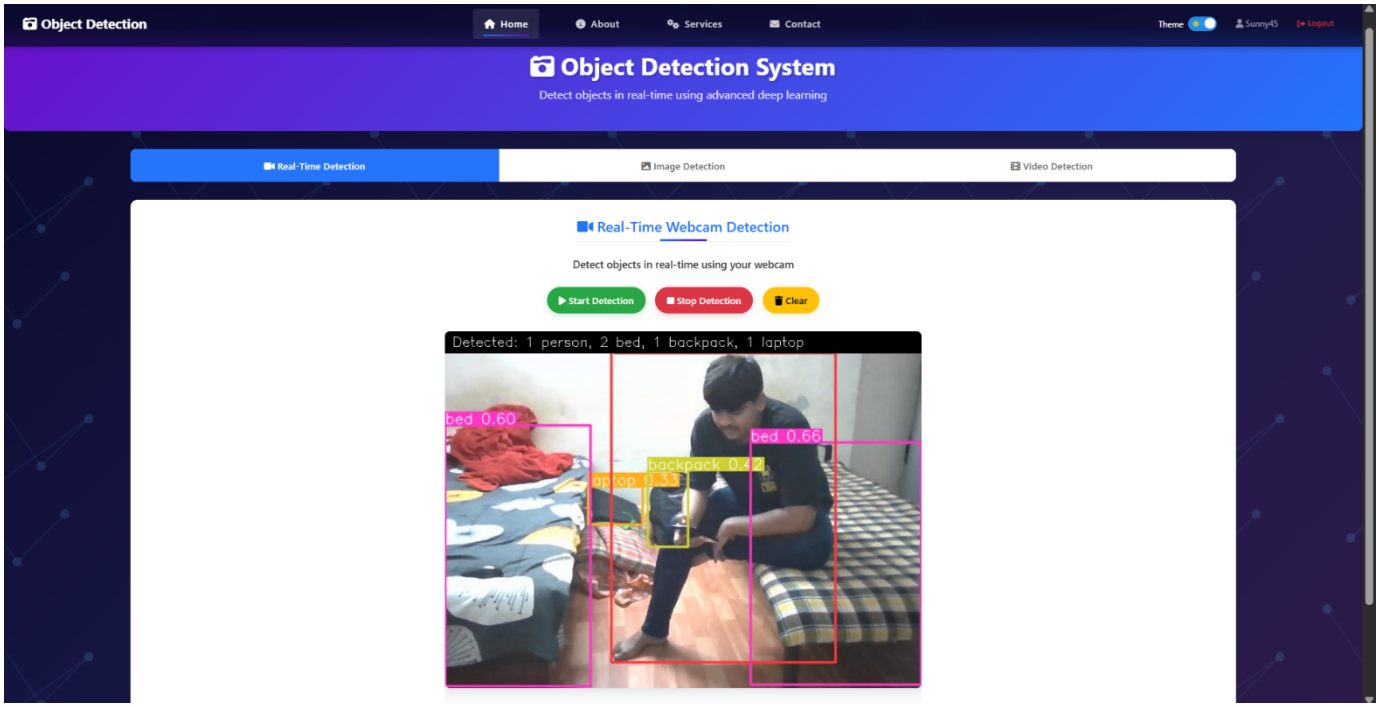


Fig.3 Real-Time Object Detectionsample result 2

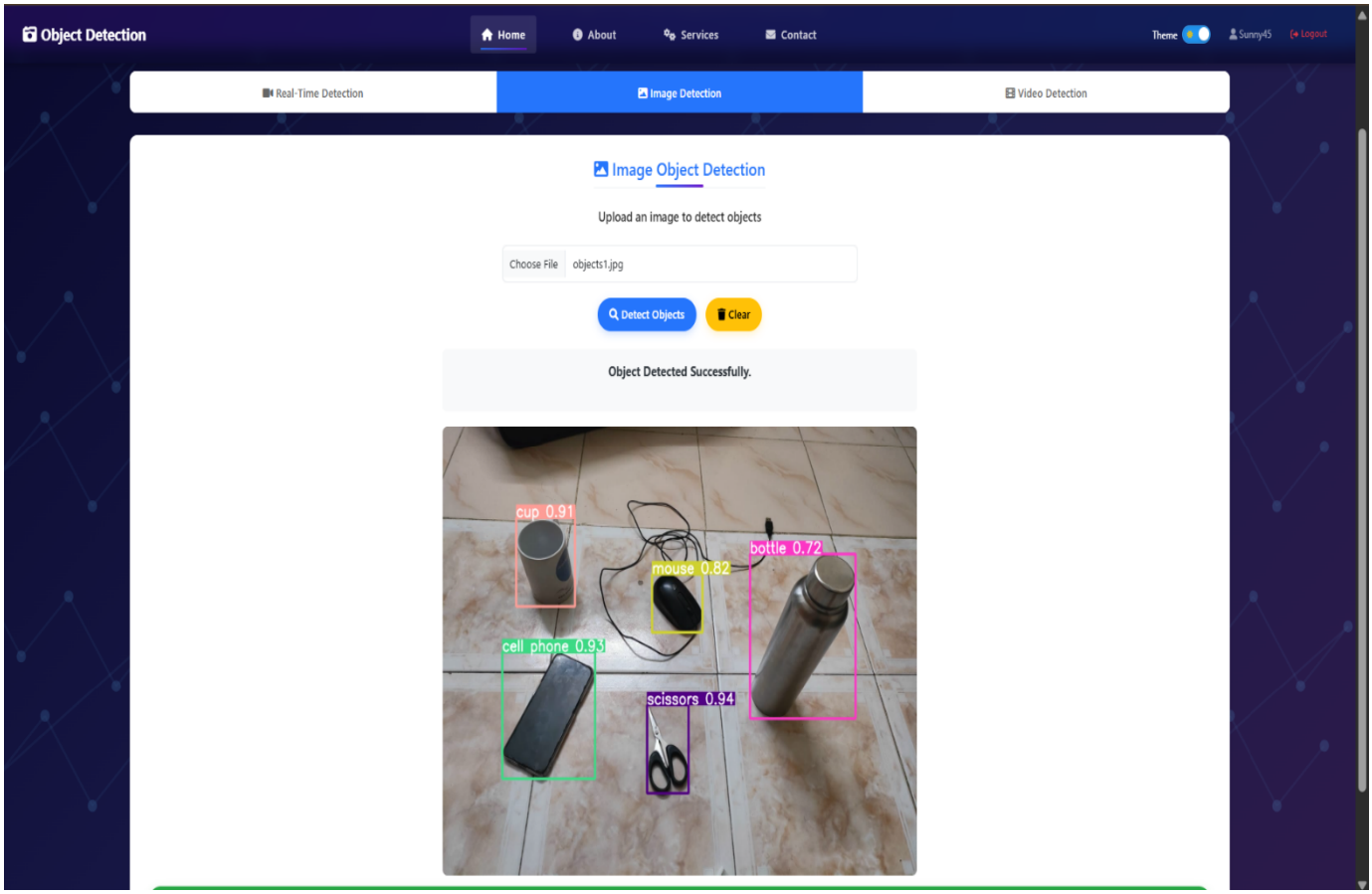


Fig.4Image Object Detection

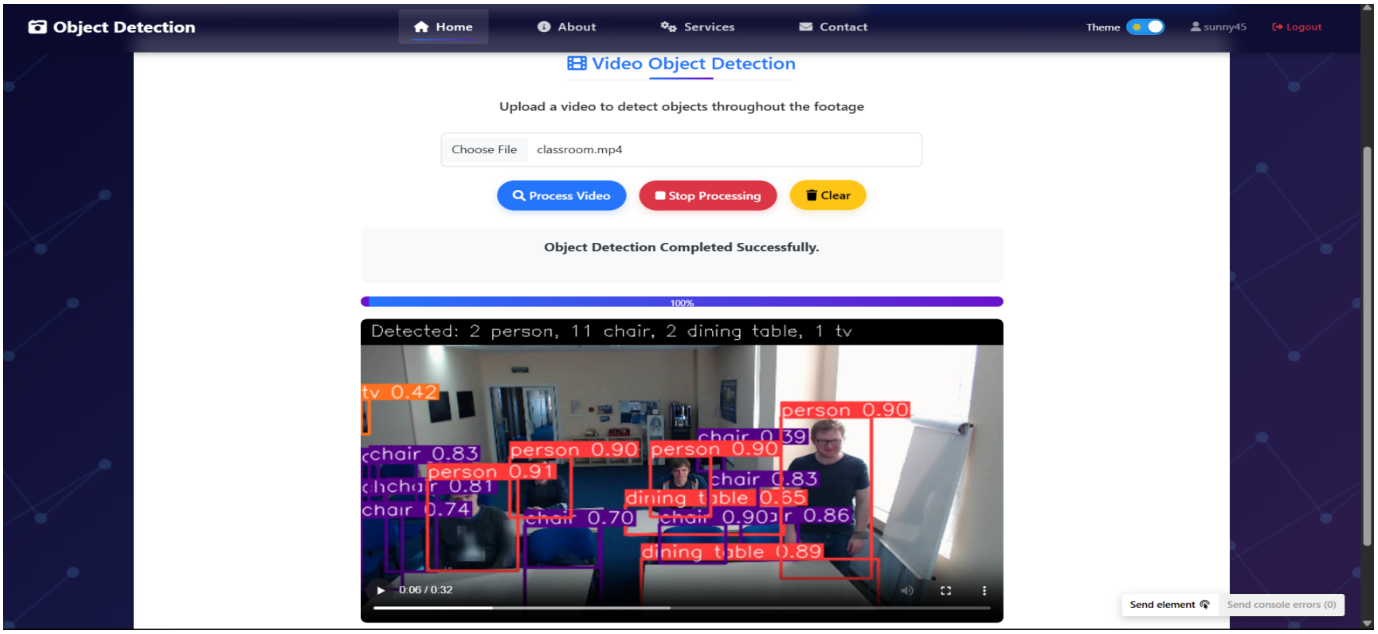


Fig.5Video Object Detection

V. CHALLENGES AND LIMITATIONS

The integration of YOLOv8 with the Flask web application presented several technical challenges. Ensuring smooth video streaming while maintaining detection accuracy required careful optimization of the Flask application's architecture and resource management. Implementing secure user access while maintaining real-time performance proved challenging, necessitating a careful balance between security measures and system responsiveness. Ensuring consistent performance across different browsers and devices required extensive testing and adaptation of the web interface, highlighting the complexities of cross-platform compatibility.

The quality and diversity of the training dataset posed significant challenges. Collecting and annotating a representative dataset of target objects required substantial time and resources, particularly for scenarios involving motion and varying environmental conditions. While augmentation techniques improved model generalization, they also introduced potential biases that required careful validation to ensure they didn't distort the underlying data distribution. The dataset's limitations in capturing all real-world scenarios, particularly edge cases and extreme conditions, could impact practical deployments.

The computational requirements for training and deploying the YOLOv8 model were substantial. Training the YOLOv8m model on the custom dataset required significant GPU resources and time, which could be a barrier for smaller organizations or individual developers. Maintaining real-time performance (25-30 FPS) required dedicated GPU hardware, limiting deployment options to systems with appropriate computational capabilities. Efficient memory management during both training and inference was crucial, especially when processing high-resolution video streams, to ensure optimal performance.

The evaluation scope was limited to specific experimental conditions. The evaluation was primarily conducted on a specific GPU (NVIDIA GTX 1660 SUPER) and CPU (Intel i7-10700F) configuration, which may not fully reflect performance on other hardware setups. While the system was tested across webcam, image, and video inputs, it may not fully capture the performance characteristics when deployed in different real-world scenarios. The evaluation focused on specific object classes in the custom dataset, which may not represent all potential use cases for object detection.

These challenges highlight the need for careful consideration and ongoing research to enhance the system's capabilities and broaden its applicability across different scenarios and contexts.

VI. CONCLUSION

This study successfully designed and evaluated a real-time object detection system using the YOLOv8 model, integrated into a Flask-based web application. The model, trained on a custom dataset, showed promising results, achieving a Mean Average Precision (mAP) of 75% at an IoU of 0.5. It performed best on static images, while slightly lower performance was noted on real-time webcam streams, primarily due to challenges such as motion blur and inconsistent lighting. Despite these, the system maintained smooth performance at 25–30 FPS with GPU support, demonstrating its capability for real-time applications.

In contrast, inference using only a CPU highlighted the performance gap, underlining the critical role of hardware acceleration. The Flask web interface provided an intuitive and secure platform for users to interact with the system across different input types—webcam, image uploads, and video files—demonstrating its flexibility for diverse practical use cases. The outcomes of this study underscore the importance of a robust and diverse dataset, as well as the need to address edge-case challenges like low-light and occlusion scenarios. This project also illustrates the potential of combining modern deep learning models with lightweight web frameworks to build accessible and scalable computer vision solutions. Looking ahead, improvements such as model variant selection (e.g., YOLOv8n for speed or YOLOv8x for higher accuracy), more extensive dataset expansion, and backend optimizations for concurrent users could further enhance system performance and adaptability. Overall, the integration of deep learning with web-based interfaces marks a meaningful step toward making real-time object detection systems more practical, scalable, and accessible for real-world deployment.

REFERENCES

[1] R. K, G. B, K. R. P, M. K. R, G. M and D. K.G, "A Perspective Study of Real-Time Object Detection Using Deep Learning by Applying Design Thinking Approach," 2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon), Pune, India, 2024, pp. 1-5, doi: 10.1109/MITADTSoCiCon60330.2024.10575232.

[2] G. M. B. Catedrilla, "Mobile-Based Navigation Assistant for Visually Impaired Person with Real-time Obstacle Detection Using YOLO-based Deep Learning Algorithm," 2022 5th Asia Conference on Machine Learning and Computing (ACMLC), Bangkok, Thailand, 2022, pp. 63-67, doi: 10.1109/ACMLC58173.2022.00020.

[3] U. Dwivedi, K. Joshi, S. K. Shukla and A. S. Rajawat, "An Overview of Moving Object Detection Using YOLO Deep Learning Models," 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024, pp. 1014-1020, doi: 10.1109/ICDT61202.2024.10489800.

[4] S. Borkar, U. Singh and S. S, "Dynamic Approach for Object Detection using Deep Reinforcement Learning," 2024 IEEE Space, Aerospace and Defence Conference (SPACE), Bangalore, India, 2024, pp. 393-397, doi: 10.1109/SPACE63117.2024.10667858.

[5] T. Sarkar, M. Rakhra, V. Sharma, S. Takkar and K. Jairath, "Comparative Study of Object Recognition Utilizing Machine Learning Techniques," 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), Gautam Buddha Nagar, India, 2024, pp. 726-731, doi: 10.1109/IC3SE62002.2024.10593475.

[6] S. A. Babu Parisapogu, N. Narla, A. Juryala and S. Ramavath, "YOLO based Object Detection Techniques for Autonomous Driving," 2024 Second International Conference on Inventive Computing and Informatics (ICICI), Bangalore, India, 2024, pp. 249-256, doi: 10.1109/ICICI62254.2024.00049.

[7] B. Jaison, A. J. G, J. J and D. P. C, "You Only Look Once(YOL O) Object Detection with COCO using Machine Learning," 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), Bengaluru, India, 2022, pp. 1574-1578, doi: 10.1109/IIHC55949.2022.10059737.

[8] T. S. Gunawan, I. M. M. Ismail, M. Kartiwi and N. Ismail, "Performance Comparison of Various YOLO Architectures on Object Detection of UAV Images," 2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Melaka, Malaysia, 2022, pp. 257-261, doi: 10.1109/ICSIMA55652.2022.9928938.

[9] A. Senapati et al., "Identification of blurred objects in real time Video using deep learning neural networks," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-4, doi: 10.1109/ICCCNT54827.2022.9984429.

[10] J. Wang, W. Hongjun, J. Liu, R. Zhou, C. Chen and C. Liu, "Fast and Accurate Detection of UAV Objects Based on Mobile-Yolo Network," 2022 14th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 2022, pp. 01-05, doi: 10.1109/WCSP55476.2022.10039216.

[11] L. Kuhlane, D. Brown and M. Marais, "Real- Time Detecting and Tracking of Squids Using YOLOv5," 2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 2023, pp. 1-5, doi: 10.1109/icABCD59051.2023.10220521.

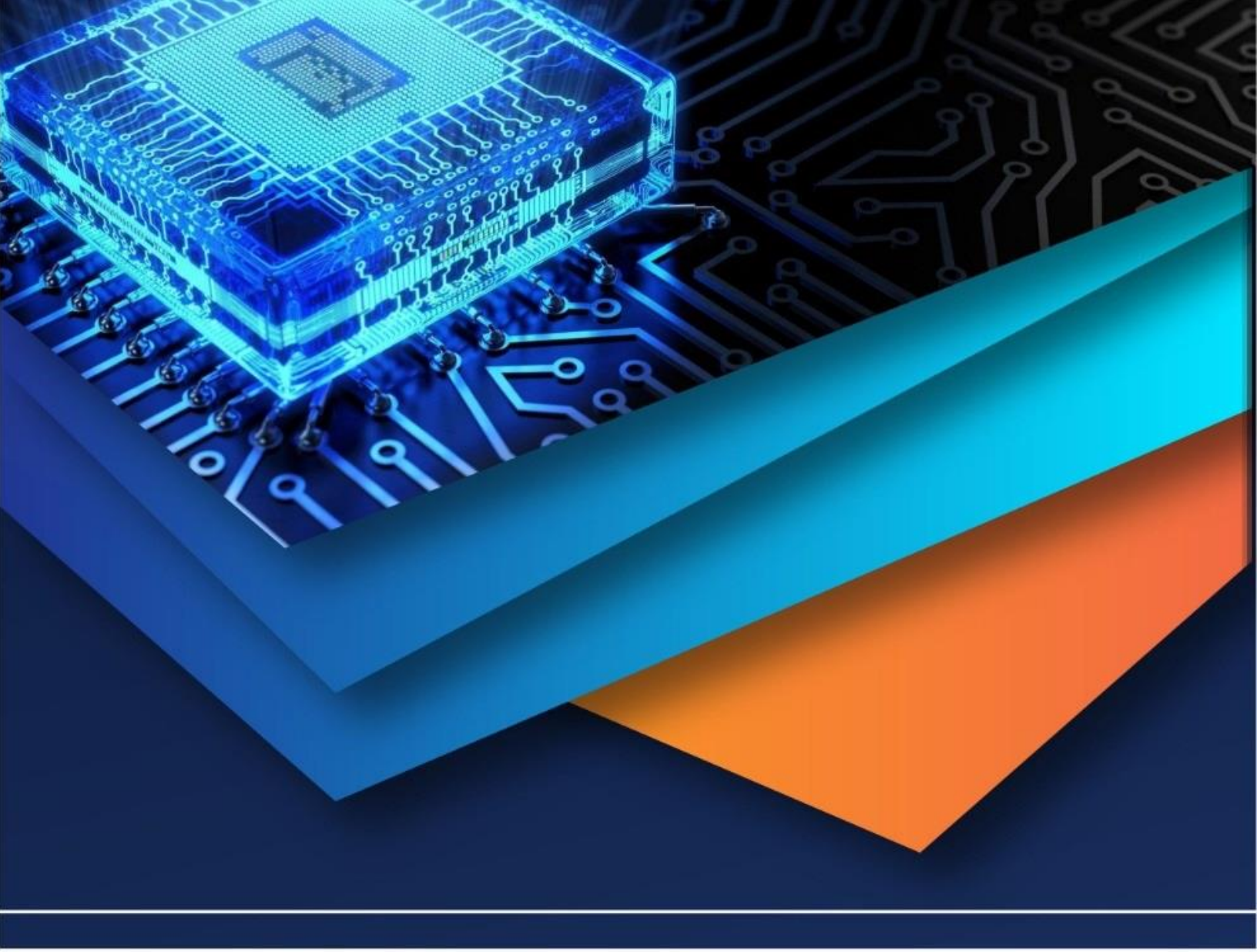
[12] P. C. Manojkumar, L. S. Kumar and B. Jayanthi, "Performance Comparison of Real Time Object Detection Techniques with YOLOv4," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-6, doi: 10.1109/IConSCEPT57958.2023.10169970

[13] V. A. Rajan, S. Sakhamuri, A. P. Nayaki, S. Agarwal, A. Aeron and M. Lawanyashri, "Optimizing Object Detection Efficiency for Autonomous Vehicles through the Integration of YOLOv4 and EfficientDet Algorithms," 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, 2024, pp. 1-5, doi: 10.1109/TQCEBT59414.2024.10545157.

[14] N. Chatterjee, A. V. Singh and R. Agarwal, "You Only Look Once (YOLOv8) Based Intrusion Detection System for Physical Security and Surveillance," 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2024, pp. 1-5, doi: 10.1109/ICRITO61523.2024.10522139.

[15] A. Afdhal, K. Saddami, S. Sugiarto, Z. Fuadi and N. Nasaruddin, "Real-Time Object Detection Performance of YOLOv8 Models for Self-Driving Cars in a Mixed Traffic Environment," 2023 2nd International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), Banda Aceh, Indonesia, 2023, pp. 260-265, doi: 10.1109/COSITE60233.2023.10249521.

[16] X. Peng, L. Zeng, W. Zhu and Z. Zeng, "A Small Object Detection Model for Improved YOLOv8 for UAV Aerial Photography Scenarios," 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 2024, pp. 2099-2104, doi: 10.1109/AINIT61980.2024.10581840.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)