iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83918

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 14 Issue VI June 2026- Available at www.ijraset.com

Objects Detection System Using Yolo and Open CV

Anusha A S[1], Preethi C K[2], Rithu Patel R[3], Thanushree B S[4], Dr. Anand M J[5]
Department of ECE, PESCE, Mandya, India

Abstract: ***This mini-project focuses on a real-time object detection system for images and live video. OpenCV is used for the processing of frames, and the YOLO algorithm with TensorFlow is used for instant identification and display of the names of the items. It is very flexible and works great with pre-trained and custom-trained models using inputs from webcams or video files. Once the object is detected, the live labelling is performed, and a bounding box surrounds the object. It helped to get practical deep learning skills and was a real solution for smart surveillance, automated traffic, and intelligent applications requiring fast visual recognition.***

Keywords: Object Detection, YOLO(You Only Look Once), OpenCv, TensorFlow, Deep learning, Webcam, Visual Recognition.

I. INTRODUCTION

Object detection has become one of the most significant advances in computer vision. for computers to identify and find objects of interest in images and video streams. As more modern applications increasingly rely on visual data for decision-making, the demand for efficient, accurate object detection systems continues to grow. grow. From autonomous cars to smart security systems Real-time object recognition is revolutionizing the way machines see the world around them. Deep learning techniques have significantly boosted the development of object detection systems, especially convolutional neural networks (CNNs). There are a number of different models available, including the YOLO(You Only Look Once) The You only look Once architecture is famous for its higher speed and accuracy. YOLO performs detection in a single pass. Unlikely that processes are in multiple stages; it strongly super pass Real-time applications. This project aims to design and implement an object detection system capable of recognizing specific objects such as people, vehicles and animals in image or video streams. The system enables using pre-trained YOLO models as well as custom trained models, providing flexibility depending on the desired application. The system can be extended with real-time processing with webcam/video input for continuous detect objects and tag them in the scene with high reliability. Object detection is still a very important task in the rapid development of artificial intelligence, especially in enhancing automation, safety and efficiency across a wide range of fields. This implementation of the system shows the potential of deep learning-based detection of real-world scenarios and highlights the increasing importance of intelligent visual monitoring technology.

II. LITERATURE SURVEY

1) J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” CVPR, 2016. This paper introduced the YOLO algorithm, which performs object detection as a single regression problem. Unlike region-based methods, YOLO processes the entire image at once, achieving high speed and good accuracy for real-time applications. However, it faced challenges in detecting small objects.

2) R. Girshick, “Fast R-CNN,” ICCV, 2015. Fast R-CNN improved the speed and accuracy of object detection by processing the entire image through a CNN and extracting region features from a shared feature map. Although faster than R-CNN, it still relied on Selective Search for region proposals, limiting real-time performance.

3) J. Huang et al., “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors,” CVPR, 2017. This study compared popular object detection models such as Faster R-CNN, SSD, and R-FCN using the TensorFlow Object Detection API. The authors highlighted the trade-off between detection speed and accuracy, helping developers choose suitable models for real-time or high-accuracy applications.

4) T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” ECCV, 2014. This paper introduced the MS COCO dataset, a large-scale benchmark for object detection and image segmentation. Its diverse and richly annotated images enable the training and evaluation of robust deep learning models for real-world object detection tasks.

5) G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000. This paper introduced OpenCV (Open-Source Computer Vision Library), which is a widely used library for computer vision and image processing applications. OpenCV provides many built-in functions for tasks such as reading and displaying images, capturing video from cameras, detecting objects, tracking movement, and performing image transformations. It is especially useful for real-time applications because it

©IJRASET: All Rights are Reserved

processes images and video frames quickly and efficiently. OpenCV can be easily integrated with deep learning frameworks like TensorFlow, allowing developers to use trained models for object detection and recognition. In practical projects, OpenCV is commonly used to capture live video from webcams, preprocess images by resizing or converting colour formats, and draw bounding boxes around detected objects. Because of these capabilities, OpenCV plays a very important role in building real-time object detection systems.

III. PROBLEM STATEMENT AND OBJECTIVE

A. *Problem Statement*

Existing Object Detection systems are often suffering from the problems of high computational cost, low detection speed and reduced accuracy in real-time environments. Many current solutions rely heavily on cloud-based processing, making them unsuitable for offline or low-latency applications. Such limitations degrade performance in applications such as surveillance, traffic monitoring and automation where timely detection is important. Hence there is a need of fast, efficient and cost effective object detection system able to process video streams in real-time, accurately recognize multiple objects and run without using external servers independently.

B. *Objective*

1) To develop a real time system to accurately detect and classify the static and live media.
2) Implement a TensorFlow-based real-time object detection pipeline utilizing pre-trained YOLO model.
3) Utilize OpenCv for real-time video capture image processing and visualization of detection results from webcam and file-based streams.

IV. METHODOLOGY

We developed this system to detect and identify objects in real time using the YOLO object detection algorithm. The system captures input from a webcam, video file, or image dataset and preprocesses the frames for analysis. The YOLO model then detects objects, predicts their locations, and classifies them with confidence scores. Finally, the detected objects are displayed with labeled bounding boxes, enabling fast and efficient real-time monitoring.

A. *Data Acquisition*

The system receives input either from a webcam, video file, or image dataset. Frames are continuously captured and forwarded to the processing module for analysis.

B. *Image Processing*

The captured images are pre-processed to improve detection accuracy. This includes resizing images to the required input dimensions, normalizing pixel values, and converting image formats when necessary. These steps ensure compatibility with the object detection model.

C. *Object Detection and Classification*

The pre-processed frames are passed to the YOLO (You Only Look Once) object detection model. YOLO processes the entire image in a single forward pass and identifies multiple objects simultaneously. For each detected object, the model predicts:

- Object class (e.g., person, car, bottle, chair)
- Bounding box coordinates
- Confidence score

D. *Performance Evaluation*

The system's performance is evaluated using metrics such as detection accuracy, precision, recall, mean Average Precision (mAP), and Frames Per Second (FPS). These metrics help assess both detection quality and real-time processing capability.

V. BLOCK DIAGRAM

Block diagram representing the working process of a real time object detection system. The system begins with a video input(webcam, video file) that outputs continuous frames for analysis. These frames are then sent to the image preprocessing stage, where operations such as frame capture, resize and normalize to convert the data to a format for detection accuracy.

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

After preprocessing, the frames are passed to the object detection module (TensorFlow & YOLO), where the YOLO deep learning model takes the image in one go and identifies objects like people, vehicles or animals. It is fast, efficient and accurate for object detection. Lastly, the output stage of detected objects displays the result by drawing bounding boxes around detected objects in real time and label them on-screen. This entire process also allows for fast, accurate and real time object detection for surveillance and monitoring application.

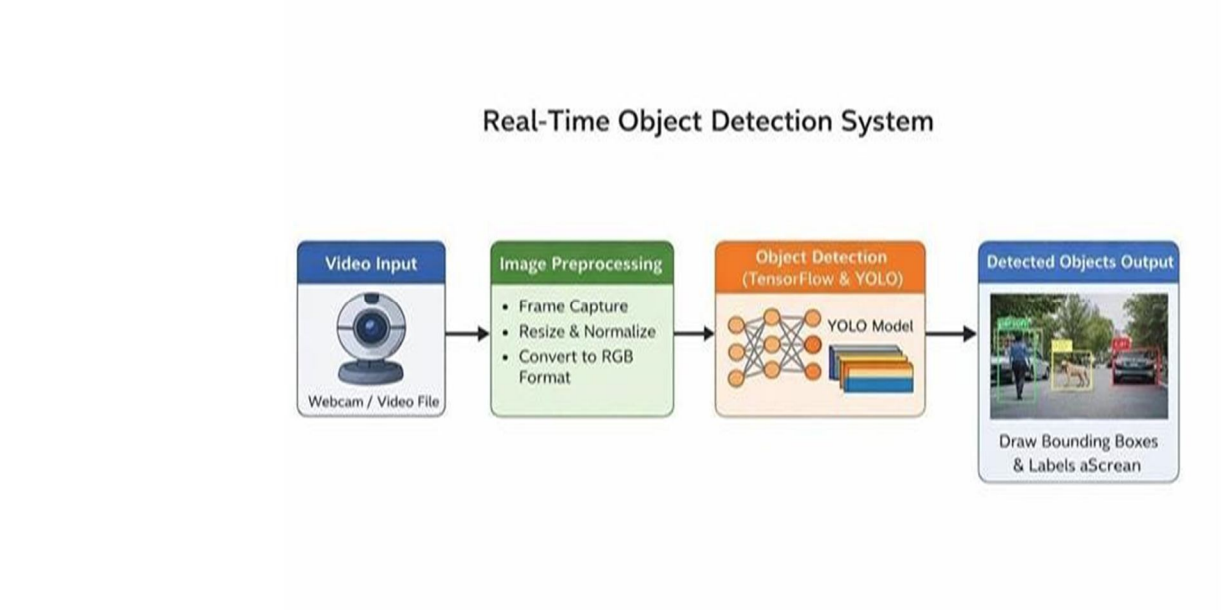


Fig. 1. Block Diagram of Object Detection System

A. Video Input and Frame Acquisition

The system begins by capturing real-time video through a webcam or by reading frames from a prerecorded video file. Each frame acts as an individual image that is forwarded to the processing pipeline. Continuous frame acquisition ensures that moving objects can be detected and tracked in real time. The captured frames are temporarily stored in memory and passed to the preprocessing module for further enhancement.

B. Image Preprocessing

Before object detection is performed, each input frame undergoes a preprocessing stage to improve detection accuracy and computational efficiency. The frame is resized to match the input dimensions required by the YOLO model and normalized to ensure consistent pixel intensity values. Color conversion from BGR to RGB format is also carried out because the TensorFlow implementation of YOLO expects RGB images. These preprocessing operations reduce noise and prepare the image data for efficient feature extraction.

C. Object Detection Using TensorFlow and YOLO

After preprocessing, the prepared frame is fed into the YOLO (You Only Look Once) object detection model implemented using TensorFlow. The model processes the entire image in a single forward pass through a deep neural network and extracts high-level visual features. It then predicts object classes, confidence scores, and bounding box coordinates simultaneously. The detected objects are filtered using confidence thresholds and Non-Maximum Suppression (NMS) to eliminate duplicate detections and improve prediction quality. This approach enables fast and accurate real-time object recognition.

D. Detected Objects Visualization and Output

The final stage displays the detection results on the output frame. Bounding boxes are drawn around the identified objects, and corresponding class labels with confidence scores are displayed on the screen. The annotated frames are continuously rendered to provide a live visualization of detected objects. This allows users to monitor object presence and movement in real time, making the system suitable for surveillance, automation, and smart monitoring applications.

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

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

VI. RESULT AND DISCUSSION

The Real-Time Object Detection System successfully detected and classified objects from both webcam streams and video files using the YOLO algorithm. The preprocessing stage improved input quality, enabling faster and more accurate detection. Detected objects were displayed with bounding boxes and labels in real time, providing clear visual identification. The system achieved efficient performance with minimal delay, making it suitable for live monitoring applications. Overall, the integration of TensorFlow, YOLO, and OpenCV demonstrated reliable object detection with high accuracy and real-time responsiveness.

VII. CONCLUSION

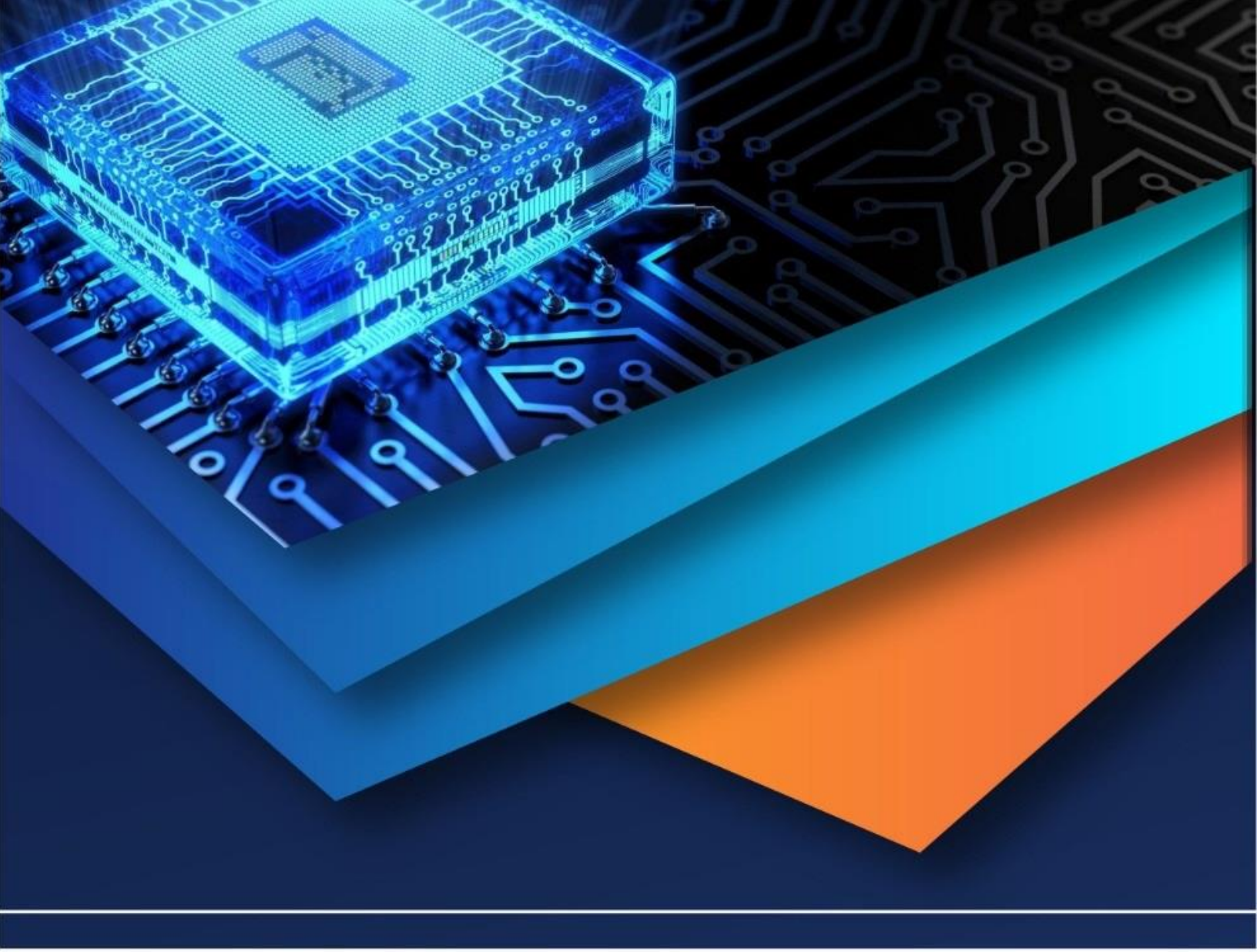
The Real-Time Object Detection System was successfully developed using OpenCV, TensorFlow, and the YOLO algorithm to identify and classify objects in images and live video streams. By incorporating image preprocessing techniques and a deep learning-based detection model, the system was able to detect multiple objects accurately while maintaining real-time performance. The use of YOLO enabled fast processing speeds, making the system suitable for applications that require immediate object recognition and monitoring.

The project demonstrates the effectiveness of modern deep learning techniques in computer vision tasks. The generated output, which includes bounding boxes and object labels, provides clear visual feedback to users. Overall, the system achieved its objective of delivering efficient, reliable, and real-time object detection. Future enhancements may include custom object training, object tracking, and deployment on embedded or edge devices for wider practical applications.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016
[2] R. Girshick, “Fast R-CNN,” Proceedings of the IEEE International Conference on Computer Vision(ICCV), 2015.
[3] J. Huang et al., “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[4] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” European Conference on Computer Vision (ECCV), 2014.
[5] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

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



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)