iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71512

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 V May 2025- Available at www.ijraset.com

Combining YOLO and R-CNN for Robust Object Detection

Khushi Singh[1], Miss. Ankita Dubey[2], Somil Singh[3], Anand Kumar Verma[4], Abhishek Pathak[5]
Goel Institute Technology & Management Lucknow

Abstract: ***This research focuses on the development of a real-time object detection system using deep learning techniques. The system is designed to accurately identify and localize multiple objects within video frames in real-time, making it suitable for applications such as surveillance, autonomous vehicles, and smart environments. Real-time object is a complex area and fundamental of computer vision. Due to its increased utilization in face recognition, tracking system, robotics, augmented reality and surveillance used in security and many others applications like live streaming filters (Snapchat, Instagram). The goal is to identify objects Which is done with the help of YOLO to locate object using bounding boxes [1]. YOLO (You Only Look Once) is a new approach to object detect, it outperforms other detection methods, including DPM and R-CNN when generalizing from natural images to other domains like artwork. In recent years, the integration of real- time object detection with edge computing and IoT has gained traction. This combination allows smart devices to make on-the spot decisions without relying heavily on cloud services enhancing privacy and reducing latency. [2]It is briefly describes the development process of the YOLO algorithm, summarizes the methods of target recognition and feature selection, Besides, this paper contributes a lot to YOLO and other object detection literature.***

Keywords: R-CNN, YOLO,CNN, Real-time object detection.

I. INTRODUCTION

Object detection is a transformative technology which is the branch of computer vision focuses on locating and identifying objects within an image or video stream as quickly as possible using bounding box Tasks belonging to such application like industrial robots, driving identify authentication smart health care, surveillance of visual etc. There are two task involve for process classification, Which identify what the object is (e.g. car, cow, bicycle) and localization, which Identifies where the object is in the frame using bounding boxes. Real-time object detection is made possible by advance deep learning algorithms and powerful hardware as in figure[1] Popular like YOLO, R-CNN and single shot detector which commonly used .Basically, YOLO [3]being particularly noted for its speed and efficiency or accuracy also it is a new approach to real time object detection repurposes classifiers to perform detection. As shown in figure 1.

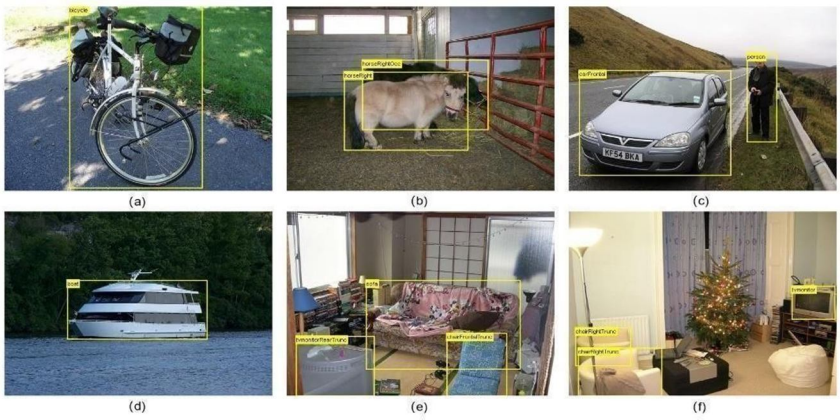


Figure1:-The YOLO Detection System. Processing images with YOLO is simple and straightforward.

The core idea of detection is still the same:- detect objects in one pass, but the architecture and tricks have gotten way better by YOLO**.** Real-time object detection is a cutting-edge field in computer vision and artificial intelligence that enables machines to identify and locate multiple objects within an image or video stream as it happens. Unlike traditional image recognition, which only labels a scene, real-time detection provides both object classification and spatial awareness.

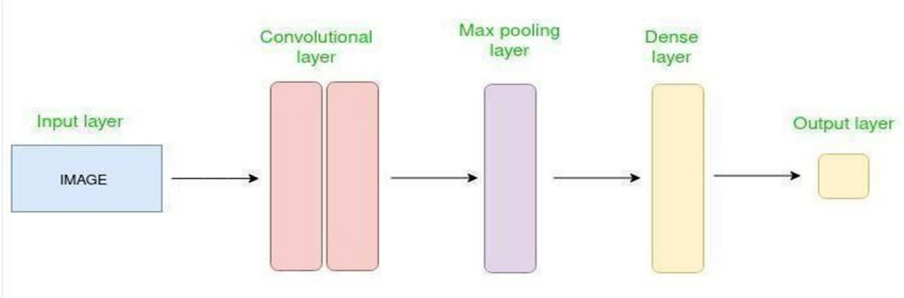
II. RELATED WORK

The efficiency of Object detection growth in the past few years, Single-stage detectors for such as yolo object (You only look once) and SSD (Single shot multibox Detector) have gain significant attention for real-time object detection due to their ability to balance speed and efficiency. Unlike two-stage detectors, which first generate region proposals and then classify them, single-stage methods predict object classes and bounding boxes in one forward pass through the network. YOLO, introduced by Redmon et al., pioneered this approach and evolved through versions (YOLOv2, YOLOv3, YOLOv4, and more recently YOLOv5 and YOLOv8) [9]to significantly improve accuracy while maintaining high frame rates. These models are particularly well-suited for real- time scenarios, such as surveillance, autonomous vehicles, and robotics, where low latency is critical. Recent improvements incorporate advanced backbone architectures (e.g., CSPDarknet, MobileNet) and optimization techniques (e.g., quantization, pruning), making them even more suitable for deployment on edge devices. Other single-stage models such as SSD (Single Shot MultiBox Detector) and RetinaNet have also contributed to this area, with SSD emphasizing multi-scale feature maps and RetinaNet introducing the focal loss to handle class imbalance. However, YOLO remains dominant in real-time scenarios due to its high frames-per-second (FPS) performance and relatively compact model size.Recent research also explores lightweight and mobile- optimized versions of YOLO (e.g., YOLO-Nano, Tiny-YOLO) for deployment on edge devices. These advancements demonstrate that single-stage detectors continue to evolve toward enabling real-time object detection in resource-constrained environments without significantly sacrificing accuracy.

III. METHODOLOGY

This research implements and evaluates real-time object detection systems using two prominent deep learning models: You Only Look Once (YOLO) and Region- based Convolutional Neural Networks (R-CNN). The methodology involves dataset preparation, model selection and implementation, training and evaluation, and performance comparison,neural networks in modern object detection systems act as a visual brain, their hierarchical structure enables the automatic extraction of spatial and semantic features from images, which are essential for detecting and classifying objects accurately [12]. Traditional CNNs are usually used for image classification, [7] not detection they output just one label per image. It is primarily used for feature extraction in object detection pipelines. Early layers detect simple patterns like edges and textures, while deeper layers capture complex patterns such as shapes and object parts. These multi-level features help the model understand what is in the image and where it is located. CNNS consist of multiple layers like the input layer, convolutional layer, pooling layer, and fully connected layers the step- by-step flow of data, starting from the input image, through convolutional and pooling layers for feature extraction, followed by flattening and fully connected layers, and ending with a softmax output layer that predicts the class of the image. The above diagram of CNN works by taking an input image, applying convolutional layers to extract features, using activation functions like ReLU to add non-linearity, downsampling the data with pooling layers, flattening the feature maps into a vector, passing it through fully connected layers, and finally using a softmax layer to classify the image. Several research papers have explored the use of Convolutional Neural Networks (CNNs) for detection tasks [8],But there is One major drawback is their high computational complexity, deep CNN architectures often demand substantial processing power and memory, which poses challenges for deployment on resource-constrained devices such as mobile platforms or embedded systems. Additionally, CNNs typically require large volumes of labelled training data to achieve optimal performance. [15] This dependence on extensive datasets can be problematic in domains where annotated data is scarce or expensive to collect. These are negatively impact detection accuracy. In response to these challenges, the computer vision community has explored alternative architectures, most notably the emergence of Vision Transformers and DEtection Transformer.

Figure 2



A. *R-CNN (Regions with Convolutional Neural Networks)*

R-CNN marked a significant advancement in object detection by introducing a twostage framework which combines traditional region proposal methods with deep learning. [11] Initially, A set of region proposals that are likely to contain objects. Each of these regions is then independently passed through a convolutional neural network to extract deep features, which are subsequently classified using support vector machines and refined with bounding box regressors. While R-CNN demonstrated improved accuracy over previous techniques. R-CNN Works in multiple steps — first generates region proposals, then classifies each region.

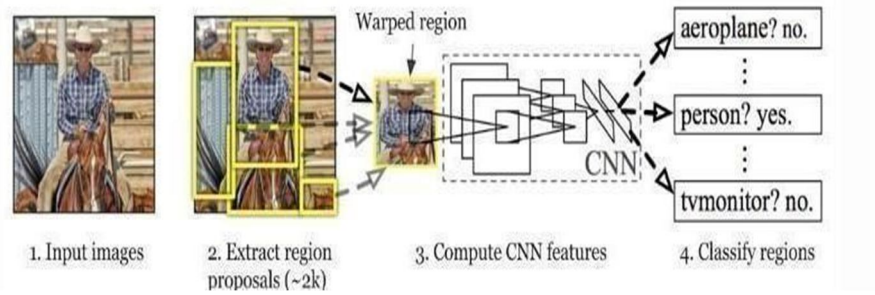


Figure 3

This R-CNN diagram 3 shows how the model works in four steps: first, it takes an input image and generates about 2,000 region proposals where objects might be. Then, each region is cropped and resized (warped) to a fixed size and passed through a CNN to extract features. Finally, each region's features are sent to a classifier (like an SVM) to determine if it belongs to a specific class, such as "person", "aeroplane" or "tvmonitor." A deeper understanding of this approach was facilitated by the comprehensive analysis in the original R-CNN report [2].There are some main functions and aims of R-CNN in real-time object detection:-

- Generate candidate regions in an image that might contain objects using algorithms like Selective Search.
- Extract deep features from each proposed region using a pre- trained Convolutional Neural Network (e.g., AlexNet).
- Classify each region as a specific object class (e.g., car, person, dog) or as background using a classifier like SVM.
- Classify each region as a specific object class (e.g., car, person, dog) or as background using a classifier like SVM.
- Achieve higher detection accuracy than older methods.
- Introduced deep learning into the object detection pipeline, bridging classification and localization tasks etc.
- Bounding Box Regression**.**

B. *YOLO (You Only Look Once)*

You Only Look Once is a real-time object detection algorithm that identifies and classifies objects in an image using a single neural network pass, making it fast and efficient for applications. YOLO is end-to-end and faster because it doesn't need a separate region proposal step. There are few algorithm development review paper [3]. There are several versions, each better than the last: There are few research paper from YOLOv1 to YOLOv10 [4] [5]

YOLOv1:- YOLOv1 treats detection as a single regression problem and predicting bounding boxes , class probabilities directly from the entire image using a single convolutional neural network, making it extremely fast but with some trade-offs in localization accuracy, especially for small objects. YOLOv1 divides the input image into a grid, and each grid cell predicts bounding boxes and class probabilities for objects whose centers fall inside that cell, it treats detection as a single regression problem and processes the entire image in one forward pass through a convolutional neural network, making it extremely fast and efficient for real-time object detection.

YOLOV2(YOLO9000):- It is an improved version of YOLOv1 that enhances both accuracy and speed by introducing anchor boxes, batch normalization, dimension clustering, and a higher-resolution classifier, allowing it to detect multiple objects more precisely while remaining efficient for real-time applications.

YOLOv3:- YOLOv3 improves upon YOLOv2 by using a more powerful backbone (Darknet-53), multi-scale predictions, and better bounding box and class predictions, allowing it to detect smaller objects more accurately and handle multiple object sizes while maintaining real-time speed.

YOLOv4:- YOLOv4 builds upon YOLOv3 by incorporating Mosaic data augmentation, advanced techniques like CSPDarknet53 as the backbone, PANet for better feature pyramid networks, and dropblock regularization, significantly improving accuracy and robustness while maintaining high speed for real-time object detection.

YOLOv5:- YOLOv5, developed by the Ultralytics team, is an unofficial yet popular PyTorch-based implementation of YOLO, offering improved ease of use, faster training, and better performance with multiple model sizes, advanced augmentation techniques, and easy deployment to edge devices.

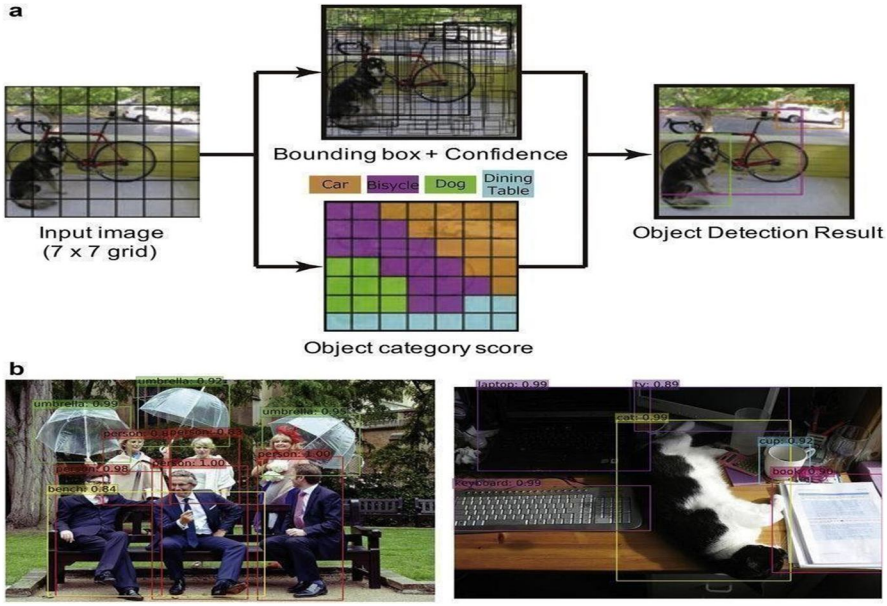
YOLOv6 focuses on optimizing YOLO for real-time deployment on edge devices, such as smartphones and IoT devices, where computational power is limited. It introduces techniques like the Decoupled Head to improve performance without sacrificing efficiency, enabling faster training and more accurate predictions. This version is ideal for applications that require quick, on-site inference without heavy computational resources.

YOLOv7 takes YOLO to the next level by achieving state-of-the-art (SOTA) performance on benchmark datasets. It introduces several advanced techniques, including Re-parameterization, which improves the model's flexibility and efficiency, and Dynamic Convolution, which adapts the convolution process dynamically, improving the overall performance of the model. YOLOv7 balances efficiency and speed, making it suitable for high-performance applications while maintaining real-time detection.

YOLOv8, developed by the Ultralytics team (the same team behind YOLOv5), focuses on simplifying and improving the user experience when deploying and training models. This version is more user-friendly, provides better support for PyTorch and TensorRT, and incorporates optimizations that increase the accuracy of predictions and overall performance. YOLOv8 is particularly suited for edge and mobile applications, where deployment flexibility and efficient model inference are critical.

YOLOv9: YOLOV9 is a new model scaling techniques and architectural changes improve accuracy and flexibility, catering to diverse needs in speed and efficiency.

YOLOv10:YOLOv10 is a advance model of YOLOv9, enhancing customization options and optimizing performance for different use cases. YOLOv10 work in figure[1] where only requires one pass to detect objects, it can process videos or image streams quicker than other models. where speed is critical to handle YOLOv10 make it effective and accurate, like traffic monitoring. In the below figure it detects objects by dividing an input image into a grid, predicting bounding boxes and confidence scores for each cell, and assigning category scores to identify object like people ,cat cup, keyboard all in a single fast pass, making it highly efficient for real- time application.



C. Real-time object detection

Real-time object detection is the process of identifying and locating objects within an image or video stream as it happens — typically using a camera feed. This involves detecting objects and drawing bounding boxes around them with associated class labels (e.g., "person," "car," "dog") in real-time, ideally with minimal latency. [14] To achieve real-time performance, object detection systems use optimized deep learning models like YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), or lightweight versions such as MobileNet- SSD. These models are capable of processing video frames in milliseconds, making them suitable for live applications [10].

The typical process involves capturing a video frame, preprocessing it (such as resizing and normalization), running it through the

detection model, and then displaying the results with visual annotations.

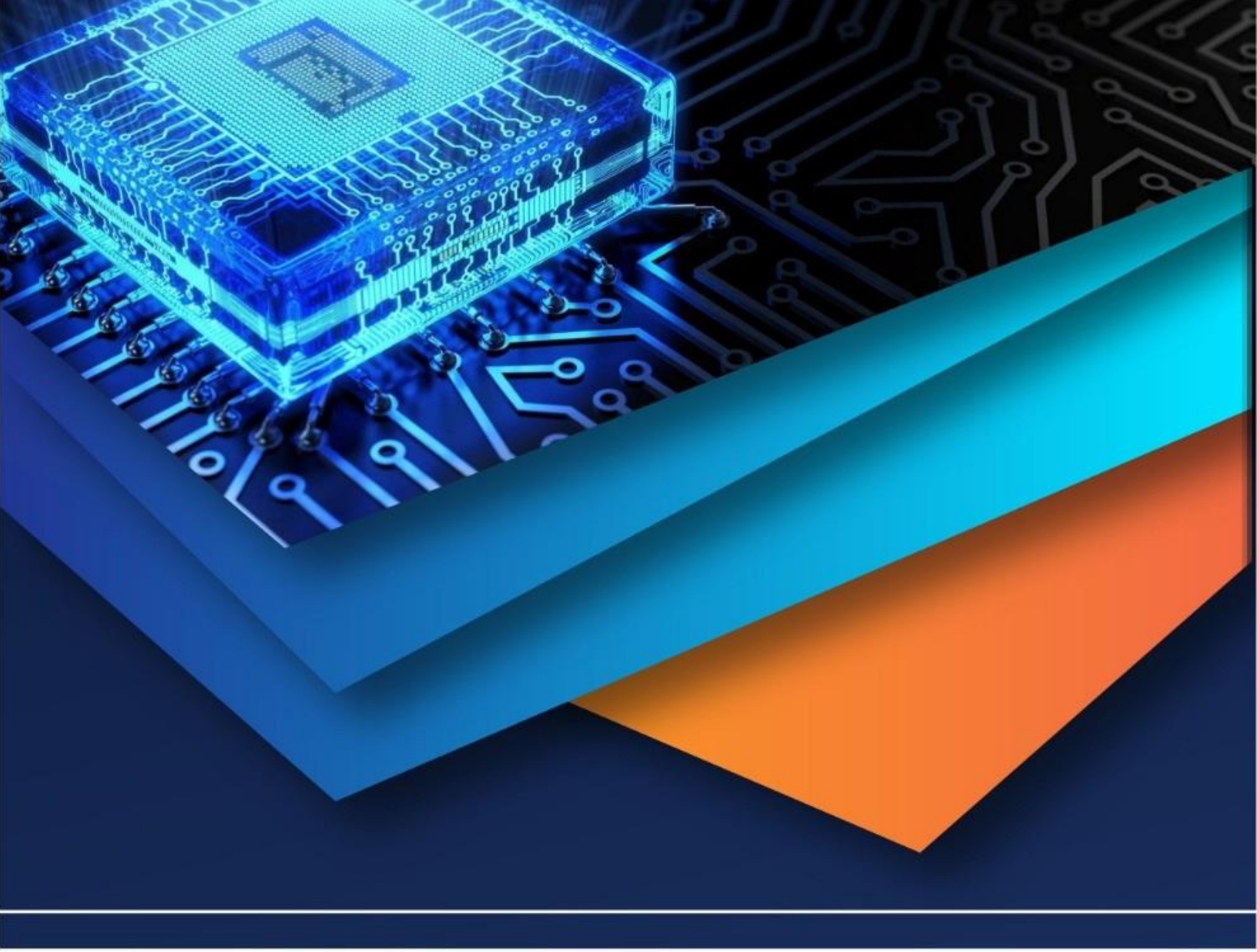
Frameworks like TensorFlow [13]PyTorch, and OpenCV are commonly used to build real-time detection systems. The combination of speed and accuracy allows developers to create intelligent systems that can react to the environment in real time, making object detection a cornerstone of modern AI- powered solutions.realtime object detection plays a vital role in modern AI applications by enabling faster, smarter, and more automated solutions that improve efficiency, safety, and user convenience in everyday life. In the modern era, real-time object detection has become an essential technology across various industries due to its speed, accuracy, and ability to process visual data instantly.

IV. CONCLUSION

This paper gives us a review of the YOLO version and the work of CNN, R-CNN. Here we draw the following remarks. In which First CNN work in object detection and after that know about R-CNN, the step of work, aim and function. YOLO version has a lot of differences. YOLO stands out as the most practical solution for real-time object detection, making it ideal for applications like autonomous driving, surveillance, and robotics and it is more adopted in modern applications. Future advancements may see a convergence of these approaches, combining the accuracy of region based methods with the speed of single-shot detectors to meet the ever increasing demands of intelligent systems.

REFERENCES

[1] V. S. K. N Mittal, " Object detection and classification using Yolo," academia, 2019.
[2] S. D. R. G. J Redmon, "You only look once: Unified, real-time object detection," foundation, 2016.
[3] Y. J. W. C. Y. H. Y. Z. X Xu, "DAMO-YOLO : A Report on Real-Time Object Detection Design," arxiv., 2023.
[4] C. R. K. R. A. Viswanatha V, "Real Time Object Detection System with YOLO and CNN Models: A Review," cs.CV], 2022.
[5] H. L. CY Wang, " YOLOv1 to YOLOv10: The fastest and most accurate real-time object detection systems," nowpublishers, 2024.
[6] K. M. SV Kothiya, "SV Kothiya, KB Mistree," researchgate, 2015.
[7] K. H. R. G. J. S. Shaoqing Ren, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," .neurips, 2015.
[8] M. I.-K. -. I. A. S Sambolek, "Automatic person detection in search and rescue operations using deep CNN detectors," ieeexplore, 2012.
[9] L. K. HDI Upulie, "Real-time object detection using YOLO: a review," researchgate., 2021.
[10] A. R. RK Chandana, "Real time object detection system with YOLO and CNN models," arxiv, 2022.
[11] Y. W. H. S. R. F. J. X. T. H. Bowen Cheng, "Revisiting RCNN: On Awakening the Classification Power of Faster RCNN," thecvf, 2018.
[12] G. Z. D. S. W Zhang, "Real-time accurate object detection using multiple resolutions," ieeexplore, 2007.
[13] A. P. B. B. A Talele, "Detection of real time objects using TensorFlow and OpenCV," asianssr, 2019.
[14] K. S. G Khekare, "Real time object detection with speech recognition using tensorflow lite," researchgate, 2022.
[15] C. P. H. T. J. P. R. V. D Bhatt, " CNN variants for computer vision: History, architecture, application, challenges and future scope," mdpi., 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY