International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

YOLOv5: A Comprehensive Technical Review of a Paradigm-Shifting Object Detector

Mr. Harish M[1], Rakshita A[2], Ramya[3], Rohit P G[4], Sachith R[5]

[2,3,4,5]BE Students, Computer Science and Design Department, PES Institute of Technology and Management, Shivamogga, Karnataka, India.
[1]Professor, Computer Science and Design Department, PES Institute of Technology and Management, Shivamogga, Karnataka, India.

ABSTRACT
Real-time object detection is of paramount importance for robotics, autonomous vehicles, video analytics, and surveillance. This work presents the design, implementation, and evaluation plan of a practical real-time object detection system that balances a tradeoff between inference speed (in FPS) and quality of detection (in mAP). The system includes a lightweight detection backbone, optimized detection head, anchor-free YOLO family or lightweight DETR variants, and engineering optimizations such as model pruning, quantization, TensorRT/ONNX conversion, and multi-scale NMS. We detail the architecture, training regimen on COCO/transfer datasets, deployment strategy to edge GPUs, and experimental protocol for benchmarking latency, throughput, and accuracy. Modern state-of-the-art YOLO-family and efficient one-stage detectors provide very competitive speed-accuracy trade-offs in real-time applications, as revealed by results from the literature.

Keywords: Feedback Classification, Natural Language Processing (NLP), Machine Learning (ML), Logistic Regression, College Help Desk, Automation, Sentiment Analysis.

1. INTRODUCTION

1.1 Problem Statement :

Real-time object detection remains one of the most challenging tasks because it requires a fine balance between high accuracy and low latency under rapidly changing environmental conditions. Most traditional object detectors are not designed to process video streams at fast frame rates while maintaining reliable detection performance, especially under changing lighting, partial occlusions, and multiple objects per frame. Several state-of-the-art systems have computational resource-heavy requirements, which easily limit their deployment on edge devices for real-world applications like surveillance, autonomous navigation, and monitoring traffic flow. There is thus a need for the development and optimization of a lightweight and efficient object detection system, capable of real-time analysis of video input, detection of objects with good enough accuracy, and easy execution on generally available hardware without notable performance degradation.

1.2 Objectives :

The main objectives of this project include:

- This, in turn, can allow the real-time detection and classification of objects with minimum latency, ensuring smooth and continuous performance on live.

- To achieve high detection speed, targeting 25–30+ FPS by using optimized AI/ML models that are suitable for real-time applications.
- To develop methods for improving system efficiency by utilizing lightweight architecture, model compression, and hardware acceleration techniques.
- Reliable performance in many different settings needs to be eensuredrecurring problems enhance such as lighting, movement, occlusions, crowded scenes, etc.

This will result in model training and fine-tuning for enhanced accuracy, with sure-shot detection performance on both general and domain-specific datasets. The primary goal of real-time object detection systems is to accurately identify objects in live video streams and classify them with minimum delay, ensuring smooth and continuous performance. High-speed detection, at 25-30 frames per second or higher, is a target achieved by the inclusion of efficient machine learning architectures and other optimization techniques. Furthermore, enhancing the robustness of detection in challenging environments subjected to variable lighting, motion blur, occlusions, and crowded scenarios is another key objective. The system will be able to handle resource-constrained hardware such as edge devices, embedded platforms, and low-power processors to make real-time detection feasible and extendable. For practical applications in surveillance, autonomous navigation, and traffic monitoring, the system should enable tracking, event detection, and timely notifications. The system will also strive for higher accuracy through model fine-tuning, adaptive learning, and domain-specific training. Finally, it is important to have a flexible and modular framework that can easily be integrated or extended for a wide range of real-world use cases, considering its long-term usability and adaptability.

2. LITERATURE REVIEW

Real-time object detection has rapidly evolved from classical two-stage detectors, e.g., Faster R-CNN, to highly optimized one-stage detectors that emphasize speed with limited sacrifice of accuracy. One-stage architectures, in particular the YOLO family, have dominated practical real-time applications by framing detection as a single end-to-end regression/classification problem; successive YOLO releases and community forks have targeted the speed–accuracy tradeoff through architectural refinements and training “bag-of-freebies” techniques, and have become a common baseline in real-time systems.

Real-time object detection is a fundamental and challenging task in the domain of computer vision, differing from the usual object detection due to strict computational time constraints [1]. The main objective is not just to recognize (classify) an object with a high degree of accuracy and pinpoint it (localize) within an image, but also to do so at a high enough speed to process a live video feed without lag [2]. In other words, this usually requires a speed higher than 30 FPS processing rate, which creates an impending trade-off challenge between speed and accuracy [6]. This technology lies at the heart of many modern applications, including, but not limited to, autonomous driving, robotics, video surveillance, and manufacturing automation [2, 7].

It is common to categorize the evolution of real-time detection based on its architectural approach: two-stage versus one-stage detectors.

Historically, two-stage detectors such as those in the Region-based Convolutional Neural Network family have set the standard for accuracy [5]. Models such as Faster R-CNN work by first having a Region Proposal Network (RPN) scan the image to find areas likely to contain an object. A second stage then analyzes only these "regions of interest" for classification. While highly accurate, this sequential, multi-stage process creates a computational bottleneck that generally prevents true real-time performance [5].

In this regard, the You Only Look Once algorithm actually redefined object detection as a single regression problem [8]. Instead of a two-step process, YOLO looks at the entire image just once and predicts all bounding boxes and class probabilities simultaneously. This single-pass, unified architecture is very fast, making it a foundational model for many real-time tasks [1, 9]. Apart from YOLO, there emerged another powerful one-stage approach known as the Single Shot MultiBox Detector. The key innovation in SSD was the use of multi-scale feature maps, which allowed detecting objects of various sizes in a single pass with the support of a set of default "anchor" boxes [4].

It is interesting to note how the rapid evolution of the field so far mainly revolves around improvements to these one-stage design ideas. Specific attention could be drawn to many iterations of the YOLO algorithm, such as YOLOv3, YOLOv4, and YOLOv8, where each new version has tried (and often succeeded) in pushing the limits even further on the accuracy/speed trade-off [3, 9]. Of these, the more modern versions, such as YOLOv8, stand at the state of the art, offering several model sizes that let a developer choose an appropriate balance for a given application, from small, fast models suitable for edge devices to larger, more accurate models suitable for high-end GPUs [3].

Transformers, the architecture powering large language models, have also recently been adapted for vision. Models such as DETR and its real-time versions, RT-DETR, are setting new state-of-the-art results, often by eliminating the need for hand-tuned components, such as anchor boxes, allowing for further streamlining of the detection process [10]. In summary, the literature manifests a clear and imperative movement from accurate-but-slow two-stage methods [5] to extremely efficient one-stage architectures [4, 8]. The key open issue remains the "speed-accuracy trade-off" [6], which is the gold standard for all new models. The ongoing evolution from the original YOLO [8] to more modern architectures like YOLOv8 [3] and RT-DETR [10] shows continuous and fruitful research work to let computer vision systems see and react instantaneously to the world..

3. METHODOLOGY

The undertaking utilizes an organized, intricate pipeline that consists of multiple stages and is aimed at the processing of data in an efficient manner, along with attaining superbly accurate text classification. The entire setup consists of capturing raw data at the very beginning and goes all the way to deploying a validated classification model for the purpose of real-time analysis at the end.

3.1 System Architecture Overview:

The system architecture is a linear workflow process, and the main stages involved are as follows:

Data Collection: Gathering raw feedback from the source database (Airtable).

Data Preprocessing: Cleaning and normalizing the text data.

Feature Extraction: Converting text into numerical vectors (TF-IDF).

Model Training and Evaluation: Applying ML algorithms (Logistic Regression) and verifying performance.

Dashboard & Classified Output: Presenting the final categorized results in the User Interface (UI).

Figure: Block Diagram of A Real-time Object Detection System

The flow chart(Figure 1) depicts the operation of the Smart Feedback Classifier for the College Help Desk System. The whole activity starts with gathering feedback and comes to an end with monitoring and decision-making by the administration in real-time.

Feedback Submission by User (Google Form / Web Forms): Complaints or opinions of students and staff are delivered through an online form. The text is mostly unstructured, with the users explaining their problems in detail (e.g., "The internet connection in the hostel is down").

Text Preprocessing (Tokenization, Stopword Removal): The text that has been collected is going through preprocessing. Tokenization splits the sentences into single words. Stop-word Removal takes away the frequent words "the", "is", and "that" that do not contribute to the meaning. This process purifies and normalizes the text for subsequent processing. Feature Extraction (TF-IDF): The text is translated into numerical values via TF-IDF (Term Frequency–Inverse Document Frequency) after preprocessing. TF-IDF allows the model to detect major terms in the feedback (e.g., Wi-Fi, hostel, canteen, marks), thus increasing the precision of the classification.

Classification Model (Logistic Regression): The TF-IDF result is given to a Logistic Regression classifier, which unerringly sorts the feedback according to the rightful department:

Hostel / Canteen, Academic IT / Technical Examination Cell. Consequently, there is no need for manual checking and forwarding.

Airtable Database (Storage The feedback that has been classified, together with its status, urgency level, and timestamps, is kept in the Airtable database. This acts as a unified feedback store for the administration.

Admin Dashboard – Real-Time Analytics: A real-time dashboard is the means by which the feedback data is stored and shown graphically. The dashboard displays: The number of complaints for each department, Levels of urgency, resolved vs Pending, and overall student satisfaction trends.

3.2 Input Source: Camera / Video Stream:

Real-time object detection starts by first capturing a continuous visual feed from a camera, CCTV, drone, or live video stream. As raw input, this generally contains a sequence of frames coming in rapid succession, often 25–30 frames per second. Each frame becomes the basic unit that the system must process instantaneously. The system should be able to perform well across various resolutions, dynamic backgrounds, lighting changes, and motion changes while maintaining smooth real-time performance.

Feature Engineering and Model Training:

The training of a model in a real-time object detection system revolves around the capability of a deep learning model in recognizing objects and their locations within an image. First, it requires a labeled dataset where each image is bounded with a rectangular box and is further labeled with its class. Such feature patterns essentially include shapes, edges, textures, and color distribution, which the model learns by passing images through convolutional layers during training. It makes predictions through forward propagation and adjusts the errors through backpropagation with the help of a loss function that comprises a classification loss and a localization loss. Techniques such as data augmentation, learning-rate scheduling, and transfer learning are also employed in order to achieve better accuracy in less time. Over several epochs, the model gradually starts adjusting to the dataset and develops the ability to predict object locations and categories with high speed and precision, therefore finding its utilization in real-time applications.

Pre-Processing Stage :

Each incoming frame is pre-processed before any detection by the model, which improves detection acc-

uracy and reduces the computational load. Prevalent operations include resizing the frame to model-specific dimensions (e.g., YOLO at 640×640), normalizing pixel intensity values, and doing basic filtering to reduce noise. Pre-processing ensures uniform input quality, lowers unnecessary computation, and prepares the frame for efficient feature extraction inside the neural network.

3.5 Detection Model (Deep Learning Algorithm):

The core of the system is where AI/ML models like YOLO, SSD, Faster R-CNN, or EfficientDet detect objects present in the frame. The model extracts features using convolutional layers, analyzes the pattern, and predicts the location of an object and its class category. Real-time models have a steady balance of speed without sacrificing too much accuracy. Modern architectures, mainly YOLO and EfficientDet, are optimized to process multiple objects during a single forward pass of computation, so that high-speed inference is enabled on edge devices too.

3.6 Post-processing (NMS & Thresholding):

After making raw predictions, the model undergoes a post-processing stage to further refine the results. It eliminates duplicate bounding boxes over the same object using Non-Maximum Suppression. Thresholding based on the confidence removes weak or low-probability predictions, making sure that only the best and most meaningful detections are presented. This greatly improves clarity, reduces false positives, and enhances the quality of the final output.

4. RESULTS

Figure 2: A screenshot demonstrating the detection performance during live video processing. Each detected object is highlighted with a colored bounding box and a corresponding probability percentage indicating the model's confidence.

Figure 2 represents the visual analysis identifies two main factors :

Overview Summary of YOLOv5 Output: The output of a YOLOv5 (You Only Look Once version 5) real-time object detection system is not a single image, but rather a **structured numerical tensor** that lists every detected object, its location, and what the model thinks it is.

In simple terms, for every frame of video it processes, YOLOv5 returns a list of "bounding boxes." Each box tells you: "I found an object here, it is this big, I am X% sure it is a [Class Name] (e.g., Person, Car, Dog)."

Advantages :

- **Decisions happen instantly:** The biggest selling point here is speed. When you're dealing with things like self-driving cars or security feeds, you can't afford a delay. Real-time detection processes visual data the moment it happens, allowing the system to react immediately rather than reviewing footage after the fact.
- **You don't have to sacrifice accuracy for speed**: In the past, you usually had to choose: do you want it fast, or do you want it right? Modern algorithms like YOLO and EfficientDet have largely solved

this. They manage to keep frame rates high without losing the ability to spot details, which is crucial for time-sensitive tasks.

- **It takes the load off human operators:** Staring at a screen for hours is exhausting and leads to mistakes. These systems automate the boring stuff. Whether it's watching a traffic camera or inspecting products on an assembly line, the software handles the monitoring 24/7 without getting tired or distracted.
- **It takes the load off human operators:** Staring at a screen for hours is exhausting and leads to mistakes. These systems automate the boring stuff. Whether it's watching a traffic camera or inspecting products on an assembly line, the software handles the monitoring 24/7 without getting tired or distracted.
- **It works almost anywhere:** This technology isn't limited to just one niche. You see it popping up everywhere—from farmers using it to monitor crops, to retailers tracking inventory, to hospitals managing patient safety.
- **A massive boost for safety:** In high-risk environments, this tech acts as a second set of eyes. It can spot a worker without a helmet, unauthorized entry into a dangerous zone, or a hazard on the road much faster than a human can, triggering alarms before an accident happens.
- **Future Scope:**
  While the proposed system has the power to automatically classify feedback and perform the monitoring of administrators in real-time, there are still some possibilities of improvements that are going to be able to further increase the performance and usability of the system:
- **Moving everything to the "Edge"**: We are moving away from relying on big servers. The future is about running powerful AI on small, efficient chips (like the Jetson Nano) right on the device. This means faster processing and no need for a constant internet connection to work.
- **The 5G revolution:** When you combine real-time detection with 5G speeds, things get interesting. We're looking at ultra-low latency that could enable things we can't quite do yet, like reliable remote surgery or massive networks of connected vehicles talking to each other instantly.
- **Handling the "messy" real world**: Right now, things like heavy rain, bad lighting, or crowded streets can confuse detectors. The next generation of research is focused on making these algorithms robust enough to "see" clearly even when conditions are terrible.
- **Seeing with more than just cameras:** Cameras have limits. Future systems are going to be "multi-modal," meaning they will combine video data with sound, radar, and LiDAR. This gives the AI a much more complete, 3D understanding of the world, which is essential for safe robotics.
- **Learning on the fly:** Currently, retraining a model takes time. Future systems will likely feature "adaptive learning," where the AI can learn to recognize a new object or face instantly in the field without needing a full system update.

5. CONCLUSION

At its core, the development of real-time object detection represents a massive shift in how computers interpret the world. We have moved past the era where image analysis was a static, post-processing task and entered a phase where machines can perceive and react to their environment instantly. This capability is critical because, in the real world, timing is everything. Whether it is an autonomous car identifying a pedestrian or a manufacturing robot spotting a defect, the value of the information decays rapidly if it isn't

processed the moment it happens. The transition from standard detection to *real-time* processing bridges the gap between simple observation and actionable intelligence.

Technologically, this journey has been defined by the struggle to balance speed with accuracy. For a long time, developers had to choose one or the other, but recent advancements in one-stage architectures, particularly the YOLO family of models, have largely solved this dilemma. We have successfully moved away from the computationally heavy, multi-step processes of the past toward streamlined models that can run on everyday hardware without significant lag. This democratization of the technology means that high-level computer vision is no longer restricted to massive data centers but can now function reliably on smaller, local devices.

In conclusion, the goal is no longer just to draw a box around an object, but to understand its behavior and context in a crowded, dynamic environment. Ultimately, real-time object detection is evolving from a novelty into a fundamental piece of infrastructure that will silently power the smart cities and autonomous systems of tomorrow.

6. REFERENCES

1. S. P. V. S. Rao and M. Ramakrishna, "Real Time Object Detection System," *International Journal for Research Trends and Innovation (IJRTI)*, vol. 8, no. 4, pp. 266-269, 2023.
2. A. Kumar, S. Sharma, and R. Patel, "Real-Time Object Detection: Overview, Advancements, Challenges and Applications," *2023 International Conference on Computing and Communication Systems (ICCCS)*, pp. 45-52, 2023.
3. G. Solawetz, "YOLOv8 Guide: Architecture, Performance, and Comparison," *Viso.ai Computer Vision Technical Reports*, 2023.
4. J. Hui, "SSD Object Detection: Single Shot MultiBox Detector for Real-Time Processing," *Advances in Deep Learning Architectures*, 2018.
5. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.
6. N. Malviya, "Evaluation of Object Detection Models: FLOPS, FPS, Latency, and Memory," *Journal of Computer Vision and Pattern Recognition*, 2020.
7. M. H. T. Saiwa, "Real-World Applications of Object Detection Techniques in Industry," *Saiwa Artificial Intelligence Review*, 2023.
8. S. T. E. F. W. P. L. H. X. V. "Performance evaluation of machine learning models for customer [8] K. L. V. "YOLO Object Detection: Evolution and Applications in Computer Vision," *SuperAnnotate Technical Journal*, 2022.
9. J. Wang, Y. Chen, and Z. Liu, "Improved Algorithms for Real-Time Object Detection," *Preprints.org*, Manuscript 202510.2019, pp. 1-12, 2019.
10. G. Jocher, "RT-DETR: Real-Time Detection Transformer Model Architecture," *Ultralytics AI Research*, vol. 1, pp. 1-5, 2023.