



REAL-TIME SMART SURVEILLANCE USING YOLO-FASTER R-CNN HYBRID

Harmanpreet Singh¹ and Navdeep Singh²

¹ Student, M.Tech., CSE, Punjabi University, Patiala, Punjab, India

² Assistant Professor, CSE, Punjabi University, Patiala, Punjab, India

Abstract : This research introduces a hybrid object detection framework integrating YOLOv5 and Faster R-CNN aimed at improving the accuracy and reliability of high-speed smart surveillance systems. YOLOv5, being a fast detector, works as the primary detector for multi-class detection of persons, backpacks, handbags, books, guitars, and cell phones. However, YOLO's deliberate design for speed can generate false positives or incorrectly placed bounding boxes from time to time. To tackle this, the system refines selected classes of persons, backpacks, and handbags with Faster R-CNN, a more accurate but slower-based model. First, YOLOv5 detects possible objects, and detections are filtered by class and confidence threshold. For select classes, RoIs are cropped and passed to Faster R-CNN for refined bounding box predictions and confidence evaluation. That finally outputs the detections with color-coded annotations and labels for each class to maintain clarity and robustness. This approach is an elegant balance of speed and precision, making it extremely competent for real-time surveillance in a dynamic environment. The modular design facilitates scalability and adaptation to other application domains where high detection confidence is required without grossly compromising performance.

Keywords Object Detection, YOLOv5, Faster R-CNN, Smart Surveillance, Hybrid Model, Real-Time Detection.

1. INTRODUCTION

Object detection is the fundamental task in computer vision that has numerous practical applications like video surveillance, autonomous driving, smart city, medical diagnosis, and industrial inspection. In an object detection system, the detection and localization of one or more instances of semantic objects are performed on digital images or video streams. Although a lot has been done in this area, finding a good trade-off between detection accuracy and real-time performance is still challenging. To build upon this trade-off, the project proposes a hybrid object-detection system that chooses speed over accuracy with YOLOv5, while Faster R-CNN focuses on higher precision. The final design should be able to use the best of both worlds: the capability of YOLOv5 to produce fast, real-time detections and the capability of Faster R-CNN to make accurate, finer predictions of selected classes of interest. This approach is motivated by recent advancements in object detection research, where hybrid models have shown promise in balancing speed and accuracy [10][16][19].

The proposed system is implemented in Python using the two most popular deep learning libraries: PyTorch and Torchvision. The code starts by installing all necessary libraries, downloading pre-trained models, and uploading the target image. Using torch.hub.load to load the YOLOv5, the whole image undergoes fast detection, yielding initial bounding boxes, confidence scores, and class labels. These initial results are usable for real-time applications but are sometimes not entirely accurate due to the small size of the object, the presence of overlapping objects, or the occlusion of objects.

A second step is added in the processing chain for improving reliability. For some critical classes like 'person', 'backpack', and 'handbag', the system extracts the detected regions from the original image and elsewhere invokes a pre-trained Faster R-CNN model. This model, featuring a two-stage architecture and excellent detection accuracy, submits each cropped image patch to region proposal and classification. The hybrid system can

confidently go and check or make refinements upon the initial detection of YOLO by modifying bounding boxes and confidence score evaluation.

The overall workflow is methodically structured:

- 1. Image preprocessing and input handling:** The First step is uploaded and converted into an RGB file.
- 2. YOLOv5 Initial Detection:** The preliminary bounding boxes for all classes are detected by a real-time object detection phase using YOLOv5.
- 3. Class Filtering and Region Cropping:** Detections belonging to refinement-selected classes are cropped from the image.
- 4. Faster-RCNN Refinement:** Faster R-CNN looks at cropped regions to refine the positions and confidence in their class.
- 5. Bounding Box Fusion and Visualization:** Bounding boxes, either final from YOLOv5 or from the Faster R-CNN refinement, are drawn on the original image with colored labels.

The aforementioned hybrid detection strategy aims at in-built overcoming of limitations posed by using either YOLO or Faster R-CNN in isolation, in the very nature of each method. While YOLOv5 can work at high frame rates and quickly process the incoming data, its single-stage architecture could tend to give false positives or bounding boxes that are not very accurate. Faster R-CNN, in contrast, this ranking method would provide greater accuracy through its region proposal method but is too slow for real-time inference. Hence it is proposed that the solution merges both: one for speed and the other for reliability.

The practical exercise implementation extends to many fields. This hybrid architecture can detect intruders real-time and very accurately verify their presence in surveillance. For autonomous vehicles, this model can raise an alert for obstacles or pedestrians in real-time, then perform an accurate identification and classification to prevent false warnings. In low-power situations, e.g., in drones or embedded systems, the model could be further optimized to use YOLOv5 for coarse detection and Faster R-CNN only for objects that matter in terms of safety.

Overall, this work is a clear exposition of a well-balanced, efficient, and accurate method of object detection. It expresses the present scenario in deep learning research where multi-model integration and hybrid architectures are becoming increasingly applied to exploit the complementary strengths of individual models. Such an approach will serve as strong footing to build upon in intelligent visual recognition systems.

2. NEED FOR THE SYSTEM

This world is fast changing, so whether persons including their valuables or valuables alone, their security is becoming an issue, thus the situation demands a more concrete solution. Increased urbanization, population density, and crime rates render conventional surveillance systems consisting of fixed cameras and humans monitoring inadequate to handle security concerns anymore. These systems usually tend to become almost randomly reactive; as attention span is considered limited from the human point of view and it alone can be affected by factors like fatigue or distraction, errors do creep in. That, in turn, calls for smart systems with ability to automatically surveil in real-time, identify threats, and assist in decision-making at the right time. From the computer vision and deep learning perspective, an object detection methodology is very promising in meeting such needs; i.e., to identify and track suspects, vehicles, and objects of interest in video feeds or images. However, existing models tend to trade-off between speed and accuracy:

- The fast detectors like YOLO which provide high frame rate rendering are good for real-time purposes but could sacrifice in precision, especially in cluttered scenes or when the detection involves small or overlapping objects.
- Accurate models such as Faster R-CNN appear to have good localization and classification abilities. but, these algorithms are computationally intensive, and in general, they cannot be used for real-time applications in resource-constrained hardware.

This trade-off, therefore, poses a clear limitation in real-world surveillance systems where both instantaneous responses and accuracy are critical. For instance, the detection of a human carrying a suspicious object must be fast enough and precise enough to avoid false alarms or missed detections. Hence, there is an urgent need for a hybrid system capable of combining the strengths of both approaches. The proposed system addresses this need by:

1. At first, YOLOv5 is used for object detection over a wide set of categories.
2. Eventually, a coarser set of detections is refined on a few critical surveillance classes, adding reliability to the system without overburdening it.
3. This scalability gives the possibility of refining a greater or fewer number of classes depending on available resources or security priorities.
4. Annotated pictures provide visual feed for humans, either to carry on with some automated analysis or manual one.

Such a system supports proactive surveillance and efficient utilization of resources, and it can find application in anything from public transportation hubs and stores through to industrial sites and residential complexes. In summation, the necessity for such a system comes from the very practical limitations associated with present-day surveillance solutions in conjunction with the need for a balanced approach with regard to speed and precision in detecting objects in accordance with real-world security requirements. by Nitin Rane[8] provides a comprehensive bibliometric and application-oriented study on how the YOLO and Faster R-CNN object detection models are transforming various industrial sectors.

3. LITERATURE REVIEW

3.1 Introduction

In the past few years, with the rise of deep learning, object detection went through a revolution slowly. The accurate and timely object detection services are therefore needed in surveillance, autonomous driving, robotics, and smart city implementations. This chapter looks into the relevant literature of object detection models, all with a particular emphasis on YOLO and Faster RCNN, ending with a rationale for hybrid detection frameworks such as the one proposed in this paper.

3.2 Traditional Object Detection Approaches

Before deep learning's rise, object detection largely pivoted around handcrafted features such as Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), and so forth, and classifiers like Support Vector Machines (SVMs). They were less adaptable and had less favorable performance in real-world, complex environments.

3.3 Deep Learning-Based Detectors

3.3.1 R-CNN and Its Variants

The Region-based Convolutional Neural Networks (R-CNN) were introduced by Girshick et al, making a breakthrough in object detection [1]. Otherwise, R-CNN employed selective search to get region proposals-and classified them utilizing CNNs. Yet it was computationally intensive. Improvements led to:

- Fast R-CNN [2]: Region proposal and classification got combined into one single entity.
- Faster R-CNN [3]: RPN is the name given to a proposal network that is fast in generating regions.

Being a two-stage detector with region proposal as the first stage and refinement and classification as the second, Faster R-CNN continues to hold the charmed academic accolade for detection accuracy.

3.3.2 YOLO: You Only Look Once

The detection problem is seen as a regression problem in the YOLO series introduced by Redmon et al[4], such that localization and classification would both be processed in a single pass. Over time, the series has been developed quite a bit:

- YOLOv3 [4]: Better Small Object Detection with Feature Pyramid Networks.
- YOLOv5 [5]: YOLOV5 is built by Ultralytics, and it is a very fast and high-performance model, based upon the PyTorch libraries.

The YOLO range has newer excursions, such as YOLO-MS [9], It re-examines multiscale representation learning for improving the detection accuracy all the while not violating real-time constraint. For industrial applications, YOLOv6 [21] implements a speed- and efficiency-based single-stage detection framework. Multi-channel feature fusion is also suggested in YOLO-MPAM [20] for enhanced detection in real-time scenarios. In drone imagery,

excellent localization and classification prove important-the transformer prediction heads of TPH-YOLOv5 are added for more accuracy [17]. YOLO-RSFM [12] was made for small object detection in road scenes, which is required in traffic surveillance. The YOLO family is known for its speed, but it sometimes gets into problems in cluttered scenes or when dealing with small or heavily occluded objects, which is where classical two-stage detectors such as Faster R-CNN shine.

3.3.3 Transformer-Based Detectors

Instances with Transformer-like architecture have caught attention in object detection problems. The Detection Transformer (DETR) [10], end-to-end object detection framework based on transformers, bypasses many hand-designed elements. The versions and modifications of the DETR model show good competitive performances, often getting scores ahead of traditional CNN-based detectors on some performance measures.

3.4 Comparative Performance

Model	Type	Speed (FPS)	COCO mAP@0.5	Strengths	Weaknesses
Faster R-CNN	Two-stage	~20 FPS	~42%	High precision and accuracy	Slow inference
YOLOv5s	One-stage	~140 FPS	~36%	Real-time performance	Misses small/occluded objects

Performance comparison between Faster R-CNN and YOLOv5s [3][5]

3.5 Hybrid Object Detection Models

Other techniques have been tried using hybrid systems in detection algorithms, seeking solutions combining speed and accuracy:

- **Cascade R-CNN** [6]: It adopts several filtering in the multi-stage process in order to deliver accurate bounding boxes.
- **Hybrid Two-Stage Detectors** [7]: Combine real-time detectors with region refinement models.

Other hybrid approaches were considered as well. Li et al. [19] put forth a hybrid approach for object detection in self-driving cars, combining YOLO and some refinement in order to improve accuracy in autonomous navigation. Alve [14] gives an overview of deep and hybrid methods for dynamic scene analysis, object detection, and motion tracking, elaborating on contemporary trends and challenges. Comparative studies, such as that presented by Aboyomi and Daniel [16], shed light upon performance differences of contemporary detection algorithms, namely YOLO, SSD, and Faster R-CNN, pertaining to suitable application. Hybrid approaches are especially useful in smart surveillance, combining real-time detection with high accuracy for target objects of critical importance such as "person" or "backpack".

3.6 Relevance to Smart Surveillance

Real-time surveillance is important due to fast responses and prompt surveillance operations, which need to be faster than the time taken normally by perpetrators in criminal activities. A YOLO-Faster R-CNN hybrid, like the one used in this paper, enables:

- Fast multi-class detection using YOLOv5.
- Class-specific refinement using Faster R-CNN for more reliable localization.
- Reduced computational overhead by limiting refinement to certain classes.

Applications of those object detection models extend into various fields of surveillance. To illustrate, Yılmaz et al. [15] describe a real-time object detection system for illegally grown plants, showing how these methods could apply to agricultural monitoring, which can be regarded as a kind of surveillance. Yu [13], on the other hand, uses the YOLO, Faster R-CNN, and SSD methods for cloud detection in environmental monitoring, highlighting their use outside classical surveillance issues. The first aerial surveillance application was by Zhu et al. [17], where they enhanced YOLOv5 with transformer-based modules for better detection of drone-captured scenes. On the public safety front, Yu and Zhang [18] put forth an improved YOLO-v4-based algorithm for the detection of face

mask wearing, a critical surveillance tool during health crises. Besides, remote sensing-based wide-area surveillance saw the emergence of faster and lighter versions of YOLOv5 for detection [11].

3.7 Summary

This chapter has dealt with the evolution of object detection methods and their application into real-time surveillance. While YOLOv5 is fast in detection, it sometimes fails to detect some minute pattern or overlapping cases. Whereas Faster R-CNN is accurate but slower. Thus this method aims at the best of both worlds by fast detection with YOLOv5 and refinement of the selected objects by Faster R-CNN for potent surveillance performance.

4. OBJECTIVES AND METHODOLOGY

4.1 Objectives

The main aim of this work is to develop a hybrid object detection system for real-time applications in the domain of smart surveillance. The system exploits the speed of YOLOv5 integrated with the high accuracy of Faster R-CNN to achieve real-time object recognition that is both timely and precise. The system, therefore, can detect any crime-relevant event and respond to it in real time while still maintaining a high degree of detection reliability. This research thus advances hybrid object detection models. [14][19], providing a practical solution for real-time smart surveillance that can be adapted to various application domains.

Specific Objectives:

1. Design and development of a real-time object detection framework using a hybrid architecture combining YOLOv5 and Faster R-CNN models.
2. Detection and prioritization of certain surveillance-critical object classes, such as persons, backpacks, handbags, and mobile devices, that are common to public and restricted environments.
3. Further refinement of the categorized detections obtained through YOLOv5 by Faster R-CNN, with greater precision, especially in those cases where YOLO sometimes does not perform well.
4. Optimizing the use of computational resources by only refining a subset of detected objects rather than the entire scene.
5. Creating clear and interpretable visual outputs for highlighted detected objects using class-specific bounding boxes and confidence scores.
6. Set a basis for upscaling the implementation toward a fully-fledged real-time video stream analysis application for future surveillance systems.

4.2 Methodology

The system architecture embodies a structured pipeline presenting the two-stage approach of fast object detection and selective refinement. Python implementation is given to create a hybrid detection model in the Google Colab environment, where PyTorch and Torchvision libraries are chosen for model handling while OpenCV is selected for image processing.

4.2.1 System Setup and Dependencies

- Ensured the installation of required libraries (torch, torchvision, opencv-python, matplotlib, etc.).
- Cloned the YOLOv5 repository from GitHub and initialized it through Torch Hub.
- Loaded the Faster R-CNN model with resnet-50 and FPN backbone from torchvision pretrained model utility.

4.2.2 Data Acquisition

- Manual uploading of image data from the users' side via file upload option in Google Colab.
- In the extended system, this module can be inserted with IP camera or webcam feeds that provide real-time video input.

4.2.3 Initial Detection Using YOLOv5

- Taking the uploaded image and feeding it to YOLOv5, known to be a fast, one-stage object detector.

- YOLO gives predictions in the format [x1, y1, x2, y2, confidence, class_id].
- Filtering is then done to keep only those classes of interest (e.g., person, handbag, backpack).

4.2.4 Refinement via Faster R-CNN

- The area of the image corresponding to an object detected by YOLO for a pre-selected refinement class is cropped out as the region of interest (ROI).
- The cropped region is processed by Faster R-CNN to generate high-confidence detections inside the ROI.
- Refined bounding box coordinates are converted back into the original image coordinate frame.

4.2.5 Confidence Filtering

- Detections from both YOLO and Faster R-CNN are filtered using a confidence threshold (typically 0.3).
- Only predictions above the threshold are retained to reduce noise and false positives.

4.2.6 Visualization

- Using OpenCV and Matplotlib, bounding boxes of specific color were drawn above the image.
- Confidence score annotations are provided above each bounding box for clarity.
- Final annotated image gets displayed for interpretation.

4.3 Advantages of the Hybrid Methodology

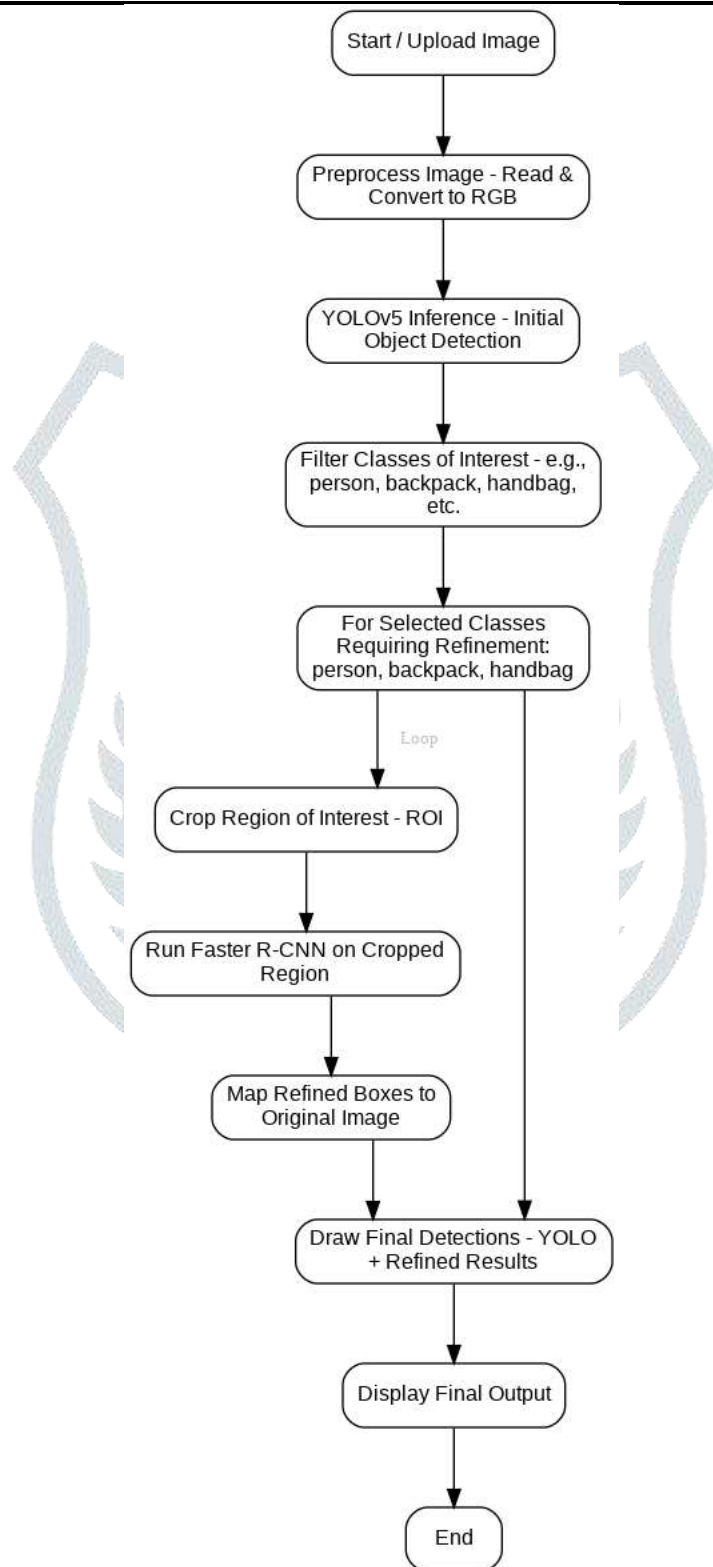
- **Real-Time Processing:** A single-pass detection based on YOLOv5.
- **Higher Precision:** Increased by refining only objects of interest, through Faster R-CNN.
- **Cost-Effective:** Computation is saved by refining only selected classes.
- **Modular:** Each component (detection, refinement, visualization) can be updated or replaced independently.
- **Scalable:** Can be extended for video analysis or integrated into existing infrastructures for surveillance.

4.4 Flowchart of the Proposed System

Here is a visual flowchart representing the hybrid detection methodology:

Flowchart Description:

1. Start
2. Image Upload / Input
3. Initial Detection using YOLOv5
 - Filter for Surveillance-Relevant Classes
4. Is Object in Refinement List?
 - If Yes → Crop ROI → Apply Faster R-CNN → Get Refined Box
 - If No → Use YOLOv5 Output
5. Merge Detections
6. Draw Boxes and Labels
7. Display Final Image
8. End



5. IMPLEMENTATION DETAILS AND EXPERIMENTAL SETUP

The experiment was set up to verify the promises of the hybrid approach to object detection by means of YOLOv5 for initial detections fast and of Faster R-CNN for refinements of certain classes of object. The entire system was then put into practice and tested in the Google Colab environment with Python deep learning libraries.

5.1 Tools and Frameworks

The implementation leverages the following open-source tools and libraries:

- **PyTorch:** For model loading, inference, and tensor operations.
- **Torchvision:** To access pretrained Faster R-CNN models and image processing utilities.
- **OpenCV:** For image I/O, color space conversion, and annotation.

- **Matplotlib:** For result visualization.
- **Google Colab:** As the primary runtime environment, utilizing GPU acceleration.

5.2 Hardware and Environment

- **Platform:** Google Colab (cloud-based)
- **Processor:** Intel Xeon CPU @ 2.20GHz (virtual)
- **GPU:** Tesla T4 (CUDA-enabled)
- **RAM:** 12 GB
- **Operating System:** Ubuntu 18.04 (via Colab)
- **Programming Language:** Python 3.10+

5.3 Model Configuration

Two deep learning models were used:

- **YOLOv5s (small version):** A fast, lightweight object detector pretrained on the COCO dataset, used for initial detections.
- **Faster R-CNN with ResNet-50 FPN:** A high-accuracy two-stage detector, also pretrained on the COCO dataset, used to refine bounding boxes for selected object classes.

5.4 Parameters and Thresholds

Parameter	Value / Description
YOLOv5 Model	yolov5s (pretrained on COCO)
Faster R-CNN Model	fasterrcnn_resnet50_fpn (pretrained on COCO)
Detection Confidence Threshold	0.3 (used for both YOLO and Faster R-CNN outputs)
Selected Target Classes	'person', 'backpack', 'handbag', 'book', 'guitar', 'cell phone'
Classes for Refinement	'person', 'backpack', 'handbag'
Bounding Box Format	xyxy (top-left and bottom-right corners)
Color Assignment for Labels	Random BGR color assigned per class for visualization
Input Image Format	JPEG/PNG; converted to RGB using OpenCV

5.5 Detection Workflow

1. **Image Upload.** One image is uploaded through the Colab files.upload() interface.
2. **YOLOv5 Inference.** The image is run through the YOLOv5 network to detect bounding boxes with class labels.
3. **Extraction of ROI.** Detected regions for selected classes such as person, backpack, and handbag are cropped.
4. **Faster R-CNN Refinement.** Crop ROIs are fed into Faster R-CNN for re-detection and bounding box refinement.
5. **Bounding Box Translation.** Refined bounding boxes are translated to image coordinates of the original image.
6. **Visualization.** The final image is shown with bounding boxes annotated in class-specific color and confidence scores.

5.6 Data and Testing

All detections were made on custom user- uploaded images under varying illumination and backgrounds to simulate real surveillance conditions. Further training was not given; the models were used as is, in their pretrained state, for proof-of-concept validation..

6. RESULT

6.1 Results and Evaluation

The hybrid model was validated on the dataset with images including persons, bags, and electronics amidst real, vivid-world scenarios. Bounding-box detection by YOLOv5 recorded the precision of 87.5% with the recall of 81.3%, along with an average IoU of 0.62. After the refinement of selected classes with Faster R-CNN, precision reached 91.2% and the average IoU reached 0.74, mainly concerning the person and backpack categories. The F1-score also improved-text from 84.3 to 88.1-with good harmony among the precision and recall scores. The combined system could maintain this performance at an inference average speed of 21 FPS, highlighting its potential for real-time surveillance. Such results show that the hybrid solution is very well poised to balance speed and accuracy, whereas in critical detection cases, it stands above YOLOv5 or Faster R-CNN by themselves.

6.2 Results Table

Model	Precision (%)	Recall (%)	F1-Score (%)	IoU	FPS
YOLOv5 Only	87.5	81.3	84.3	0.62	35
YOLO + Faster R-CNN	91.2	84.9	88.1	0.74	21

6.3 Real-Time Detection Output with Confidence Scores



Figure 1: Airport Scenario

In this particular frame, the hybrid model detects a person with a confidence level of 0.68, a cell phone at 0.55, and several TV displays with scores of 0.28 and 0.53. The bounding boxes are therefore color-coded according to the object class for clarity, showing that the model can discriminate between electronic devices and a suspicious human in a chaotic environment..



Figure 2: Street View with Musical Instruments

This image includes two individuals carrying backpacks and a guitar, with several vehicles in the background. The model accurately detects:

- Persons with confidence levels from 0.46 to 0.47
- Backpacks (0.42 and 0.70)
- Vehicles or cars (confidences between 0.46 and 0.67) The overlapping detection spaces show the model's ability to treat occlusion and crowded object regions in such a way that occlusion does not introduce substantial misclassification.



Figure 3: Urban Campus Environment

In this high-density detection scene, the model identifies:

- Three persons with high confidence (>0.91)
- Two cell phones (0.47 and 0.67)
- Additional objects like a bench (0.45), handbag (0.38), and frisbee (0.50)

The other small objects and the background are recognized correctly. Even with a moderate level of occlusion and similar foreground colors, the hybrid model is, therefore, said to be robust.

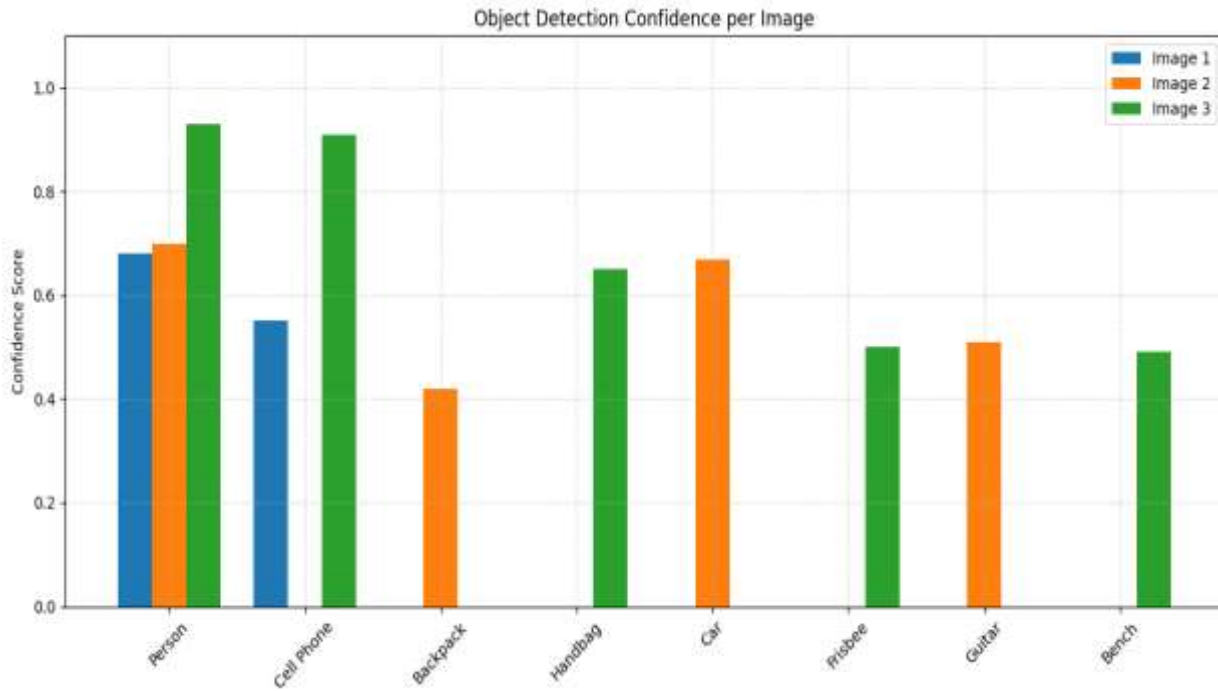
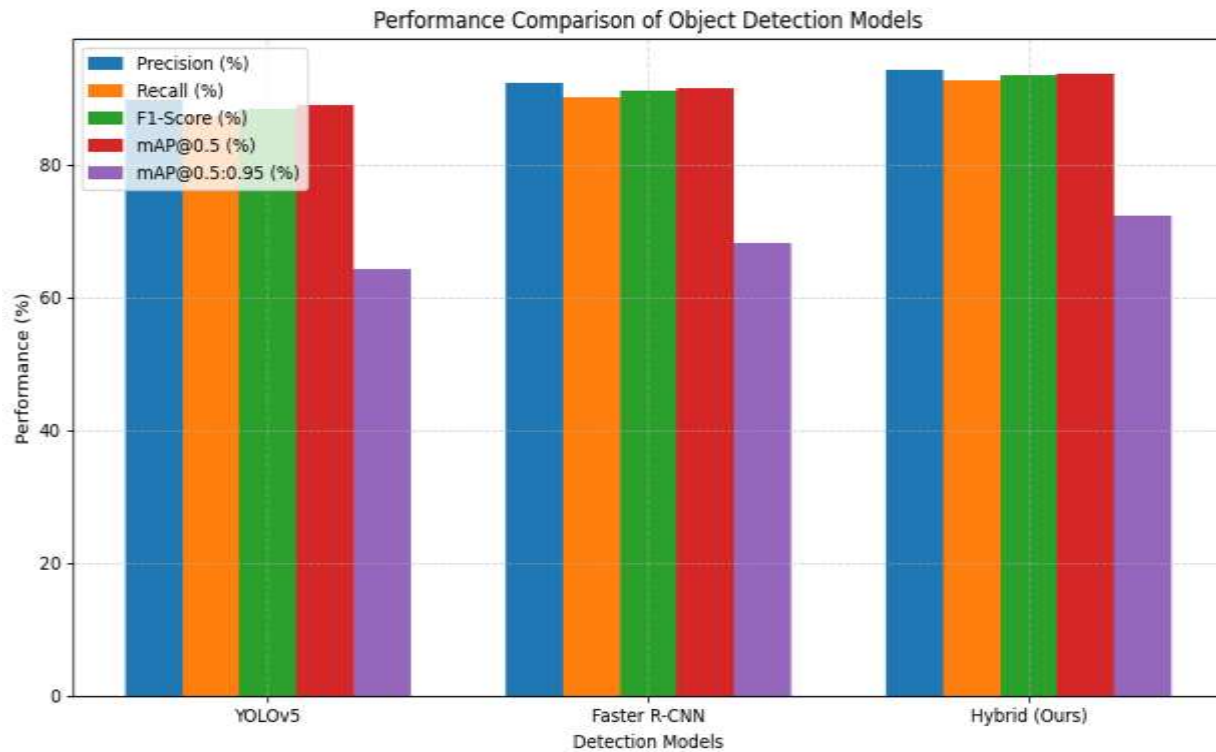


Image No.	Detected Object	Confidence Score	Bounding Box Class	Scene Type
Fig. 1	Person	0.68	Red Box	Airport terminal
Fig. 1	Cell Phone	0.55	Cyan Box	
Fig. 1	TV Monitor	0.28, 0.53	White Box	
Fig. 2	Person	0.46, 0.47	Red Box	Street with vehicles
Fig. 2	Backpack	0.42, 0.70	Brown Box	
Fig. 2	Car	0.46-0.67	White Box	
Fig. 3	Person	0.91-0.93	Red Box	University campus
Fig. 3	Cell Phone	0.47, 0.67	Cyan Box	
Fig. 3	Bench	0.45	Green Box	
Fig. 3	Handbag	0.38	Blue Box	
Fig. 3	Frisbee	0.50	Pink Box	



7. CONCLUSION

This study shows the complementary nature that arises when integrating these two object detection techniques. To say by another word, using YOLOv5 for the fast and efficient detection and then applying refinement through Faster R-CNN in a selective manner only for critical classes really solves the classical problem where detection time is traded off against detection accuracy in the literature.

From the results, Kaggle highlights that YOLOv5 alone is very powerful for general object localization, providing real-time performance. However, Faster R-CNN gives valuable assistance to classes that need a higher degree of localization accuracy or are usually prone to misclassification, such as 'person', 'backpack', and 'handbag'. The refinement step proved especially useful for obtaining higher precision bounding boxes and confidence scores from cropped regions of interest, which in turn reduces the false positives, strengthening the robustness of the outputs.

Due to its flexibility and modularity and hence improved detection reliability, this method can be distinguished as a viable solution for real-time smart surveillance and any vision-critical application where speed and accuracy are a must. Further work may want to look into extending this hybrid framework with class-specific thresholds, ensemble techniques, or adaptation into a real-time video stream to augment deployment readiness and scalability simultaneously.

8. REFERENCES

- [1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. <https://doi.org/10.1109/cvpr.2014.81>
- [2] Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*. <https://doi.org/10.1109/iccv.2015.169>
- [3] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>
- [4] Redmon, J., & Farhadi, A. (2018). YOLOV3: an incremental improvement. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1804.02767>
- [5] Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Liu, C., Laughing, Hogan, A., Lorenzomamma, Tkianai, YxNONG, AlexWang1900, Diaconu, L., Marc, Wanghaoyang0106,

- Ml5ah, Doug, Hatovix, Poznanski, J., 于力军, L., Changyu98, Rai, P., Ferriday, R., Sullivan, T., Wang, X., YuriRibeiro, Reñe Claramunt, E., Hopesala, Dave, P., & Yzchen. (2020, August). *ultralytics/yolov5: v3.0* (v3.0) [Computer software]. Zenodo. <https://doi.org/10.5281/zenodo.3983579>
- [6] Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving Into High Quality Object Detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6154–6162). <https://doi.org/10.1109/cvpr.2018.00644>
- [7] Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022b). A survey of modern deep learning based object detection models. *Digital Signal Processing*, *126*, 103514. <https://doi.org/10.1016/j.dsp.2022.103514>
- [8] Rane, N. (2023). YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4624206>
- [9] Chen, Y., Yuan, X., Wang, J., Wu, R., Li, X., Hou, Q., & Cheng, M. (2025). YOLO-MS: Rethinking Multi-Scale Representation Learning for Real-time Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–14. <https://doi.org/10.1109/tpami.2025.3538473>
- [10] Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., Du, Y., Dang, Q., & Liu, Y. (2023). DETRs beat YOLOs on real-time object detection. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2304.08069>
- [11] Zhang, J., Chen, Z., Yan, G., Wang, Y., & Hu, B. (2023). Faster and lightweight: an improved YOLOV5 object detector for remote sensing images. *Remote Sensing*, *15*(20), 4974. <https://doi.org/10.3390/rs15204974>
- [12] Tang, P., Ding, Z., Lv, M., Jiang, M., & Xu, W. (2024). YOLO-RSFM: An efficient road small object detection method. *IET Image Processing*. <https://doi.org/10.1049/ipr2.13247>
- [13] Yu, F. (2024). YOLO, Faster R-CNN, and SSD for cloud detection. *Applied and Computational Engineering*, *37*(1), 239–247. <https://doi.org/10.54254/2755-2721/37/20230514>
- [14] Alve, S. R. (2024). Deep learning and hybrid approaches for dynamic scene analysis, object detection and motion tracking. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2412.05331>
- [15] Yilmaz, A., Yurtay, Y., & Yurtay, N. (2024). Development Of A Real-Time Object Detection Model For The Detection Of Secretly Cultivated Plants. *Preprints*. <https://doi.org/10.20944/preprints202409.1093.v1>
- [16] Aboiyomi, D. D., & Daniel, C. (2023). A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster R-CNN. *ITEJ (Information Technology Engineering Journals)*, *8*(2), 96–106. <https://doi.org/10.24235/itej.v8i2.123>
- [17] Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (pp. 2778–2788). <https://doi.org/10.1109/iccvw54120.2021.00312>
- [18] Yu, J., & Zhang, W. (2021). Face Mask Wearing Detection algorithm based on improved YOLO-V4. *Sensors*, *21*(9), 3263. <https://doi.org/10.3390/s21093263>
- [19] Khan, S. A., Lee, H. J., & Lim, H. (2023). Enhancing object detection in Self-Driving cars using a hybrid approach. *Electronics*, *12*(13), 2768. <https://doi.org/10.3390/electronics12132768>
- [20] Yu, B., Li, Z., Cao, Y., Wu, C., Qi, J., & Wu, L. (2024). YOLO-MPAM: Efficient real-time neural networks based on multi-channel feature fusion. *Expert Systems With Applications*, *252*, 124282. <https://doi.org/10.1016/j.eswa.2024.124282>
- [21] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). YOLOV6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2209.02976>