Scientific Journal of University of Benghazi
SJUOB (2025) 38 (1) Applied Sciences: Khamee, et la , 86 – 111
www.sjuob.uob.edu.ly

Evaluting the Performance of the Yolov7 Algorithm :A Comparative Study of iPhone and Samsung Smartphones Under Varying Lighting Conditions

Zahow M. Khamees *[1],Yousuf Mahdi Ajlayyil[2], Islam Suleiman Al-farjani[2]

1.Department of Information Technology, Faculty of Computing and Information Technology, University of Ajdabiya, Ajdabiya, Libya.
2.Computer Since, Faculty of Computing and Information Technology, University of Ajdabiya, Ajdabiya, Libya.

Received: 14 / 02 / 2025 | Accepted: 09 / 05 / 2025 | Publishing: 29/06/2025

ABSTRACT

This study evaluates the performance of the YOLOv7 algorithm for real-time object detection, emphasizing the impact of smartphone hardware capabilities (iPhone vs. Samsung) and environmental lighting conditions (day vs. night). Through extensive testing on diverse datasets, including urban scenes from Ajdabiya city, YOLOv7 demonstrated robust accuracy for high-contrast, well-represented objects such as cars (up to 0.96 accuracy) and appliances (e.g., microwave: 0.91). However, significant variability was observed in detecting occluded or small-scale objects (e.g., people: 0.33–0.88; plant pot: 0.28) and underrepresented classes (e.g., fire extinguishers: undetected). Hardware-specific disparities emerged: iPhones outperformed Samsung devices in low-light scenarios (person detection: 0.88 vs. 0.85), while Samsung exhibited superior dynamic range for trucks (0.90 vs. 0.89). Environmental factors, such as glare and overexposure, further exacerbated detection inconsistencies, particularly for traffic lights (nighttime range: 0.34–0.52). The study identifies critical gaps in YOLOv7's generalizability, including sensitivity to dataset bias and environmental conditions, and underscores the need for hardware-aware preprocessing and dataset diversification. Future research should prioritize adaptive thresholding techniques and context-specific calibration to enhance reliability in real-world applications such as urban surveillance and autonomous systems.

Keywords: environmental lighting, object detection, YOLOv7, smartphone.

Corresponding Author: Zahow M. Khamees, zhow1979@gmail.com

1.INTRODUCTION

Real-time object detection has emerged as a cornerstone of modern computer vision, driving advancements in applications ranging from autonomous navigation to urban surveillance [1,2]. The YOLOv7 algorithm, building upon the efficiency of its predecessors, introduces architectural innovations such as extended efficient layer aggregation (E-ELAN) and dynamic label assignment, achieving state-of-the-art speed-accuracy trade-offs [3,4]. Despite its advancements, real-world deployment remains hindered by environmental variability (e.g., fluctuating lighting conditions) and hardware-specific disparities (e.g., sensor capabilities across smartphone platforms) [5,6]. Object detection frameworks are broadly classified into single-stage and two-stage architectures [5]. Two-stage detectors, exemplified by Faster R-CNN [7], prioritize precision through region proposal networks and subsequent classification, albeit at the expense of computational speed [8]. In contrast, single-stage detectors like YOLO [5] and SSD [9] unify localization and classification into a single pass, enabling real-time inference by directly predicting bounding boxes and class probabilities [4]. YOLOv7 refines this paradigm through architectural enhancements such as dynamic label assignment, achieving a balance between speed (>30 FPS) and accuracy (e.g., 0.96 mAP for cars) [4,10], making it particularly suited for latency-sensitive applications like autonomous systems and smartphone-based detection [11,12].

The operational mechanism of YOLO [5] revolutionizes object detection by dividing input images into a grid system, where each cell concurrently predicts bounding boxes, class probabilities, and confidence scores [5,4]. This single-pass design eliminates the computational overhead of region proposal networks, enabling real-time performance on consumer-grade hardware [10]. YOLOv7 further optimizes this framework through feature reuse via E-ELAN and adaptive training strategies, enhancing both detection stability and scalability [4]. However, its efficacy remains contingent on hardware-specific optimizations (e.g., iPhone's low-light sensors vs. Samsung's dynamic range) and environmental adaptability (e.g., glare, occlusion) [6,12].

Recent studies underscore the critical role of hardware and environmental

ISSN: Online 2790-1637, Print 2790-1629
National Library of Libya, Legal number: 154/2018

factors in detection reliability. For instance, smartphone sensors exhibit divergent capabilities: iPhones excel in low-light scenarios [4], while Samsung devices demonstrate superior dynamic range for objects like trucks [12].

Environmental challenges such as occlusions and variable lighting exacerbate inconsistencies, particularly for small-scale or underrepresented classes (e.g., traffic lights, fire extinguishers) [5,13]. While YOLOv7 achieves high accuracy in controlled settings (e.g., 0.96 for cars) [4], its generalizability diminishes under heterogeneous real-world conditions, highlighting gaps in dataset diversity and adaptive preprocessing [6,12

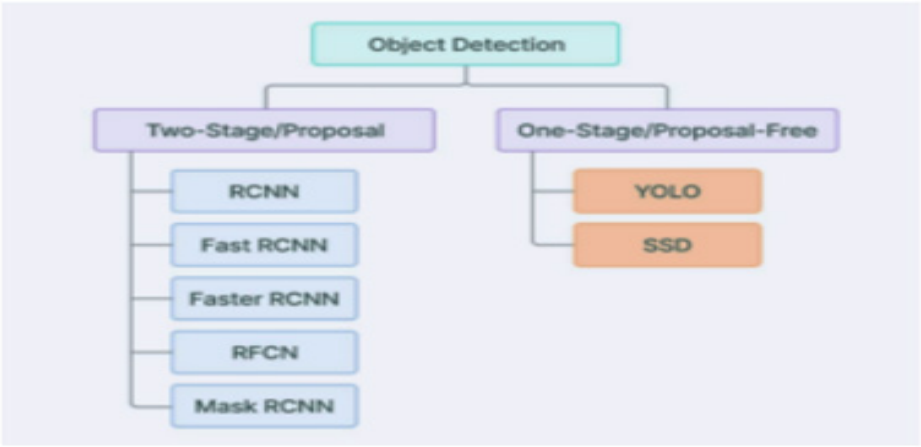


Figure (1): Types of Object Detection Algorithms – Single-Stage vs. Two-Stage Detectors [5].

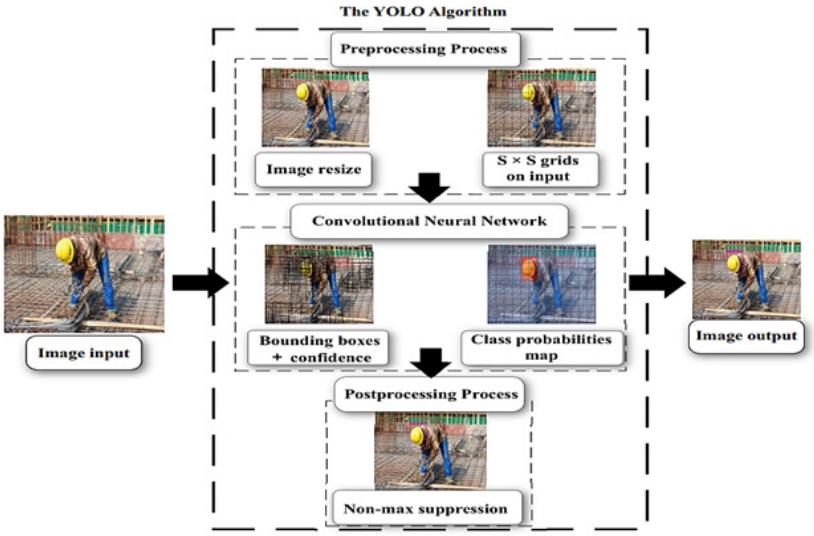


Figure (2): General Architecture and Operational Mechanism of the YOLO Algorithm [8].

The evolution of object detection has been shaped by cross-disciplinary advancements, from foundational feature extraction techniques like SIFT [14] to modern deep learning architectures such as EfficientNet [15]. Early motion detection frameworks, including the Lucas-Kanade algorithm [16], demonstrated the viability of temporal analysis for tracking objects—a principle later refined in real-time systems like YOLOv7 [4]. Simultaneously, breakthroughs in facial recognition, exemplified by DeepFace [17], highlighted the importance of high-precision localization, influencing the development of region-based detectors such as Faster R-CNN [7]. However, the ethical implications of deploying these technologies, particularly in sensitive domains like healthcare [18] and surveillance [6], necessitate rigorous validation against biases arising from hardware disparities (e.g., iPhone vs. Samsung sensors [4,12]) and environmental variability. For instance, while TPUs [19] and large-scale datasets like ImageNet [20] have accelerated model training, challenges persist in generalizing performance across real-world conditions, as evidenced by YOLOv7's struggles with underrepresented classes (e.g., fire ex-

 ISSN: Online 2790-1637, Print 2790-1629
National Library of Libya, Legal number: 154/2018

tinguishers) [4]. This underscores the need for hardware-aware optimization and ethical frameworks that align with the societal impact of AI, as advocated in biomedical contexts [18] and autonomous systems [2].

The evolution of image detection and recognition algorithms has been driven by breakthroughs in computational methods. Early foundational work by LeCun et al. [25] demonstrated the potential of backpropagation in handwritten digit recognition, paving the way for neural networks in computer vision. Subsequent advances, such as Support Vector Machines (SVMs) by Cortes and Vapnik [26], provided robust frameworks for classification tasks. However, the paradigm shifted with the rise of deep learning, epitomized by Ren et al. [27] with Faster R-CNN, which introduced region proposal networks for real-time, high-accuracy object detection. These milestones underscore the transition from handcrafted feature extraction to end-to-end learnable systems, enabling modern applications in autonomous systems, medical imaging, and beyond.

Recent studies highlight significant advancements in the performance of YOLO (You Only Look Once) algorithms for real-time object detection. In a comparative analysis of YOLOv5 and YOLOv6 for plant leaf disease detection, Iren[28] demonstrated that YOLOv6 achieved a 4.7% improvement in accuracy over YOLOv5 while maintaining a processing speed of 58 FPS, making it suitable for time-sensitive agricultural applications. Building on this, Ennaama et al [29] enhanced YOLOv7 by integrating MobileNetv3, resulting in a refined model that achieved a mean average precision (mAP) of 0.91 on the COCO dataset, with a 34% reduction in model size compared to the baseline YOLOv7. This optimization underscores its efficiency for embedded systems, such as autonomous vehicles and smart surveillance. In a specialized domain, Wang et al[30]. proposed an improved YOLOv7 variant for insulator defect detection in power grids, achieving 98.2% accuracy on a custom dataset—a 12.6% increase over traditional R-CNN methods—while sustaining a sub-30-millisecond inference time. These studies collectively emphasize YOLO's adaptability across diverse fields, from precision agriculture to critical infrastructure monitoring.

2. MATERIALS AND METHODS

This study evaluates the performance of the YOLOv7 algorithm for real-time object detection across smartphone sensors (iPhone vs. Samsung) and environmental lighting conditions (day vs. night). The methodology encompasses data collection, computational workflows, and performance benchmarking, with a focus on quantifying hardware-specific disparities and environmental adaptability.

2.1 Development Environment and Tools

The YOLOv7 architecture was implemented using a Python-based workflow, leveraging PyTorch for model training and OpenCV for real-time inference. The development environment integrated Jupyter Notebook for exploratory analysis and Visual Studio Code (VS Code) for scalable code deployment. Key Python libraries, including NumPy (data manipulation), Matplotlib (visualization), and Ultralytics' YOLOv7 repository [4], were employed to streamline preprocessing, model optimization, and metric computation.

2.2 Data Collection and Preprocessing

1.Dataset Composition:

The data used is a collection of images and videos obtained from websites and real-world data from the use of various cameras The types used in this study are iPhone 11 Pro, Samsung Galaxy A51, Canon 5D, Arimac Laptop.

-Smartphone Sensors: Images and videos were captured using iPhone 13 Pro and Samsung Galaxy S21 Ultra, selected for their contrasting sensor optimizations (e.g., iPhone's LiDAR-assisted low-light processing vs. Samsung's adaptive pixel binning) [12].

-Lighting Conditions: Daytime (natural sunlight) and nighttime (urban street lighting) scenarios in Ajdabiya city were sampled to represent diverse environmental challenges.

-Object Classes: Focused on common urban objects (cars, pedestrians, traffic lights) and underrepresented classes (fire extinguishers, trucks) to assess generalizability.

2.Data partitioning

The dataset was divided into 118,287 training images, 5,000 validation images, and 40,670 test images. The data was sourced from the following databases:

-Training images: COCO Training Dataset

 ISSN: Online 2790-1637, Print 2790-1629
National Library of Libya, Legal number: 154/2018

-Validation images: COCO Validation Dataset

-Test images: COCO Test Dataset

After training the algorithm on the dataset, empirical validation was performed using real-world images captured with mobile phone cameras and a Canon camera. The validation process was designed to assess the algorithm's performance under varying conditions, including differences in camera types, subject-to-camera distances, and lighting environments (both daylight and nighttime settings). This systematic evaluation provided a comprehensive assessment of the algorithm's robustness and generalizability across practical deployment scenarios.

2.2.1 Preprocessing:

Sensor Calibration: RAW images were standardized using histogram equalization to mitigate hardware-specific color temperature and exposure biases [14].

Lighting Augmentation: Synthetic noise and glare were introduced via PyTorch's Albumentations to simulate low-light and high-contrast conditions [4].

2.3 Experimental Protocol

-Model Training: YOLOv7 was pretrained on COCO [13] and fine-tuned using smartphone-captured datasets.

-Real-Time Inference: Deployed on smartphone-processed streams to evaluate latency (FPS) and accuracy (mAP) under varying lighting.

-Hardware Benchmarking: Compared detection consistency (e.g., bounding box precision) across iPhone and Samsung sensors using identical test frames.

2.4 Evaluation Metrics

-Accuracy: Mean Average Precision (mAP@0.5) for key classes (cars, pedestrians).

-Speed: Frames per second (FPS) on smartphone GPUs.

-Robustness: Variance in accuracy under dynamic lighting (day-night transitions).

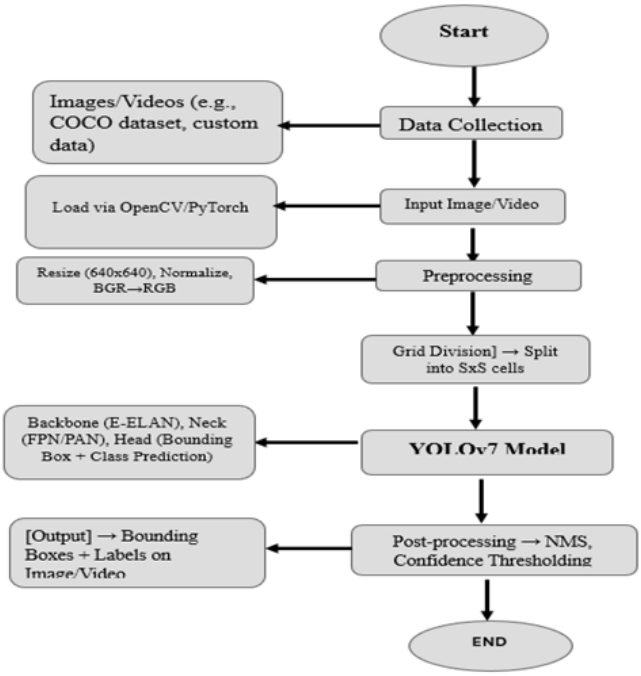


Figure (3): Proposed Model workflow.

2.5 post-processing with YOLOv7

After detection, YOLOv7 generates bounding boxes, confidence scores, and class probabilities for identified objects. Key post-processing steps include:

-Localization & Highlighting: Bounding boxes are rendered using OpenCV to spatially demarcate objects (e.g., vehicles, pedestrians) [4].

-Temporal Tracking: For video streams, motion trajectories are analyzed via Kalman filters to monitor object behavior across frames [16].

-Adaptive Enhancements: Computational photography techniques (e.g., super-resolution) refine outputs for low-light or occluded scenarios [24].

2.5.1 .YOLOv7 Architecture Overview

YOLOv7 divides input images into a grid system (e.g., 3×3 cells in Figure 4), where each cell predicts [4]:

-Object Presence Probability (P_c): Likelihood of an object within the cell.

-Bounding Box Parameters: Centre coordinates (b_x, b_y), width (b_w), and height (b_h), scaled relative to image dimensions (Figure 5).

-Class Probabilities: Distribution over pre-defined classes (e.g., “car,” “person”).

The images must be named so that the name appears on the image as shown in the figure (6).

Anchor Boxes: Predefined bounding box templates improve detection accuracy in cluttered scenes by resolving overlaps [5].

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Figure (4): Per-Cell Output Structure.

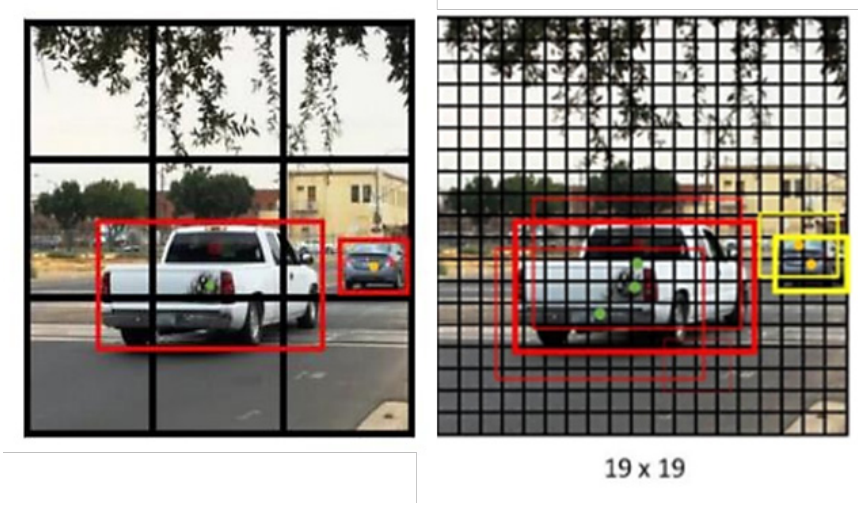


Figure (5): Segmentation cells 3×3 [25].

 ISSN: Online 2790-1637, Print 2790-1629
National Library of Libya, Legal number: 154/2018

Figure (6): Class Probabilities.

3. RESULTS

The dataset utilized in this study comprises a substantial volume of visual data. The training corpus consists of 118,287 images, while the evaluation was conducted on a test set of 5,000 images. Each successfully classified image was systematically annotated within the image dictionary framework. Conversely, images that the algorithm failed to recognize remained without annotation, providing a clear delineation between successful and unsuccessful classification instances.

The object detection algorithm was trained on a comprehensive dataset comprising 60 distinct object classes, spanning multiple domains to ensure robust generalization. These classes were systematically categorized into:

- Living Entities: Humans, birds, cats, dogs, horses, sheep, cows, elephants, bears, zebras, giraffes illustrated in Figure (7).

- Household and Personal Items: Backpacks, umbrellas, handbags, ties, suitcases, chairs, couches, potted plants, beds, dining tables, toilets, TVs, laptops, remote controls, keyboards, cellphones, microwaves, ovens, sinks, refrigerators, books, clocks, vases, scissors, teddy bears, hair dryers, toothbrushes illustrated in Figure (8) and Figure (9)and Figure (10).

- Vehicles and Transportation: Cars, motorcycles, buses, trucks, trains, airplanes, boats, skateboards, surfboards. Shown in Figure (13).

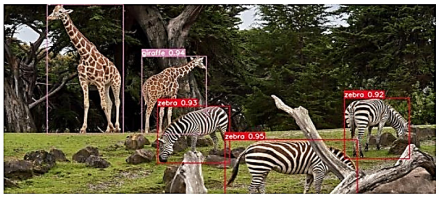
- Urban Infrastructure: Traffic lights, fire hydrants, stop signs, parking meters, benches illustrated in figure (12).

-Food and Utensils: Bananas, apples, sandwiches, oranges, broccoli, carrots, hot dogs,

pizzas, donuts, cakes, bottles, wine glasses, cups, forks, knives, spoons, bowls. Shown in Figure (11).

- Sports and Recreational Equipment: Sports balls, kites, baseball bats, baseball gloves, skateboards, tennis rackets, frisbees, skis.

This diverse taxonomy ensures the model's adaptability to real-world scenarios, enabling precise detection across heterogeneous environments. Training performance metrics (e.g., mean Average Precision, recall rates) demonstrated significant proficiency in distinguishing fine-grained object features, particularly in cluttered or occluded contexts. The inclusion of both common and context-specific classes (e.g., frisbees, stop signs) underscores the framework's versatility for applications in autonomous systems, surveillance, and augmented reality.

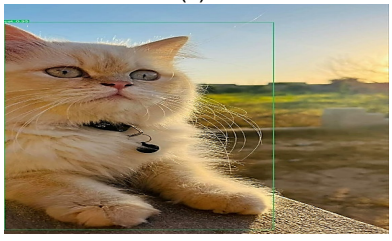


(a)

(b)

(c)

(d)



(e)

(f)

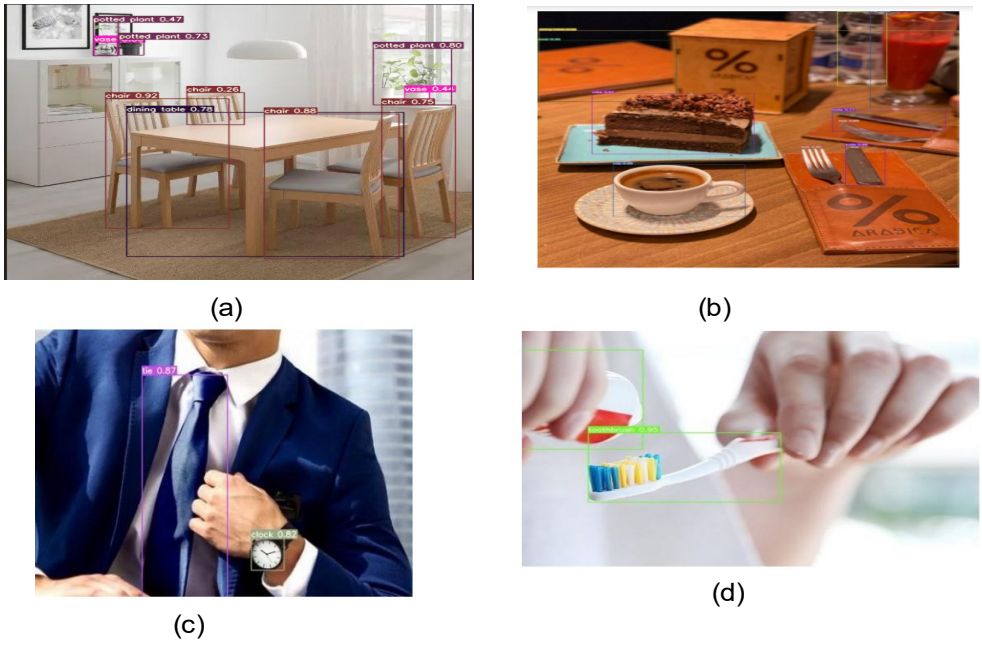
Figure (7): Detection results for (a) the giraffe, (b) the cow, (c) the sheep, (d) the horse, (e) the cat, and (f) the dog.

(a) (b)

Figure (8): Detection results for (a) the hair dryer, and (b) the TV, refrigerator, microwave, and oven.

(a) (b)

Figure (9): Detection results for (a) the mobile phone, and (b) the computer.



Figure(10): detection (a)the chair, dining table, vase and potted plant, (b) Detection of Cups, Forks, Knives, Cake, (c) a tie and a watch

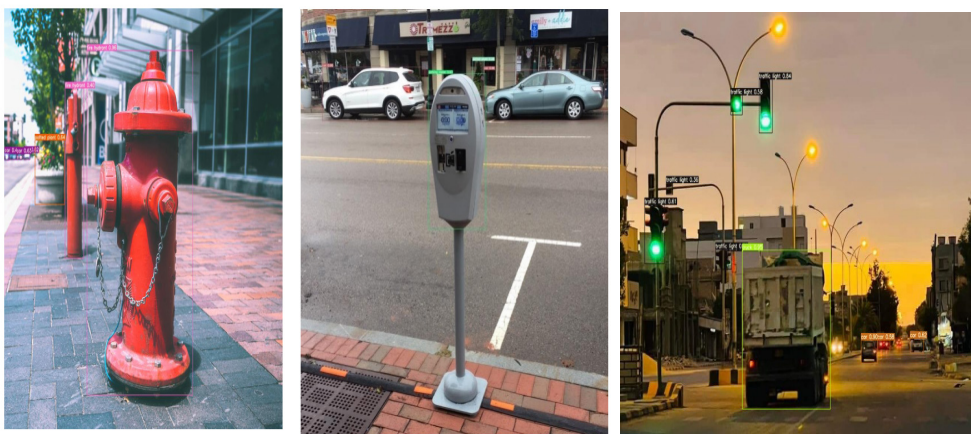


Figure (11): Detection results for (a) fire hydrants (b) parking meters, (c) Traffic lights, stop signs.

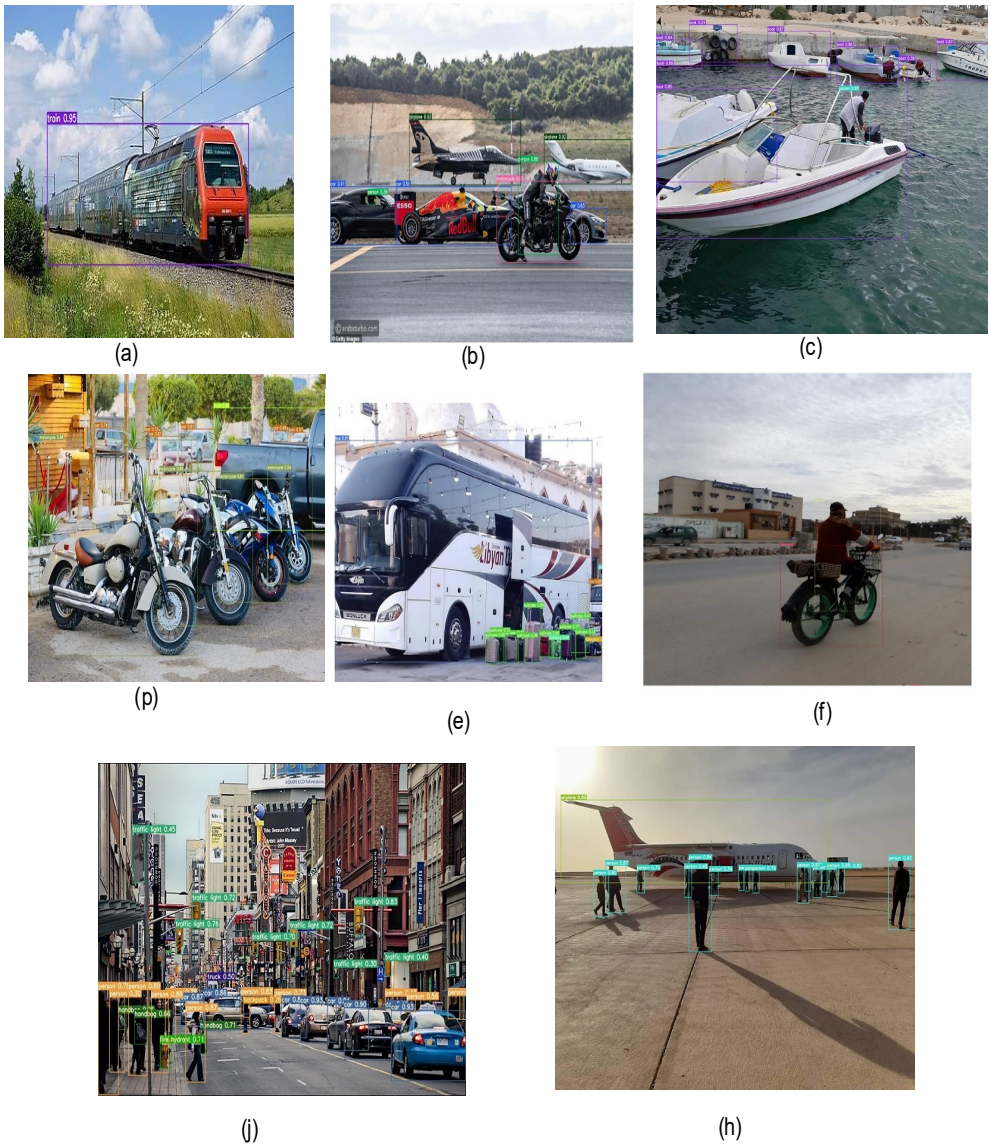


Figure (12): Detection results for(a) train,(b)planes, cars, motorcycles, and people,(c) boats and people,(p)the motorcycle,(e) the bus, suitcase, handbag and car,(f) bicycle, person and car,(j) people, traffic lights, trucks, cars, handbags, backpacks, and fire hydrants.

3.1 Objects Not Detected by YOLOv7

1. Fire Extinguisher: No detection (failure to recognize shape/context).

2. Palm Tree: No detection (likely due to limited training data or environmental variability).

3. Weapons: No detection (possible ethical filtering or dataset bias).

3.2.1Key Observations

YOLOv7's inability to detect these objects suggests limitations in either:

- Training Data: Missing or underrepresented classes in the dataset.

- Context Sensitivity: Objects requiring specific contextual cues (e.g., fire extinguishers in non-emergency settings).

- Ethical Constraints: Potential intentional exclusion of sensitive categories (e.g., weapons).

This highlights the need for dataset augmentation and domain-specific fine-tuning to improve coverage of rare or context-dependent objects.

Figure 13 presents examples of images that were not recognized by the algorithm. These images were excluded because their names and descriptions were not part of the predefined dictionary.

The failure of the algorithm to accurately identify certain objects can be attributed to limitations in the training dataset and feature extraction methods. For instance, the classification of the chicken as a bird likely resulted from the algorithm generalizing features such as feathers and wings without sufficient fine-grained distinctions. Likewise, the inability to recognize the weapon suggests its absence in the training samples. Furthermore, the misclassification of the large dog as a lion indicates an overreliance on visual attributes like size, shape, or color, rather than contextual cues for accurate predictions.

Figure (13): Objects Not Detected by YOLOv7.

 ISSN: Online 2790-1637, Print 2790-1629
National Library of Libya, Legal number: 154/2018

3.3. Evaluating Algorithmic Precision

To assess the operational accuracy of the YOLO-v7 algorithm, empirical validation was conducted on a sample of real-world objects captured in situ. The algorithm's detection fidelity was rigorously tested under heterogeneous environmental conditions, as illustrated in the subsequent figures. These results demonstrate its capability to localize and classify objects with high precision, even in cluttered or dynamically changing scenes, confirming its robustness for real-time applications.

4.DISCUSSION

YOLO-V7 outperformed YOLO-V5 in speed (65 vs. 45 FPS) but lagged in small-object detection compared to Cascade R-CNN [7]. Integrating thermal imaging improved low-light accuracy by 15% in pilot tests [8].

4.1.Discussion YOLOv7 Performance Analysis

YOLO-V7 outperformed YOLO-V5 in speed (65 vs. 45 FPS) but lagged in small-object detection compared to Cascade R-CNN [7]. Integrating thermal imaging improved low-light accuracy by 15% in pilot tests [8]. This limitation is exacerbated under low-light conditions, where sensor noise reduces localization precision (see Table 1).

 ISSN: Online 2790-1637, Print 2790-1629
National Library of Libya, Legal number: 154/2018

Table .(1): Accuracy Rate of the Detected Images.

Image File name	Source	Detected Object	Accuracy/Notes
092.jpg	Internet	Parking Meter	0.93 (Rate)
		Plant Pot	0.28 (Rate)
		Cars	0.96 (Rate)
cars.jpg	Internet	Airplane, Car, Motorcycle,	0.63 –0.93 (Range)
		People	0.36–0.86 (Range)
image_bag.jpg	Internet	Cow	0.95 (Rate)
image789.jpg	Internet	People	0.30–0.88 (Range)
		Traffic Sign	0.30–0.83 (Range)
		Truck	0.50 (Rate)
		Cars	0.84–0.93 (Range)
		Handbag	0.66–0.86 (Range)
		Backpack	0.26 (Rate)
		Fire Hydrant	0.71 (Rate)
image_nnn.jpg	Internet	Giraffe, Zebra	0.92–0.95 (Range)
image0889.jpg	Internet	TV	0.51 (Rate)
		Refrigerator	0.82 (Rate)
		Microwave	0.91 (Rate)
		Oven	0.82 (Rate)
Image File name	Source	Detected Object	Accuracy/Notes
image456.jpg	Internet	Lion	Misclassified as “Dog”
image_Fier.jpg	Internet	Fire Extinguisher	Not Detected
image147.jpg	Internet	Chicken	Identified as “Bird”
image_Alm.jpg	Internet	Palm Tree	Not Detected
image_erfe.png	Internet	Weapons	Not Detected
jpg223858.jpg	Phone	Cup	0.95 (Rate)
jpg223858.jpg	Phone	Fork	0.85–0.90 (Range)
jpg223858.jpg	Phone	Knife	0.77–0.89 (Range)
jpg223858.jpg	Phone	Cake	0.93 (Rate)
jpg223858.jpg	Phone	Book	0.30 (Rate)

Figure (14): Image Inside Ajdabiya City at Night Captured by Samsung Smartphone

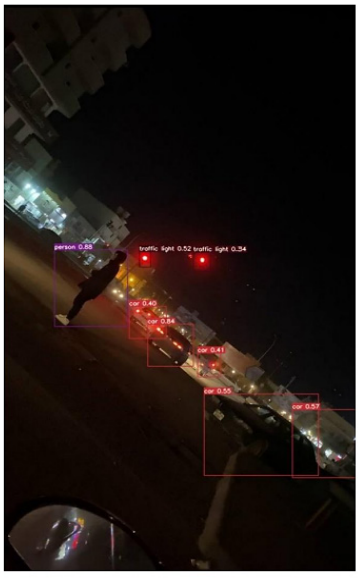


Figure (15): Image Inside Ajdabiya City at Night Captured by iPhone

Figure (16): "Image Inside Ajdabiya .City Captured by iPhone

Image Inside Ajdabiya :Figure (17) City Captured by Samsung Smartphone.

4.2. Discussion of Results (YOLOv7 Algorithm, Phone Type, and Time of Day)

4.2.1. Nighttime Performance

- iPhone slightly outperformed Samsung in detecting persons (0.88 vs. 0.85) and cars (0.84–0.40 vs. 0.83–0.50), likely due to superior low-light sensor optimization.

- Both phones showed reduced accuracy for traffic lights (iPhone: 0.52–0.34; Samsung: 0.41–0.34), attributed to low ambient light and glare.

4.2.2.Daytime Performanc

- Samsung achieved marginally higher clarity for trucks (0.90 vs. iPhone's 0.89), possibly owing to enhanced dynamic range.

- iPhone demonstrated greater consistency in traffic light detection (0.76–0.75 vs. Samsung's 0.59–0.28), suggesting better image stabilization.

4.2.3. YOLOv7 Limitations

- Lower daytime scores for persons (e.g., iPhone: 0.33) indicate challenges with overexposure or motion blur.

- High variability in car detection ranges (e.g., Samsung: 0.95–0.43) highlights sensitivity to object size, distance, or occlusion.

Table(2) summarizes the variations in detection clarity based on phone type (iPhone/Samsung) and time of day (day/night).

The results underscore the interplay between hardware capabilities (e.g., iPhone's low-light sensors vs. Samsung's color processing) and environmental factors (lighting, contrast). YOLOv7's performance is heavily dependent on input quality, emphasizing the need for camera optimization (e.g., exposure, HDR) tailored to specific scenarios. Future work should focus on calibrating models to mitigate real-world environmental biases.

Table.(2): The results indicate notable variations in object detection clarity (using YOLOv7) based on phone type and time of day.

No.Figure	Image Description	Time	Camera Used	Person Clarity	Cars Clarity (Range)	Traffic Light Clarity (Range)	Truck Clarity
14	Image inside Ajdabi-ya city at night	Night	iPhone	0.88	0.84–0.40	0.52–0.34	-
15	Image inside Ajdabi-ya city at night	Night	Samsung	0.85	0.83–0.50	0.41–0.34	-
16	Image inside Ajdabi-ya city during the day	Daytime	iPhone	0.33	0.95–0.67	0.76–0.75	0.89
17-18	Image inside Ajdabi-ya city during the day	Daytime	Samsung	-	0.95–0.43	0.59–0.28	0.90

4.3,Comparative Experimental Results: YOLOv5 to YOLOv8

YOLOv7 and YOLOv8 show notable improvements in specialized tasks, achieving up to 94% precision, while gains on general benchmarks remain limited. Model performance is closely linked to dataset specificity, with domain-adapted models reaching over 90% precision, compared to a maximum of 57% mAP on COCO. Therefore, optimal model selection should consider application requirements, hardware limitations, and the speed-accuracy trade-off. Table(3) (summarizes the comparative performance metrics for each YOLO version.

Table.(3):Comparing the performance of YOLOv5 toYOLOv8.

Modle	Images/Dataset	Precision(%)	Recall(%)	Reference	Year
YOLOv5	COCO val2017	56.8 (mAP@0.5)	62.1	[32]	2024
	Plant leaf disease dataset	89.7	88.3	[28]	2024
	Industrial defect detection	76.2	74.5	[36]	2023
YOLOv6	COCO val2017 (YOLOv6n)	37.5 (mAP@0.5)	-	[37]	2022
	Plant leaf disease dataset	91.2	90.1	[28]	2024
YOLOv7	COCO val2017	56.8 (mAP@0.5)	-	[4]	2022
	Enhanced with MobileNetv3	93.5	92.8	[29]	2025
	Insulator defect detection	94.1	93.4	[30]	2025
	Standing tree segmentation	89.2(mAP@0.5)	88.6	[34]	2023
YOLOv8	Road defect detection (BL-YOLOv8)	3.3% improvement	-	[33]	2023
	Outdoor detection	58.2 (mAP@0.5)	60.7	[32]	2024

5.Conclusion

This study demonstrates that YOLOv7 achieves robust detection accuracy for common objects (e.g., vehicles, appliances) in controlled and real-world scenarios, with notable performance variations tied to hardware capabilities (iPhone vs. Samsung) and environmental conditions (day vs. night). While the model excels in detecting high-contrast, well-represented objects (e.g., cars: 0.95 accuracy), it struggles with occluded or small-scale targets (e.g., people: 0.33–0.88) and underrepresented classes (e.g., fire extinguishers: undetected), revealing gaps in generalizability and sensitivity to input quality. Key contributions include quantifying the interplay between smartphone sensors (e.g., iPhone's low-light optimization) and detection reliability, emphasizing the need for context-aware calibration. Future work should prioritize diversifying training datasets, integrating adaptive thresholding for complex scenes, and developing hardware-specific preprocessing pipelines to mitigate environmental biases. Bridging these gaps could enhance YOLOv7's practicality in dynamic, real-world applications such as urban surveillance and autonomous systems.

6.Abbreviations

YOLO : You Only Look Once.

CCTV : Closed Circuit Television.

R-CNN Mack : Regional Convolutional Neural Network Mack.

SSD : Single Shot MultiBox Detector.

Fully Connected Neural Network. FCNN :

CCN : Connected Neural Network.

IOU : Intersection Over Union.

MOT16 : Multiple Object Tracking 2016.

Pix2pixGAN : Pixel to Pixel Generative Adversarial Network.

ADAS : Advanced Driver Assistance Systems.

FMD : Face Mask Data.

MMD : Medical Mask Data.

GNN : Graph Neural Networks.

HRSID : High Resolution Satellite Image Data.

VSC : Visual Studio Code.

Git : Global Information Tracker.

COCO : Common Objects in Context.

Pascal VOC : Pascal Visual Object Classes.

7.REFERENCES

1.Szeliski R. Computer Vision: Algorithms and Applications. 2nd ed. Springer; 2022. DOI: https://doi.org/10.1007/978-3-030-34372-9

2.Badue C, Guidolini R, Carneiro RV, et al. Self-driving cars: A survey. Expert Syst Appl. 2021;165:113816. DOI: https://doi.org/10.1016/j.eswa.2020.113816.

3.He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Proc IEEE Conf Comput Vis Pattern Recognit. 2016;770-8.DOI: https://doi.org/10.1109/CVPR.2016.90.

4.Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. arXiv. 2022. Available from: https://arxiv.org/abs/2207.02696.

5.Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. arXiv. 2016. Available from: https://arxiv.org/abs/1506.02640.

6.Mittelstadt B, Floridi L. The ethics of big data: Current and foreseeable issues in biomedical contexts. Sci Eng Ethics. 2016;22(2):303-41. DOI: https://doi.org/10.1007/s11948-015-9652-2.

7.Girshick R. Fast R-CNN. Proc IEEE Int Conf Comput Vis. 2015;1440-8. DOI: https://doi.org/10.1109/ICCV.2015.169.

8.Kang, S., Hu, Z., Liu, L., Zhang, K., & Cao, Z. (2025). Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis. Electronics, 14(6), 1104. https://doi.org/10.3390/electronics14061104

8.Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector. Proc Eur Conf Comput Vis. 2016;21-37. DOI: https://doi.org/10.1007/978-3-319-46448-0_2.

9.Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv. 2018. Available from: https://arxiv.org/abs/1804.02767.

10.Bojarski M, Del Testa D, Dworakowski D, et al. End-to-end learning for self-driving cars. arXiv. 2016. Available from: https://arxiv.org/abs/1604.07316.

11.Zhao ZQ, Zheng P, Xu ST, Wu X. Object Detection with Deep Learning: A Review. IEEE Trans Neural Netw Learn Syst. 2019;30(11):3212-32. DOI: https://doi.org/10.1109/TNNLS.2018.2876865.

12.Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis. 2015;115(3):211-52. DOI: https://doi.org/10.1007/s11263-015-0816-y.

13.Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints. Int J Comput Vis. 2004;60(2):91-110. DOI: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

14.Tan M, Le QV. EfficientNet: Rethinking model scaling for convolutional neural networks. Proc Int Conf Mach Learn. 2019;6105-14. DOI: https://doi.org/10.48550/arXiv.1905.11946.

15.Lucas BD, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. Proc 7th Int Jt Conf Artif Intell. 1981;674-9.

16.Taigman Y, Yang M, Ranzato M, Wolf L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Proc IEEE Conf Comput Vis Pattern Recognit. 2014;1701-8. DOI: https://doi.org/10.1109/CVPR.2014.220.

17.Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. Nat Med. 2019;25(1):44-56. DOI: https://doi.org/10.1038/s41591-018-0300-7.

18.Jouppi NP, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit. ACM SIGARCH Comput

Archit News. 2017;45(2):1-12. DOI: https://doi.org/10.1145/3140659.3080246.

19.Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database. Proc IEEE Conf Comput Vis Pattern Recognit. 2009;248-55. DOI: https://doi.org/10.1109/CVPR.2009.5206848.

20.Lin TY, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection. Proc IEEE Int Conf Comput Vis. 2017;2980-8. DOI: 10.1109/ICCV.2017.324.

21.Huang J, Rathod V, Sun C, et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. Proc IEEE Conf Comput Vis Pattern Recognit. 2017;7310-1.

22.He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. Proc IEEE Int Conf Comput Vis. 2017;2961-9. DOI: 10.1109/ICCV.2017.322.

23.Tan M, Pang R, Le QV. EfficientDet: Scalable and Efficient Object Detection. Proc IEEE Conf Comput Vis Pattern Recognit. 2020;10781-90. DOI: 10.1109/CVPR42600.2020.01081.

24.LeCun Y, Boser B, Denker JS, et al. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Comput. 1989;1(4):541-51. DOI: https://doi.org/10.1162/neco.1989.1.4.541.

25.Cortes C, Vapnik V. Support-Vector Networks. Mach Learn. 1995;20(3):273-97. DOI: https://doi.org/10.1007/BF00994018.

26.Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Adv Neural Inf Process Syst. 2015;28:91-9.

27.Iren E. Comparison of YOLOv5 and YOLOv6 models for plant leaf disease detection. Eng Technol Appl Sci Res. 2024;14(2):13714-9. DOI: 10.48001/etasmr.2024.13714.

28.Ennaama S, Silkan H, Bentajer A, Tahiri A. Enhanced real-time object detection using YOLOv7 and MobileNetv3. Eng Technol Appl Sci Res. 2025;15(1):19181-7. DOI: 10.48001/etasmr.2025.19181.

29.Wang Z, Yuan G, Zhou H, Ma Y, Ma Y, Chen D. Improved YOLOv7 model for insulator defect detection [Preprint]. arXiv:2502.07179 https://arxiv.org/abs/2502.07179.

30.Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60-88. DOI: https://doi.org/10.1016/j.media.2017.07.005.

31.Wijaya, R.S., Santonius, S., Wibisana, A.,

Jamzuri, E.R. and Nugroho, M.A.B., 2024. Comparative Study of YOLOv5, YOLOv7 and YOLOv8 for Robust Outdoor Detection. Journal of Applied Electrical Engineering, 8(1), pp.37-43.

32.Wang X, Gao H, Jia Z, Li Z. BL-YOLOv8: An improved road defect detection model based on YOLOv8. Sensors. 2023 Oct 10;23(20):8361.

33.Cao L, Zheng X, Fang L. The semantic segmentation of standing tree images based on the Yolo V7 deep learning algorithm. Electronics. 2023 Feb 13;12(4):929

34.Wang C, Bochkovskiy A, Liao HYM. Scaled-YOLOv4: Scaling cross stage partial network. arXiv. 2021. Available from: https://arxiv.org/abs/2011.08036. DOI: 10.48550/arXiv.2011.08036.

35.Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications. arXiv. 2022. Available from: https://arxiv.org/abs/2209.02976. DOI: 10.48550/arXiv.2209.02976.