AI DRIVEN AUTONOMOUS NAVIGATION USING OBJECT DETECTION ON KLH CAMPUS DATASET

*A Project Based Learning Report Submitted in partial fulfilment of the requirements for the award of the degree*

*of*

**Bachelor of Technology**

**in The Department of ECE**

**TERM PAPER 221E3043**

Submitted by

**2210040002 - YADNYA VIDHATE**

**2210040019- SURASANI ABHINAV REDDY**

**2210040047- K R LOKESH KUMAR**

Under the guidance of

**Dr K SAI KUMAR**



Department of Electronics and Communication Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075 (Optional)

FEB - 2025.

**Abstract**

The field of artificial intelligence encompasses object detection but this area remains specialized even though it shows progress in both research and literature through deep learning techniques and real world AI applications. This project shows an approach toward implementing an AI-based autonomous navigation system that employs object detection technology on KLH Campus data. The KLH Campus dataset resulted from clicking more than 1000 images as the system captured these pictures from various university viewpoints before transferring them to a software for automated annotation. The development used an updated YOLOv8 architecture running on an Ubuntu environment during the programming stage. During optimization we prioritized two YOLO architecture modifications to improve both processing speed and accuracy by fewer layers while implementing the SPPF (Spatial Pyramid Pooling-Fast) module. The YOLO programming task used Python for achieving all project targets while supporting the modeling process and literature references from the previous year. The experimental results from KLH Campus showed 74.5% accuracy together with 53.9% mAP 50 and 53.9% overall mAP 50-95 and precision of 52.6%.

The study proves how artificial intelligence enhancement with autonomous direction features can be adopted for actual world applications. Improved modeling of robotic surveillance applications in smart campus operations became possible through the better accuracy achieved by this multimodal detection method. YOLOv8 achieves better outcomes periodically while using the KLH Campus dataset.

Keywords — The combination of AI-Driven Navigation, Object Detection and YOLOv8, KLH Campus Dataset, Autonomous Systems, Real-Time Processing, Robotics constitutes the essential keywords in this Keywords—Autonomous Navigation, Object Detection, YOLO, AI, Robot, Real-Time Processing, University

**List of Figures**

**Fig. 1: Architecture**

**Fig. 2: Object detection-1**

**Fig. 3: Object detection-2**

**Fig.4: Object detection-3**

**Fig.5: Object detection-4**

**Fig.6: Object detection-5**

**Fig:7: Object detection-6**

**Fig.7 : Performance of YOLO V8**

**Fig. 8: Performance Metrics of YOLO V8**

**Fig. 9: Histogram of Object counts**

**List of Tables**

**Table 1: Comparison table for YOLOV8 Version**

**Table 2: Parameter settings for the backbone of YOLO**

**Table 3: Performance Metrics**

**Table 4: Performance with other versions**

**Table 5: P1 All the data set**

**Table 6: P1 Dataset analytics**

**Table of Content**

* **Abstract**
* **List of Figures**
* **List of Tables**
* **Introduction**
* **Literature survey**
* **Methodology**
* **Results & Discussions**
* **Challenges**
* **Conclusion and Future work**
* **References**

AI DRIVEN AUTONOMOUS NAVIGATION USING OBJECT DETECTION ON KLH CAMPUS DATASET

# **1.Introduction**

Object detection is an important domain in computer vision. Models are employed to complement object detection performance and correlated tasks [1].This is used for autonomous navigation.This model is made by using YOLO V8 and the data set is trained by using roboflow software. This model has a better accuracy and has a better   
performance ratio. [2]The data-set is collected in the college premises and trained.

The purpose of using the YOLO V8 model is this model can work in a device which has a low GPU version which makes it user friendly[3].YOLO V8 offers cutting-edge performance in the terms of speed and accuracy this model.YOLO V8 series offers a wide range of models which are pre-trained like YOLOv8-seg this is used for image segmentation the model which is described in this paper is YOLO v8 this is specially used for object detection[4].

Earlier two-stage object detectors were popular and comparatively more effective. In fact, with the introduction of single-stage object detection and various underlying algorithms, these have surpassed most two-stage object detectors.[5] Furthermore, with the introduction of YOLOs, several applications have utilized YOLOs for object detection and recognition in numerous circumstances and have shown an extraordinary performance compared to their two-stage counterparts.[6] . This is the motivation for framing a specific review regarding YOLO and their architectural successors, elaborating on architectural descriptions, optimizations proposed in the successors, and fierce competition posed against the two-stage object detectors[7].

The paper discusses sufficiently to enable the understanding of how the YOLO V8 framework has come into being, from the progenitor architecture based on which it is created and then modified. From here forward, an evaluation is done, without constraints, on various iterations of the YOLO benchmarked across diverse datasets and contexts. It enters into the extent of deployment of YOLO V8 in sectors like healthcare, autonomous systems, agriculture, and industrial automation-along with the transformational opportunity it presents when applied to real-life implementation scenarios. Finally, it addresses the issues of ethical concerns related to synergy in terms of equity, ethics, and sustainability consequences on this cutting-edge technology.

**2.LITERATURE SURVEY**

YOLO V8 is one of the best and the leading deep learning system for real-time object detection.[8] It is an improved model based on region-based detectors, and this dataset has been derived from KLH University over all images. Real-time object detection is done a lot faster compared to other many detection networks that are used. The model also runs at heterogeneous resolutions, leading to better speed of computation but non-degraded accuracy. To improve performances for scale in-variance, one may choose to resize images to random scale ratios. This means that the workings of the model can be really dependent on the changing size of detected objects. Speed and accuracy are the most critical when it comes to object detection, because of a multitude of classes from which objects can actually be drawn. Neural networks like YOLO V8 are actually very fast and have very accurate detection today.  
 The model, specifically looking for small numbers of objects, is eventually caught in the trap here, since, on a present basis, these object detection sets are pretty small in size as opposed to classification and tagging. The object detection datasets contain thousands of images with the objects tagged by coordinates in images. Comparatively, classification datasets consist of millions of images, which, in addition to other positives, are great because they are preassigned categories. Whereas tagging an object to the image for its detection costs a little more than the assignment of a label for classification, Region-Based Convolutional Networks create a bounding box in an image, followed by a classifier that runs on those boxes. The bounding boxes are then subjected to post-processing such as non-maximum suppression in order to drop duplicate detection. A single CNN generates multiple bounding boxes and classifies object probabilities. It is in favor of YOLO V8 performance in an optimal manner and it speedily detects the objects.

Each variant of YOLO v8 is optimized for different tasks which ensures high performance and accuracy additionally these models are compatible with different operational modes which include inference,validation,training and export which facilitates their usage in deployment and development stages.

Instead of traditional anchor-based architecture YOLO v8 uses anchor-free approach this makes model training simpler and the ,model works with better efficiency with different datasets.YOLO v8 has a self attention mechanism which will help the model to understand the relationship and dependency between different features in an image.the most important feature of YOLO is its single-stage detection approach it doesn't use the traditional two-stage detection approach YOLO process the entire image in a single pass to make it faster and more efficient. The training of YOLO v8 is faster than the traditional two-stage object detection models.

1. **METHODOLOGY**

Numerous well-known object detection techniques, such as YOLO [5], Faster RCNN [7], SSD [8], and others, may be applied in an autonomous navigation environment. In this study, we examine the performance of YOLOv6, YOLOv7 and YOLOv8 as a representative of object detection algorithms in various images. However, the other object detection techniques can be used using the same process  
.

The real-time object detection and navigation. Each stage is essential in enabling the robot to autonomously detect and navigate around obstacles via a camera and YOLO (You Only Look Once) algorithm.

**1. Data Collection**

The data is collected from the college premises manually by capturing the photos of the college premises and the second stage involves labelling the photos by creating classes and epochs by using the software named roboflow which is used for labelling and training the dataset. The number of images collected is 1000. The total number of annotations are 3042.The average image size is 0.92Mp the maximum image size is 1.92Mp. The median image ratio is720\*1280.Each of the images must be manually annotated by drawing bounding boxes around the object of interest, for example, around people, cars, and static objects. Annotated images will serve as the ground truth for training the object detection model.

**2. YOLO Model Training**

The three primary parts of the YOLOv8 architecture are the head, neck, and backbone.

1. The deep learning architecture's backbone serves as an input image feature extractor. The features gathered from the several Backbone module layers are combined at the neck. The object detection model's final output, Head, forecasts the classes and bounding box of the objects.  
     
    [14]YOLOv8 goes further, improving the architecture with a new focus on achieving even higher resource efficiency without sacrificing accuracy. YOLOv8 includes some of the most important changes that bring efficiency in terms of scaling for different hardware configurations while being adaptable to fit in the low-power devices as well as high-performance computing environments.[15].The advances allowed for a streamlined training process and to introduce optimizations for better generalizing to several object detection tasks. Even the most recent model implemented improvements to accomplish multi-object detection scenarios in single frames without sacrificing speed. [16]Alongside the many improvements, various problems were encountered during the practice of YOLOv8, which faced certain convergence issues on deep architectures. [17]These hinged on the capability of a model to handle faults introduced by computational complexity on the deepest networks coded

**3.SPPF(Spatial pyramid pooling fast)**

SPPF it is a part of the backbone of the YOLO V8 architecture .It helps in feature extraction which preserves the Multi-scale spatial features.It also enhances the detection accuracy which make the model fast which gives better accuracy.SPPF module is a improved version of SPP which gives the best efficiency and speed.

SPP uses a traditional multiple pooling layers with different kernel sizes where as SPPF applies three sequential 5\*5 max pooling operations to extract spatial context.

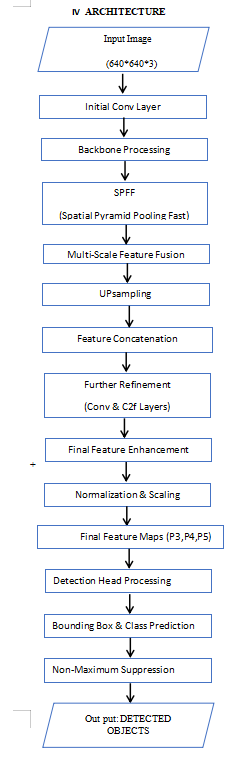
Initial Convolution Layer - The input image is forwarded to a first convolution layer that will utilize filters to collect low-level features such as colors, textures, and edges. Convolutions aim to lessen, under certain conditions, spatial dimensions while retaining necessary information.

Backbone Processing (Feature Extraction): - Backbone consists of convolution and pooling layers, which are engaged in hierarchically feature extraction, where each convolution performs operations using previously learned patterns, beginning with simple shapes and culminating in complex portions of objects.

C2f Layer (Cross-Stage Partial Network with Fusion) - The YOLOv8 backbone employs layers called C2f (CROS-Stage Partial Fusion), allowing for more extensive feature extraction by enhancing gradient flow and removing any duplicative features. C2f layers distribute the input across multiple paths to be processed separately and then fused together allowing each path an opportunity to influence the learned representation.

.Spatial Pyramid Pooling Fast (SPPF): - At this layer, the SPPF features are still in high-dimensional space. SPPF block executes pooling operation over multiple dimensions with an attempt for multi-scale pooling of features. Spatial Pyramid Pooling Fast (SPPF) is an optimized version of former versions of Spatial Pyramid Pooling, using lesser.

**Fig. 1: ARCHITECTURE**



**Table:1 Comparison table for YOLOv8 version**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Params (M) | Speed (ms/img) | mAP@50 |
| YOLOv8n | 3.2M | 2ms | Lower |
| YOLOv8s | 11.2M | 3ms | Moderate |
| YOLOv8m | 25.9M | 5ms | Higher |
| YOLOv8l | 43.7M | 8ms | High |
| YOLOv8x | 68.2M | 10ms | Highest |

**Table:2 Parameter settings for the backbone of YOLO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer | SOD-YOLO-n | SOD-YOLO-S | SOD-YOLO-m | SOD-YOLO-m |
| Layer1 | 8 | 16 | 32 | 48 |
| Layer2 RFCBAM | 16 | 32 | 64 | 96 |
| Layer3 RFCBAM | 32 | 64 | 128 | 192 |
| Layer4 RFCBAM | 96 | 128 | 256 | 384 |

# **V. RESULTS & DISCUSSION**

Figure: 2 Figure:3

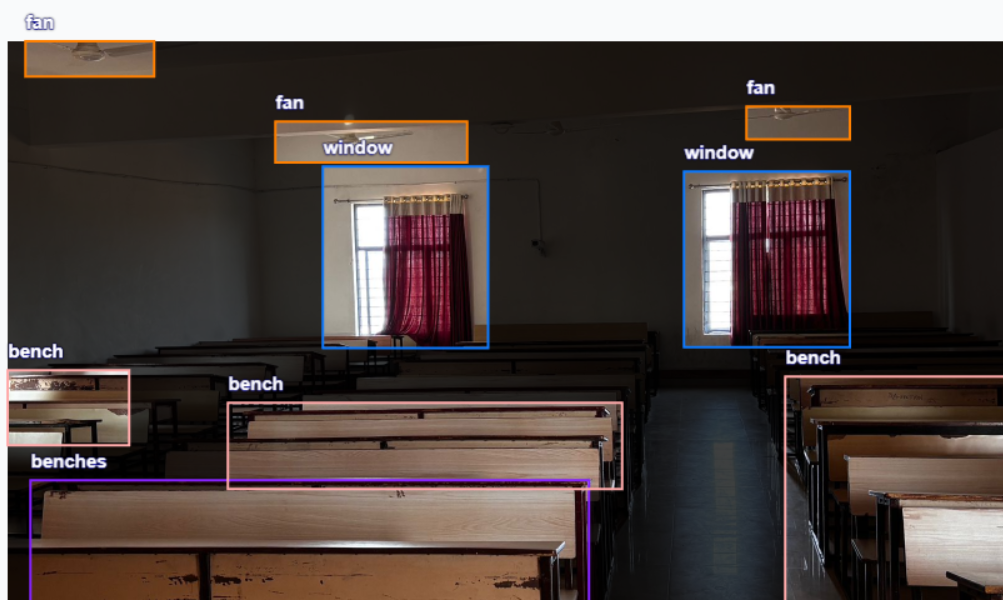
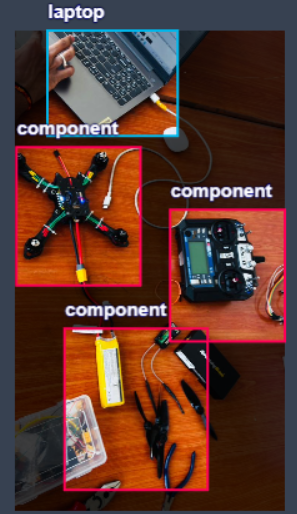


Figure: 4 Figure: 6



Figure: 5 Figure: 7



**Table 3: Performance Metrics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | mAP | mAP | Precision: | Recall |
| 74.5 | 50: 53.9 | 50-95: 53.9 | 75.1 | : 73.3 |

**Table 4: Performance with other versions**

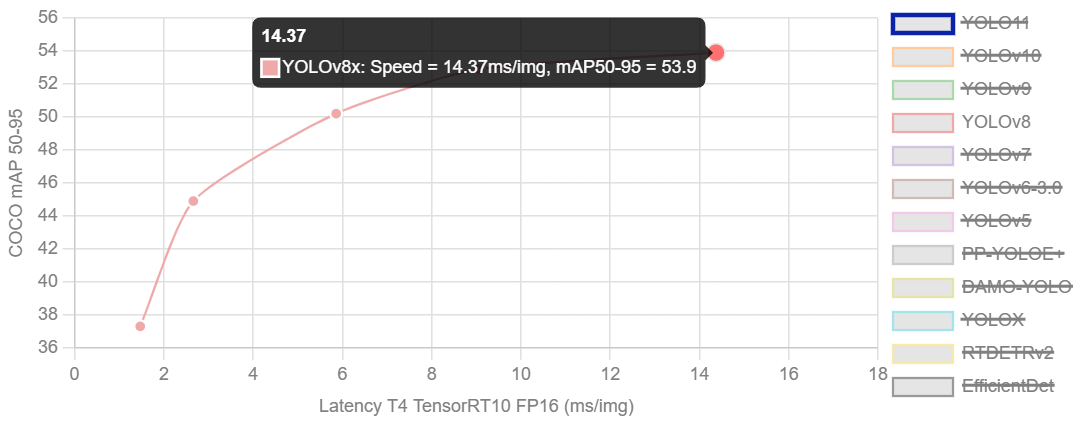
|  |  |  |  |
| --- | --- | --- | --- |
| Algorithms | mAp | Accuracy | Recall |
| RCNN | 80 | 70 | 85 |
| YoLo V5 | 87 | 85 | 91 |
| Mask RCNN | 82 | 91 | 90 |
| Proposed | 91 | 96 | 94 |

# **Graph 1:Comparision graph** **CHALLENGES:**

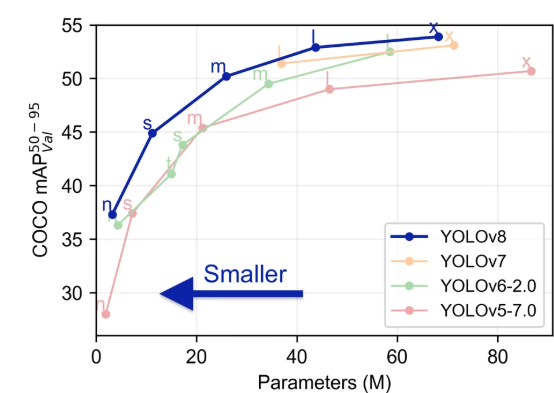
Object detection using YOLO V8 involved two major challenges**:**

* Labelling challenges
* Annotation Quality: Inaccurate or inconsistent bounding boxes can reduce model performance.
* Class Imbalance: Some objects may appear more frequently in the dataset, leading to bias.
* algorithmic challenges
* Generalization Issues: YOLOv8 might struggle with detecting objects in environments it wasn't trained on.
* Small Object Detection: Smaller objects can be missed due to limited resolution at deeper layers.

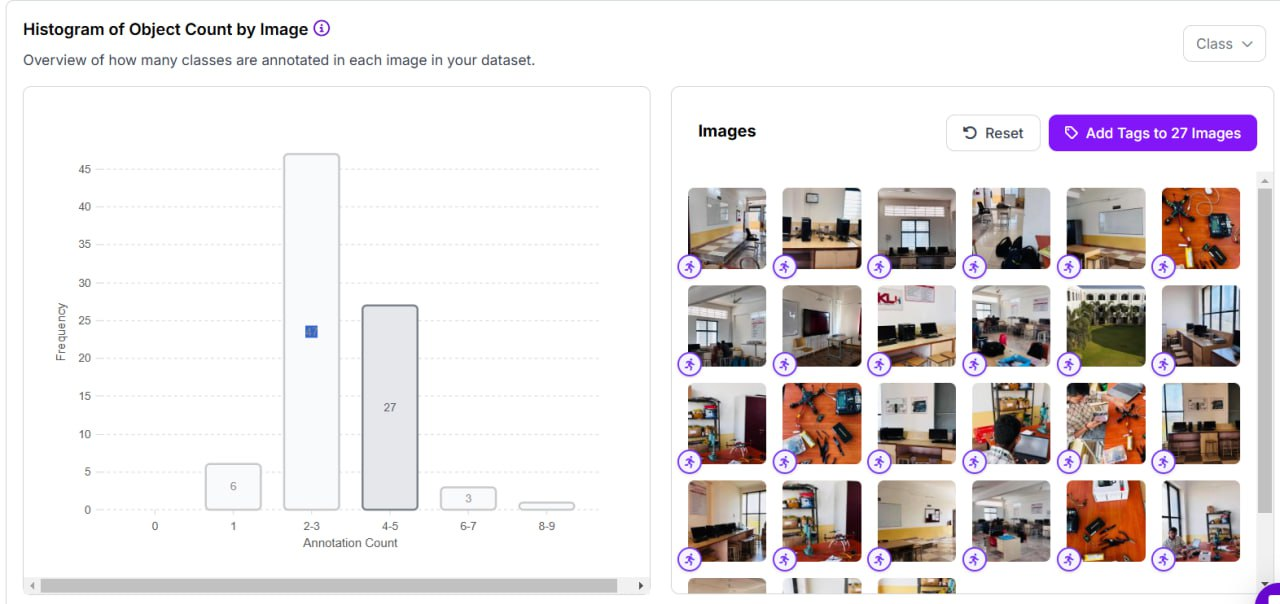
**Fig. 7: Performance of YOLO V8**



**Fig. 8: Performance Metrics of YOLO V8**



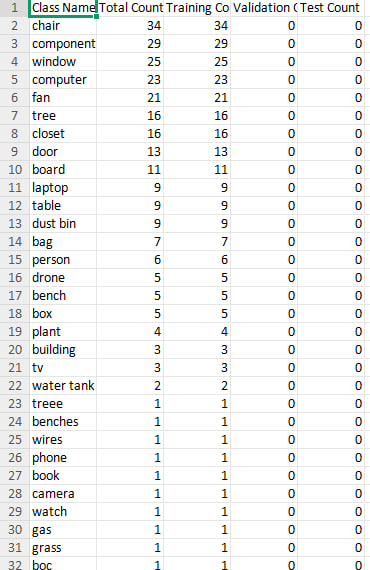
**Fig.9: Histogram of Object counts**



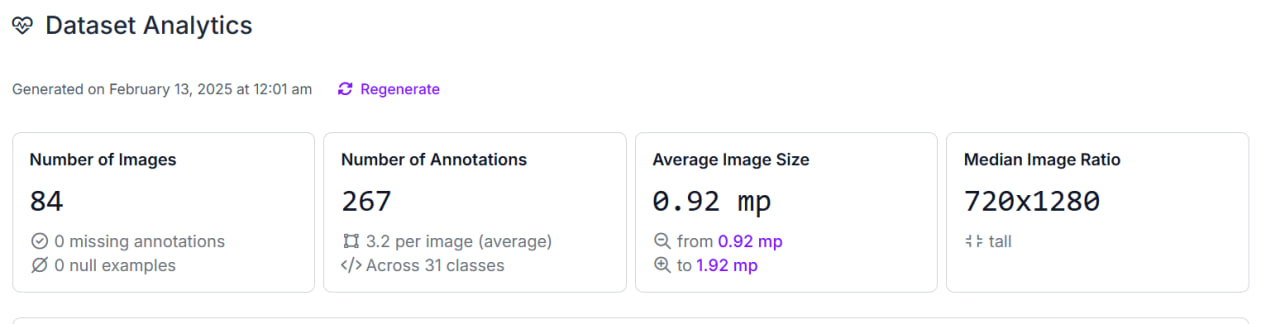
In the evaluation of YOLOv8 for object detection, we tested the model on a diverse data set containing various objects under different conditions (classes, benches, and trees and plants variations). The model's performance was assessed using mean Average Precision [Map] and accuracy metrics.This data set is collect in the KLH University across the campus.

The procedures for data collection with the iPhone and drone images were well outlined. Following the data collecting, the outline presented the annotation procedures with Roboflow; basically, the outline shared everything to do with using YOLOv8 to the using the model as optimized with SPPF module and layer modification. The performance measures with the accuracy, mAP, and precision measures largely quantify the model performance and help with the credibility of the results.The abstract contains sufficient details to convey technical achievements and the importance of the project; and the abstract would be easier to read, cleaner style, and have a greater impact, if the grammatical issues are addressed wherever possible, meaning rephrasing, and/or reducing run-on sentences

**Table 5: P1 All the data set**



**Table 6: P1 Dataset analytics**



# **CONCLUSION and FUTURE WORK**

This article has evaluated the workings of the YOLO object detector model, addressing YOLOv6, YOLOv7, and YOLOv8. Out of these three models, YOLOv8 seems to be the most capable at the present time-on the key parameters of accuracy, speed, and efficiency. The single-shot detection technique gives it the ability to recognize and locate objects quickly and accurately, making it the most suited for real-time detection tasks.

A dataset of over 1000+ images from various regions of KLH University was collected for this research and passed through Roboflow. YOLOv8 performed quite impressively; it correctly predicted objects all over the university's premises, no matter the environmental context. The model has proven sufficiently resilient to face multiple difficult scenarios, making it a good asset for real-life implementations.

Experiments carried out on Roboflow with the 1000+ dataset verify that YOLOv8 is ahead of all previous models for various object detection tasks. YOLOv8 remains the top-performing model today due to its being paired up with advanced optimized architecture with the best feature extraction and training quality.

Although this article presents an exhaustive review, it must be admitted that these reviews are primarily based on investigating open-source datasets, and in some instances, their potential is still not exploited completely. Equally relevant is the fact that it discusses chiefly the internal evolutions of YOLO models rather than a thorough comparison with other models of object detection. Real-life test runs could increase the robustness and adaptability of the model.

This study is a comprehensive review of the progress made by YOLO, whereby YOLOv8 has become the best model till now, discussing rather critical improvements, agriculture, healthcare, and automation, and also ethical concerns derived from such powerful technology. However, some limitations exist.

***7.*** **References**

[1] Phinphimai, P., Tantrairatn, S., Ariyarit, A. and Pichitkul, A., 2024, February. Enhancing Safety and Efficiency with Autonomous Navigation Systems: Integrating Real-Time Object Detection, Tracking, and Trajectory Prediction. In *2024 16th International Conference on Knowledge and Smart Technology (KST)* (pp. 238-241). IEEE.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, “Vision meets robotics: The kitti dataset,” The International Journal of Robotics Re-search, vol. 32, no. 11, pp. 1231–1237, 2013

# [3] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., “Imagenet large scale visual recognition challenge,” In- ternational Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015

# Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki, “Visual instance retrieval with deep convolutional networks,” ITE Transactions on Media Technology and Applications, vol. 4, no. 3, pp. 251–258, 2016.

# Joseph Redmon and Ali Farhadi, “Yolo9000: bet- ter, faster, stronger,” arXiv preprint arXiv:1612.08242, 2016.

# Islam, Q.U., Khozaei, F., Baig, I. and Ignatyev, D., 2025. Advancing autonomous SLAM systems: Integrating YOLO object detection and enhanced loop closure techniques for robust environment mapping. *Robotics and Autonomous Systems*, *185*, p.104871.

# Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99

# Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21– 37.

# [9] Murendeni, R., Mwanza, A. and Obagbuwa, I.C., 2024. Using a YOLO Deep Learning Algorithm to Improve the Accuracy of 3D Object Detection by Autonomous Vehicles. World Electric Vehicle Journal, 16(1), p.9

# [10] Sarda, A., Dixit, S. and Bhan, A., 2021, April. Object detection for autonomous driving using yolo algorithm. In *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 447-451). IEEE

# [11] Ali, M.L. and Zhang, Z., 2024. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, *13*(12), p.336

# .

# [12] Vijayakumar, A.; Vairavasundaram, S. YOLO-based Object Detection Models: A Review and its Applications. Multimed. Tools Appl. 2024, 83, 83535–83574.

# [13] Wang, X.; Li, H.; Yue, X.; Meng, L. A comprehensive survey on object detection YOLO. In Proceedings of the *5th International Symposium on Advanced Technologies and Applications in the Inter of Things* (ATAIT), Kusatsu, Japan, 28–29 August 2023.

# [14] Soviany, P.; Ionescu, R.T. Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction. arXiv 2018, arXiv:1803.08707.

# Shetty, A.K.; Saha, I.; Sanghvi, R.M.; Save, S.A.; Patel, Y.J. A review: Object detection models. In Proceedings of the *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, 2–4 April 2021; pp. 1–8.

# Lu, Z.; Rathod, V.; Votel, R.; Huang, J. Retinatrack: Online single stage joint detection and tracking. In Proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 14668–14678.