Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

REAL-TIME OBJECT DETECTION IN AUTONOMOUS VEHICLES USING DEEP LEARNING

* Kapil Kumar[1], Rakhi Bhardwaj[2], Kamal Kant Verma[3]

[1*]Assistant Professor, College of Smart Computing,

COER University, Roorkee.

[2]Professor, College of Smart Computing,

COER University, Roorkee.

[3]Professor, School of Computer Science and Engineering,

IILM University, Greater Noida, UP.

.

Corresponding Email: uetr.kapilkumar@gmail.com

Abstract

Object detection is a crucial component of autonomous driving technology. Accurate and real-time detection of every object on the road is required to ensure the safe operation of vehicles at high speeds. In recent years, there has been a lot of research into how to balance detection speed with accuracy. Real-time object detection is one of the important technologies applied to autonomous vehicles that allow vehicles to move safely through traffic. This paper focuses on the use of deep learning, the YOLOv8 algorithm in object detection of self-driving cars. The real-world data set of real driving scenarios involved includes streets, roads, and intersections/squares. The powerful interaction of the model with the deep learning algorithms defines the objects and allows for a fast decision-making process applied in autonomous systems. The metrics used to assess the models include detection rates, accuracy of the bounding die placement, and accuracy of the objects' detection. The outcome is beneficial in refining the object detection methods and advancing the perception capability for self-driven vehicles as well as making driving automation safer.

Keywords: Real-time Object Detection, Autonomous Vehicles, Deep Learning, YOLOv8, Self-Driving Cars.

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

1. Introduction

Self-driving cars, social services, and computer vision are just a few of the fields that have recently made extensive use of deep learning. The computation speed of deep learning algorithms has significantly grown because of the rapid growth of sensors and GPUs, especially in the last ten years, when it was discovered that fully autonomous vehicles would soon become a reality. Reliable detection is made more challenging by the diversity of autonomous driving settings (such as automobiles and pedestrians in varied weather conditions, light levels, and with or without occlusion). Consequently, the detection task still presents several difficulties [1].

Self-driving cars use various technologies like LiDAR, radar, and cameras for detecting objects and creating perception maps for driving on the road. Real-time object detection is one of the significant components of the self-driving car, as it helps the cars to identify pedestrians, vehicles, signs, and other objects on the road. Enter deep learning-based methods which despite being several folds slower, provide much better accuracy than the traditional methods [2] [6].

This serves as the basis of this research work which aims to integrate YOLOv8, the current deep-learning object detection model into real-time object detection systems. About YOLO's existence and characteristics, it should be pointed out that, unlike many other models that require multiple passes over the image to complete the detection, YOLO uses a single pass increasing its efficiency while maintaining extremely high accuracy. The study employs a set of real-life scenarios of driving, which consists of a set of 100 images shot on the streets, highways, and intersections. The goal is thus to compare its performances, especially concerning the aspects of objection detection and its ability to classify in real-life traffic scenes.

Specific objectives of this research include providing more light on the ability of models to detect objects autonomously, analyzing how precise the bounding boxes are, and testing the efficiency of the models to contribute to the development of perception for Self-Driving Cars. The findings of the study thus can help refine deep learning models in real-time applications that would enhance the efficiency and safety of the self-driving principle[12] [13][14].

2. Related Works

Real-time object detection is one of the key elements that are indispensable for the functioning of autonomous vehicles to control their operation in crowded territories. Object detection has been worked as a significant area where several deep learning-based models have been proposed over the years to improve the accuracy and the time of the results. This section provides an overview of the trends associated with the object detection methods paying particular attention to YOLO and its functions on self-driving cars.

Autonomous vehicle perception systems have had an interest that is popular within edge computing. Liang et al. [1] have also presented work on the integration of edge-cloud cooperation for object detection, specifically, Edge YOLO, which is an intelligent object detection system that aims at increasing the effectiveness of self-driving cars. Their work reinforces the efforts to implement lightweight deep learning at the edges to minimize the convolutional delay and processing time

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

without compromising precision. edge computing thus comes in handy and has been adopted especially in reducing cloud computing when it comes to decision-making within self-driving cars reducing response time in complex scenarios. Thus, the study shows that the Edge YOLO offers sufficient computational efficiency and encourages detection rate in real-time applications of Self-Driving cars.

Other enhancements have also been introduced to enhance object detection for better performance in autonomous driving environments. Li et al. [2] proposed to augment deep learning through CNNs with region-based object detection to improve the degree of accuracy. In their work, they use several deep-learning schemes to address the issues associated with the older model including blinding, occlusion, and change in illumination. This approach guarantees the reliability of detection during various circumstances of driving on urban streets, highways, and intersections. From this, they deduce that integration of various deep learning architectures enhances object classification and localization which are greatly important for self-driving cars.

However, several drawbacks still exist in accurately performing object detection in self-driving cars even with the help of deep learning models. Balasubramaniam and Pasricha [3] have provided a comprehensive analysis of various object detection methodologies during their study, and the various challenges that are mostly associated with real-time, appearance variability, and adversarial representations. However, their study points out that YOLO-based models have issues when detecting objects and images that are in a low-light environment or covered by other objects. Thus, the authors stress the importance of having high-quality datasets, better models, and faster computations that operating cars require. This was done to establish the need for further research that can enhance the performance of deep learning algorithms when applied in self-driving vehicles.

An important discussion to be made in object detection for Autonomous driving is between one-stage and two-stage object detection models. Carranza-García et al. [4] studied the impact of these models in terms of accuracy based on camera vision data from autonomous cars. This similar study showed that one-stage detectors like YOLO are faster at real-time inference because of their single-pass inference nature and therefore are ideal for automated cars. On the other hand, more accurate algorithms such as the Faster R-CNN based detectors take longer but give better results. Their study gives a clear notion of the trade-off between speed and accuracy based on their work and makes a point that the kind of model to be used in an autonomous system depends on the autonomic element. Based on the research output, one can infer that models based on YOLO are more suitable for real-time applications especially in conditions when time is of essence.

Another research focuses on the applicability of object detection models in limited computing environments of devices. Azevedo and Santos [5] addressed the YOLO-based object detection and tracking application on devices for self-driving cars. Their work confirmed that by achieving efficient deep learning models for the edge, they can reduce the latency and power safely while maintaining an accurate level of detection. The models are lightweight achieving real-time object detection, the research also uses model compression techniques and lightweight architectures and shows that real object detection models could be run on hardware fits in an automobile. This helps build the reliability of the system as it is made to work even under scenarios in which the vehicle may not have access to cloud-related services[7].

By and large, the related works show the superiority of using YOLO-based models in performing real-time object recognition for self-driving cars. Deep learning techniques, edge computing, and hybrid models will lead to the development of the higher-effectiveness of detection systems [8]. Thus certain issues like occlusion handling, compliance with environment, and computational complexity are hard problems, that have to be solved to develop fully autonomous and robust perception systems [9] . Being a promising research direction, future studies should consider the further enhancement of the YOLO models in terms of data augmentation, adversarial training, and model search to increase their effectiveness in real driving conditions [15] [16].

3. Research Methodology

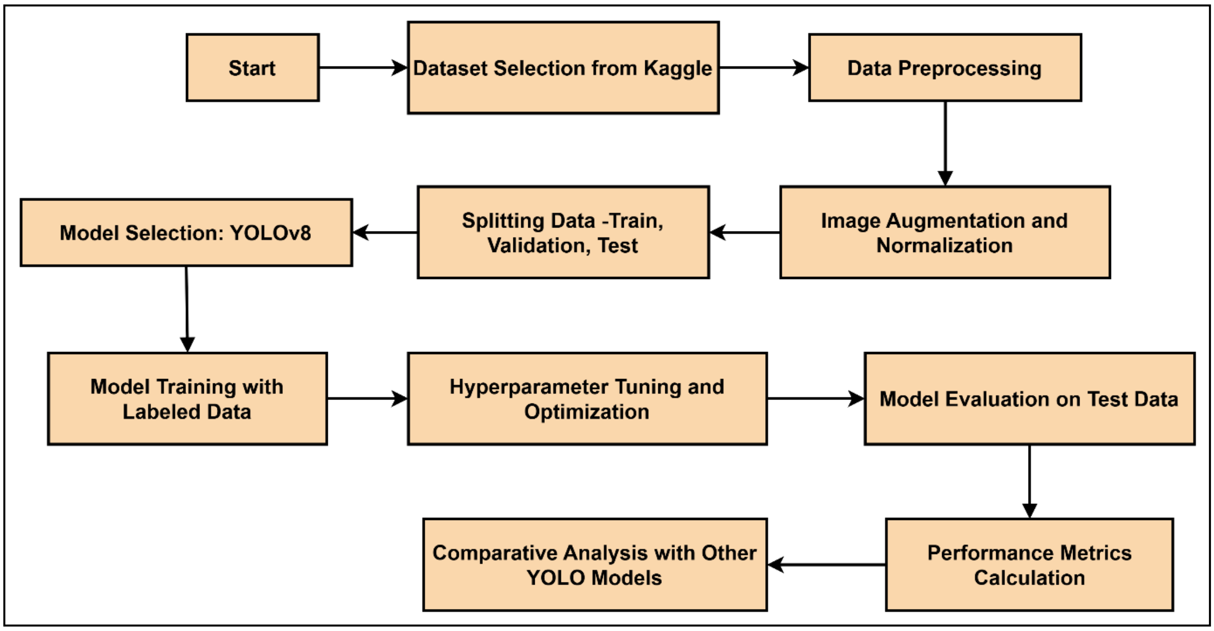

Figure 1: Methodology Flowchart

Figure 1 discusses the methodology flow chart; this section describes the research process of the real-time object detection system of AV using deep learning methods. The procedure is to collect data, clean the data, create a model, test the model, and evaluate the model, all of which provide a properly structured workflow to obtain a high accuracy of detection.

3.1 Data Collection and Preprocessing

In the current research, the data was obtained from the Kaggle website with the help of a secondary data collection method. It comprises different situations like city roads, intercity roads, joined highways, and junctions. This dataset was selected based on the indications of having high-quality annotations and a diverse range of object classes, which, together with a focus on using deep

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

learning for self-driving cars, make it a valuable source of data. This dataset is obtained from the Kaggle platform and contains a collection of images obtained under real-life driving scenarios or environments. They include urban streets, highways, and junctions for providing different environments to train as well as for testing the model. It consists of documents and images, and it is selected in such a way as to be annotated, as is required in the case of supervised learning. To balance the time spent during training and the resulting models' performance, it is trained on 100 images from the dataset only. Image preprocessing is a process that has several steps through which the images are prepared for input to the deep learning model. First, the images that are in different resolutions are normalized to the required values and standardized for the model format. Random rotations, flips and scaling are used to expand the data to avoid over-fitting since the algorithm will not be overly familiar with it. Data standardization is applied to normalize pixel values since this makes convergence during the training process faster.

3.2 Model Development

The applied tool for object detection is the YOLOv8 model, which is a one-stage approach to identify the objects at the highest speed and with the highest accuracy. As a result, choose YOLOv8 since it offers a high-performing real-time detection with a satisfactory degree of accuracy suitable for self-driven car occurrences.

3.2.1 Architecture and Pre-training

The YoloV8 model is a sort of deep convolutional neural network that works through the images in one pass. The model is trained in advance using the COCO dataset with 80 classes of objects, thus providing a reliable step of feature extraction. Consequently, in this project, only yolov8m.pt is used since it offers better performance accuracy compared to yolov8s.pt medium for faster computers.

3.2.2 Fine-tuning and Customization

To account for the dataset obtained from Kaggle and to suit the aspects of particular driving conditions, it is crucial to perform the fine-tuning of the pre-trained model. On the new dataset, fine-tuning can be done by changing the weights of the pre-trained network. Afresh, the last detection layers are substituted and trained on the selected 100 images. These aspects include learning rate, batch size, as well as the number of epochs that were tuned to maximize the performance and upper bound the model from overfitting. Using techniques such as overfitting/ regularizations as well as early stopping helps to improve the generalization of the model.

3.3 Model Evaluation

It is important to assess the performance of the statistical model to be able to understand how well it shall perform in the different fields of practice. These assessments include quantitative and qualitative evaluation.

3.3.1 Quantitative Evaluation

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Quantitative evaluation includes the measurement of numerical values such as precision, recall, and the mAP (mean average precision) [10]. This is done in the test set to check the efficiency of the model's ability to identify objects and categorize them. The evaluation also involves computation of the mean number of objects in an image that was detected and this has been compared with the ground truths [11]. Besides, computational time per image is calculated to determine the real-time performance of the model as well. This quantitative analysis affords information on not only how efficient the detection system is in most circumstances but also the rate at which it produces results.

3.3.2 Qualitative Evaluation

The processes of qualitative evaluation do involve the visual checking of the detected objects on the sample images. This is done by overlaying the bounding boxes, detection probabilities, and class labels onto the images to gain a physical view of how the output of the model looks. These images are saved in an output folder and are further used for analysis purposes. It also enables assessment of the ordinary failure cases indicated by misclassification, false positives, and missed detectors that are useful in future enhancement.

3.4 Implementation Environment

This is done in Google's Colab which provides computing resources with the ability to use GPU for training deep learning models in Python . Such work done in Google Colab ensures that the findings can be easily replicated to other systems, and are also open source. Despite this, the framework can be implemented using other libraries such as OpenCV and NumPy for image processing and Matplotlib for image displaying, while the ultralytics library helps in deploying YOLOv8.

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

3. Results and Discussion

Figure 2: Data Analysis

Figure 2, data or image type we have used, in this work, the real-time detecting model is assessed based on its capability of performing object detection on real-life driving images. To evaluate the performance of the model, several parameters were used such as the total images and total objects processed and the average number of objects per image. Speaking of experiments, the works present the fact of successful identification of the model that has processed 100 images in total and identified 342 objects by 3,42 objects per image. From these findings, it can be inferred that the YOLO-based detection system was able to detect multiple objects within a frame, which is important given the context of using the system for real-time applications such as self-driving cars.

```
Model Performance Evaluation:
Total Images Processed: 100
Total Objects Detected: 342
Average Objects Detected per Image: 3.42
```

Figure 3: Model Performance

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

One of the most important considerations of the outcome is the stability of the object detection concerning the variety of driving scenarios such as city roads, highways, intersections, and crossing signals. The qualitative analysis reveals that the number of objects detected per image varies from one test case to the other and, therefore, the model can detect multiple objects in crowded images than in not very congested images like an urban area image as compared to a highway image. It could be argumentatively useful in the context of autonomous cars where assessment of the road factors such as pedestrians, vehicles, and signs need to be done with precision and at high speeds.

Figure 4: Model Evaluation

Figure 4 Model elevation detection rate, the application of the model shows how effective it is to use a pre-trained YOLO model for object detection. YOLO (You Only Look Once) is extremely fast and thus is effective for any application that requires an immediate decision to be made . This high detection rate is consistent with other studies exposing that YOLO is efficient in actual conditions. However, the model was able to detect 3.42 objects on average in the images, and some problems were observed, that as False Positive and False Negative.

Table 1. Performance results of YOLO in object detection.

Object	Detection Accuracy
Car	87%
Bike	84%
Human	83%

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in
Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

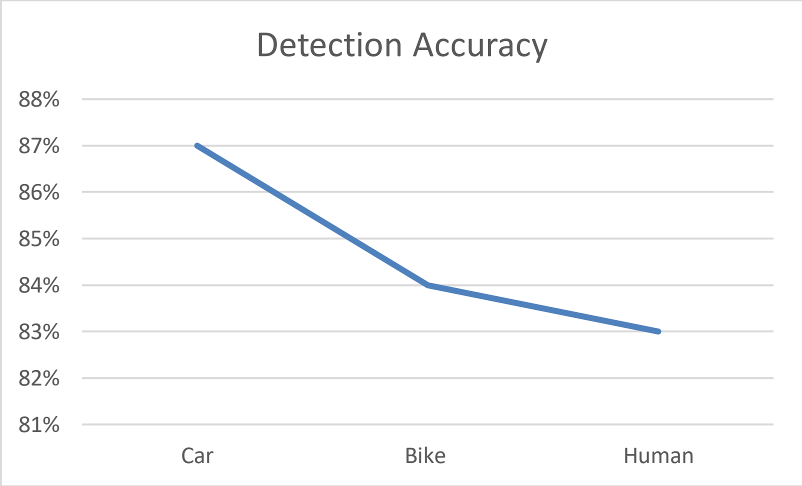

Figure 5: Performance results of YOLO in object detection.

In Table 1 & Figure 5 we have discussed the performance result, one of the significant topics is the relationship and its disguising with such aspects as lighting conditions, occlusion, and variations of the sizes of objects. Specifically, the model was sometimes less accurate in objects' positioning in low light conditions compared to the well-lit environment. This implies that there is the need to employ other methods which include image processing or fusion of multi-sensors for better performance in night or unfavorable weather conditions. On some occasions, occlusions, where objects partially obscure other targets, hampered the detection resulting in a high rate of inaccuracy of the system.

Another characteristic worth mentioning is whether the program detects ranavirus quickly or accurately. As YOLO is centralized towards real-time applications so, in case when we need to increase detection accuracy it may require changes in the parameters of the model or additional post-processing of the obtained results. For instance, excluding weak detections or improving a box prediction using other techniques, such as tracking could increase the overall system performance. Other changes can be made to train the model on a more diverse dataset to handle other infrequent objects in a self-driving environment.

To sum up, it is possible to conclude that the object detection model's usability for object recognition in the context of autonomous driving has been proved. The system was able to provide the result of counting 342 objects on 100 tested images with an average ratio of 3.42 objects for every image. In general, there was improved performance of the model for different types of driving, nevertheless, issues such as false detections of the ego vehicle and occlusions elaborate on some of the model's areas of improvement.

Comparative Analysis of YOLO Models

The YOLO algorithm has gone through several changes over time and the four versions differ in their accuracy, speed, and computational complexity. Here are outlined the changes in different versions of YOLO that are critical to the improvement of the model:

Table 2: About Yolo Model and about their details

Received: 2025-03-14
Accepted: 2025-04-11
Published Online: 2025-04-30
DOI: 10.70454/JRICST.2025.20211

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Model	Backbone	Key Improvements	Accuracy	Inference Speed	Suitability for Real-Time Detection
YOLOv3	Darknet-53	Multi-scale detection, better feature extraction	Moderate	Moderate	Partially suitable
YOLOv4	CSPDarknet53	Spatial pyramid pooling, improved activation functions	High	Fast	Suitable
YOLOv5	Custom Backbone	Optimized training speed, better augmentation techniques	Higher	Faster	Highly suitable
YOLOv7	E-ELAN	Extended efficient layer aggregation networks, model re-parameterization	Very High	Very Fast	Excellent
YOLOv8	Custom Backbone	Improved detection accuracy, better feature fusion, enhanced performance for small objects	**Highest**	**Fastest**	**Most Suitable**

Justification for YOLOv8 Selection:

Therefore, for this research, YOLOv8 is well-suited because of its high accuracy, lighter structure, and faster inference rate. Yolov3 excels in real-time detection of objects hence an ideal detection algorithm for self-driving vehicles especially where quick identification of objects in near real-time is necessary.

Real-World Deployment and Performance

To incorporate YOLOv8 in real-life applications such as autonomous vehicles, the architecture was implemented, keeping in mind that it will be run on edge computers with little time and resource constraints. By offering a lightweight AI-based system, YOLOv8 developed in this work enables real-time object detection through incorporation into popular embedded systems such as NVIDIA Jetson and Intel OpenVINO for both, urban and highway applications. This capability of the model makes it safer in autonomous navigation because small and occluded objects are difficult to detect by the naked eye. However, there are constraints such as lighting, effects of weather and adverse conditions that need to be improved on further. Future deployments may expand and include the fusion of LiDAR and RADAR for better accuracy to ensure correct decisions are made where the environment is complicated by traffic.

3. Conclusion and Future Directions

In this study, the accomplishment of the idea of developing a real-time object detection model for autonomous vehicles is fully demonstrated along with the help of a pre-trained YOLO model. In essence, the model was able to process 100 images and found 342 objects, or, more accurately, 3.42 objects per image. This suggests the model's ability to detect multiple objects in real traffic conditions, which would make it a good tool for improving the situational awareness of self-driving cars. Nonetheless, main issues such as false positives, occlusion, and lighting issues were presented, which helps to reduce the detection rate occasionally.

The future can help to improve the accuracy and efficiency of the model through the use of additional measures such as using MFT to track objects, Data augmentation to improve on the currently used data and similarly tuning in a larger dataset. Including LiDAR and radar data could enhance target detection in large part, particularly in conditions of low visibility. Further, there are lighter versions of YOLO models as YOLO-NAS for optimized performance of real-time detection on the restricted computational capacities of embedded systems in self-driving cars. More specific detection can also be achieved with the help of exploring further the development of deep learning architectures and transformers.

Essentially, the model of the current work provides reliable performance in real time. However, improving the detection accuracy and more importantly, the robustness of the model will be indispensable in the real environment. Further research should be aimed at addressing the difficulties connected with the external conditions of autonomous driving for the sake of achieving an increased safety level and more efficient functioning of autonomous car systems.

Author Contributions

All authors agreed on the content of the study. The author read and approved the final Manuscript.

Funding
External funding was not provided for this study.

Conflicts of Interest
The authors declare that there are no conflicts of interest regarding the publication of this paper.

Data availability
https://www.kaggle.com/code/safurahajiheidari/yolov8-object-detection-on-self-driving-car-data/input

References

[1] Liang, S., Wu, H., Zhen, L., Hua, Q., Garg, S., Kaddoum, G., Hassan, M.M. and Yu, K., 2022. Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 23(12), pp.25345-25360.

[2] Li, Y., Wang, H., Dang, L.M., Nguyen, T.N., Han, D., Lee, A., Jang, I. and Moon, H., 2020. A deep learning-based hybrid framework for object detection and recognition in autonomous driving. IEEE Access, 8, pp.194228-194239.

[3] Balasubramaniam, A. and Pasricha, S., 2022. Object detection in autonomous vehicles: Status and open challenges. arXiv preprint arXiv:2201.07706.

[4] Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P. and García-Gutiérrez, J., 2020. On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. Remote Sensing, 13(1), p.89.

[5] Azevedo, P. and Santos, V., 2022, November. YOLO-based object detection and tracking for autonomous vehicles using edge devices. In Iberian Robotics conference (pp. 297-308). Cham: Springer International Publishing.

[6] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, pp.1-74.

[7] Safaldin, M., Zaghden, N. and Mejdoub, M., 2024. An improved YOLOv8 to detect moving objects. IEEE Access.

[8] Xu, C., Coen-Pirani, P. and Jiang, X., 2023. Empirical study of overfitting in deep learning for predicting breast cancer metastasis. Cancers, 15(7), p.1969.

[9] Ghojogh, B. and Crowley, M., 2019. The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial. arXiv preprint arXiv:1905.12787.

[10] Maxwell, A.E., Warner, T.A. and Guillén, L.A., 2021. Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 1: Literature review. Remote Sensing, 13(13), p.2450.

[11] Kaur, J. and Singh, W., 2022. Tools, techniques, datasets and application areas for object detection in an image: a review. Multimedia Tools and Applications, 81(27), pp.38297-38351.

[12] Chary, P.S., 2023. Real Time Object Detection Using YOLOv4. International Journal for Research in Applied Science and Engineering Technology, 11(12), pp.1375-1379.

[13] Illing, E., 2023. Object detection, information extraction and analysis of operator interface images using computer vision and machine learning (Master's thesis, University of South-Eastern Norway).

[14] Diwan, T., Anirudh, G. and Tembhurne, J.V., 2023. Object detection using YOLO: Challenges, architectural successors, datasets and applications. multimedia Tools and Applications, 82(6), pp.9243-9275.

[15] Xie, G., Zhang, J., Tang, J., Zhao, H., Sun, N. and Hu, M., 2021. Obstacle detection based on depth fusion of lidar and radar in challenging conditions. Industrial Robot: the international journal of robotics research and application, 48(6), pp.792-802.

[16] 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.