Educational Administration: Theory and Practice
2024, 30(4) 10401-10411
ISSN: 2148-2403
https://kuey.net/

Research Article

Comparison of YOLO Models for Object Detection from Parking Spot Images

Yash Doshi [1*][0009–0007–6376–0508], Khushi Shah [2][0009–0005–7722–8024], Neha Katre [3][0000–0001–8320–7071], Vinaya Sawant [4][0000–0003–4661–9231], Stevina Correia [5][0000–0001–6915–9712]

[1*,2,3,4,5]Dept. of Information Technology, SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai-400056, India.

Citation: *Yash Doshi* et al (2024), Comparison of YOLO Models for Object Detection from Parking Spot Images, *Educational Administration: Theory and Practice*, *30(4)* 10401-10411
Doi: 10.53555/kuey.v30i4.7025

ARTICLE INFO

ABSTRACT

This paper compares several You Only Look Once (YOLO) models for object detection in parking lot images. Surveillance, independent vehicles, and intelligent cities are some of the applications that demand object recognition. The YOLO algorithm has undergone various iterations to improve real-time performance and increase accuracy. Effective parking space management is one of the major players in reducing traffic congestion in cities. Computer vision-based systems show us a way forward by automatically identifying free parking slots. The present paper compares using the YOLOv3, YOLOv5, YOLOv7, and YOLOv8 models to test images taken from car parks in different periods of the year and under various meteorological conditions. The results of the conducted tests show all the strengths and weaknesses of each YOLO model type, considering statistical elements such as precision, recall, evaluation time, and ease of use.

Keywords: Car Detection · Object Detection · Parking · YOLO (You Only Look Once)

Introduction

Object detection is one of the most critical components in many domains, such as surveillance and security, autonomous vehicles, and smart cities. The You Only Look Once algorithm (YOLO) is one of the most famous used in the object detection niche. YOLO has been selected because of its real-time solution and accuracy. The YOLO algorithm has undergone several versions during its existence, and each of them has made many changes and optimizations throughout this experience, as object detection is a continually evolving task.

Efficient parking space management is critical for reducing congestion in urban areas and maximizing space use. A computer vision-based methodology, which enables the automatic determination of unoccupied parking areas is one potential approach to achieving such tasks. YOLO is one of the object detection algorithms that helps to quickly consider the location of everything through photographs or videos. Hence, it can be used to detect parking place occupancy.

Overall, the paper features a comparative analysis of multiple YOLO algorithm versions using performance metrics in terms of detecting empty parking spots on parking lot images. The versions in question include YOLOv3, YOLOv5, YOLOv7, and YOLOv8, and the performance is evaluated using the detection of whether the parking lot space is free. Therefore, the goal of the research is to investigate and evaluate all tested YOLO versions to identify the most appropriate one for the object detection.

The study uses two extensive datasets [11] [12] containing pictures of parking lots taken under different conditions such as varied lighting, weather types, and parking lot layouts. Because of these reasons, it's possible to have multiple YOLO algorithm versions that yield different performance results on this project.
It is necessary to utilize many versions of YOLO algorithms while implementing them alongside these datasets so as to investigate how they compare with each other during real-time detection of parking lots. The findings

obtained from the above procedure might find application within computer vision communities especially those dealing with surveillance systems among other fields.

Literature Review

The paper [1] provides an extensive review of the YOLO algorithm and its extensions, representing the conceptual foundation, topic, and object of study of the entire paper. YOLO was later significantly improved and modified, resulting in the YOLOv2, YOLOv3, YOLOv4, YOLOv5 models along with YOLO-LITE. The key objective is to compare the main types of YOLO implementations in terms of design reasons, feature development, limitations, connections, and importance.

It provides a technical comparison and analysis based on public data and valuable YOLO trends and related queries. The paper concludes reflections on the insights from the YOLO versions review. There are several major differences In the found YOLO versions, which are listed above. However, the commonalities also exist, which proves the similarity and relevance of the YOLO versions identified in the examined data. The major implication of the commonalities and found YOLO versions is the potential for different research and other numerous improvements, especially in scenarios.

The paper [2] introduces the ACPDS dataset for image-based parking space occupancy classification, addressing the growing need for efficient parking management systems to reduce congestion and emissions. Unlike prior datasets, ACPDS provides systematically annotated images from unique views, ensuring diversity and realism in parking lot scenarios across train, validation, and test sets. The paper identifies the limitations of existing models trained on generic object-detection datasets or limited application-specific datasets and introduces ACPDS as a challenging benchmark for evaluating model generalization. The conclusion summarizes the significance of ACPDS in facilitating research and development of practical models for parking space occupancy classification, achieving over 98% accuracy on unseen parking lots and paving the way for future improvements in parking management systems.

The paper [3] contributes to the field of digital image processing by exploring the efficacy of YOLOv2 and YOLOv3 algorithms in object detection from aerial photographs. With the increasing popularity of deep learning algorithms in various disciplines, object detection from aerial or terrestrial images has gained significant attention. Leveraging the DOTA dataset, the study evaluates the performance of YOLOv2 and YOLOv3 in detecting nine class objects, including large vehicles, small vehicles, planes, and more. Through precision, recall, and F1-score analyses, the study reveals that YOLOv2 outperforms YOLOv3 in five out of nine classes, while the latter excels in recognizing small objects. Notably, YOLOv3 demonstrates superior time performance, detecting objects in an average of 2.5 seconds compared to YOLOv2's 43 seconds. By shedding light on the strengths and weaknesses of both algorithms, the study provides valuable insights for future research in object detection from aerial images, emphasizing the importance of expanding training datasets and incorporating images of varying scales to enhance detection accuracy.

A study in the paper [4] has thoroughly summarized the entire YOLO family of algorithms. This specifically targets the investigation of the latest model for real-time object detection, YOLOv7. The research aims to compare the resolution performance of YOLOv7 in object detection with other YOLO models and state-of-the-art models like YOLOv5, YOLO-X, and YOLO-R.The results indicate that YOLOv7 has high accuracy but low frames per second (FPS) when compared to other models in the YOLO family. In spite of its limited FPS, YOLOv7's accuracy makes it suitable for business applications, and therefore could lead to wider use cases for real-time object detection across different domains.

The paper [5] addresses the challenge of real-time object detection on Unmanned Aerial Vehicles (UAVs) with limited computing resources. Their efficacy in terms of mean Average Precision (mAP) and Frames Per Second (FPS), crucial metrics for real-time applications, were evaluated by studying various YOLO series models on the Pascal VOC dataset. YOLOv3, YOLOv3-tiny, YOLOv3SPP3, YOLOv4 and YOLOv4-tiny were among the models evaluated. It was found out that YOLOv4 outperformed the rest by achieving an mAP of 87.48% and an FPS of 72. Nevertheless, despite having lower accuracy, YOLOv3-tiny was considered suitable for real-time UAV applications if speed in processing is of high essence.

Methodology

The primary motivation for conducting this comparative analysis of YOLOv3, YOLOv5, YOLOv7, and YOLOv8 is to systematically evaluate and benchmark their performance in the context of object detection. Each iteration of the YOLO model introduces advancements and optimizations, making it essential to understand their respective strengths and weaknesses. By comparing these models, the study aims to identify which version offers the best balance of accuracy, efficiency, and computational resource requirements.

This analysis not only provides valuable insights into the evolution of YOLO models but also guides practitioners and researchers in selecting the most suitable model for their specific application needs. It helps in determining whether newer versions offer significant improvements over their predecessors and if those improvements justify the potential increase in computational demands. Ultimately, the comparative study serves as a comprehensive resource for informed decision-making in the deployment of object detection systems. Figure 1 represent the system architecture of our Exposure Therapy AR application.

3.1Data Used

Two datasets were use for this investigation. It was sourced from Kaggle under the title “Parking Lot Database, for YOLO" [11] created by Duy Thanh Nguyen referred as “Dataset 1" and “Aerial View Car Detection for YOLOv5" [12] by BraunGe referred as “Dataset 2".

Dataset 1 encompasses images taken in parking lots which exhibit various possible scenarios as well as configurations necessary for object detection models’ training and testing. This data is categorized into two - NoXML and HasXML based on availability of annotation files. A variety of conditions in parking lot scenes under different weather and day-time settings are depicted in the dataset. The captured images include scenes of days, cloudy and rainy weather. For real world deployment scenarios, it is important that the models trained can effectively adapt to different surroundings. This diversity ensures that trained models can adapt effectively to settings, which is crucial, for real world deployment scenarios.The images, in the dataset are arranged in order making it easier to analyze them over time. This organization allows for the identification of trends or patterns in detection performance. Sample images from Dataset 1 are shown in Figure 1.

Fig.1. Samples from Parking Lot Dataset

Dataset 2 includes a richly-annotated high-resolution aerial images dataset marking the location of cars, created for training and testing YOLOv5 car detection models. We varied the dataset by using images taken from different altitudes and angles in different environments (urban areas, highways, and parking lots). This dataset and annotations are given in YOLO format,and each image has a text file with bounding box mentioned as normalised cordinates. The dataset consists of several thousand images, each with multiple bounding boxes around cars and a field of view from the top, proving to be a useful tool for the development of aerial car detection technology. Sample image from Dataset 2 is shown in Figure 2.

Fig.2. Samples from Aerial View Car Detection Dataset

3.2 YOLO

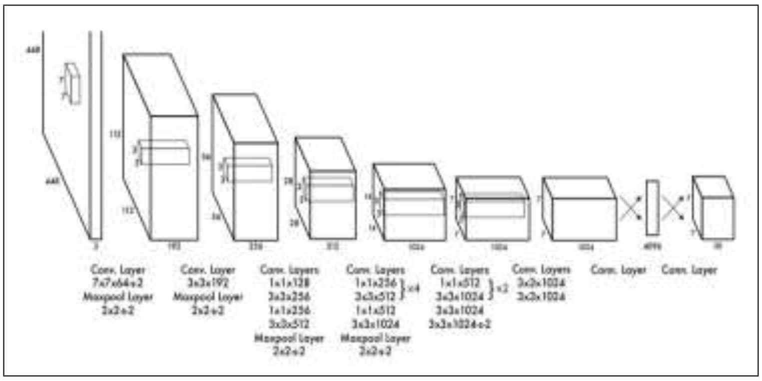

Fig.3. Architecture of YOLO algorithm (Redmon et al., 2017) [3]

You Only Look Once was developed and released by Joseph Redmon and Santosh Divvala in 2016 and is a leap forward concept in object detection. The application enables marking and detection across the frame in a single network pass, which means it both runs very quickly while remaining highly accurate.

YOLO is divided into a grid where each grid cell predicts a bounding box and its confidence score for some class. This is compared to other methods that need several distributed proposals before classifying an image. In order to greatly simplify object detection, YOLO predicts both bounding boxes which define object locations and object classes within the grid structure itself. YOLO predicts locations by regressing from spatial dimensions using convolutional neural networks (CNNs) while also predicting classes with logistic regression over categories.

Figure 3 illustrates the architecture of the YOLO algorithm. [3] The diagram highlights YOLO's single-stage object detection process, which divides the input image into a grid. Each grid cell predicts bounding boxes and their corresponding confidence scores for objects within that cell. This structure simplifies object detection by integrating both localization and classification tasks within a single convolutional neural network (CNN) pass, unlike other methods that require multiple proposal stages. The architecture also features anchor boxes to handle objects of different scales and aspect ratios, optimizing the model for both speed and accuracy in detecting various objects in an image.

A series of subsequent variations were produced, all with the same goal of improving its efficiency and functionality. The YOLO technique is based on the use of a personalized loss function as part of a Convolutional Neural Network structure that was created specifically to perform two tasks at once. The single-stage approach streamlines the process by removing the need for an object proposal step. To localize objects, standard size boxes (called anchors) that have been modified according to scales and aspect ratios of objects present in the training set are used to predict bounding boxes within grid cells.

3.3 Implementation

YOLOv3, YOLOv5, YOLOv7, and YOLOv8 algorithms are trained on parking Lot dataset. To train the model for Dataset 1, 186 parking lot images were used. A further 26 parking lot images were used for validation, while 20 were used for testing. Objects such as vehicles parked at different spaces defined by lines on both sides among other things are part of the dataset. Finally, the weather conditions in the picture library includes cloudy, rainy and sunny weather. Dataset 2 was segregated as 254 images used for training the model, 36 for validation and 18 for testing the model. The position of each object in the dataset is denoted by boxes which have been represented as "x1, y1, x2, y2, x3, y3, x4, y4" in the dataset.The implementation of YOLOv3, YOLOv5, YOLOv7 and YOLOv8 algorithms has been carried out on the Google Colaboratory platform with free high GPU (Graphics Processing Unit) support [6]. All models have been trained for 25 epochs, completing 25 full passes through the training dataset to adjust their parameters.

Fig.4. Sample Output using YOLOv8 Model on Dataset 1

Implementing YOLOv3, YOLOv5, and YOLOv8, which were directly released by Ultralytics, was facilitated by the availability of pre-trained models, training scripts, and comprehensive documentation provided by the Ultralytics. The implementation process began with the setup and installation of required dependencies such as PyTorch and NumPy. The dataset for object detection was then prepared, organized into training, validation, and testing sets, ensuring a balanced distribution of object classes and diverse scenarios. In the case of YOLOv7, which was not directly released by Ultralytics but implemented from online GitHub sources, a similar implementation process was followed with some differences. GitHub repositories containing the YOLOv7 implementation were identified based on community contributions and popularity within the computer vision community.

Once phases of control had been successfully concluded, twenty images were converted into output files from each of the four models, so that algorithms could be compared and evaluated. The output consisted of images processed using YOLOv3, YOLOv5, YOLOv7 and YOLOv8 for evaluation, and each algorithm was assessed individually against the twenty pictures mentioned earlier. Figure 4 shows output of an image from Dataset 1 where YOLOv8 model is used for object detection.

Results and Discussion

4.1Evaluation Metrics

During the evaluation of the results, four metrics were determined; evaluation time, precision (Eq. 1), recall (Eq. 2) and mAP. These metrics were calculated on the basis of the confusion matrix. The distribution of object detection is shown by the confusion matrix. [7] The confusion matrix has 4 parameters: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). TP: Model accurately predicts a positive data point. TN: model accurately predicts a negative data point. FP: model predicts a positive data point incorrectly. FN: Model wrongly predicts a negative data point. [8] Precision talks about the number of the correct detections made by the method; recall is the metric for correctly detected objects that actually exist.

$$Precision = TP/(TP + FP) \quad (1)$$

$$Recall = TP/(TP + FN) \quad (2)$$

Along with that, mean Average Precision (mAP) computed at an Intersection over Union (IoU) [9] threshold of 0.5, and across a range of IoU thresholds from 0.5 to 0.95, providing a broader evaluation of object detection performance by considering varying degrees of overlap between predicted and ground truth bounding boxes are calculated. Box Loss, Classification Loss (CLS) and Distribution Focal Loss (DFL) [10] both at Training and Validation phase are also used for evaluation.

The reason why we use precision, recall, and evaluation times as metrics to evaluate YOLO models is because they play a crucial role in assessing how well object detection algorithms perform. Precision tells us the ratio of true positive detections to all the positive detections made by the model, showing us the model's accuracy. On the other hand, recall measures the ratio of true positive detections to all the actual positive instances, indicating the model's ability to capture all the relevant objects.

The evaluation time shows how fast the models are in terms of performing calculations. A shorter time of model evaluation makes it possible for the model to carry out accurate object detection in real time crucial in the real life situations. Comparing the evaluation times of different models, we can assess the trade-offs between performance and computational efficiency.

4.2 Results

Recall and precision were utilized to assess the performance of YOLOv3, YOLOv5, YOLOv7, and YOLOv8 in parking spot detection. Additionally, mean Average Precision (mAP) at a threshold of 0.5 and mAP across a threshold range from 0.5 to 0.95 provided insights into detection performance under different confidence thresholds. Evaluation also included analysis of Box Loss, Classification (CLS) Loss, and Distributed Focal (DFL) Loss, both during training and validation of the model.

Figure 5 shows performance results of YOLOv3 model on Dataset 1. The precision and recall rates gradually increase, indicating better results with higher epochs. The mAP increases, while both box loss and classification loss go down, signifying improved model optimization and accuracy.

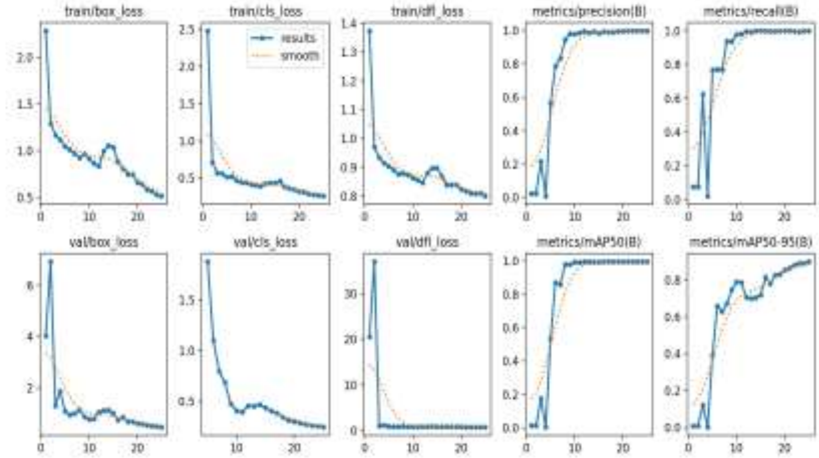

Fig.5. Results of YOLOv3 Model on Dataset 1

Figure 6 shows the evaluation metrics of the YOLOv5 model on Dataset 1. Compared to YOLOv3, this model has a much steeper and more stable curve of precision and recall, which is representative of stronger initial performance and better consistency. Higher mAP throughout training and decreasing box loss and classification loss during training are indicative of better convergence and object detection performance.

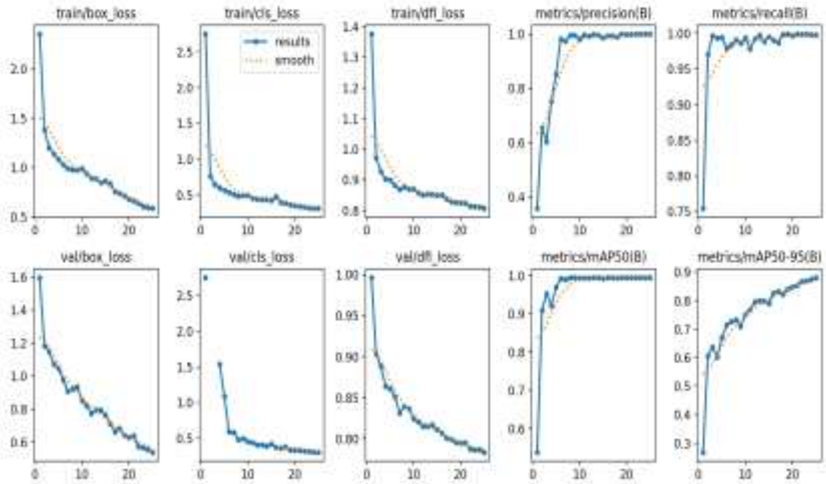

Fig.6. Results of YOLOv5 Model on Dataset 1

Figure 7 shows evaluation metrics on Dataset 1 for the YOLOv7 model. Initially, the precision and recall are low but they improve drastically. The mAP increases while box and classification loss decreases as number of epochs increase.

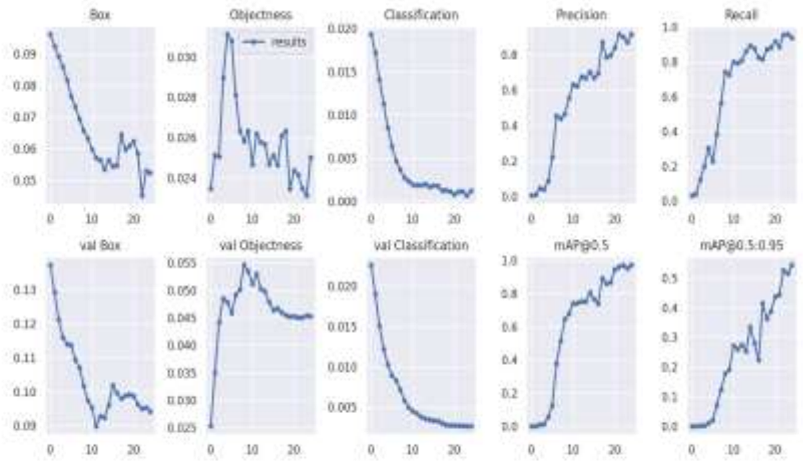

Fig.7. Results of YOLOv7 Model on Dataset 1

Figure 8 shows evaluation metrics for YOLOv8 model on Dataset 1. The rate of precision and recall are high across epochs, indicating superior performance and accuracy. Even when the number of epochs is less, better performance and accuracy is observed. mAP is significantly high compared with other models. Box loss and classification loss are at the lowest, which signifies its efficiency.

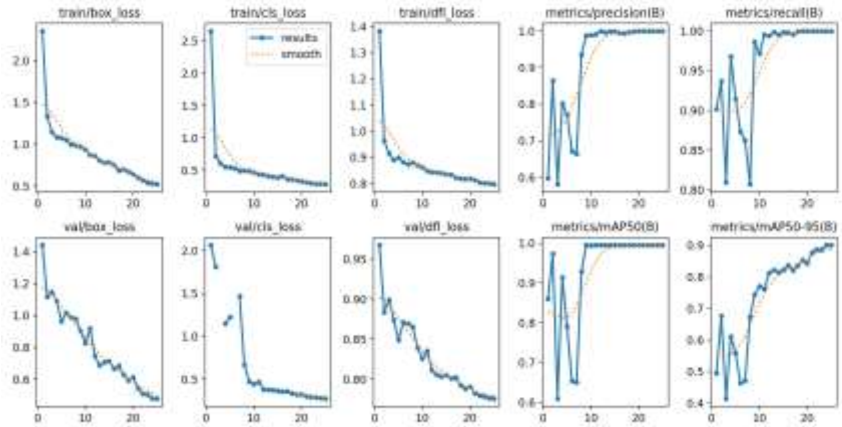

Fig.8. Results of YOLOv8 Model on Dataset 1

Figure 9 represents evaluation metrics of YOLOv3 on Dataset 2. The precision and recall curves get better with each epoch the same way it was with the case of Dataset 1. Meanwhile, mAP increases gradually, and box loss and classification loss are decreasing over epochs.

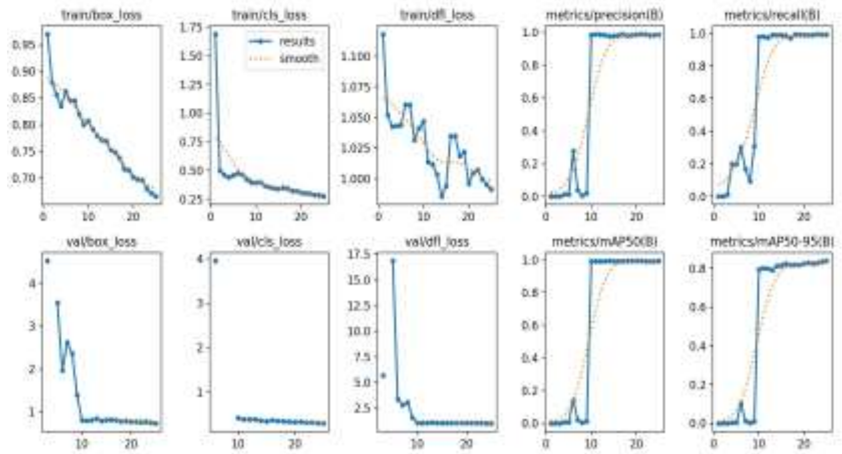

Fig.9. Results of YOLOv3 Model on Dataset 2

Figure 10 shows the evaluation metrics of YOLOv5 on Dataset 2. It shows higher and more stable precision and recall rates. Higher mAP across all epochs and low box loss and classification loss show consistency in its performance and model efficiency

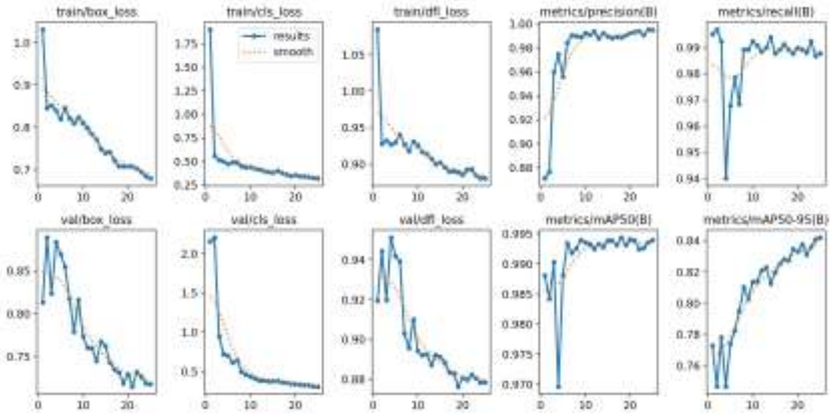

Fig.10. Results of YOLOv5 Model on Dataset 2

Figure 11 shows the evaluation metrics for the YOLOv7 model on Dataset 2. Precision and recall are both lower initially, but show significant improvement over the epochs. The mAP curve rises steadily, and the losses decrease over time.

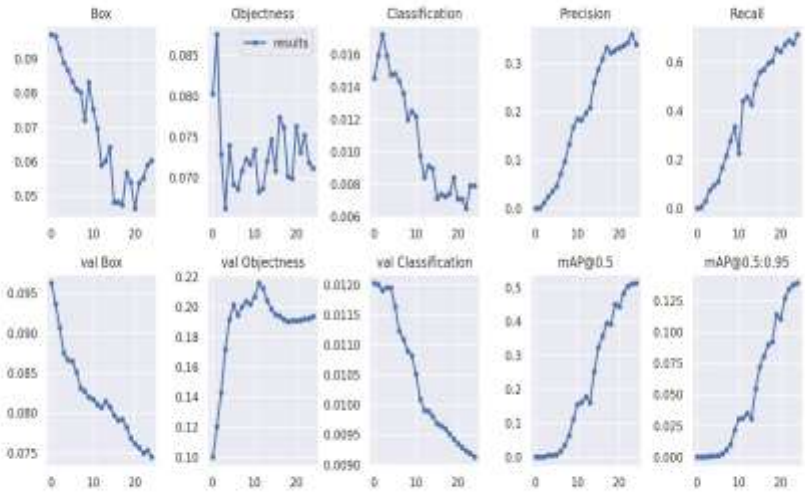

Fig.11. Results of YOLOv7 Model on Dataset 2

Figure 12 presents the results for the YOLOv8 model on Dataset 2. The precision and recall rates stay consistently high, similar to the performance on Dataset 1. The mAP remains significantly high, and the losses are minimal, proving the model's reliability and accuracy.

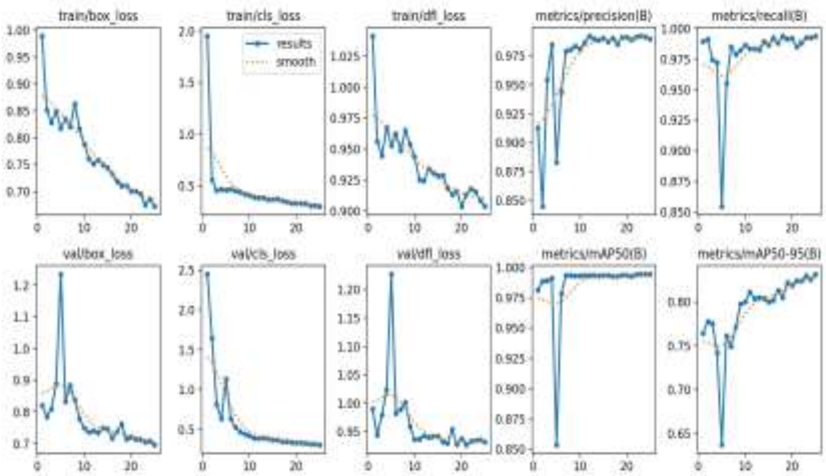

Fig.12. Results of YOLOv8 Model on Dataset 2

4.3 Discussion

Table 1. Evaluation Times of YOLO Models for 25 epochs in hours.

YOLO Version	Dataset 1	Dataset 2
YOLOv3	0.190	0.231
YOLOv5	0.109	0.111
YOLOv7	0.273	0.408
YOLOv8	0.111	0.108

The evaluation times for various YOLO models (YOLOv3, YOLOv5, YOLOv7, YOLOv8) across two datasets for 25 epochs are presented in Table 1. These times provides a measure of the computational efficiency of each model. YOLOv3 provided decent evaluation time, being outperfromed by YOLOv8 and YOLOv5 but considering it was one of the earliest version produced still performance being fairly decent. YOLOv5 and YOLOv8 are the most time-efficient models, with YOLOv5 outperforming YOLOv8 on Dataset 1, while YOLOv8 having and an edge over YOLOv5 on Dataset 2. In contrast, YOLOv7 had the longest evaluation times, requiring 0.273 hours for Dataset 1 and 0.408 hours for Dataset 2. Major reason behind this would be the lack of official version release of YOLOv7 making updates and tuning extremeley less frequent as compared to YOLOv5 and YOLOv8 which were launched and maintained by Ultralytics and are used more widely.

Table 2. Precision of YOLO Models.

YOLO Version	Dataset	Epoch 1	Epoch 5	Epoch 10	Epoch 15	Epoch 20	Epoch 25
YOLOv3	1	0.02455	0.56311	0.97876	0.98412	0.99619	0.99735
	2	0	0.01115	0.98287	0.97888	0.9852	0.98403
YOLOv5	1	0.3579	0.85111	0.98074	0.98511	0.99708	1
	2	0.8711	0.95594	0.99233	0.98948	0.99231	0.99497
YOLOv7	1	0	0.03559	0.1675	0.2592	0.3267	0.3389
	2	0.005127	0.08807	0.5546	0.7029	0.7952	0.9144
YOLOv8	1	0.59679	0.77009	0.98871	0.99793	0.99926	0.99933
	2	0.91278	0.88293	0.9807	0.99039	0.99101	0.98955

The precision and recall metrics for each model over 25 epochs are shown in Table 2 and Table 3 respectively, providing insights into their performance in object detection tasks. YOLOv8 consistently achieves the highest precision and recall across both datasets, particularly in the later epochs. For instance, by epoch 25 on Dataset 1, YOLOv8 achieves a precision of 0.99933 and a recall of 1.00000. Similarly, on Dataset 2, YOLOv8 attains a precision of 0.98955 and a recall of 0.99385.

Table 3. Recall of YOLO Models.

YOLO Version	Dataset	Epoch 1	Epoch 5	Epoch 10	Epoch 15	Epoch 20	Epoch 25
YOLOv3	1	0.07542	0.76681	0.97725	0.99777	0.9974	0.99937
	2	0	0.19802	0.9769	0.98845	0.9888	0.99175
YOLOv5	1	0.75456	0.9946	0.99358	0.99488	0.99698	0.99782
	2	0.99503	0.96779	0.99233	0.98773	0.98989	0.98773
YOLOv7	1	0	0.09235	0.3349	0.5091	0.6544	0.7127
	2	0.03093	0.3041	0.7268	0.8918	0.8807	0.9381
YOLOv8	1	0.9016	0.91437	0.97161	0.99863	0.99982	1
	2	0.98951	0.85457	0.98651	0.98651	0.99191	0.99385

YOLOv5 also performs exceptionally well, with precision and recall values closely matching those of YOLOv8. By epoch 25, YOLOv5 achieves a precision of 1.00000 and a recall of 0.99782 on Dataset 1, and a precision of 0.99497 and a recall of 0.98773 on Dataset 2.

YOLOv3 shows significant improvement over the epochs, especially in precision. Starting from a low precision of 0.02455 at epoch 1, YOLOv3 improves to 0.99735 by epoch 25 on Dataset 1. However, its initial precision and recall values are considerably lower compared to YOLOv5 and YOLOv8. For Dataset 2, YOLOv3 improves from a precision of 0.00000 at epoch 1 to 0.98403 by epoch 25, and from a recall of 0.00000 to 0.99175 over the same period.

YOLOv7 has the lowest initial precision and recall values. However, it shows significant improvement by epoch 25. For instance, in Dataset 1, the precision increases from 0.00000 at epoch 1 to 0.33890 at epoch 25, and the recall from0.00000 to 0.71270 during the same period. On the other hand, in Dataset 2, YOLOv7's precision rises from 0.005127 at epoch 1 to 0.91440 at epoch 25 while its recall goes up from 0.030930 to 0.93810.

This comparative study's findings demonstrate the strengths and weaknesses of each YOLO model. YOLOv8, introduced by Ultralytics in 2023 is the most efficient and effective model. It attained the highest precision and recall across both datasets while requiring minimal evaluation time. This shows YOLOv8, the latest version of the software, has been tailored for optimal performance as well as computational efficiency.

YOLOv5 is also a strong performer, with metrics almost at par to those of YOLOv8. Its slightly longer evaluation times compared to YOLOv8 on Dataset 2 are offset by its high accuracy, making it a reliable model for applications where both speed and accuracy are critical. Interestingly, Ultralytics actively maintains YOLOv5 as an open-source project with over 250 contributors and frequent improvements [13]. It is user-friendly for training, deployment, and implementation, but might soon be replaced by YOLOv8 over better optimisation and faster processing.

YOLOv3, released back in 2018, although starting with lower initial precision and recall, shows significant improvement over time. Therefore YOLOv3 may require more epochs to reach optimal performance levels. However, its lower starting values indicate that it might not be the best choice for tasks requiring immediate high accuracy.

YOLOv7, while showing the most significant improvement in precision and recall over the epochs, still lags the other models in terms of initial performance and overall evaluation times. Infrequent updates, relying mainly on community support makes it less suitable for applications where both speed and high accuracy are required from the start.

Conclusion

YOLOv3, YOLOv5, YOLOv7, and YOLOv8 methods were used to detect objects in the parking lot images used in this study. Parking Lot dataset which contains numerous classes of parking spot images was used. This study has revealed valuable insights about the strengths and weaknesses of each YOLO model variation when it comes to recognizing parking spots under different circumstances through thorough analysis and experimentation.

Our findings demonstrate that YOLO models provide effective and efficient solutions for parking space detection tasks. YOLOv3, one of the pioneers of YOLO models, has been outperformed by its successors and remains inaccurate and inefficient over fewer epochs. YOLOv5, despite breaking away from the traditional YOLO lineage, achieves remarkable performance thanks to its single-stage detector architecture and simplified design, making it one of the quickest and most accurate techniques available, backed by substantial community support. YOLOv7, driven by the community, underscores the collective contributions made by the computer vision field in expanding object detection capabilities, though it's not as fast as YOLOv5 and lacks proper documentation and ease of use. It still boasts decent precision and recall rates. YOLOv8, the newest release, features advanced mechanisms, including attention mechanisms and data augmentation strategies, increasing detection speed and maintaining high accuracy and robustness levels.

Identifying through comparison, this research evaluated different iterations of the YOLO model, finding strengths as well as areas needing improvement, among other things velocity, accuracy, and resource efficiency were assessed to install object detecting systems suited for parking lots.

Based on our findings we can conclude that, YOLOv8 stands out as the most efficient and effective model, combining high computational efficiency with superior performance metrics, making it the best choice for most object detection tasks. YOLOv5, while slightly less efficient in terms of evaluation times, is also highly accurate and a close second to YOLOv8. YOLOv3 and YOLOv7, despite improvements, are less competitive due to lower initial performance and longer evaluation times. Thus, for applications demanding both speed and accuracy, YOLOv8 is recommended as the top-performing model.

References

1. P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A Review of Yolo Algorithm Developments,
2. *Procedia Computer Science, vol. 199, 2022, pp. 1066-1073, ISSN 1877-0509,* https: //doi.org/:10.1016/j.procs.2022.01.135
3. M. Marek, Image-Based Parking Space Occupancy Classification: Dataset and baseline, arXiv (Cornell University), Jan. 2021, https://doi.org/10.48550/arxiv.2107.12207
4. M. E. Atik, Z. Duran, and R. Özgünlük, Comparison of YOLO Versions for ObjectDetection from Aerial Images, *International Journal of Environment and Geoinformatics*, vol. 9, no. 2, pp. 87–93, Jun. 2022, https://doi.org/10.30897/ijegeo.1010741
5. D. C. Wyld et al. (Eds), AIFZ, SIGPRO, NWCOM, CLSB, MLNLP, DTMN,BDIOT, CSITY - 2022, pp. 17-28, 2022. CS & IT - CSCP 2022, https://doi.org/10.5121/csit.2022.121602

6. S. Wang, Research towards Yolo-Series Algorithms: Comparison and analysis ofobject detection models for Real-Time UAV Applications, *Journal of Physics. Conference Series*, vol. 1948, no. 1, p. 012021, Jun. 2021, https://doi.org/10.1088/1742-6596/1948/1/012021
7. S. M. Thwin, S. J. Malebary, A. W. Abulfaraj, and H.-S. Park, AttentionBased Ensemble Network for Effective Breast Cancer Classification over Benchmarks, *Technologies*, vol. 12, no. 2, p. 16, 2024, https://doi.org/10.3390/ technologies12020016
8. F. Gönültaş, M. E. Atik, and Z. Duran, Extraction of Roof Planes from DifferentPoint Clouds Using RANSAC Algorithm, *International Journal of Environment and Geoinformatics*, vol. 7, pp. 165-171, 2020, https://doi.org/10.30897/ijegeo.715510
9. S. O. Atik, and C. Ipbuker, Integrating Convolutional Neural Network and Multiresolution Segmentation for Land Cover and Land Use Mapping Using Satellite Imagery, *Applied Sciences*, vol. 11, no. 5551, 2021, https://doi.org/10.3390/ app11125551
10. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, GeneralizedIntersection Over Union: A Metric and a Loss for Bounding Box Regression, in *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 658-666, https://doi.org/10.1109/CVPR.2019.00075
11. X. Li et al., Generalized focal loss: learning qualified and distributed boundingboxes for dense object detection, arXiv (Cornell University), Jan. 2020, https: doi.org/10.48550/arxiv.2006.04388
12. D. T. Nguyen, Parking Lot Database for YOLO, Kaggle, https://www.kaggle. com/datasets/duythanhng/parking-lot-database-for-yolo
13. J. Braun, Aerial View Car Detection for YOLOv5, Kaggle, https://www.kaggle. com/datasets/braunge/aerial-view-car-detection-for-yolov5
14. SuperAnnotate, The evolution of YOLO: Object detection algorithms, https:// www.superannotate.com/blog/yolo-object-detection