doi: 10.5937/str2600007Z

Research of the Impact of the Number of Epochs and Dataset Size on the Performance of the YOLO Algorithm

Snežana Zurovac [1)]
Jelena Jevremović [1)]
Ninko Miletić [1)]
Vasilija Joksimović [1)]

This study investigates the influence of dataset size and the number of training epochs on the performance of YOLOv8 object detection models. Three model variants: YOLOv8s, YOLOv8m, and YOLOv8xl, were trained using five datasets of increasing size (200 to 1000 images) and five epoch configurations ranging from 10 to 50 epochs. Model performance was evaluated using standard validation metrics, including mean Average Precision at an IoU threshold of 0.5-0.95 (mAP50-95), precision, and recall. The results indicate that increasing dataset size and training duration generally improves detection performance, with performance gains diminishing beyond approximately 40 training epochs. Model generalization capability was further assessed using a separate test set consisting of 32 manually annotated images, employing a similarity coefficient that integrates localization accuracy, classification correctness, and detection completeness. Additional analyses of true positives (TP), false positives (FP), and false negatives (FN) were conducted at confidence thresholds of 0.25, 0.50, and 0.85 to examine detection robustness under varying confidence constraints. Among the evaluated models, YOLOv8m provided the best balance between detection accuracy and computational efficiency. YOLOv8s, although less accurate, remained well suited for real-time inference and deployment on resource-constrained systems. In contrast, YOLOv8x did not exhibit consistent performance improvements relative to its increased computational complexity.

Key words: **YOLOv8, Object detection, Dataset size, Training epochs, Model performance.**

Introduction

YOLO (You Only Look Once) is a widely adopted and computationally efficient architecture for one-stage object detection. In contrast to two-stage methods, such as R-CNN and its variants, which decouple region proposal generation from object classification, YOLO formulates object detection as a single regression problem and performs inference in a single forward pass of a convolutional neural network. This unified design enables high inference speed while maintaining competitive detection accuracy. The fundamental principle of YOLO is to partition the input image into a fixed grid of cells, each of which is responsible for detecting objects whose centers lie within its spatial extent. For each grid cell, the network jointly predicts bounding box coordinates, a confidence score indicating the presence of an object, and class probability distributions. By jointly optimizing localization and classification in an end-to-end manner, YOLO effectively captures both spatial and semantic information within a single detection framework [1].

Owing to its architectural design and optimization strategies, YOLO is particularly well suited for applications that require an effective balance between detection speed and accuracy, rendering it highly suitable for real-time systems and resource-constrained embedded platforms. The training process of a machine learning model is inherently iterative and is primarily governed by factors such as the size of the training dataset and the number of training epochs. An epoch is defined as one complete pass of the entire training dataset through the neural network, encompassing both forward and backward propagation. The number of epochs plays a critical role in determining the degree of model convergence, as well as in mitigating the risks of overfitting and underfitting [2, 3].

The dataset employed for training, validation, and testing comprises images with accurately annotated objects and their corresponding class labels. Both the size and diversity of the dataset are critical factors for achieving robust model generalization. The performance of the object detection models is assessed using a validation dataset and standard evaluation metrics, including mean Average Precision at an IoU threshold of 0.5-0.95 (mAP50-0.95), precision, and recall.

To achieve an optimal balance between detection accuracy, training time, and computational efficiency, particularly in applications where real-time performance is essential, it is necessary to perform a comparative analysis of the relationship between dataset size and the number of training epochs. Such an analysis enables the identification of training configurations that yield the best overall performance while minimizing unnecessary computational overhead.

This paper is organized into four chapters. Following the introduction, the second chapter presents a review of the relevant literature concerning the impact of training epoch

[1)] Military Technical Institute, Ratka Resanovića 1, 11000 Belgrade, Serbia
Correspondence to: Snežana Zurovac, e-mail: zurovac@medianis.net

count and dataset size. The third chapter describes the adopted methodology and reports the experimental results, followed by a discussion and concluding remarks.

Related Work

Previous research in the field of deep learning–based object detection has identified training duration and dataset characteristics as key factors influencing model performance. However, these factors have predominantly been examined in isolation, while their combined effects remain insufficiently explored, particularly for modern one-stage detectors.

The number of training epochs plays a critical role in model convergence and generalization. Foundational studies in deep learning have shown that insufficient training leads to underfitting, whereas excessive training may result in overfitting or performance saturation, without corresponding gains in generalization performance [4]. In the context of object detection, early YOLO-based studies demonstrated that increasing the number of training iterations improves detection accuracy up to a convergence point, after which additional training yields diminishing returns while significantly increasing computational cost [5]. Similar behavior has been observed in later YOLO architectures, where extended training beyond optimal convergence does not consistently translate into improved detection performance, despite higher model capacity [6]. Furthermore, theoretical and empirical analyses have shown that deep neural networks are capable of fitting training data perfectly given sufficient training epochs, yet such behavior does not necessarily correlate with improved generalization [7].

In parallel, numerous studies have emphasized the importance of dataset size and diversity in achieving robust object detection performance. Benchmark evaluations have shown that larger and more diverse datasets lead to improved localization accuracy and reduced false detections, particularly in complex visual scenes [8]. Subsequent large-scale empirical studies demonstrated that increasing dataset size often yields greater performance gains than increasing model complexity, especially when model capacity is not the primary limiting factor [9]. These findings are particularly relevant for practical detection tasks, where data collection and annotation are resource-intensive and model complexity must be carefully balanced against computational constraints.

Despite these established findings, existing studies typically assume fixed training schedules or sufficiently large datasets, and do not systematically analyze how dataset size interacts with training duration across multiple model variants. In particular, widely adopted YOLO training pipelines focus on architectural optimization and data augmentation strategies, while the joint influence of dataset scale and epoch count is largely treated implicitly [6]. Moreover, most evaluations rely exclusively on standard validation metrics, such as mAP, precision, and recall, without explicitly assessing generalization performance on independent test sets or incorporating complementary similarity-based measures.

Motivated by these limitations, this work provides a comprehensive and controlled analysis of the combined effects of dataset size and training epoch count on multiple YOLOv8 model variants of differing complexity. By evaluating performance saturation, generalization behavior, and accuracy–efficiency trade-offs under identical experimental conditions, the study aims to address an important gap in the current object detection literature.

Methodology

The YOLOv8 architecture was employed to investigate the effects of dataset size and the number of training epochs on model performance. All experiments were conducted using the Ultralytics library implemented within the PyTorch framework. Three YOLOv8 model variants—YOLOv8s, YOLOv8m, and YOLOv8x—were trained and evaluated. These variants differ primarily in network depth and width, resulting in an increasing number of trainable parameters and computational complexity. Specifically, YOLOv8s contains 11.2 million parameters, YOLOv8m 25.9 million parameters, and YOLOv8x, 68 million parameters [10].

The primary objective of this research is to identify, based on the obtained experimental results, the optimal model for real-time deployment. The selected model must provide high detection accuracy with a minimal rate of incorrect predictions. Furthermore, the selected model must meet minimal hardware requirements and low energy consumption constraints to ensure suitable real-time deployment on unmanned aerial vehicles. Due to the required model characteristics for real-time operation, this research was conducted. All three models were evaluated under identical conditions: they were trained on the same dataset, using the same number of training iterations, and ultimately evaluated on the same set of images, ensuring that the obtained results are directly comparable.

The dataset consisted of images extracted from video recordings captured under real-world field conditions, representative of the model's intended operational environment. All images were manually annotated using the Label Studio annotation tool. From the complete dataset of 1,000 annotated images, five training subsets were constructed, containing 200, 400, 600, 800, and 1,000 images, respectively. Each YOLOv8 model variant was trained using five different epoch configurations—10, 20, 30, 40, and 50 epochs—resulting in a comprehensive evaluation of the combined influence of dataset size and training duration on detection performance.

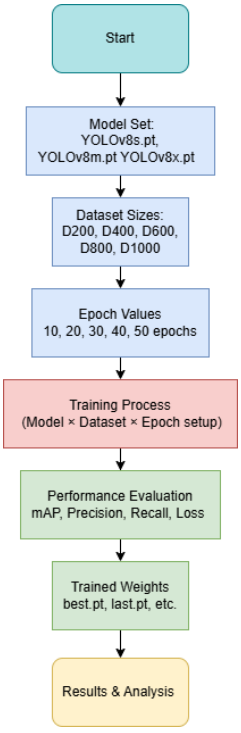

Figure1. Training workflow for the YOLOv8 models

The research methodology, that is, the workflow of the study, is illustrated in Figure 1. The models were trained using the loss function defined within the YOLOv8 architecture, which comprises three components: bounding box regression loss (bbox loss) for penalizing localization errors, classification loss (cls loss) for object class prediction, and Distribution Focal Loss (DFL) to enable more precise bounding box regression.

The training process employed stochastic gradient descent (SGD) as the optimization algorithm, with a momentum of 0.937 and a weight decay of 0.0005. The initial learning rate was set to 0.01, and a batch size of 16 was used. During training, the dataset was split into training and validation subsets. Model performance was evaluated on the validation set using standard object detection metrics, including mean Average Precision at an IoU threshold of 0.5 (mAP50), precision, and recall. These metrics facilitated a comprehensive comparison of performance across different dataset sizes, numbers of training epochs, and YOLOv8 model variants.

Upon completion of training for each model variant (YOLOv8s, YOLOv8m, and YOLOv8xl), the model achieving the highest mAP50 score was selected for each dataset size (200, 400, 600, 800, and 1,000 images).

To further assess generalization performance, a separate evaluation set consisting of 32 images not included in the training or validation process was prepared. These images were manually annotated to enable direct comparison with ground-truth labels. A similarity coefficient was defined to quantify the agreement between the predicted labels generated by the trained models and the manually annotated ground-truth labels. This coefficient was computed individually for each evaluation image, and the average similarity coefficient was subsequently calculated for each examined model. In addition, for each image, the total number of predicted objects was compared with the ground-truth annotations, and the numbers of true positives (TP), false positives (FP), and false negatives (FN) were determined. The results present, for each model, the total number of ground-truth objects across all 32 manually annotated images, as well as how many of these objects were correctly detected, incorrectly detected, or missed by the model.

The Following equipment was used for the experiment
- GPU: NVIDIA RTX A4500 (20 GB VRAM)
- CPU: AMD EPYC 7543P (32 cores)
- RAM: 256 GB
- Operating System: Windows Server 2019
- Environment: Python 3.11, CUDA 12.9

Results and Analysis

Training of all three YOLOv8 models (YOLOv8s, YOLOv8m, and YOLOv8x) was conducted using the Ultralytics framework, which performs automatic validation after each training epoch. For each dataset size, the validation subset was used to compute key performance metrics, including mean Average Precision at an IoU threshold of 0.5 (mAP50), precision, and recall. The validation results for all model–dataset–epoch configurations are presented in the following sections using tables and figures, which provide the most effective formats for illustrating and comparing the experimental outcomes.

For clarity and completeness, the results are reported separately for each of the three model variants. Performance trends, metric variations across different epoch configurations, and the influence of dataset size are shown independently for YOLOv8s, YOLOv8m, and YOLOv8x, facilitating a direct and systematic comparison of their behavior under identical experimental conditions.

4.1 Training Results of the YOLOv8s Model

This section presents the training performance of the YOLOv8s model, showing how mAP50-95, precision, and recall vary across different dataset sizes and epoch configurations. The corresponding figures are provided below.

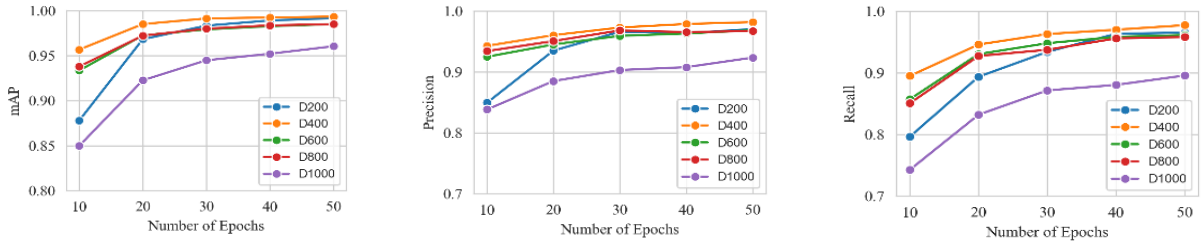

Figure 2. Validation curves for the YOLOv8s model (mAP50-95, precision, recall)

Figure 2 illustrates the validation performance of the YOLOv8s model across five training epoch configurations (10, 20, 30, 40, and 50 epochs) and five dataset sizes (D200–D1000). The three subplots depict the variations in mAP50, precision, and recall, respectively.

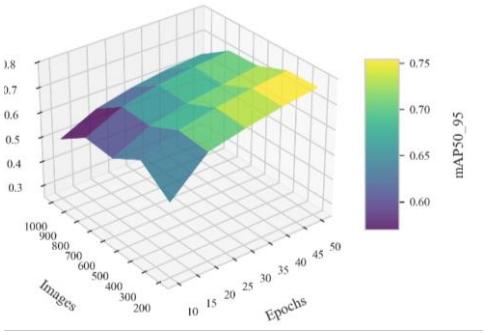

Figure 3. Shows 3D plot mAP50-95 for the YOLOv8m Model

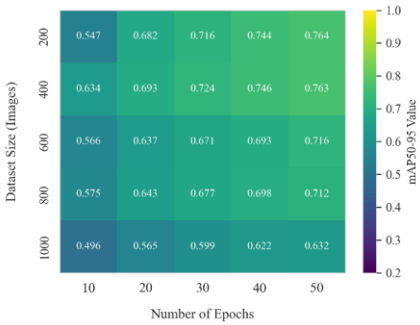

Figure 4. Heatmap of mAP50-95 for Dataset Size vs. Number of Epochs(YOLOv8s Model)

The 3D surface plot in Figure 3 illustrates the variation of mAP50 as a function of both dataset size and the number of training epochs. Complementarily, the heatmap in Figure 4 provides a grid-based visualization of mAP50 across all dataset size–epoch combinations, facilitating a clearer comparison of performance trends.

4.2 Training Results of the YOLOv8m Model

The next subsection presents the results obtained for the YOLOv8m model, with the corresponding figures provided below for reference.

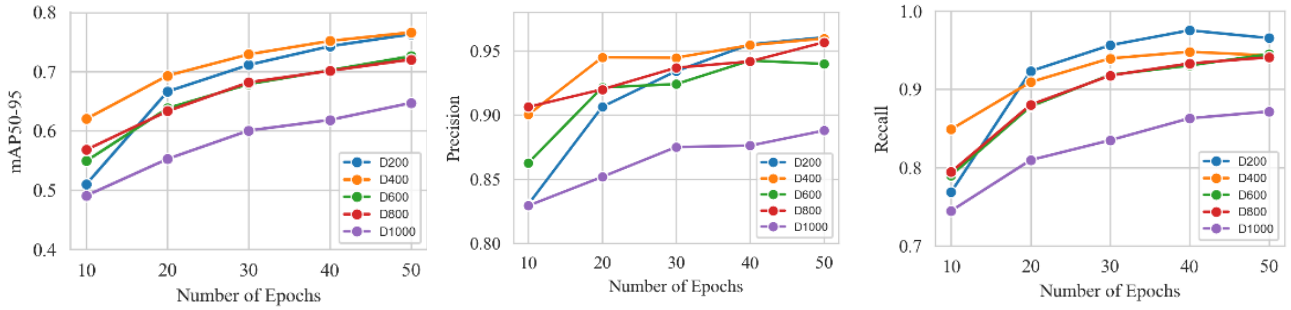

Figure 5. Validation curves for the YOLOv8m model (mAP50-95, precision, recall)

Figure 5 illustrates the validation performance of the YOLOv8m model across five training epoch configurations (10, 20, 30, 40, and 50 epochs) and five dataset sizes (D200, D400, D600, D800, and D1000). The three subplots depict the variations in mAP50, precision, and recall, respectively.

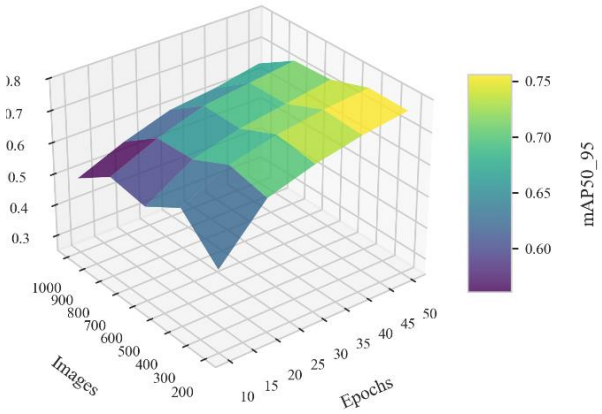

Figure 6. Shows 3D plot mAP50 for the YOLOv8m Model

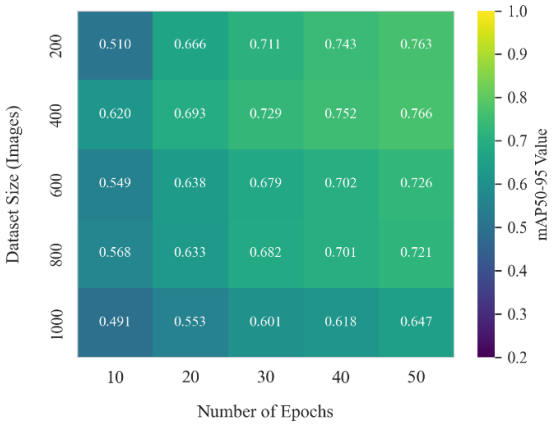

Figure 7. Shows mAP50 for Dataset Size vs. Number of Epochs (YOLOv8m Model)

Figures 6 and 7 provide a summarized view of the variation in mAP50 as a function of dataset size and the number of training epochs. The overall trends observed for the YOLOv8m model are consistent with those previously reported for YOLOv8s.

4.3 Training Results of the YOLOv8x Model

The YOLOv8x model was evaluated using the same experimental setup as the previously described models. The following figures present the corresponding training and validation results.

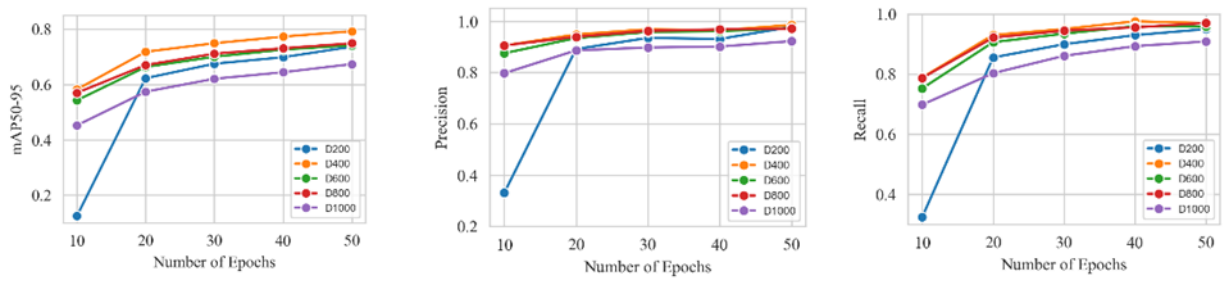

Figure 8. Validation curves for the YOLOv8x model (mAP50-95, precision, recall)

Figure 8 illustrates the variation of validation metrics (mAP50, precision, and recall) for YOLOv8x across all epochs and dataset configurations.

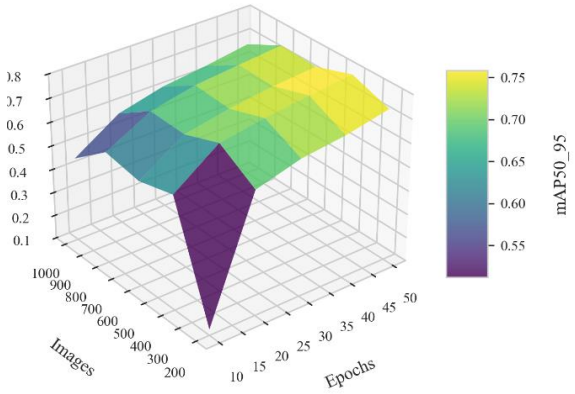

Figure 9. Shows 3D plot mAP50-95 for the YOLOv8x Model

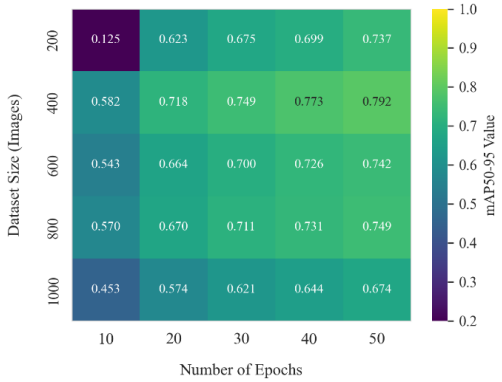

Figure 10. Shows mAP50-95 for Dataset Size vs. Number of Epochs (YOLOv8x Model)

Figures 9 and 10 present the mAP50 response surface and heatmap for the YOLOv8x model. The observed trends are consistent with those of the previous models, showing progressively higher performance with increasing dataset size and a greater number of training epochs.

Evaluation of Trained Models on an External Image Set

Upon completion of the training phase for all three YOLOv8 models (YOLOv8s, YOLOv8m, and YOLOv8x), the model achieving the highest mAP50 score from each architecture was selected for further evaluation. These best-performing models were tested on an independent dataset comprising 32 manually annotated images, which were not used during training or validation. This evaluation aimed to assess the generalization capability of the selected models under conditions representative of real-world deployment.

The following images show the predictions performed on the images from the evaluation data set, with conf=0.25.

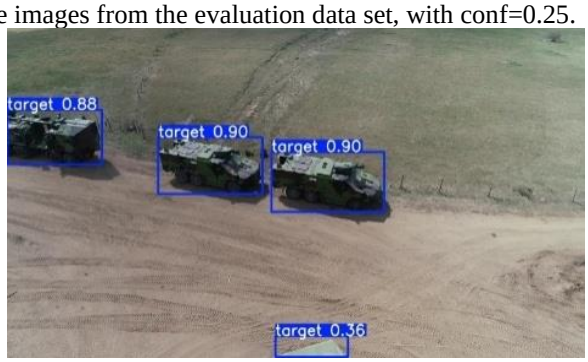

Figure 11. Example of prediction results obtained using the YOLOv8s model

Figure 12. Example of prediction results obtained using the YOLOv8m model

Figure 13. Example of prediction results obtained using the YOLOv8x model

To systematically quantify the agreement between the predicted detections and ground-truth annotations, a similarity coefficient was defined. This metric jointly captures localization accuracy, classification correctness, and completeness of detections. For each ground-truth object, the prediction with the highest IoU is selected as a match. Each matched pair receives a point value computed as:

$$\text{points} = 0.6 \cdot \text{IoU} + C$$

where the class-matching term C is defined as:

$$C = \begin{cases} 0.4 & \text{if the predicted class} \\ 0 & \text{otherwise} \end{cases}$$

This formulation ensures that bounding box alignment contributes up to 60% of the score, while correct classification contributes an additional fixed 40% [11].

The total similarity index for an image is obtained by summing all points from matched pairs and dividing by the total number of detection events that must be evaluated. This includes: matched pairs, unmatched ground-truth objects (false negatives), unmatched predictions (false positives).

$$\text{Similarity} = \frac{\sum \text{points}}{\text{matched pairs} + \text{false negatives} + \text{false positives}}$$

A similarity score close to 1 indicates accurate, correctly classified, and complete detections, whereas a value near 0 reflects misalignment, misclassification, or missed/extra detections [12].

In addition to the similarity coefficient, the evaluation included counting the number of true positives (TP), false positives (FP), and false negatives (FN) for each model. These measures provide a comprehensive overview of how accurately each model identifies objects (TP), how often it produces incorrect detections (FP), and how many objects it fails to detect (FN) [13].

The evaluation results for all three models are presented in the Tables 1 to 3 and Figures 14 to 18. Predictions were generated using three confidence thresholds: 0.25, 0.50, and 0.85, enabling analysis of model behavior under different detection strictness levels.

Table 1. Evaluation Results of the YOLOv8s Model at Confidence Thresholds 0.25, 0.50, and 0.85

YOLOv8s															
Data set	**Conf 0.25**					**Conf 0.5**					**Conf 0.85**				
	Sim.indx	**TP**	**FP**	**FN**	**Time**	**Sim.indx**	**TP**	**FP**	**FN**	**Time**	**Sim.indx**	**TP**	**FP**	**FN**	**Time**
200	0.5132	34	34	85	34.26	0.5446	29	13	90	34.61	0.4812	5	0	144	34.98
400	0.5175	45	30	74	34.57	0.5975	40	15	79	34.31	0.5492	10	1	109	35.07
600	0.5843	50	20	69	34.39	0.7059	46	10	73	34.39	0.575	12	0	107	34.44
800	**0.6703**	56	18	63	34.68	**0.6904**	50	5	69	34.83	**0.6743**	11	0	108	34.78
1000	0.6603	64	16	55	34.37	0.6232	54	9	65	34.39	0.6676	12	0	107	34.78

Table 2. Evaluation Results of the YOLOv8m Model at Confidence Thresholds 0.25, 0.50, and 0.85

YOLOv8m															
Data set	**Conf 0.25**					**Conf 0.5**					**Conf 0.85**				
	Sim.indx	**TP**	**FP**	**FN**	**Time**	**Sim.indx**	**TP**	**FP**	**FN**	**Time**	**Sim.indx**	**TP**	**FP**	**FN**	**Time**
200	0.4776	35	36	84	35.82	0.6181	24	15	95	36.32	0.5554	6	1	113	36.62
400	0.6152	59	28	60	36.18	0.5972	47	15	72	36.35	0.6072	11	1	108	35.96
600	0.5639	50	18	69	36.06	0.6398	43	9	76	36.09	0.5476	11	3	108	35.87
800	0.6237	64	22	55	36.09	0.6386	60	11	59	36.16	0.5972	14	1	105	36.08
1000	**0.7077**	77	22	42	35.98	**0.6406**	62	11	57	36.16	**0.819**	10	0	109	36.83

Table 3. Evaluation Results of the YOLOv8xl Model at Confidence Thresholds 0.25, 0.50, and 0.85

YOLOv8xl															
Data set	**Conf 0.25**					**Conf 0.5**					**Conf 0.85**				
	Sim.indx	**TP**	**FP**	**FN**	**Time**	**Sim.indx**	**TP**	**FP**	**FN**	**Time**	**Sim.indx**	**TP**	**FP**	**FN**	**Time**
200	0.4266	28	36	91	40.82	0.5947	20	14	99	40.84	0.6023	6	0	113	40.61
400	0.5826	41	16	78	41.21	0.6583	34	9	85	40.77	0.5017	8	1	111	41.3
600	0.5809	55	20	64	41.1	0.5859	45	9	74	40.45	0.5641	12	0	107	41.83
800	**0.66**	59	18	60	40.6	**0.6549**	52	8	67	40.38	0.5648	14	1	105	41.92
1000	0.628	58	17	61	40.89	0.6189	48	7	71	41.99	**0.7022**	10	0	109	41.74

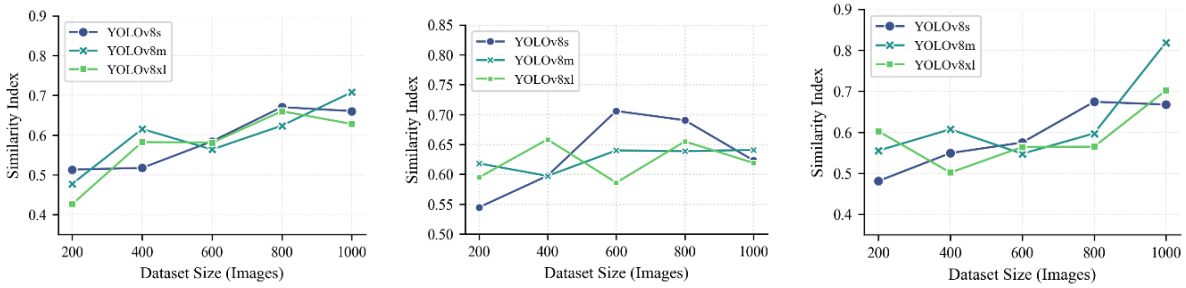

Figure 14. Similarity index vs Dataset at confidence thresholds 0.25, 0.50, and 0.85

Figure 14 (Similarity Index vs Dataset at confidence 0.25, 0.50, 0.85) shows how detection quality changes with dataset size under different confidence thresholds.

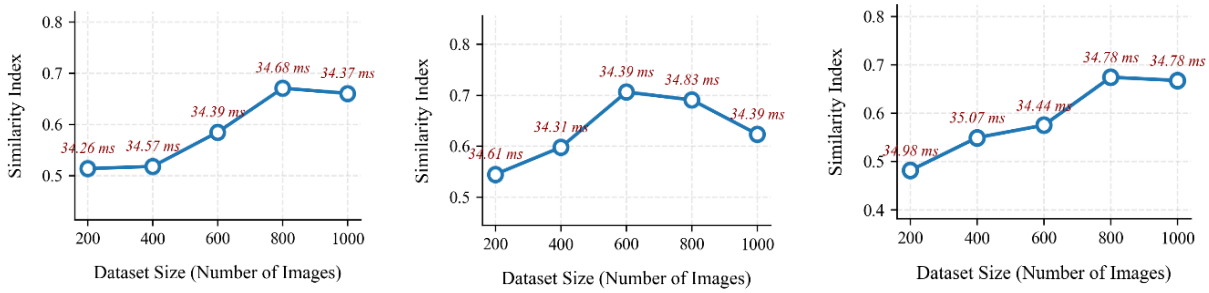

Figure 15. Similarity index vs Dataset vs Inference Time at confidence thresholds 0.25, 0.50, and 0.85 for the YOLOv8s

Figure 15 (YOLOv8s: Similarity Index vs Dataset vs Inference Time) shows how inference time changes with dataset size under different confidence thresholds for YOLOv8s model.

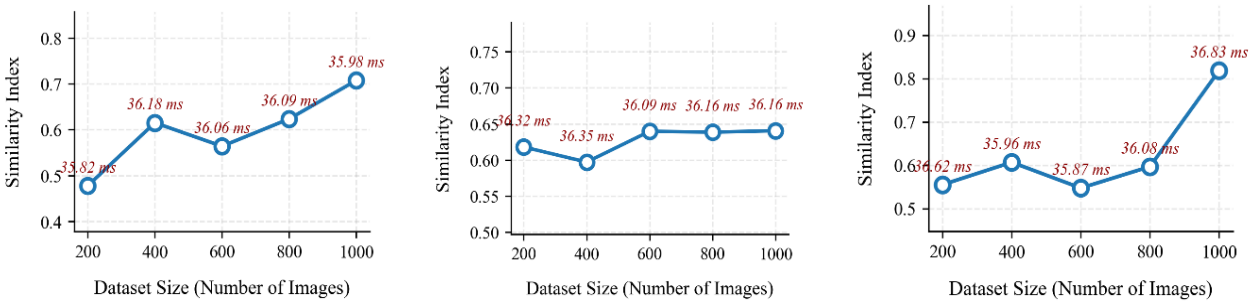

Figure 16. Similarity index vs Dataset vs Inference Time at confidence thresholds 0.25, 0.50, and 0.85 for the YOLOv8m

Figure 16 (YOLOv8m: Similarity Indes vs Dataset vs Inference Time) shows how inference time changes with dataset size under different confidence thresholds for YOLOv8m model.

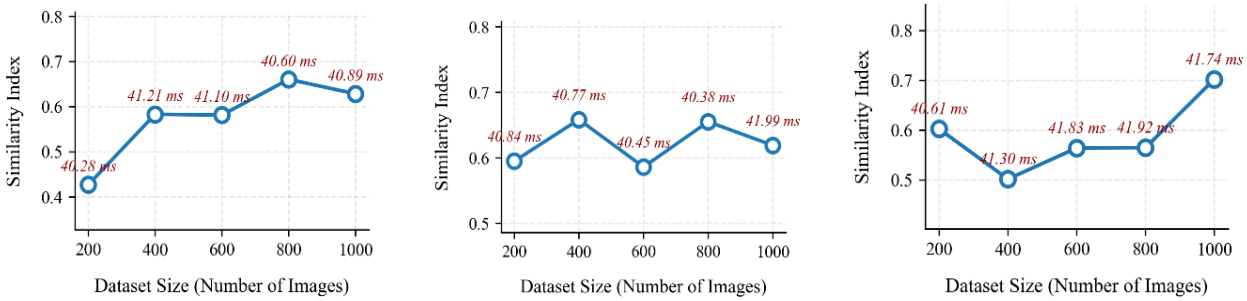

Figure 17. Similarity index vs Dataset vs Inference Time at confidence thresholds 0.25, 0.50, and 0.85 for the YOLOv8x

Figure 17 (YOLOv8l: Similarity Indes vs Dataset vs Inference Time) shows how inference time changes with dataset size under different confidence thresholds for YOLOv8x model.

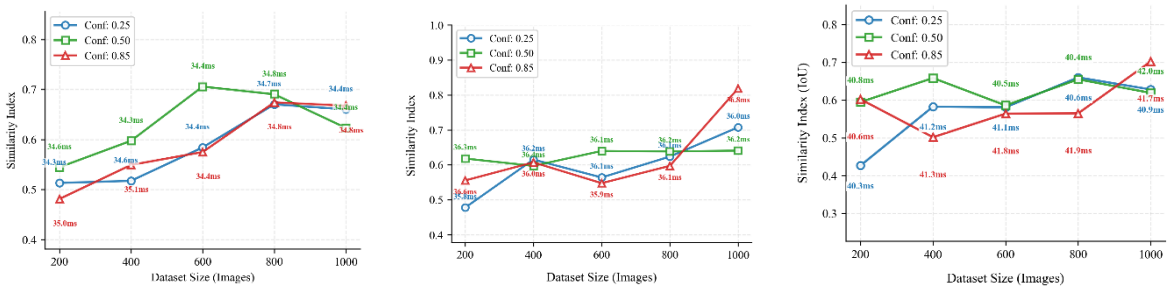

Figure 18. Similarity index vs Dataset vs Inference Time at confidence thresholds 0.25, 0.50, and 0.85 (YOLOv8s, YOLOv8m, YOLOv8x)

Figure 18 provides a joint comparison of similarity index, dataset size, confidence threshold, and inference time for the three YOLOv8 variants.

The experimental results presented in this study provide a comprehensive overview of how dataset size and the number of training epochs influence the performance of three YOLOv8 architectures: YOLOv8s, YOLOv8m, and YOLOv8x. The diagrams and tables in Section 4 clearly indicate a consistent and systematic trend across all models: larger datasets and longer training durations lead to improved performance, reflected through increases in mAP50-95, precision, and recall.

The validation curves (Figures 2, 5, and 8) reveal that performance grows steadily from 10 to approximately 40 epochs, after which the gains begin to plateau. This saturation effect is visible across all dataset sizes, suggesting that beyond 40–50 epochs, additional training provides minimal benefit. Similarly, mAP50-95 heatmaps and surface plots (Figures 3–4, 6–7, and 9–10) show a clear gradient where performance rises with dataset size, particularly between 600 and 1000 images.

In Section 5, which presents the evaluation of the trained models, it can be clearly observed that dataset size has a significant impact on the performance of the trained models. As shown in Table 1, an increase in the number of training samples leads to a consistent improvement in the similarity coefficient across all evaluated YOLOv8 variants.

For example, for the YOLOv8s model at a confidence threshold of 0.85, the similarity coefficient increases from 0.4812 for a dataset size of 200 samples to 0.6676 when the dataset size is increased to 1000 samples. A similar trend is observed for the YOLOv8m model, where the similarity coefficient rises from 0.5554 to 0.819 at the same confidence threshold, indicating a substantial performance gain with increased training data.

The most important conclusion arises from the evaluation performed on a separate set of 32 images that were not used during the training process. This evaluation demonstrates that the models trained on the largest dataset (1000 images) achieved the best overall performance, indicating superior generalization capability despite the observed validation precision trends.

As shown in Tables 1, 2, and 3, the number of true positive detections increases with the growth of the training dataset, while the numbers of false positives and false negatives decrease accordingly.

The longest inference times are observed for the YOLOv8x model, which is expected given the higher complexity of its architecture and the larger number of parameters. In contrast, significantly shorter inference times approximately 12% lower are required for the YOLOv8m and YOLOv8s models.

Conclusion

Based on the results obtained throughout the training and validation stages, a set of general conclusions can be summarized for the three evaluated models.

The YOLOv8s model demonstrates stable performance and the shortest inference time, with an execution time of approximately 34–35 ms, making it suitable for real-time applications and environments with limited computational resources. Although it achieves slightly lower maximum similarity index values compared to larger models, YOLOv8s represents a favorable compromise between detection accuracy and inference speed.

YOLOv8m achieves the best overall performance in most of the evaluated configurations, particularly for larger dataset sizes (800 and 1000 samples), where it attains the highest similarity index values and the largest number of true detections. The increased model complexity relative to YOLOv8s results in a moderate increase in inference time (approximately 36 ms), while providing a substantial improvement in detection accuracy.

YOLOv8x, as the most complex architecture, does not exhibit consistently superior performance compared to YOLOv8m, despite its significantly longer inference time (approximately 41 ms). In certain configurations, especially when trained on smaller datasets, a degradation in performance is observed, indicating potential overfitting and reduced efficiency under limited data conditions.

In conclusion, the experimental results indicate that YOLOv8m represents the most suitable model for the considered application, providing the highest overall detection reliability while maintaining moderate computational requirements. YOLOv8s, although exhibiting lower detection accuracy, remains a viable choice for real-time and resource-constrained applications due to its shorter inference time and lower hardware demands. In contrast, YOLOv8x does not demonstrate consistent performance improvements relative to its increased architectural complexity and computational cost.

Future work may extend this study by incorporating multi-class datasets, evaluating robustness under challenging environmental conditions, and benchmarking performance on embedded hardware platforms such as NVIDIA Jetson or other edge-AI systems.

References

[1] J. Terven, D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS", *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[2] T.-W. Sung, J. Li, C.-Y. Lee, Q. Fang, "Improvement of YOLOv8 Object Detection Based on Lightweight Neck Model for Complex Images", Image Analysis & Stereology, vol. 44, pp. 69–86, 2025, doi: 10.5566/ias.3514.

[3] Z. Fan, Z. Qin, W. Liu, M. Chen, Z. Qiu, "SS-YOLOv8: A Lightweight Algorithm for Surface Litter Detection", Applied Sciences, vol. 14, no. 20, art. no. 9283, 2024, doi: 10.3390/app14209283.

[4] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[6] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[7] C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, "Understanding Deep Learning Requires Rethinking Generalization," ICLR, 2017.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[9] C. Sun, A. Shrivastava, S. Singh and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in the Deep Learning Era," in Proc. IEEE Int. Conf. Computer Vision (ICCV), Venice, Italy, 2017, pp. 843–852, doi: 10.1109/ICCV.2017.97.

[10] Ultralytics, "Explore Ultralytics YOLOv8 — YOLOv8 Model Overview and Tasks," *Ultralytics YOLO Docs*, 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/. [Accessed: Dec. 30, 2025].

[11] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), pp. 303–338.

[12] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788.

[13] Hoiem, D., Chodpathumwan, Y., & Dai, Q. (2012). Diagnosing Error in Object Detectors. In European Conference on Computer Vision (ECCV), pp. 340–353.

Received: 12.01.2026.
Accepted: 20.02.2026.

Istraživanje uticaja broja epoha i veličine skupa podataka na performanse YOLO algoritma

Ova studija ispituje uticaj veličine skupa podataka i broja epoha treniranja na performanse modela za detekciju objekata YOLOv8. Tri varijante modela — YOLOv8s, YOLOv8m i YOLOv8xl — trenirane su korišćenjem pet skupova podataka rastuće veličine (od 200 do 1000 slika) i pet konfiguracija epoha u rasponu od 10 do 50. Performanse modela su evaluirane pomoću standardnih validacionih metrika, uključujući srednju prosečnu preciznost pri IoU pragu od 0,5–0,95 (mAP50–95), preciznost i odziv.

Rezultati pokazuju da povećanje veličine skupa podataka i dužine treniranja generalno poboljšava performanse detekcije, pri čemu se dobit u performansama smanjuje nakon približno 40 epoha treniranja. Sposobnost generalizacije modela dodatno je procenjena korišćenjem zasebnog test skupa koji se sastoji od 32 ručno anotirane slike, uz primenu koeficijenta sličnosti koji integriše tačnost lokalizacije, ispravnost klasifikacije i potpunost detekcije.

Dodatne analize tačnih pozitivnih (TP), lažno pozitivnih (FP) i lažno negativnih (FN) rezultata sprovedene su pri pragovima pouzdanosti od 0,25, 0,50 i 0,85, kako bi se ispitala robusnost detekcije pod različitim ograničenjima pouzdanosti. Među evaluiranim modelima, YOLOv8m je pokazao najbolji balans između tačnosti detekcije i računarske efikasnosti. YOLOv8s, iako manje tačan, ostaje pogodan za izvođenje u realnom vremenu i primenu na sistemima sa ograničenim resursima. Nasuprot tome, YOLOv8x nije pokazao konzistentna poboljšanja performansi u odnosu na povećanu računarsku složenost.

Ključne reči: **YOLOv8, detekcija objekata, veličina skupa podataka, epoha treniranja, performanse modela.**