Improved YOLOv8 Algorithms for Object Detection

Elvianto Dwi Hartono 1
Universitas 17 Agustus 1945 Surabaya, elvianto.evh@untag-sby.ac.id:
Bagus Hardiansyah 1
Universitas 17 Agustus 1945 Surabaya, bagushardiansyah@untag-sby.ac.id:

Abstract
Current Over the past decade, deep neural networks have progressed rapidly, with computer vision consistently achieving new performance milestones and becoming a part of daily life. In target detection, the YOLO model stands out as a popular real-time detection algorithm, known for its speed, efficiency, and accuracy. This study focuses on enhancing the latest YOLOv8 model to improve small object detection and compares it with other YOLO versions. Using YOLOv8 as a foundational deep learning algorithm, we introduced several optimizations, including redefining the detection head, narrowing its perceptual field, and increasing the number of detection heads to better capture fine details in small objects. We then compared this optimized YOLOv8 model with established YOLO models, such as YOLOv3 and YOLOv5n. The experimental results indicate that our optimized model achieves higher accuracy in detecting small objects. This research offers a promising approach for small object detection with strong application potential. As this technology evolves, the YOLO algorithm is expected to remain a key solution in object detection, serving various real-time applications effectively.

Keywords: object detection, small object, YOLO, detection head.

Introduction

Object detection is a fundamental task in computer vision, and it plays a critical role in various applications, such as real-time video analysis, autonomous driving, and intelligent security systems. Traditional object detection algorithms relied on manually extracting features, a process that was not only time-consuming but also prone to instability. The state-of-the-art (SOTA) algorithm at the time, the Deformable Part-based Model (DPM) [1], offered relatively fast inference speeds and was capable of adapting to slight deformations. However, it could not handle large-scale rotations effectively and demonstrated low robustness, limiting its performance in more challenging detection tasks.

In recent years, the advent of convolutional neural networks (CNNs) has led to significant advancements in deep learning-based object detection algorithms. This progress has given rise to two major approaches: anchor-free methods and anchor-based methods [2], which have substantially improved the accuracy and efficiency of object detection. Anchor-based methods are divided into two categories: one-phase algorithms and two-phase algorithms. The two-phase approach involves generating region proposals from an image, followed by producing bounding boxes from these proposals. Algorithms like RCNN are representatives of this type and achieve high Mean Average Precision (mAP). However, their inference speed is slow due to the need to train multiple networks to perform different tasks in various inference stages, making RCNN unsuitable for real-time applications [3].

Several improvements, such as Fast-RCNN [4], Faster-RCNN [5], and MaskRCNN [6], have been introduced to accelerate RCNN's inference speed by modifying network structures. Nevertheless, the frame rates remain low, limiting their real-time effectiveness.

In contrast, the You Only Look Once (YOLO) [7] algorithm, a one-stage method, has garnered significant attention. YOLO uses a single network to simultaneously predict bounding box coordinates and classification probabilities, resulting in exceptional speed. Despite its efficiency, the original YOLO model had some limitations, including slightly lower mAP and challenges in detecting a large number of closely grouped objects.

To address these shortcomings, researchers have developed improved versions of YOLO, with the latest being YOLOv8. While YOLOv8 achieves remarkable performance in both real-time processing and accuracy, there remains room for further enhancements to optimize its capabilities even further.

YOLOv8 faces challenges in detecting small and dense targets, often resulting in missed detections and overlapping detection frames, especially for objects smaller than 8×8 pixels. This is due to its use of a predefined detection head, which lacks the precision needed for small target details and struggles with densely packed objects.

To address these issues, this paper proposes optimizing the detection head by reducing its perceptual field and increasing the number of detection heads. After reconstruction, the improved YOLOv8 demonstrated significantly better performance in detecting grouped small objects.

1. Improved Performance on Dense Targets: The datasets used for inference included an average of more than 30 objects per image. Despite the potential for a significant time cost to predict every bounding box parameter, the optimized model achieved excellent efficiency, maintaining an average frame rate of 30 fps.
2. Higher Recall Rate: While the original YOLOv8 achieved a recall rate of less than 60%, the improved model successfully detected nearly all objects, with a recall rate exceeding 80%.
3. Practical Applications: The optimized model serves as an example of a "counting machine," capable of performing tasks such as "counting sand in a desert." This showcases its potential for handling other similar tasks that are traditionally time-intensive and prone to human error.

These enhancements demonstrate the model's capability for efficiently detecting small and dense targets, making it suitable for real-world applications requiring precision and speed.

Methodology

Model Architecture

The YOLOv8 architecture builds on the advancements of previous YOLO versions [8], featuring two fundamental components: the backbone and the head.

Backbone: YOLOv8 utilizes a revised Cross-Stage Partial (CSP) architecture as its backbone. This architecture consists of 35 convolutional layers and employs cross-stage fractional connections to improve data transfer between layers. These connections enhance feature representation and processing efficiency.

Head: The YOLOv8 head is responsible for predicting bounding boxes, object confidence scores, and class probabilities for detected objects. It consists of a series of convolutional layers followed by fully connected layers. A key enhancement in YOLOv8's head is the integration of a self-attention mechanism [9], allowing the model to focus on different regions of the image and adjust the importance of features based on their relevance.

Multi-Scale Object Recognition: Another notable feature of YOLOv8 is its ability to detect objects at multiple scales, facilitated by a characteristic hierarchy network [10]. This network includes multiple layers that are specifically designed to detect objects of varying sizes, enabling the model to reliably recognize small, medium, and large objects within the same image.

These architectural improvements make YOLOv8 more robust and accurate, particularly in handling diverse object sizes and complex scenes. Figure 1 illustrates the typical structure of the YOLOv8 model.

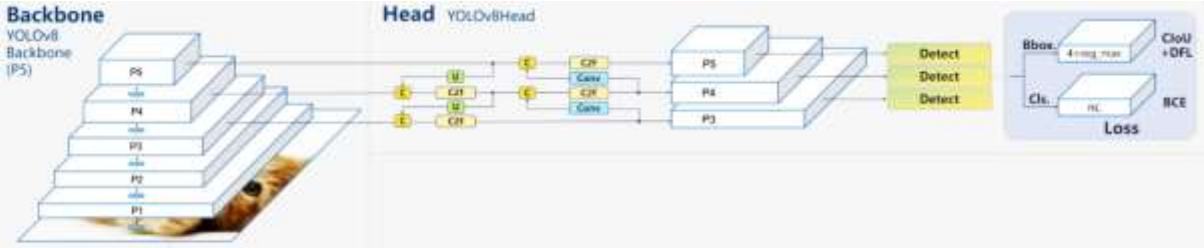

Figure 1. Structure of YOLOv8

In YOLOv8, the "head" refers to the top-level hierarchical structure of the neural network, which processes the feature map extracted by the backbone. This part plays a critical role in converting feature maps into detection results. The YOLOv8 "head" consists of three primary components:

Detection Layers:

These layers transform feature maps into bounding boxes and corresponding category prediction probabilities using convolution operations. Each detection layer is associated with anchor boxes, enabling the detection of objects at various scales. By handling feature maps at different resolutions, the detection layers ensure accurate predictions across diverse object sizes.

Up-Sample Layers:

These layers increase the resolution of feature maps through deconvolution operations, effectively converting low-resolution maps into high-resolution ones. This process enhances the model's ability to detect small-sized objects, improving accuracy in scenarios where fine details are critical.

Route Layers:

Route layers connect feature maps from different levels within the network. They merge feature maps from earlier layers with those from later layers, facilitating multi-scale feature fusion. This approach combines information from different resolution levels, enabling the model to detect objects of varying sizes and types with greater precision.

The "head" in YOLOv8 is a pivotal component that translates feature maps into detection results through a combination of detection layers, up-sample layers, and route layers. This design ensures effective multi-scale feature fusion, allowing the model to efficiently and accurately detect objects of different sizes and types in diverse scenarios.

Results and Discussion

Performance

In object detection, three critical metrics are used to evaluate model performance: recall rate, precision, and mean average precision (mAP). This paper highlights the importance of comparing these metrics to comprehensively assess the effectiveness of object detection models.

Recall Rate:

Recall rate measures the proportion of actual objects in an image that the model successfully detects. A higher recall rate indicates the model's ability to identify most of the objects present, reducing missed detections.

Precision:

Precision calculates the proportion of detected objects that are correctly identified as true objects, avoiding false positives. A high precision value ensures the reliability of the model's detections.

Mean Average Precision (mAP):

The mAP metric provides a comprehensive evaluation by summarizing the trade-off between precision and recall across multiple classes in object detection tasks. Specifically, mAP 90% Intersection Over Union (IoU) evaluates the model's performance when the predicted bounding boxes perfectly match the ground truth.

By analyzing these three metrics, this paper provides a detailed evaluation of object detection models, ensuring a balance between accurate identification (precision), comprehensive coverage (recall), and overall effectiveness across multiple object categories (mAP).

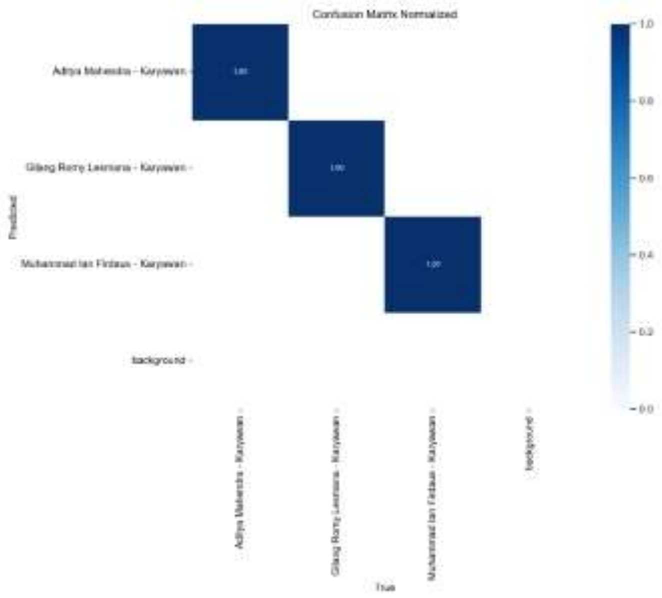

Figure 2. Confusion Matrix Normalized

represents a normalized confusion matrix, illustrating the Performance of a classification model on a validation dataset. Each cell in the matrix indicates the proportion of predictions for a given class relative to the total true samples of that class. Normalization scales the values from 0 to 1, making it easier to interpret and compare the performance across different classes, regardless of variations in class sample sizes.

In this confusion matrix, the diagonal elements (e.g., 1.00 for "Aditya Mahendra - Karyawan," "Gilang Romy Lesmana - Karyawan," and "Muhammad Ian Firdaus - Karyawan") represent true positive rates, indicating that the model accurately predicted the correct labels for these classes in 100% of cases. The absence of significant off-diagonal values demonstrates that there were no misclassifications or false positives for any class. This reflects an exceptional performance, where the model has successfully distinguished between the labeled categories with perfect precision.

The inclusion of the "background" class, which also achieves a value of 1.00, suggests that the model effectively identifies and separates irrelevant or non-human elements in the dataset from the target classes. This level of accuracy is especially crucial in scenarios requiring high precision, such as person identification, employee tracking, or access control in secure environments. Overall, the normalized confusion matrix highlights the robustness and reliability of the model for its intended application.

Results and Discussion

To enhance the performance of YOLOv8, this paper introduces an additional detection head to the model's head, while retaining the structure of the backbone. This modification enables the model to detect small objects as tiny as 4×4 pixels. Compared to the original YOLOv8, the improved model demonstrates:

- A 4.2% increase in precision and a 4.0% increase in recall rate for detecting bacterial colonies.
- A significant 9.2% rise in mAP.

The model's ability to visually detect almost all bacterial colonies indicates it has successfully achieved its primary goal: counting anything. The experiments confirm that incorporating an additional detection head improves YOLOv8's capability to detect small objects.

Applications:

The modified model has versatile applications, including:

- Estimating current traffic flow using satellite cameras.
- Monitoring bacterial colony growth.

Limitations and Suggestions:

1. Performance Trade-Off:

Adding too many detection heads could slow down the training and inference processes, which is a potential drawback of this method. To address this, the paper suggests:

- Reducing the number of detection heads and tailoring YOLOv8 for specific tasks.
- Adding a simple preprocessing layer before the input layer, capable of automatically receiving prompt words to modify the detection head for task-specific optimizations.

2. Object Overlap:

The model does not yet effectively handle overlapping objects, especially small items. This is a challenging issue that requires further research and development. Addressing this limitation could represent a significant step toward the ultimate goal of "counting sand," a complex task involving highly overlapping small objects.

Future Work:

The paper identifies handling object overlap and dynamic customization of detection heads as key areas for future research to improve the model's performance and adaptability further.

References

[1] P. F. 01. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010, doi: 10.1109/TPAMI.2009.167.

[2] L. 02. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, no. 3, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[3] R. 03. Girshick, J. Donahue, T. Darrell, and J. Malik, "R-CNN: Regions with CNN features," *Proc. ieee Conf. Comput. Vis. pattern Recognit.*, 2014, [Online]. Available: http://gwylab.com/pdf/rcnn_chs.pdf

[4] R. 04. Girshick, "快速的区域卷积网络方法(Fast R-CNN)," 2015, [Online]. Available: https://gwylab.com/pdf/fast-rcnn_chs.pdf

[5] S. 05. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," pp. 1–14.

[6] M. 06. R-cnn, P. Doll, and R. Girshick, "Mask R-CNN".

[7] J. 07. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once : Unified , Real-Time Object Detection".