



Benchmarking YOLOv8 Variants with Transfer Learning for Real-Time Detection and Classification of Road Cracks and Potholes

Dede Kurniadi^{1*}, A. Abdul Latif², Asri Mulyani³, Hilmi Aulawi⁴

^{1,2,3}Department of Computer Science, Institut Teknologi Garut, Garut, Indonesia

⁴Department of Industrial Engineering, Institut Teknologi Garut, Garut, Indonesia

¹dede.kurniadi@itg.ac.id, ²2106145@itg.ac.id, ³asrimulyani@itg.ac.id, ⁴hilmi_aulawi@itg.ac.id

Abstract

Road damage, including potholes and cracks, is a significant issue frequently encountered in road infrastructure in many regions. Such conditions accelerate road degradation, increase the risk of traffic accidents, and significantly increase the maintenance and repair costs. Although several deep learning models have been proposed for road damage detection, few studies have systematically compared the performance of lightweight YOLOv8 variants using a consistent dataset. To address this gap, this study proposes a road defect detection and classification model based on the YOLOv8 series, which is enhanced using transfer learning to improve performance and efficiency. The dataset, obtained from Roboflow, comprises 3,846 images categorized into training, validation, and testing sets. Three YOLOv8 variants—YOLOv8n, YOLOv8s, and YOLOv8m—were benchmarked for performance. A performance evaluation was performed using the metrics of precision, recall, and mean Average Precision (mAP). Results show that YOLOv8m achieved the highest precision (99.00%), recall (98.40%), and mAP (99.50%). In the pothole category, precision reached 98.70% and recall 99.30%; in the crack category, precision was 99.30% and recall 97.60%. The findings demonstrate that YOLOv8, particularly the YOLOv8m variant, is highly effective for real-time road damage detection and classification, offering a viable solution for intelligent transportation systems and automated infrastructure monitoring. This research has the potential to revolutionize infrastructure monitoring by enabling scalable, real-time, and cost-effective assessments of road conditions. It minimizes reliance on manual inspections, reduces human errors, and contributes to the development of intelligent transportation systems and predictive maintenance strategies.

Keywords: classification; deep learning; road damage detection; transfer learning; YOLOv8

How to Cite: D. Kurniadi, A. A. Latif, A. Mulyani, and H. Aulawi, "Benchmarking YOLOv8 Variants with Transfer Learning for Real-Time Detection and Classification of Road Cracks and Potholes", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 4, pp. 771 - 780, Aug. 2025.
Permalink/DOI: <https://doi.org/10.29207/resti.v9i4.6710>

Received: June 2, 2025

Accepted: July 23, 2025

Available Online: August 15, 2025

*This is an open-access article under the CC
BY 4.0 License
Published by Ikatan Ahli Informatika
Indonesia*

1. Introduction

The Roads are a critical component of transportation infrastructure, playing a strategic role in supporting national economic growth by facilitating the movement of goods and services [1]. However, increasing traffic volumes, environmental factors, and inadequate maintenance practices have led to widespread road deterioration in many regions, including Indonesia. This calls for early detection and more effective maintenance strategies to ensure that road infrastructure continues to support community mobility and economic growth optimally [2].

The Indonesian Ministry of Public Works and Housing (PUPR) plans to 2024 a budget for road maintenance

every year in line with efforts to repair infrastructure that various factors, such as excessive vehicle loads and extreme weather conditions, have damaged. In addition, according to the Directorate General of Highways, the Ministry of PUPR has allocated a budget of Rp14.6 trillion for regional road repairs, planned to start in July [3]. Increasing road damage is caused by high vehicle volumes, extreme weather conditions, inadequate construction quality, and lack of regular maintenance. This problem affects transportation efficiency and can hinder sustainable economic development. Therefore, strategic measures involving integrated planning, construction, and maintenance of road infrastructure are needed to ensure its continued function in supporting mobility and national economic growth [4].

One of the solutions that can be applied to increase the effectiveness in detecting and handling road damage is the utilization of artificial intelligence-based technology, especially machine learning [5]. Using machine learning to detect road damage allows analyzing road conditions to be carried out more quickly, accurately, and efficiently [6]. Machine learning technology can process road image data from sensors or drones to identify various types of damage, such as cracks, potholes, etc. This technology not only helps reduce the time required for manual inspections but also minimizes the risk of human error when assessing the level of damage. In addition, machine learning can also optimize road maintenance planning so that the government and related parties can take more proactive steps to deal with damage before it worsens.

The purpose of this research is to conduct a benchmark study on the performance of three variants of the YOLOv8 model, namely YOLOv8n (lightweight), YOLOv8s (balanced), and YOLOv8m (high-performance), in detecting and classifying road defects, especially the crack and pothole categories. YOLOv8 was selected for its balance of accuracy, speed, and deployment readiness. While YOLOv11 is the latest release, this study focuses on a systematic evaluation of YOLOv8 sub-variants YOLOv8n (lightweight), YOLOv8s (balanced), and YOLOv8m (high-performance), which remain underexplored in current literature. This focus offers novel contributions to real-world road damage detection.

In previous research conducted by Jakubec et al [7], hole detection was performed using CNN models such as R-CNN and YOLO under various weather and lighting conditions; YOLOv7 performed best in sunny and wet scenarios, while R-CNN excelled in low-light conditions. Sasmito et al [8] Implemented CNN to detect cracks, holes, and white lines from video footage, achieving 88% accuracy, with potential for improvement through increased datasets. Wang et al [9] developed BL-YOLOv8 using 4,373 images, which improved mAP@0.5 by 3.3%, reduced parameters by 29.92%, and minimized computational load, making it efficient for lightweight devices. Ali et al [10] used the GLCM method with Neural Network classification on 29 images, resulting in an accuracy of 0.79, with suggestions for improvement through dataset expansion. Djulyansyah et al [11] applied YOLOv8 to detect road conditions in real-time to assist the visually impaired, achieving over 96% accuracy for most classes, but only 62% for potholes, requiring further dataset optimization.

This study implements transfer learning techniques to improve the performance and efficiency of the model and compares its effectiveness based on the model's complexity and implementation needs in the field. While YOLOv8 is no longer the latest version in the YOLO series, it remains one of the most widely adopted object detection models due to its balance of accuracy, speed, and ease of deployment. Unlike previous studies

that primarily used traditional CNN-based approaches or earlier YOLO versions such as YOLOv4, YOLOv5, and YOLOv7, this study focuses on systematically benchmarking three sub-variants of YOLOv8 (YOLOv8n, YOLOv8s, and YOLOv8m) using a unified dataset and consistent evaluation metrics to determine optimal trade-offs between accuracy, model complexity, and real-time applicability.

Furthermore, this study addresses two critical research gaps identified in previous works. The first concerns the lack of comprehensive evaluation of the generalization capability of object detection models, particularly in recognizing subtle and fine-grained crack patterns that are often misclassified under real-world conditions. The second pertains to the absence of systematic comparisons among YOLOv8 sub-variants to assess their respective effectiveness for real-time deployment in practical scenarios. To overcome these limitations, this study employs transfer learning along with advanced data balancing and augmentation techniques to enhance model robustness and improve detection performance, especially in crack classification, which has yielded suboptimal results in prior research. In line with these objectives, the study investigates which YOLOv8 variant (YOLOv8n, YOLOv8s, or YOLOv8m) provides the most optimal balance between detection accuracy, model complexity, and feasibility for real-world implementation. The hypothesis is that YOLOv8m provides the highest detection accuracy due to its deeper architecture, while YOLOv8n performs better in terms of inference speed and computational efficiency. This clear formulation of the research question and hypothesis provides a strong methodological foundation and reinforces the scientific validity of the comparative analysis.

2. Methods

This research adopts an experimental study strategy to test a pothole and crack road defect detection and classification model using YOLOv8 and Transfer Learning to evaluate its performance. This process includes designing the steps applied during the research. The research framework is presented in Figure 1.

Based on Figure 1, the research flow begins with Data Acquisition, including Data Collection and Identification. Road damage images were obtained from [Roboflow](#), using the free *Public Plan*, which ensures open access by listing datasets and trained models publicly. This plan supports transparency and reproducibility without requiring the restricted features of the *Premium Trial*. All images were manually reviewed for label accuracy, class balance, and COCO format compliance [12].

The preprocessing stage was conducted to prepare the dataset to fit the needs of the YOLOv8 [13]. This process includes three primary methods: data balancing, labelling, and splitting. Data balancing is done to

maintain the data balance in each class so that the model is not biased. Data Labelling is redone using the YOLO annotation format to consistently define the bounding box on the image. After that, Data Splitting is

performed by dividing the dataset into three subsets, namely training, validation, and testing data, to ensure that the model evaluation is carried out objectively and supports the ability to generalize to new data.

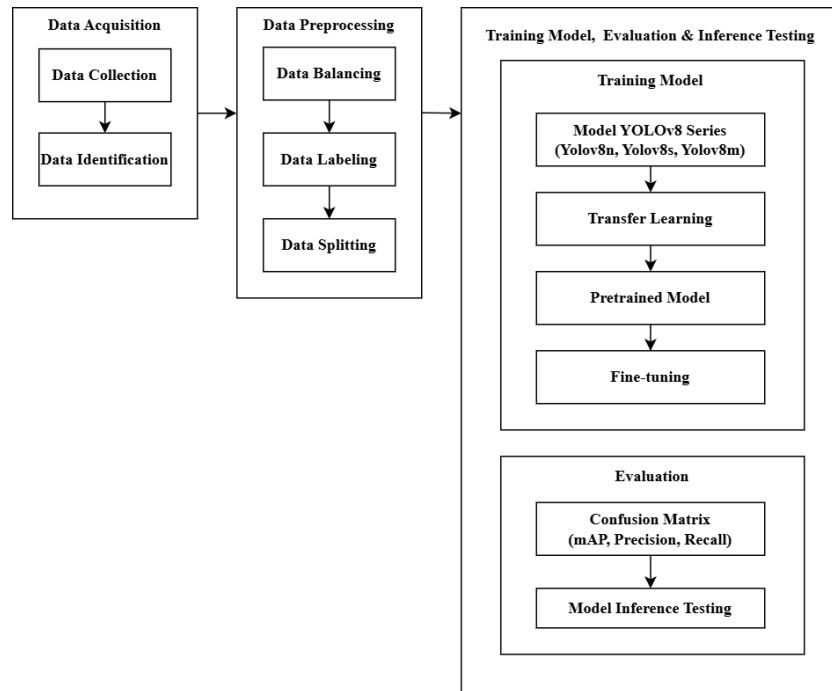


Figure 1. Research Framework

The YOLOv8 series models (YOLOv8n, YOLOv8s, and YOLOv8m) were utilized in the subsequent stage using a transfer learning approach, which involved leveraging pretrained weights trained on large-scale datasets, followed by fine-tuning on the road damage dataset used in this study. This strategy enables the models to retain general visual features while adapting specifically to the target domain, thereby improving performance and reducing training time [14]. This technique allows the utilization of weights and features from previous training to make adapting the model to new datasets more efficient and accurate [15].

Model evaluation is performed using test data, a confusion matrix, and several key evaluation metrics, such as mAP@0.5, precision, and recall [16]. Using a transfer learning approach, the evaluation stage aims to provide an objective picture of the model's ability to detect and classify road defects, such as potholes and cracks.

The final stage in this research is model inference testing, where the YOLOv8 model trained using transfer learning techniques is tested again with new data to assess its performance under real conditions. This test determined how well the model could detect and classify road damage types when applied to real-world scenarios. The results and analysis section then describes the research flow of the YOLOv8 implementation with transfer learning as well as the effectiveness of the model in image detection and classification tasks. The research results from each

stage of the process are explained to provide a complete picture of the application of this technique for improving road defect detection.

YOLOv8 is a detection model that is highly efficient in recognizing various objects in an image. It can perform detection and classification simultaneously in a single process, making it an ideal choice for models that require high speed in real-time [17]. The main advantage of YOLOv8 lies in its computational efficiency and ability to provide accurate predictions with low latency. This makes it very suitable for application in various conditions that require fast and high-precision object detection [18]. In addition, YOLOv8's optimized architecture allows for improved performance compared to previous versions, making it one of the best solutions in the field of deep learning-based object detection [19]. Its architectural structure can be seen in Figure 2. Based on Figure 2, which is reproduced directly from previous research [20], the YOLOv8 architecture consists of three main components, namely the Backbone, the Head, and the Loss Function. In the feature extraction (feature learning) stage, the network structure receives the image as initial input and processes it through several levels of the feature pyramid (P1–P5). The Backbone performs this process by applying convolution and pooling layers to extract and represent essential features in a numerical feature map. Each step in this layer strengthens the image representation before it is passed on to the prediction stage.

In the prediction stage, the Head plays a role in connecting the extracted features with the detection layer to generate object predictions at various scales [21]. This layer consists of several fully connected neurons that receive input from the feature map

generated by the backbone [22]. Then, these predictions are adjusted through a series of hidden layers, resulting in more accurate classification and object localization results.

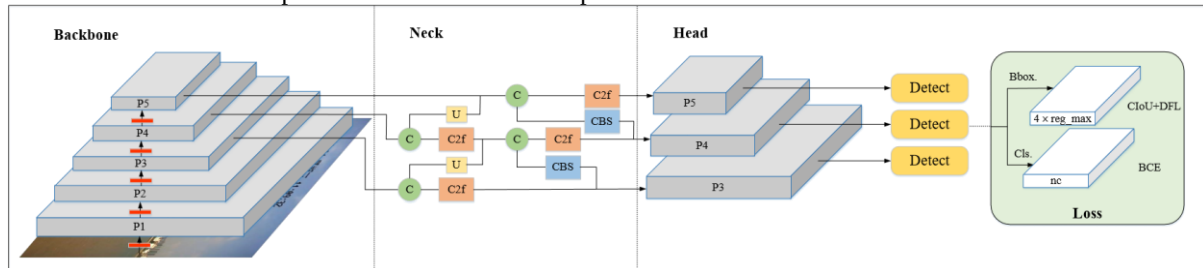


Figure 2. YOLOv8 Architecture

The convolution layer is the main component in a CNN network and performs convolution operations on the previous layer's output. It contains a series of filters that are learned to extract essential features from the input image, allowing the model to recognize more complex patterns. A pooling or subsampling layer usually follows the convolution layer. Its function is to reduce the dimension of the output of the convolution layer in the height and width spatial dimensions [23]. In this way, the number of parameters that the network has to learn is reduced, helping to overcome overfitting and improving the efficiency and accuracy of the network [24]. Activation functions such as ReLU (Rectified Linear Unit) are used to introduce nonlinear properties in the network [25]. This function retains positive values and assigns zero values to negative inputs, assisting the model in capturing complex relationships in the data. In YOLOv8, there is Lucky ReLU, which aims to improve this activation function.

YOLOv8's three detection heads may limit its ability to detect partially obscured or small objects [26]. Therefore, adding Dynamic Detection Heads (DyHeads) is proposed to improve the detection capability. DyHeads incorporate an attention mechanism with three main components: scale-aware attention, spatial-aware attention, and task-aware attention. By applying these mechanisms, the YOLOv8 model can be more optimized in detecting objects of various sizes and in more complex conditions [27]. To optimize the processing of multi-scale semantic information with high resolution, Dynamic Heads (DyHeads) are used in YOLOv8 [28]. It aims to improve classification accuracy and object localization by utilizing richer and more relevant features for each prediction [29].

In the optimization process, YOLOv8 uses a Loss Function consisting of a combination of Complete Intersection over Union (CIoU) and Distribution Focal Loss (DFL) for bounding box regression, and Binary Cross Entropy (BCE) for classification [30]. However, several evaluation metrics are used to assess the model's performance in accurately detecting objects, namely mean Average Precision (mAP), Precision, and Recall [31]. The mAP is calculated using Equation 1.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (1)$$

Where N is the dataset's total number of object classes, and AP_i is the average precision for the i -th class. The Precision is calculated using Equation (2), and the Recall is computed using Equation 3.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Where TP is the number of correct positive predictions, FP is the number of incorrect positive predictions, and FN is the number of positive objects the model does not detect.

The dataset used for the YOLOv8 model with the Transfer Learning technique, with the name Cap DP Dataset and Pre-Trained Model by TS obtained from Roboflow [32]. This dataset consists of 3,846 images with two categories of road damage, namely potholes and cracks. The dataset is divided into 2,689 images for training data, 767 for validation data, and 390 for test data. Each image in the dataset has been labeled according to the road damage category, which will be used to train and test the detection and classification model for potholes and cracks.

The stages of the machine learning process are as follows.

3. Results and Discussions

3.1 Results

The results of this research will be explained through several stages in the YOLOv8 model with Transfer Learning. Each stage, from data acquisition to model evaluation and inference testing, is essential for accurate analysis. The following section elaborates on each stage of the process in detail.

The first step in the data acquisition stage is collecting the dataset from an online provider, Roboflow. This dataset consists of 3,846 images with two categories of road damage, namely potholes and cracks. The dataset is divided into 2,689 images for training data, 767 for

validation data, and 390 for test data. Each image in the dataset has been labeled according to the road damage category, which will be used to train and evaluate the model.

In the data preprocessing stage, three main steps are performed: data balancing, labeling, and splitting. Data balancing was performed by undersampling the majority class, pothole, to equalize the number with the minority class, crack, 1,554 images each, to create a balanced data distribution and improve the model's generalization ability. Next, all images were annotated using the YOLO format with the help of tools such as Roboflow, which provides accurate bounding box information to detect the location of road defects, either potholes or cracks. After labeling, the dataset is divided into three subsets with a ratio of 70:20:10 for training, validation, and testing, that the model can learn optimally.

The model training stage uses 768×768 -pixel resolution images for 200 epochs with a batch size 16. The process uses the AdamW optimizer with an initial learning rate of 0.00015 that decreases gradually through a cosine scheduler scheme until it reaches 0.01 and is supported by regularization in the form of momentum of 0.937, weight decay of 0.001, and dropout of 0.2 to prevent overfitting. The applied data augmentation includes color augmentation (hue, saturation, brightness), horizontal inversion, as well as complex augmentation such as mosaic, mixup, and copy-paste, followed by geometric augmentation such as rotation, translation, scale, shear, and light perspective to improve the generalization ability of the model to visual variations. The YOLOv8 model uses a pre-training architecture during retraining, where the initial layers are retained to identify basic features. In contrast, the final layers are adjusted to be more relevant to road defect classification. The model is retrained with the processed dataset and monitored using mAP, precision, and recall evaluation metrics. The best model selection is based on validation performance; the final results obtained show the stability of the training and the model's ability to detect and classify road defects on new data. The results of model training with several scale variants are shown in Table 1.

Based on Table 1, the YOLOv8m model provides the best results with a precision value of 99.00%, a recall of 98.40%, and mAP of 99.50%, indicating a highly optimized detection and classification capability of road defects, both potholes and cracks. The model consistently outperformed across all evaluation metrics in both classes. Meanwhile, the YOLOv8s model also performed very well with a precision of 95.00%, a recall of 95.00%, and mAP of 98.10%, indicating a fairly optimal balance, although still below the performance of YOLOv8m. On the other hand, the YOLOv8n model has the lowest performance among the three, with a precision value of 92.50%, a recall of 89.80%, and mAP of 95.40%, although it still shows decent performance in the light detection scenario. Thus, it can be concluded

that the YOLOv8m model is the best choice for the road defect detection and classification task as it can provide an optimal balance between precision, recall, and mAP across all classes. After the retraining was completed, an evaluation was conducted to assess the improvement in the model's performance. Analysis of the learning curves showed that the model avoided overfitting or underfitting. The confidence threshold is adjusted to optimize predictions, while the precision-recall curve is used to understand the balance between precision and recall. The confusion matrix shows the distribution of model predictions and detects misclassifications. Other evaluation metrics, such as mAP, precision, and recall, show significant improvement after transfer learning is applied, as shown in Figure 3.

Table 1. Summarizes Evaluation Model Results

Model	Class	Precision (%)	Recall (%)	mAP (%)
Yolov8n	Pothole	94.70	94.20	98.00
	Cracks	90.20	85.50	92.80
	All Class	92.50	89.80	95.40
Yolov8s	Pothole	95.90	98.10	99.30
	Cracks	94.10	92.00	97.00
	All Class	95.00	95.00	98.10
Yolov8m	Pothole	98.70	99.30	99.50
	Cracks	99.30	97.60	99.40
	All Class	99.00	98.40	99.50

Figure 3 is derived from the training process of the YOLOv8 model and illustrates the various evaluation metrics and loss values that occur during the training and validation process. Figure 3(a) Training Loss Metrics displays the values of train/box_loss, train/cls_loss, and train/dfloss. These three graphs consistently decrease as the number of epochs increases, indicating that the optimization process is running well. The decrease in box loss suggests that the model is more accurate in predicting the position and size of objects. The decline in classification loss reflects the increased accuracy of the model in classifying objects into the appropriate class. In contrast, the decrease in distribution focal loss indicates that the model is more precise in estimating the spatial distribution of objects. In Figure 3(b), Validation Loss Metrics are also seen in the graphs of loss values against validation data, namely val/box_loss, val/cls_loss, and val/dfloss, shown in the second row. The decrease in these three graphs shows that the model has improved performance on the training data and has good generalization ability to new data. This indicates that the model does not experience performance beyond the training data.

In addition to the loss values, Figures 3(a) and (b) also show graphs of the evaluation metrics, namely precision(B), recall(B), mAP50(B), and mAP50 95(B). The precision graph shows an increase in value over time, indicating that the model generates fewer false positives. Meanwhile, the recall graph has also increased, indicating that the model can detect more relevant objects and minimize false negatives. The mean Average Precision (mAP) values at the IoU

threshold of 0.5 (mAP50) and the range of 0.5 to 0.95 (mAP50-95) show a significant upward trend, reflecting the model's good performance in detecting and

classifying objects with high precision at various levels of precision.

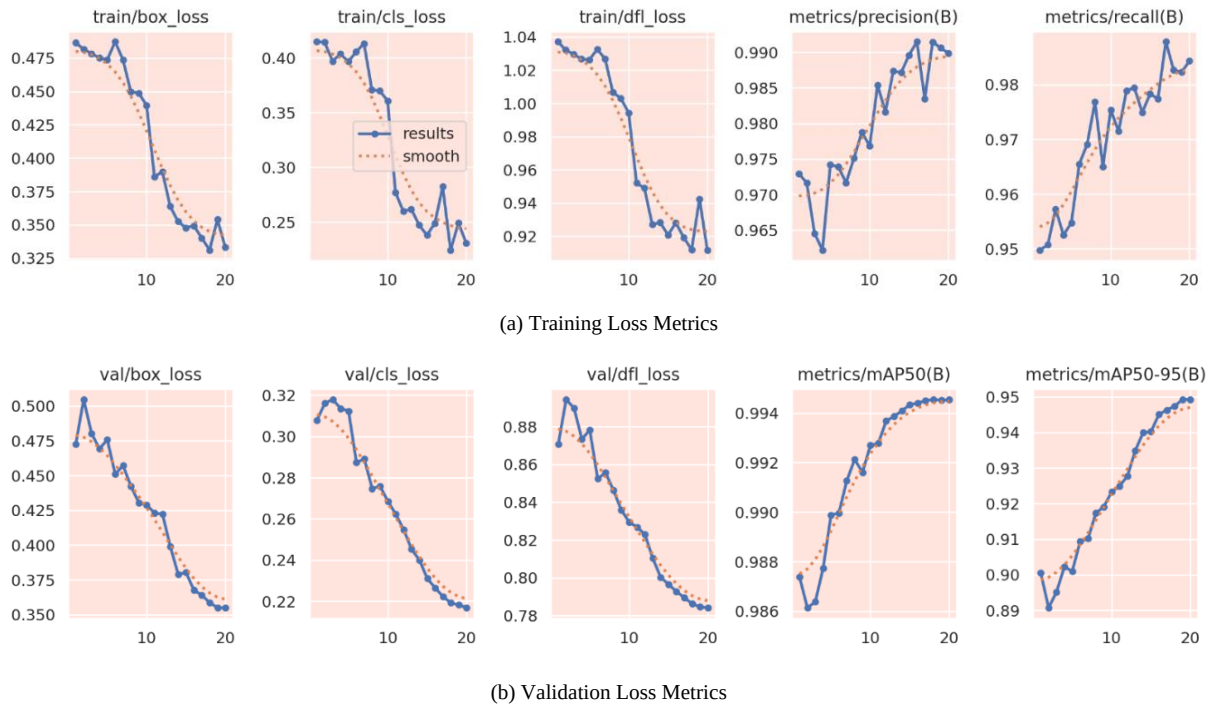


Figure 3. Graphs, (a) Training Loss Metrics, (b) Validation Loss Metrics

Overall, this graph shows that the model has gradually improved in performance, with a good balance between the training process and the ability to generalize new

data, as shown in the confidence curve, recall, precision, and precision-recall results in Figure 4.

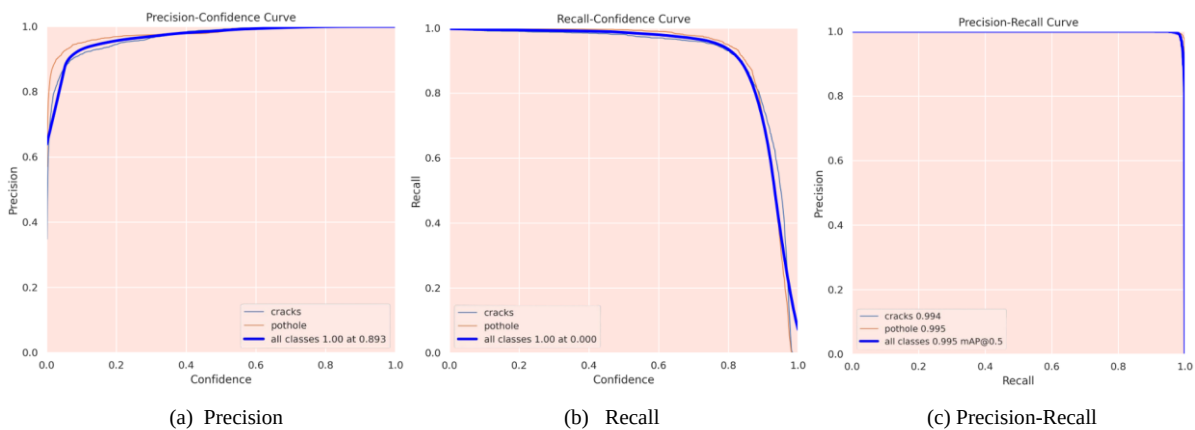


Figure 4. Confidence Curve, (a) Recall, (b) Precision, (c) Precision-Recall

Figure 4 is generated through program processing that visualizes the classification results of the model in the form of an evaluation curve. Figure 4(a), Precision-Confidence Curve, shows a precision that increases sharply and reaches a value of 1.00 at a confidence of 0.893, indicating a very accurate prediction at a high confidence threshold. Figure 4(b), Recall-Confidence Curve, shows high recall at low increases. Meanwhile, Figure 4(c), Precision-Recall Curve, shows excellent model performance with high precision and recall in both classes, and an overall mAP@0.5 of 0.995. These results indicate the model's balanced and highly reliable performance detecting objects at various confidence

levels. Finally, the evaluation with a confusion matrix is shown in Figure 5.

Based on the model evaluation results with the Confusion Matrix shown in Figure 5, the model can classify the class quite well. Figure 5(a) Confusion Matrix indicates the number of correct and incorrect predictions in absolute numbers. In contrast, Figure 5(b) Normalized Confusion Matrix presents normalized values in proportions for straightforward interpretation of model performance. Darker colors on the main diagonal indicate high accuracy for each class, while misclassification can be analyzed by distributing values outside the main diagonal.

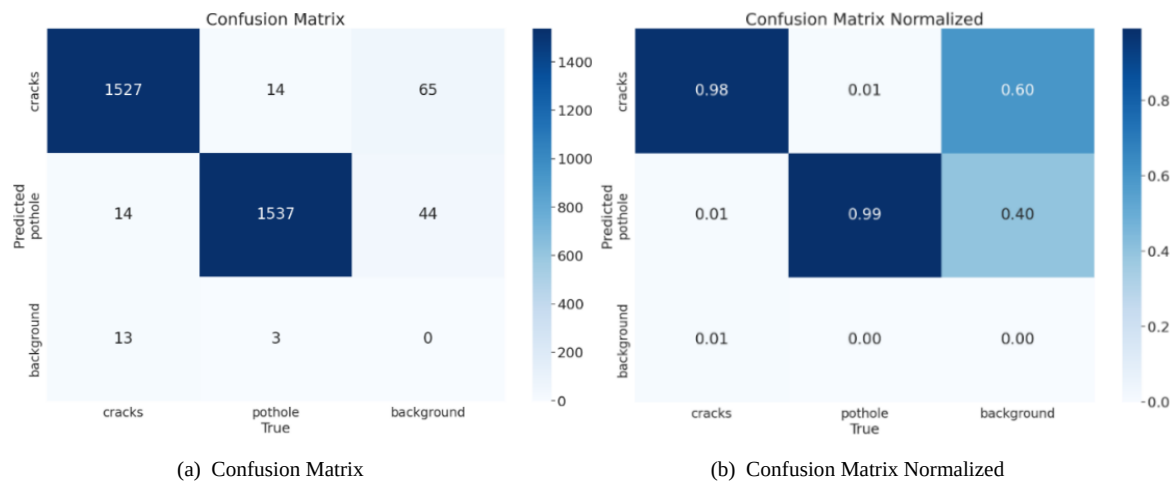


Figure 5. Diagram, (a) Confusion Matrix, (b) Confusion Matrix Normalized

Figure 5(a) is generated through program processing that visualizes the model's classification results as a confusion matrix. Based on Figure 5(a), the model has a high degree of accuracy in classifying the cracks and pothole classes, with each having 1527 correct predictions for cracks and 1537 for potholes. The dominant principal diagonal value indicates that the model can identify objects correctly. However, there are still some misclassifications, especially in the background class, which is often classified as cracks in 65 cases. This indicates that the model still has difficulty distinguishing background features from other objects.

In addition, the confusion matrix also shows that there is an exchange of predictions between cracks and potholes, with 14 cases each. However, the high number of correct predictions in both main classes indicates that the model can recognize the damaged road features well. Analysis of the error patterns is essential to identify areas that need to be optimized, for example, by adjusting the threshold or adding a variety of training data to reduce ambiguity between background and other objects.

Figure 5(b) is generated through program processing that visualizes the classification results of the model in the form of a normalized confusion matrix. Based on Figure 5(b), the model shows a high proportion of correct predictions in the pothole and cracks classes, at 0.99 and 0.98, respectively. A small proportion of misprediction is observed when objects from the cracks or pothole class are not recognized and predicted as background. For example, 0.01 of the pothole class was predicted as background. This indicates that although the model performs very well in identifying the main objects, there is still a need to improve the detection sensitivity of less obvious objects, which can be done by adding training data with variations in lighting conditions, shape, or texture of the damage.

This normalization helps to clarify the proportional distribution of prediction errors. For example, a small fraction of the cracks and pothole classes were still

mistakenly predicted as background, at 0.01 each. This indicates that the model sometimes had difficulty recognizing the features of these objects completely and thus did not give a prediction (false negative). In addition, it can also be seen that the background class often appears as a prediction result for objects not confidently recognized by the model, especially in the area of cracks. This information is essential for identifying weak points in the model's performance, which can be addressed by adjusting the confidence threshold or adding more training data on the variations of conditions that cause detection failures.

The final stage is to test the model on new data to ensure that the model can perform detection and classification well in real scenarios. Inference is performed on test validation images that were never used in training. The results of testing using a new dataset of 8 images that show the ability of the model to detect and classify two types of road damage, namely potholes and cracks, show that the model can recognize and classify objects well, as shown in Figure 6. The inference testing results in Figure 6 show that the model can detect and classify objects in the pothole and crack classes, which are characterized by bounding boxes and varying confidence scores.

In Figure 6(a), the model detects two potholes with high confidence, 0.92 and 0.89. The undetected regions are not structural potholes but surface water puddles, which do not meet the feature patterns learned by the model during training. Zooming in would not improve detection, as those areas lack the edge and depth characteristics of true potholes. Figure 6(b) displays crack detection on rough surfaces with a confidence of 0.81, indicating the model's sensitivity to thin cracks despite road textures resembling other objects. Furthermore, in Figure 6(c), the model can recognize potholes in rural areas with confidence reaching 0.87 and 0.77, despite visual distractions such as the presence of vehicles. In Figure 6(d), multiple detection of multiple potholes in one image shows the model's effectiveness in identifying objects simultaneously with

confidence levels such as 0.80, 0.78, 0.76, 0.46, and 0.35. This variation suggests robust model performance across diverse visual conditions.

In the second row, the detection of different shapes is also consistent in Figure 6(e), where the model detects one pothole with a confidence of 0.83, even though puddles cover part of the object. This demonstrates the model's ability to identify pothole contours under complex visual conditions. Figure 6(f) shows a crack detection with a vertical orientation on a relatively smooth asphalt road surface, with a confidence of 0.82. This detection shows the model's sensitivity to linear

crack shapes. Meanwhile, Figure 6(g) also shows a vertical crack with a confidence of 0.89 on a textured road surface, reinforcing the model's ability to deal with variations in surface texture. Finally, Figure 6(h) shows horizontal crack detection with a confidence of 0.87, demonstrating the model's consistency in recognizing different crack orientations and shapes. These results show that the model has good generalization power in dealing with various visual conditions, including variations in shape, direction, surface texture, and visual distractions such as puddles or the presence of other objects.

Results of Model Inference Testing

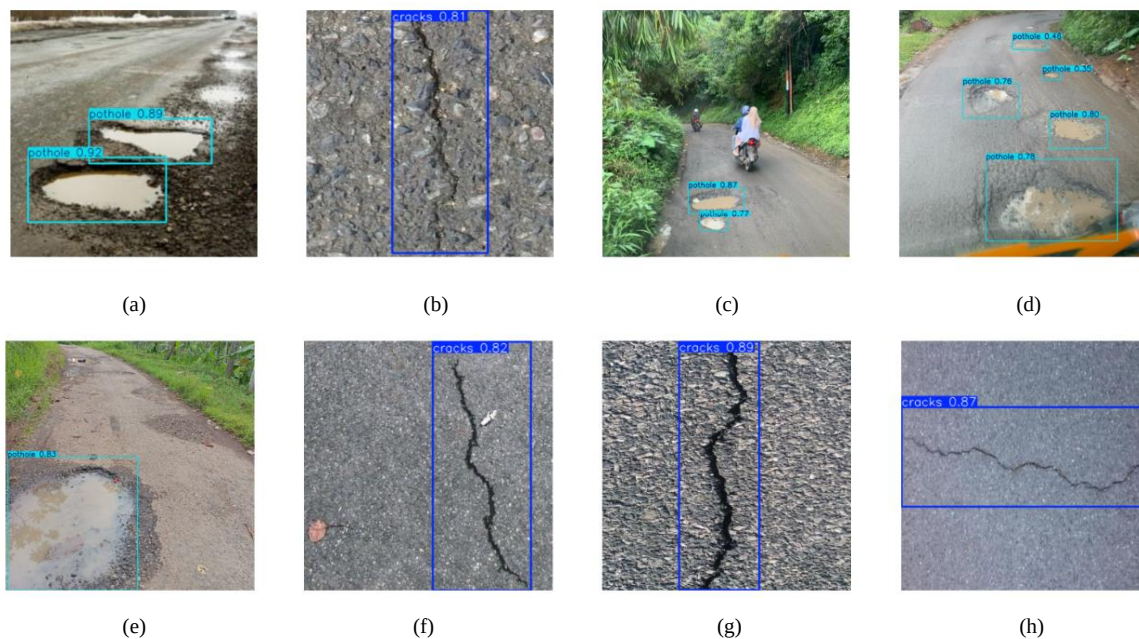


Figure 6. Model Inference Testing Result. (a) Potholed Road on Wet Surface. (b) Cracked Road on Rough Surface. (c) Pothole on Rural Road. (d) Potholes with Multiple Spots. (e) Potholes on Broken Asphalt Roads. (f) Road with Vertical Cracks on Asphalt. (g) Vertical Cracks on Asphalt Road. (h) Horizontal Cracks in Asphalt.

3.2 Discussions

This research aligns with the background that emphasizes the importance of road infrastructure in supporting mobility and economic growth and the urgency of early detection of road damage through modern technology. By applying the YOLOv8 model and transfer learning techniques, this research successfully developed an effective and efficient pothole and crack road damage detection and classification model. The evaluation results showed excellent model performance with a precision of 99.00%, a recall of 98.40%, and an mAP of 99.5%, indicating an excellent level of accuracy in the detection and classification process. This study is also consistent with several previous studies, showing improved model performance and better accuracy compared to previous studies. The complete comparison results can be seen in Table 2.

Based on Table 2, this study not only compared several road defect detection models, including YOLOv8, but also addresses performance limitations found in

previous works by optimizing YOLOv8 using transfer learning techniques.

Prior studies using models such as R-CNN (mAP 70.30%) and YOLOv7 (mAP 88.30%) [7], YOLOv4-Tiny (88% accuracy) [8], BL-YOLOv8 (mAP 90.70%, F1-score 87.00%) [9], and GLCM with Neural Network (accuracy 79%, precision 66%, recall 80%) [10], showed limited performance, especially in balancing high accuracy, precision, and generalization. In addition, a previous study using YOLOv8 achieved only 82.8% accuracy [11]. Suggesting that further enhancement was needed.

Table 2. Model Comparison Results

Research	Method	Overall Performance
This Research	YOLOv8 with Transfer Learning Technique	Precision 99.00%, Recall 98.40%, and mAP 99.50%
		R-CNN with mAP 70.30% and YOLOv7 mAP 88.30%.
[7]	R-CNN ResNet-50 and YOLOv7	
[8]	YOLOv4-Tiny	Accuracy 88%, Kappa Accuracy: 86%,

[9]	BL-YOLOv8	mAP 90.70% and F1-Score 87.00%
[10]	Grey Level Co-occurrence Matrix (GLCM) and Neural Network	Accuracy 79%, precision 66%, and recall 80%
[11]	YOLOv8	Accuracy 82.8%

This study fills that gap by applying transfer learning to the YOLOv8 model and demonstrates significant performance improvements, achieving 99.00% precision, 98.40% recall, and 99.50% mAP. These results indicate a notable advancement in detection capability, especially for complex and varied road defect types. In addition, advanced data augmentation strategies and attention to model generalization during training improved robustness, particularly in handling fine-grained crack patterns. By integrating transfer learning, this research improves existing YOLOv8 performance and contributes a more robust and generalizable solution for real-world road defect detection.

Therefore, this study extends the current understanding of YOLOv8s effectiveness and addresses previously identified weaknesses in model architecture and training strategy. Although this study focuses on model evaluation and has not yet reached the real-time deployment stage, the feasibility for real-time implementation was analyzed based on inference speed and hardware configuration. The YOLOv8n and YOLOv8s models, tested on an NVIDIA T4 GPU, achieved inference speeds of 3.9 ms (256 FPS) and 9.2 ms (108 FPS), respectively. Meanwhile, the YOLOv8m model, evaluated on a high-performance NVIDIA A100 GPU, reached 3.3 ms (303 FPS). These results significantly exceed the standard 30 FPS threshold for real-time operation, suggesting that all tested YOLOv8 variants, particularly YOLOv8n and YOLOv8s on standard GPUs, are theoretically capable of supporting real-time road monitoring applications. The YOLOv8m model also demonstrated superior accuracy with 99.00% precision, 98.40% recall, and 99.50% mAP. Although direct deployment has not been performed, these findings support the model's strong potential for future implementation in edge computing environments and embedded systems, which will be explored in subsequent research.

4. Conclusions

This study confirms the effectiveness of the YOLOv8 model, enhanced through transfer learning, in accurately detecting and classifying road defects, especially potholes and cracks. Among the three variants evaluated, YOLOv8m delivers the best performance across all metrics, making it the most suitable model for real-time implementation. The research offers practical insights for selecting the appropriate YOLOv8 variant based on deployment requirements, advancing previous research on different models YOLOv4 and YOLOv5, as well as YOLOv7. The key scientific contribution of this study is that it is the first to systematically benchmark all three YOLOv8

variants YOLOv8n (lightweight), YOLOv8s (balanced), and YOLOv8m (high-performance) for road surface defect detection, specifically targeting potholes and cracks under varied real-world surface conditions. The proposed method significantly improves generalization on real-world test data and is adaptable for real-time applications, offering a practical and efficient foundation for future automated road inspection systems. Nevertheless, this study is limited to two classes of defects, and it is therefore recommended that future research extend the model's classification capabilities to include other critical road surface issues such as alligator cracking, rutting, and edge cracking, which are prevalent in field conditions and play a vital role in road safety and maintenance prioritization.

Acknowledgements

The author declares no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- [1] Menteri PUPR, "Peraturan Menteri Pekerjaan Umum Dan Perumahan Rakyat Republik Indonesia Nomor 5 Tahun 2023 Tentang Persyaratan Teknis Jalan Dan Perencanaan Teknis Jalan," 2023. [Online]. Available: <https://binamarga.pu.go.id/index.php/peraturan/dokumen/peraturan-menteri-pekerjaan-umum-dan-perumahan-rakyat-nomor-5-tahun-2023-tentang-persyaratan-teknis-jalan-dan-perencanaan-teknis-jalan>
- [2] N. Kaiser and C. K. Barstow, "Rural Transportation Infrastructure in Low- and Middle-Income Countries: A Review of Impacts, Implications, and Interventions," *Sustainability*, vol. 14, no. 4, p. 2149, May 2022, doi: 10.3390/su14042149.
- [3] Menteri PUPR, "Anggaran yang dialokasikan untuk tahap pertama pada jalan daerah," 2024, [Online]. Available: <https://binamarga.pu.go.id/index.php/berita/anggarkan-rp146-triliun-kementerian-pupr-mulai-perbaikan-jalan-daerah-bulan-juli>
- [4] T. B. Sembiring, B. P. Ndruru, S. T. Sitepu, Z. S. Siregar, and W. W. Tarigan, "Jalan Rusak Sebagai Ajang Pungutan Liar," vol. 4, no. 1, 2024, doi: 10.56799/peshum.v4i1.6718.
- [5] S. Sulartopo, S. Kholifah, D. Danang, and J. T. Santoso, "Transformasi Proyek Melalui Keajaiban Kecerdasan Buatan: Mengeksplorasi Potensi AI Dalam Project Management," *J. Publ. Ilmu Manaj.*, vol. 2, no. 2, pp. 363–392, Mar. 2023, doi: 10.55606/jupiman.v2i2.2477.
- [6] M. Azimi, A. Eslamlou, and G. Pekcan, "Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review," *Sensors*, vol. 20, no. 10, p. 2778, May 2020, doi: 10.3390/s20102778.
- [7] M. Jakubec, E. Lieskovská, B. Bučko, and K. Záborská, "Comparison of CNN-Based Models for Pothole Detection in Real-World Adverse Conditions: Overview and Evaluation," *Appl. Sci.*, vol. 13, no. 9, 2023, doi: 10.3390/app13095810.
- [8] B. Sasmito, B. H. Setiadji, and R. Isnanto, "Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra Deep Learning di Kota Semarang," *TEKNIK*, vol. 44, no. 1, pp. 7–14, Jan. 2023, doi: 10.14710/teknik.v44i1.51908.
- [9] X. Wang, H. Gao, Z. Jia, and Z. Li, "BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8," *Sensors (Basel)*, vol. 23, no. 20, 2023, doi: 10.3390/s23208361.
- [10] M. M. Ali, M. Faishol, and K. Anwar, "Deteksi Jalan Berlubang Menggunakan Metode Grey Level Co-Occurrence Matrix Dan Neural Network," *COREAI J. Kecerdasan Buatan, Komputasi dan Teknol. Inf.*, vol. 3, no. 1, pp. 1–8, Feb. 2022,

- doi: 10.33650/coreai.v3i1.4088.
- [11] M. F. Djulyansyah, G. Fitri Laxmi, and S. Agustian H, "Model Deteksi Jalan Untuk Smart Glasses Menggunakan Algoritma Yolo," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 7963–7970, Jan. 2024, doi: 10.36040/jati.v8i4.10620.
- [12] Q. H. Nguyen et al., "Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil," *Math. Probl. Eng.*, vol. 2021, pp. 1–15, May 2021, doi: 10.1155/2021/4832864.
- [13] J. Chappidi and D. M. Sundaram, "Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction," *IEEE Access*, vol. 12, pp. 110575–110587, May 2024, doi: 10.1109/ACCESS.2024.3439230.
- [14] G. Parsons, W. Y. Liu, and T. Zahra, "A Novel Wind Turbine Fault Diagnosis Model Based on Pre-Trained YOLOv8 Deep Transfer Learning," *Eng. Res. Express*, May 2025, doi: 10.1088/2631-8695/adc3ea.
- [15] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies*, vol. 11, no. 2, p. 40, May 2023, doi: 10.3390/technologies11020040.
- [16] C. Fang et al., "Enhancing Livestock Detection: An Efficient Model Based on YOLOv8," *Appl. Sci.*, vol. 14, no. 11, p. 4809, May 2024, doi: 10.3390/app14114809.
- [17] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision," May 02, 2024, arXiv. doi: 10.48550/arXiv.2407.02988.
- [18] P. Zhao, W. Zhou, and L. Na, "High-precision object detection network for automate pear picking," *Sci. Rep.*, vol. 14, no. 1, p. 14965, May 2024, doi: 10.1038/s41598-024-65750-6.
- [19] N. R. Muntari, I. C. Nisa, A. Sriekeaningih, A. Y. A. Prasetyo, and M. Yusril, "Penerapan Algoritma YOLOv8 Dalam Identifikasi Wajah secara Real- Time menggunakan CCTV untuk Presensi Siswa," vol. 4, no. 3, 2024, doi: 10.51454/decode.v4i3.847.
- [20] Y. Li and S. Wang, "EGM-YOLOv8: A Lightweight Ship Detection Model with Efficient Feature Fusion and Attention Mechanisms," *J. Mar. Sci. Eng.*, vol. 13, no. 4, 2025, doi: 10.3390/jmse13040757.
- [21] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios," in 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), IEEE, May 2021, pp. 2778–2788. doi: 10.1109/ICCVW54120.2021.00312.
- [22] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, p. 52, May 2023, doi: 10.3390/computation11030052.
- [23] L. Alzubaidi, O. Al-Shamma, M. A. Fadhel, L. Farhan, J. Zhang, and Y. Duan, "Optimizing the Performance of Breast Cancer Classification by Employing the Same Domain Transfer Learning from Hybrid Deep Convolutional Neural Network Model," *Electronics*, vol. 9, no. 3, p. 445, May 2020, doi: 10.3390/electronics9030445.
- [24] C. F. G. Dos Santos and J. P. Papa, "Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–25, May 2022, doi: 10.1145/3510413.
- [25] A. Sushma Priya and G. Manasi, "A Review and Performance Analysis of Non-Linear Activation Functions in Deep Neural Networks," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 03, no. 03, 2021, doi: 10.1109/ISCID.2009.214.
- [26] Y. Li et al., "SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images," *Remote Sens.*, vol. 16, no. 16, p. 3057, May 2024, doi: 10.3390/rs16163057.
- [27] L. Shen, B. Lang, and Z. Song, "Infrared Object Detection Method Based on DBD-YOLOv8," *IEEE Access*, vol. 11, pp. 145853–145868, May 2023, doi: 10.1109/ACCESS.2023.3345889.
- [28] W. Rao, S. Chen, and D. Li, "Autonomous Aerial Vehicle Object Detection Based on Spatial Perception and Multiscale Semantic and Detail Feature Fusion," *IEEE Access*, vol. 13, pp. 42897–42909, May 2025, doi: 10.1109/ACCESS.2025.3547825.
- [29] J. Wei, Q. Wang, Z. Li, S. Wang, S. K. Zhou, and S. Cui, "Shallow Feature Matters for Weakly Supervised Object Localization," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, May 2021, pp. 5989–5997. doi: 10.1109/CVPR46437.2021.00593.
- [30] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, p. 336, May 2024, doi: 10.3390/computers13120336.
- [31] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," *Electronics*, vol. 10, no. 3, p. 279, May 2021, doi: 10.3390/electronics10030279.
- [32] Roboflow, "Cap DP Dataset and Pre-Trained Model by TS," 2023, [Online]. Available: <https://universe.roboflow.com/ts-25tom/cap-dp/model/1>