A Re-parameterized Multi-scale and Global Context Framework for Waste Detection in Vietnamese Street Scenes

Phat Hoang Ho[1,2], An Hoang Nguyen[1,2,3], Kien Trang[1,2,3], and Bao Quoc Vuong[1,2(✉)]

[1] School of Electrical Engineering, International University, Ho Chi Minh City 700000, Vietnam
vqbao@hcmiu.edu.vn
[2] Vietnam National University, Ho Chi Minh City 700000, Vietnam
[3] School of Information Technology, Monash University, Malaysia Campus, Subang Jaya 47500, Malaysia

Abstract. Roadside waste detection is important for smart environmental monitoring, but it remains difficult in Vietnamese street scenes because waste objects often appear with challenges such as high traffic density, diverse object scales, lighting conditions, and complex backgrounds. In this study, we propose a VGD-YOLO architecture specifically designed for complex Vietnam street scenarios. The proposed model integrates an enhanced backbone with RepHMS blocks to improve feature representation while maintaining computational efficiency, along with a refined detection head augmented with RepViT modules to strengthen global feature modeling and contextual understanding. By combining efficient structural re-parameterization and transformer-inspired representation learning, the proposed architecture enhances both localization accuracy and classification robustness. Experiments were conducted on a real-world Vietnamese roadside waste dataset of about 2,000 images covering four categories: plastic bottles, plastic bags, paper, and other waste. Results show that the proposed model outperforms several YOLO baselines. The final configuration achieves 0.487 precision, 0.447 recall, 0.433 mAP@50, and 0.290 mAP@50–95, demonstrating robust performance in complex roadside environments.

Keywords: YOLO · deep learning · ViT · street waste detection

1 Introduction

The plastic waste, often referred to as "white pollution," is one of the main threats to the global environment [1]. The severity of plastic waste is especially serious in Vietnam, driven by rapid industrialization, economic expansion, and rising consumerism [2]. According to the technical report from the World Bank [3], Vietnam discharged about 2.8 to 3.1 million tons of plastic waste annually on land, while discharging approximately 0.28 to 0.73 million tons of plastic waste to the oceans, resulting in it being one

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 413–427, 2027.
https://doi.org/10.1007/978-3-032-30491-9_26

of the five highest nations polluting the ocean [4]. In addition, approximately 10–15% of plastic waste is uncollected in urban areas and 45–60% in rural areas, while only 10% is properly managed or recycled [5]. The critical issue of plastic pollution in Vietnam requires multiple approaches to address the problem, including enforcing laws and decrees, increasing awareness, promoting recycling, and sorting and managing plastic waste through innovative technologies, etc.

In the context of Vietnam's urban and suburban roads, where waste objects vary significantly in size, shape, and appearance and are often partially occluded by traffic or surrounding structures, robust real-time detection is essential. The rapid development of Deep Learning (DL) research has enabled the deployment of DL models for plastic waste detection, enabling more efficient and scalable approaches in modern waste management and sustainability systems [6, 7]. On the other hand, You Only Look Once (YOLO)-based models have been widely applied to environmental monitoring and management systems, particularly for vision-based waste detection and classification [8–10]. In general, YOLO is an end-to-end object detection framework designed to achieve high accuracy while maintaining real-time performance. Its ability to accurately localize and classify objects in complex outdoor environments makes it well-suited for roadside trash analysis using street-level imagery.

Motivated by these advances, this study proposes an enhanced YOLO-based architecture, VGD-YOLO, specifically designed for roadside trash detection and classification in complex Vietnamese street environments, where objects often exhibit large-scale variations, irregular shapes, dense distributions, and cluttered backgrounds. The main contributions of this work are summarized as follows:

- We designed an improved YOLO backbone incorporating RepHMS modules that leverage structural reparameterization to enhance feature extraction while maintaining computational efficiency. This design improves robustness when dealing with visually complex roadside scenes.
- We enhance the detection head by integrating RepViT modules, enabling more effective global context modeling and long-range dependency capture through transformer-inspired mechanisms. This improvement is particularly beneficial for detecting small, low-contrast, and partially occluded waste objects.
- We evaluate the performance of the proposed model on an original Vietnam roadside trash dataset, against other baseline YOLO models. The experiment results demonstrate that the proposed architecture achieves superior performance across Precision, Recall, mAP 50, and mAP 50–95 metrics.

1. Related Work

Conventional approaches to the plastic waste detection problem rely on Convolutional Neural Network (CNN)-based models. Several models, including Xception, MobileNet, DenseNet, and InceptionResnet, were reported in [11] to classify the TrashNet dataset [12], with DenseNet121 and DenseNet169 achieving 95% accuracy after hyperparameter tuning. In another study [13], which implements the conventional TrashNet and ScrapNet datasets with additional plastic waste images, ResNet and EfficientNet were reported to achieve high accuracy up to 98%. Although the experimental

results from the above models are good, they are challenging for real-time applications, particularly in terms of inference time and computational resources.

In comparison to the CNN-based models, YOLO-based methods also achieve notable results in plastic waste detection and classification problems. In [9], the YOLOv7 framework is implemented for the real-time waste detection problem. The YOLOv7 architecture achieved 80.1% Precision, 86.8% mAP 50, and 61.8% mAP 50–95 on an in-house dataset. In another study [14], the proposed YOLO-VG model is based on the YOLOv8 framework for waste detection and classification. The YOLO-VG model replaces the conventional CNN architecture with the GSConv block to reduce redundant information, while introducing the ODConv module and the VoV-Ghost structure at the neck network to improve the model's adaptability and generalization. The YOLO-VG model scored 88.4% in mAP 50, achieving 24.6% better computational efficiency and reducing the architecture size by 20.4% compared to the original YOLOv8 structure. In study [8], the authors implement a YOLOv12-based framework for waste detection and classification on an in-house dataset. The YOLOv12-based model achieved 73% Precision and 78% mAP 50 after 100 training epochs, confirming its capability for waste management challenges. In study [10], the authors propose the WCD-YOLO model, based on the YOLOv10 backbone, with an additional MCA module to broaden the model's perceptual range and a FNC2f-BiFPN feature pyramid network to enhance detection of plastic waste targets. The WCD-YOLO models achieved 95.8% in mAP 50 and 74% in mAP 50–95 on an in-house dataset, with an extremely light architecture of 7.2 M parameters and GFLOPs of 8.5G.

From the above studies, it is clear that YOLO-based architectures have the potential to solve the waste detection problem, even in extremely complex street scenes in Vietnam, especially in later frameworks with more robust feature extraction and broader attention mechanisms. In this study, we propose the VGD-YOLO model, an YOLOv11-based framework for the roadsides waste detection problem. The experimental results indicate that the proposed model has outperformed conventional YOLO models and has advanced research in the field of waste detection and classification.

2 Methods

2.1 VGD-YOLO Architecture

In this study, the VGD-YOLO model is proposed for Vietnam roadsides trash detection, which has its own custom architecture, as depicted in Fig. 1. Similar to other YOLO versions, the proposed model consists of three main components: backbone, neck, and head. Based on the YOLOv11 framework, Re-parameterized Hierarchical Multi-Scale module (RepHMS) modules are introduced into the backbone to enhance multi-scale feature extraction while maintaining computational efficiency. Additionally, SPPF(Spatial Pyramid Pooling - Fast) and C2PSA (Convolutional block with Parallel Spatial Attention) modules are integrated at the end of the backbone to strengthen contextual representation and spatial attention. In the neck, feature fusion is performed using upsampling and concatenation operations combined with C3K2 (Cross Stage Partial with kernel size 2) blocks to obtain more effective multi-scale features. Furthermore, the Re-parameterized Vision Transformer (RepViT) module is incorporated before the

detection head to enhance global feature representation at deeper detection scales. The architecture of the proposed model is shown in the figure below.

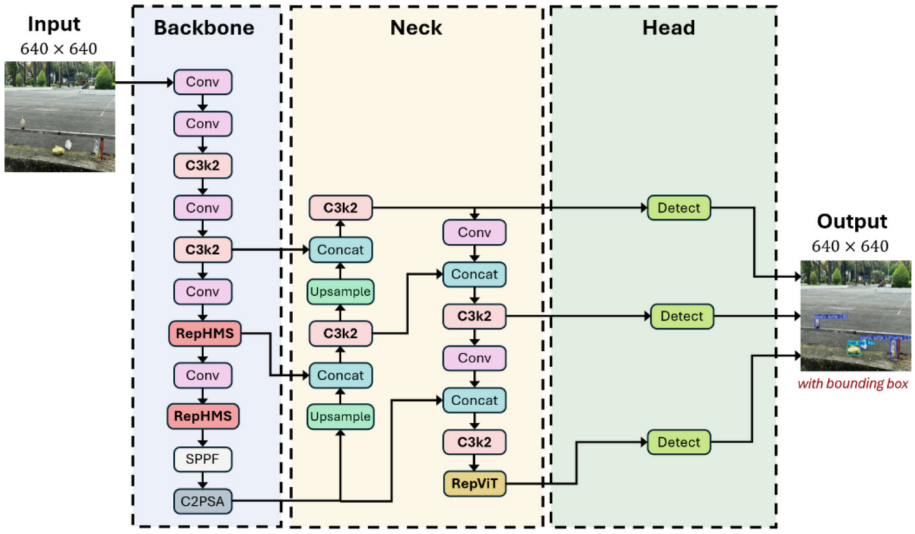

Fig. 1. Our proposed VGD-YOLO architecture

2.2 Re-parameterized Hierarchical Multi-scale Module

Framework in the proposed custom architecture, standard convolutional and C3K2 blocks are first employed to extract low- and mid-level features. To further enhance the high-level semantic representation before feature fusion, the RepHMS module is introduced at deeper stages of the backbone, as shown in Fig. 2.

RepHMS adopts a hierarchical multi-branch structure to jointly model shallow and deep features. The input feature map is first processed by a 1×1 convolution for channel adjustment and then split into multiple branches. Early branches preserve fine-grained spatial information, while deeper branches progressively extract richer semantic representations. Within each branch, feature transformation is performed using RepHDW (Re-parameterized Heterogeneous Depth-wise Convolution), which enables multi-scale spatial modeling through parallel depth-wise convolutions with heterogeneous kernel sizes. This design significantly enlarges the effective receptive field while maintaining computational efficiency.

To address the representational limitations of conventional depth-wise operations, RepHDW leverages a multi-branch training strategy that is analytically re-parameterized into a single equivalent depth-wise convolution during inference, ensuring no additional runtime overhead. The outputs of all hierarchical branches are then aggregated through feature concatenation and fused by a final 1×1 convolution, which restores channel interactions and scales the output channels to produce a compact yet expressive representation.

In the proposed architecture, RepHMS is strategically placed before the SPPF and C2PSA modules, ensuring that enriched, multi-scale, and semantically enhanced features are passed to both the spatial pyramid pooling and attention refinement stages. The overall structure of the RepHMS module is illustrated in Fig. 2, while the detailed formulation and re-parameterization mechanism of RepHDW are described in the following section.

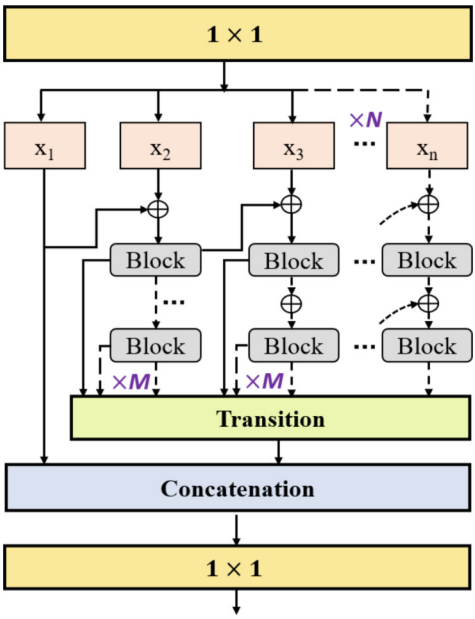

Fig. 2. RepHMS block

To be more specific, RepHDWConv is adopted as the fundamental convolutional operator within each hierarchical branch of the RepHMS module. This is achieved through a two-stage re-parameterization process. During training, it runs multiple depth-wise convolution branches in parallel, each with a different kernel size (e.g., 3×3, 5×5, 7×7). Each depth-wise convolution branch is followed by a batch normalization (BN) layer. Before merging kernels, BN must be absorbed into the convolution weights. The BN transformation is represented as Eq. (1):

$$BN\left(RepHDW\left(x\right)\right) = \gamma \cdot \frac{\left(RepHDW\left(x\right) - \mu\right)}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{1}$$

where x denotes the input feature map of the RepHDWConv block, RepHDW(x) represents the output feature map produced by the depth-wise convolution branch before batch normalization. μ and σ indicate the accumulated mean and standard deviation of the batch normalization layer, respectively, while γ and β correspond to the learned scaling factor and bias. ϵ is a small constant introduced to ensure numerical stability.

Let W_{fused} and B_{fused}, which are Eqs. (2) and (3), respectively, denote the fused convolution weight and bias after absorbing the batch normalization parameters into the

depth-wise convolution kernel. We will get a formula as

$$W_{fused} = \gamma \cdot \frac{RepHDW}{\sqrt{\sigma^2+\varepsilon}} \tag{2}$$

$$B_{fused} = \gamma \cdot \frac{(-\mu)}{\sqrt{\sigma^2+\varepsilon}} + \beta \tag{3}$$

After fusing with the BN layers, the convolution will be represented as Eq. (4):

$$BN(RepHDW(x)) = W_{fused(x)} + B_{fused} \tag{4}$$

In the second phase, the heterogeneous depth-wise convolutions must be merged into a single convolution, resulting in a feature map output. The formula of the feature map will be represented as Eq. (5):

$$O = I \otimes \left(K_{2n-1} + \sum_{i=1}^{m} K_{2n-(2i+1)} \right) + \left(B_{2n-1} + \sum_{i=1}^{m} B_{2n-(2i+1)} \right) \tag{5}$$

where I represents the input feature map of the re-parameterized convolution during inference, and O denotes the corresponding output feature map. K_n and B_n refer to the weight and bias of a depth-wise convolution with an $n \times n$ kernel, respectively, and $\otimes$ indicates the depth-wise convolution operation. Thus, n $\geq$ 3 corresponds to the largest kernel size used in the heterogeneous depth-wise convolution, while m is the largest integer satisfying $2n - (2m + 1) \geq 3$.

2.3 RepViT Block for Improving Neck Structure

RepViT is a re-parameterized vision Transformer architecture originally designed as a hierarchical feature extractor, where an initial convolutional stem reduces spatial resolution, followed by multiple stages that progressively extract higher-level semantic features through Transformer-based blocks, as shown in Fig. 3. Across these stages, RepViT leverages self-attention to model long-range dependencies while employing structural re-parameterization to significantly improve backbone and memory overhead during inference.

In the RepViT architecture, feature extraction begins with a 3 $\times$ 3 convolutional stem that reduces the spatial resolution of the input image. The network then proceeds through multiple hierarchical stages (Stages 1–4), each consisting of one or more RepViT blocks. Downsampling operations are applied between consecutive stages to progressively reduce spatial dimensions while increasing channel depth, allowing the model to concentrate on higher-level semantic features while keeping computational complexity moderate.

By applying RepViT in this manner, the model benefits from Transformer-style global dependency modeling and enhanced feature representation, while preserving the lightweight and end-to-end detection efficiency of YOLO models. This partial integration strategy allows RepViT to strengthen high-level feature interactions before the detection head, improving robustness in complex scenes without altering the original backbone structure.

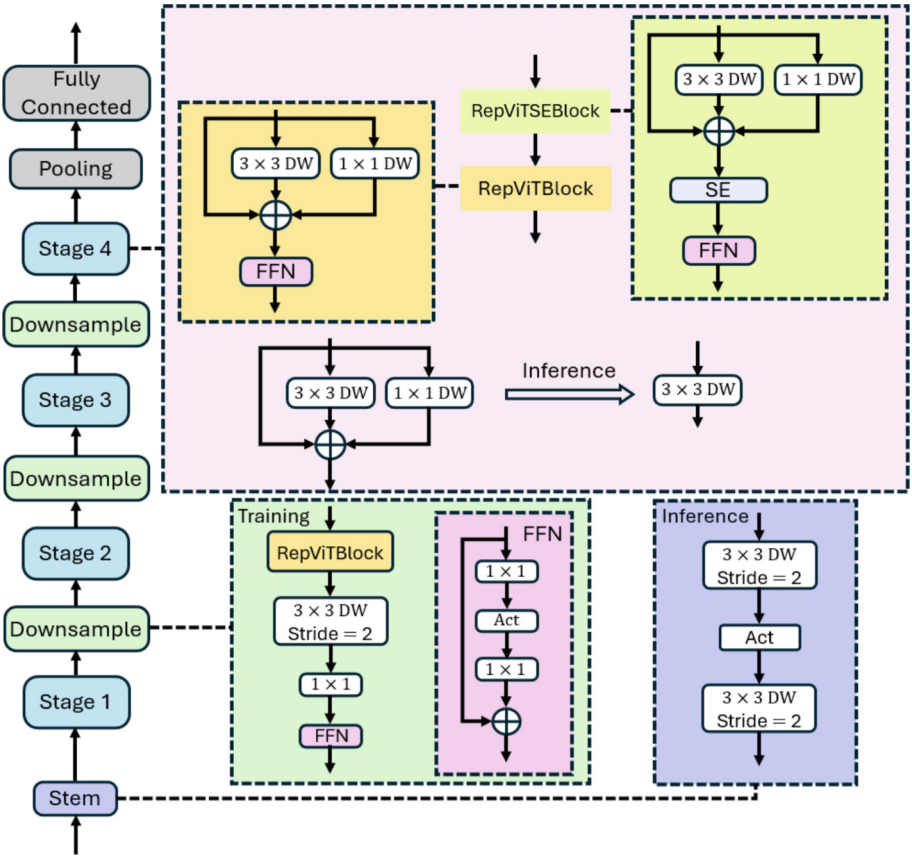

Fig. 3. RepViT architecture

3 Experiments and Results

3.1 Dataset

The dataset consists of approximately 2,000 waste images collected from real-world roadside environments in Vietnam. The images were captured across multiple locations, including Ho Chi Minh City and several other provinces, ensuring diversity in background, rain conditions, and scene complexity. The dataset contains 4 types of waste commonly found in roadside scenarios: plastic bottles, plastic bags, paper, and other types (fruit, masks, etc.). All classes were manually annotated to provide accurate ground-truth labels for model training and evaluation. This diverse data collection supports robust learning and improves the generalization capability of the proposed YOLO-based detection model. In each class, plastic bottles have 1606 instances, plastic bags have 1705, paper has 580, and others have 905, respectively.

3.2 Implementation Setting

All experiments were conducted in a Python 3.10.19 environment with PyTorch 2.5.1, accelerated by CUDA 12.1. The model was trained on a computer equipped with an NVIDIA GeForce RTX 3060 GPU with 12 GB of VRAM. The input images were resized to 640 × 640 pixels before training. The training epoch and batch size are set at 250 and 12, respectively. In addition, the ratio of the training and validation sets is 80:20, respectively.

3.3 Assessment Metrics

To comprehensively evaluate the performance of the proposed framework, several standard object detection metrics are used, including precision, recall, mAP@50, and mAP@50–95. Among these, mAP@50 and mAP@50–95 are considered the primary metrics for comparison with baseline models and for conducting ablation studies.

Precision measures the proportion of correctly predicted positive samples among all predicted positives and is defined as:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6}$$

where *TP* denotes true positive detections and *FP* represents false positive detections.

Recall evaluates the model's ability to identify all relevant objects in the dataset and is expressed as:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7}$$

where *FN* refers to false negatives, i.e., objects that are present but not detected by the model.

The metric mAP@50 represents the mean Average Precision computed at an Intersection-over-Union (IoU) threshold of 0.5, where a detection is considered correct if the IoU between the predicted and ground-truth bounding boxes is at least 50%. In contrast, mAP@50–95 is calculated by averaging the AP values over multiple IoU thresholds ranging from 0.5 to 0.95.

$$\mathbf{mAP_{50-95}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{K} \sum_{i=1}^{N} Ai, j \tag{8}$$

$$\mathbf{mAP_{50}} = \frac{1}{N} \sum_{i=1}^{N} AP_{50}, i \tag{9}$$

where N denotes the total number of object categories, and $AP_{50,\,i}$ represents the Average Precision of the i-th category evaluated at an IoU threshold of 0.50; K indicates the number of IoU thresholds used for evaluation, which is set to 10 and ranges from 0.50 to 0.95 with a step size of 0.05. $AP_{i,\,j}$ denotes the Average Precision of the i-th category computed at the j-th IoU threshold. The Average Precision is obtained from the precision–recall curve, which is determined by the number of true positives (TP), false positives (FP), and false negatives (FN) at different confidence levels.

3.4 Quantitative Results

Figure 6 shows the training progress with four evaluation metrics of our model. Almost all values reached the saturation point and fluctuated less after 100 epochs. Only mAP 50–95 reaches around 0.25, while the other metrics climb above 0.4. To evaluate the effectiveness of the proposed architecture, the VGD-YOLO model achieves the best overall detection performance among the compared methods, particularly in mAP@50 and mAP@50–95, reaching 0.433 and 0.290, respectively. These improvements aim to enhance localization accuracy and robustness across multiple IoU thresholds by using Re-parameterized Hierarchical Multi-Scale (RepHMS) modules in the backbone and the RepViT block in the neck.

In addition, RepViT is a lightweight transformer-based attention, enabling the model to capture long-range contextual dependencies. The figure below shows the comparison of the models' performance.

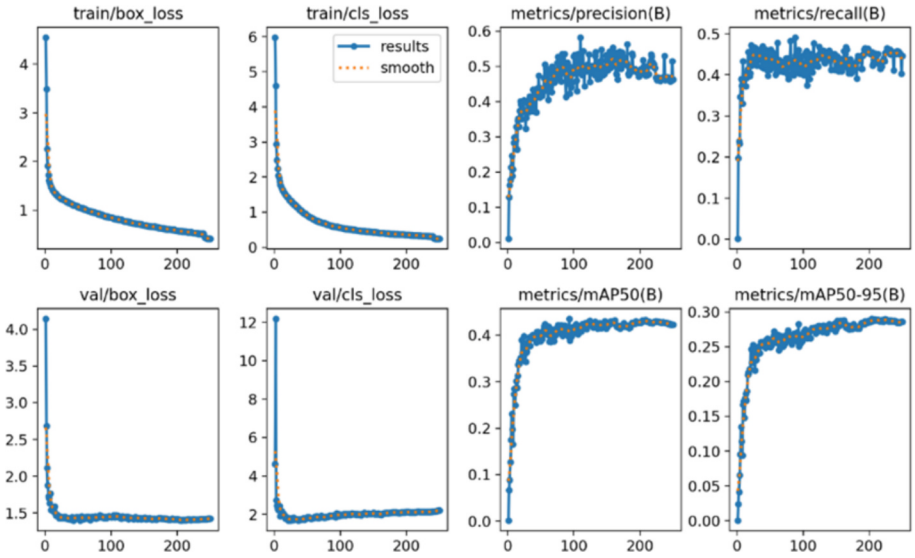

Fig. 4. The training progress of the proposed model.

Table 1. The comparison of performance between YOLO baseline and VGD

Model	Precision	Recall	mAP@50	mAP@50–95
YOLO11 [15]	0.474	0.432	0.417	0.275
YOLOv10 [16]	**0.492**	0.415	0.415	0.281
YOLOv9 [17]	0.462	**0.455**	0.422	0.280
YOLOv8 [18]	0.486	0.432	0.419	0.276
VGD-YOLO (ours)	0.487	0.447	**0.433**	**0.290**

3.5 Qualitative Results

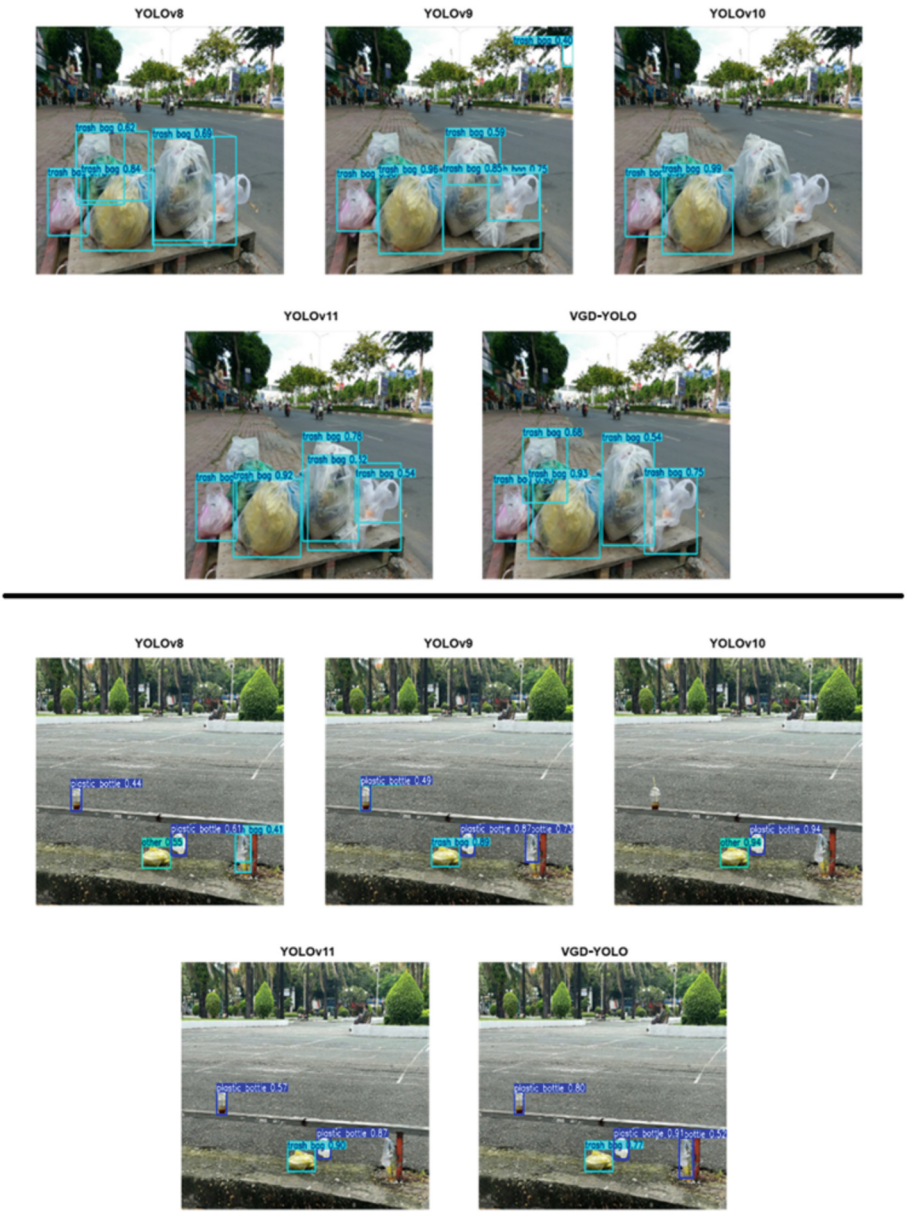

Fig. 5. The comparison of visualization between different models in daytime ideal conditions

In addition to the quantitative evaluation, we also perform detection experiments using other models to qualitatively compare their performance with that of the proposed framework. The visualization result of the proposed model in Fig. 5 shows that the custom model successfully detected all objects in the images. However, the variants of YOLO models missed the plastic bottles, which were shown in YOLOv10 and YOLOv11. Moreover, the YOLOv8 detected incorrect classes with duplicated problems and has relatively low confidence in true positive results. Similarly, in the trash bags class detection, the custom model shows superior detection capabilities than the YOLO variant model in terms of roadsides scenarios, whereas YOLO variants still struggle with the duplicated problems and missing objects in the ideal daytime conditions.

In the nighttime scenario, the proposed model still shows better performance, with a high confidence score in Fig. 6. According to the visualization results, all YOLO-based variants incorrectly detect plastic bottles, whereas VGD achieves high accuracy with a 0.76 confidence score.

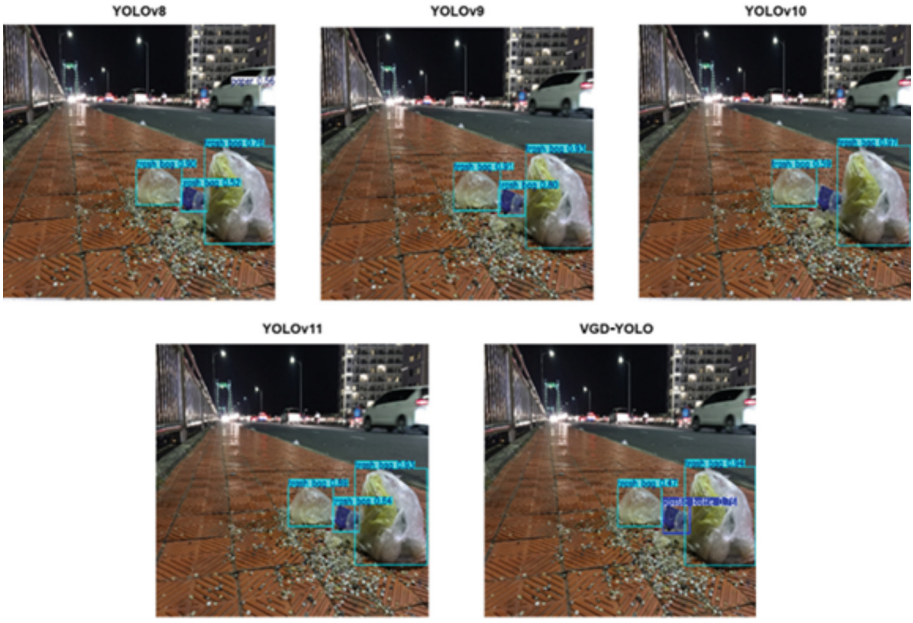

Fig. 6. The comparison of visualization between different models in nighttime ideal conditions

From the visual results of daytime non-ideal conditions, YOLOv11 can detect paper objects more correctly than the other model versions in Fig. 7. However, all models of YOLO detect the large trash bag object incorrectly instead of papers due to its white color. Overall, the proposed YOLO custom model can still detect small objects better than the other YOLO versions.

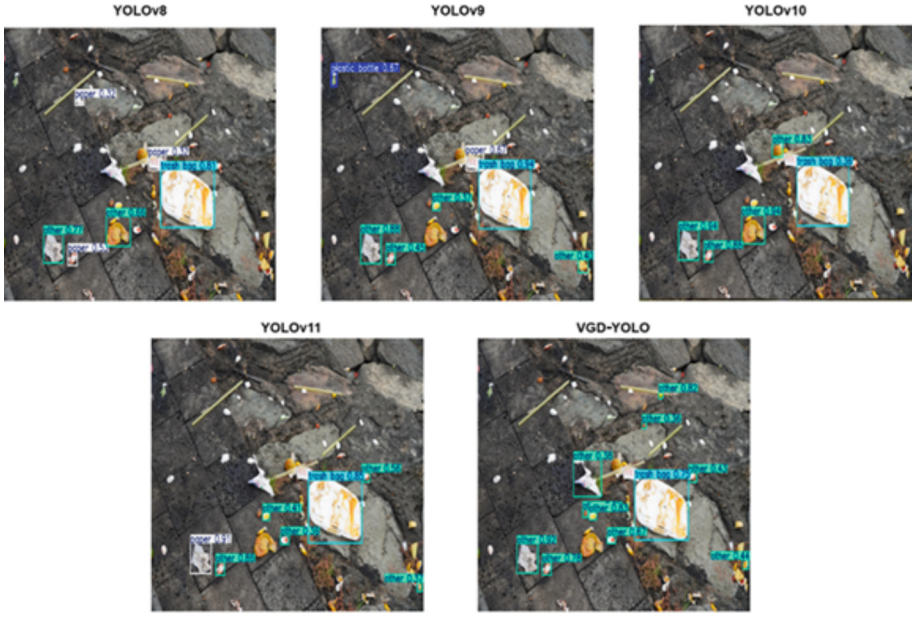

Fig. 7. The comparison of visualization between different models in daytime non-ideal conditions

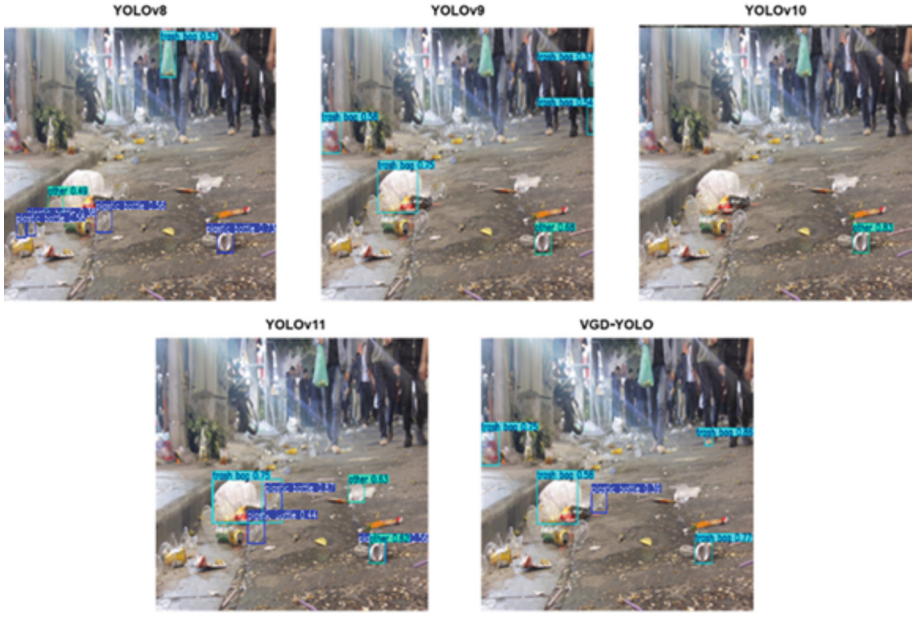

Fig. 8. The comparison of visualization between different models in nighttime non-ideal conditions

From the visual results of nighttime non-ideal conditions, YOLOv11 has superior detection in terms of plastic class. However, when detecting small objects, the VGD model shows better performance compared to YOLO based line, which is shown in Fig. 8.

3.6 Ablation Studies

In order to provide the comparative effects of implemented settings, we conduct several sets of ablation studies.

Effect of Feature Extraction Module
Based on the results shown in Table 2, the RepHMS module was selected for integration into the proposed framework. The table presents a performance comparison between several backbone block variants, including C3k2, C2f/C2fCIB, RepNCSPELAN4, and C2f. Among these alternatives, RepHMS achieves the most balanced and competitive performance across the evaluation metrics. Specifically, RepHMS obtains the highest mAP@50 of 0.432 and the best mAP@50–95 score of 0.284, indicating improved detection capability across different IoU thresholds. In terms of precision, RepHMS reaches 0.487, which is comparable to the highest precision achieved by C2f/C2fCIB (0.492), while maintaining a strong recall of 0.428. This balance between precision and recall contributes to the improved overall detection accuracy reflected in the mAP scores.

Table 2. The comparison of backbone methods between different blocks

Block	Precision	Recall	mAP@50	mAP@50–95
C3k2	0.474	0.432	0.417	0.275
C2f/C2fCIB	0.492	0.415	0.415	0.281
RepNCSPELAN4	0.462	0.455	0.422	0.280
C2f	0.486	0.432	0.419	0.276
RepHMS	**0.487**	**0.428**	**0.432**	**0.284**

Effect of Attention Mechanisms
Both proposed models benefit from the introduction of RepHMS, while Model B, which is our final proposed model, further improves performance by integrating RepViT on top of RepHMS. Model A, which uses only RepHMS, already achieves strong detection accuracy with an mAP@50 of 0.432 and an mAP@50–95 of 0.284, confirming that RepHMS effectively enhances local feature extraction and multi-scale representation. By adding RepViT, Model B achieves the best overall performance, reaching an mAP@50 of 0.433 and an mAP@50–95 of 0.290. Although the increase in mAP@50 appears modest, the mAP@50–95 shows a clear improvement, indicating that RepViT significantly improves precise bounding box localization, especially under stricter IoU thresholds. Both models maintain the same precision of 0.487, but Model B improves recall from 0.428 to 0.447, meaning that the RepViT-enhanced architecture detects more

waste objects without increasing false positives. This confirms that RepViT contributes effective global context modeling and long-range feature interaction, complementing RepHMS's strong local and multi-scale representations.

Table 3. The comparison of performance between model A **(Only RepHMS)** and model B **(RepHMS + RepVit)**

Model	Precision	Recall	mAP@50	mAP@50–95
Only RepHMS	0.487	0.428	**0.432**	**0.284**
RepHMS + RepVit	**0.487**	**0.447**	**0.433**	**0.290**

4 Conclusion

This paper presented the VGD-YOLO framework for roadside waste detection in complex Vietnamese street environments. The proposed model integrates RepHMS into the backbone to strengthen efficient multi-scale feature extraction and incorporates RepViT into the neck to enhance global contextual modeling and long-range feature interaction. Experimental results on a real-world Vietnamese roadside waste dataset show that VGD-YOLO achieves the best overall performance compared with several YOLO baselines. In particular, the final model reaches 0.487 in precision, 0.447 in recall, 0.433 in mAP@50, and 0.290 in mAP@50–95, demonstrating improved robustness and localization accuracy in challenging roadside scenes. Thus, the proposed framework shows promising potential for practical roadside waste monitoring and intelligent environmental management in Vietnam.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this paper.

References

1. Andronicеanu, A., Dragulanescu, I.-V.: A survey on the buyers' eco-responsibility and the urban white pollution. Environ. Eng. Manag. J. **15**, 481–487 (2016). https://doi.org/10.30638/eemj.2016.051
2. Salhofer, S., Jandric, A., Soudachanh, S., Le Xuan, T., Tran, T.D.: Plastic recycling practices in Vietnam and related hazards for health and the environment. Int. J. Environ. Res. Public Health. **18**, 4203 (2021)
3. World Bank: Vietnam: Plastic Pollution Diagnostics. World Bank, Washington, DC (2022)
4. Jambeck, J.R. et al.: Plastic waste inputs from land into the ocean. Science. **347**(6223), 768–771 (2015). https://doi.org/10.1126/science.1260352
5. World Bank: Toward a National Single-use Plastics Roadmap in Vietnam: Strategic Options for Reducing Priority Single-Use Plastics. World Bank, Washington, DC (2022)
6. Nahiduzzaman, M., et al.: An automated waste classification system using deep learning techniques: toward efficient waste recycling and environmental sustainability. Knowl.-Based Syst. **310**, 113028 (2025). https://doi.org/10.1016/j.knosys.2025.113028

7. Bobulski, J., Kubanek, M.: deep learning for plastic waste classification system. Appl. Comput. Intell. Soft Comput. **2021**, 6626948 (2021). https://doi.org/10.1155/2021/6626948
8. Dipo, M.H. et al.: Real-time waste detection and classification using YOLOv12-based deep learning model. Digital. **5**(2), 19 (2025). https://doi.org/10.3390/digital5020019
9. Rahmatulloh, A., Darmawan, I., Aldya, A.P.: Firmansyah Maulana Sugiartana Nursuwars.: WasteInNet: deep learning model for real-time identification of various types of waste. Cleaner Waste Systems. **10**, 100198 (2025). https://doi.org/10.1016/j.clwas.2024.100198
10. Ling, L., Chen, Y.: Zhiwu li.: WCD-YOLO: a waste classification detection model. J. Environ. Manag. **398**, 128601 (2026). https://doi.org/10.1016/j.jenvman.2026.128601
11. Aral, R.A., Keskin, Ş.R., Kaya, M., Hacıömeroğlu, M.: Classification of TrashNet Dataset Based on Deep Learning Models. In: 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, pp. 2058–2062, IEEE (2018), doi:10.1109/BigData.2018.8622212
12. Thung, G., Yang, M.: Classification of Trash for Recyclability Status (2016).
13. Masand, A., Chauhan, S., Jangid, M., Kumar, R., Roy, S.: Scrapnet: an efficient approach to trash classification. IEEE Access. **9** (2021). https://doi.org/10.1109/ACCESS.2021.3111230
14. Song, L., Yu, H., Yang, Y., et al.: YOLO-VG: an efficient real-time recyclable waste detection network. J. Real-Time Image Proc. **22**, 79 (2025). https://doi.org/10.1007/s11554-025-01655-5
15. Glenn, J.: YOLOv11. GitHub. (2024) https://github.com/ultralytics/ultralytics/releases/tag/v8.3.50
16. A. Wang et al.: YOLOv10: Real-Time End-to-End Object Detection. arXiv (2024). https://arxiv.org/abs/2405.14458
17. C.-Y. Wang, I.-H. Yeh, H.-Y. M. Liao.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv (2024). https://arxiv.org/abs/2402.13616
18. Jocher, G., Qiu, J., Chaurasia, A.: YOLOv8. Github (2023) https://github.com/ultralytics/ultralytics