2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

YOLOCrane: An Enhanced YOLOv8-Based Algorithm for Robust Crane Detection in Transmission Line Scenarios

Xiaolong Wang[1,2][0000-0003-1644-4899], Bo Jiang[2], Yanwei Zhang[3],

Genyi Wang[3][0000-0001-8634-8451], and Guocheng An[3]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, 310018, Zhejiang, China
[2] The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai, 201808, China
[3] Artificial Intelligence Research Institute of Shanghai Huaxun Network System Co., LTD., Chengdu, 610074, China
jiangbo@ecict.com

Abstract. Crane detection under transmission lines, crucial for power system safety monitoring, faces challenges in accuracy and generalization, particularly in environments with structural interference (e.g., utility poles) and dynamic vegetation occlusion. To address these issues, we propose YOLOCrane, an enhanced YOLOv8-based algorithm. First, a simplified backbone network reduces computational complexity by 33% while maintaining robust feature extraction. Second, by incorporating learnable mask parameters and multi-channel fusion, the channel-wise LBP algorithm adaptively extracts texture features across dimensions, addressing the limitations of traditional LBP in fixed 3×3 windows. Finally, a heterogeneous dual-branch attention fusion module integrates convolutional features with LBP texture patterns, enabling complementary learning of spatial and texture information. Experimental results on the CraneLine dataset demonstrate that YOLOCrane achieves an $mAP_{0.5}$ of 85.8%, surpassing YOLOv8x by 2.4%, YOLOv11x by 2.1%, and RT-DETRx by 1.7%, while improving inference speed by 9.72 FPS. These advancements underscore YOLOCrane's capability to tackle detection challenges in complex environments, providing a robust solution for real-time safety monitoring of transmission lines.

Keywords: Crane detection under transmission lines, simplified backbone (SB), channel-wise LBP, dual-branch attention Fusion (DAF).

1 Introduction

As a critical component of power systems, the safe and stable operation of transmission lines is essential for the sustainable development of the socio-economy [1]. With the continuous expansion and increasing complexity of transmission line networks, there is a growing demand for safety monitoring of surrounding construction activities [2]. This is critical in scenarios where cranes frequently operate near transmission lines. Ensuring a safe distance between cranes and transmission lines to prevent power accidents caused by accidental collisions has become an urgent and significant challenge to that must be addressed [3].

1.1 Related Work

The advancement of artificial intelligence technologies, particularly the widespread application of deep learning techniques, has opened new avenues for crane detection. For instance, object detection algorithms based on Convolutional Neural Networks (CNNs), such as Faster R-CNN [4] and the YOLO series [5-6], have demonstrated remarkable performance in general object detection tasks. Recent studies have further explored specialized methods for crane detection. Chian et al. [7] utilized CenterNet to dynamically identify potential areas for crane load drops, while Lu et al. [8] proposed a Swin Transformer-based method for crawler crane detection, leveraging drone-captured images to identify hazardous work zones and provide real-time risk information to operators. Sun et al. [9] enhanced the detection capabilities of crane hooks and personnel by improving the YOLOv5 network, albeit at the cost of increased model parameters and reduced inference speed. Xia et al. [10] introduced a rotation-based object detection paradigm using YOLOv8, achieving high-precision detection for cranes with high aspect ratios, although with limited generalization ability in complex environments.

1.2 Motivations

Despite these advancements, existing algorithms still face significant limitations in crane detection within transmission line scenarios. The primary issues stem from a singular improvement strategy that relies solely on increasing model parameters, as well as the challenges associated with generalization in complex environments. Furthermore, most current research focuses on general object detection [11-12], resulting in a notable lack of specialized studies focused on crane detection in transmission line scenarios. Given the exceptional performance of the YOLO algorithm in the field of object detection [13-14], we have improved the YOLOv8 model to create a specialized algorithm, YOLOCrane, specifically designed for crane detection in transmission line scenarios, scenarios. This development achieves state-of-the-art performance within the YOLO series for crane detection tasks. In summary, our contributions are as follows:

1) Simplified Feature Extraction Network: The optimized backbone of YOLOv8x mitigates overfitting risks and enhances computational efficiency by eliminating redundant module stacking during feature extraction.

2) Channel-wise LBP: The enhanced LBP algorithm significantly improves the

computational efficiency of the multidimensional feature maps and demonstrates its excellent texture extraction capability on different networks and datasets.

3) Dual-Branch Attention Architecture: The network adaptively integrates convolutional features with LBP features, optimized within the YOLOv8 framework, achieving exceptional generalization performance in crane detection tasks.

2 Methodology

The network architecture of YOLOCrane, as illustrated in Fig. 1, is composed of three key components: Backbone, Neck, and YOLO Head. The Backbone is primarily responsible for extracting hierarchical features from the input image at various scales. The Neck integrates feature maps from different stages of the backbone, enriching semantic information and improving the accuracy of target feature representations. Finally, the YOLO Head serves as the output layer, generating bounding boxes and category predictions based on the extracted feature maps to achieve precise object detection. This well-structured design ensures efficient feature extraction, robust feature fusion, and accurate detection output, making YOLOCrane highly effective for crane detection tasks in complex environments.

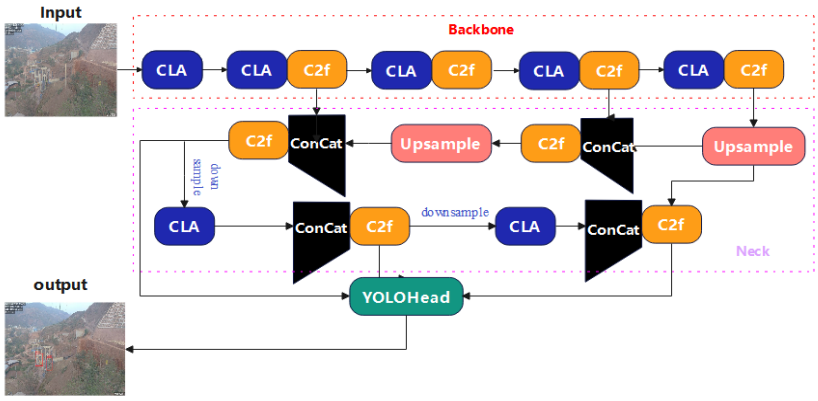


Fig. 1. Architecture of the YOLOCrane network.

Simplification of The Backbone Feature Network. The backbone feature network of the YOLOv8 algorithm is innovatively based on the CSPDarkNet-53 architecture [15], introducing the C2f structure to replace the original C3 structure. The C2f structure not only inherits the advantages of CSPDarkNet-53 but also optimizes gradient flow, thereby enhancing model performance. In the backbone feature network, the stacking counts of C2f are initially set to 3, 6, 6, and 3, while the model size can be flexibly controlled by adjusting the width and depth factors. This design ensures a balance between computational efficiency and feature extraction capability, making the network adaptable to various detection tasks.

In the original YOLOv8 backbone, the stacking configuration of the C2f layers is {3, 6, 6, 3}. When applied to the single-class task of crane detection, this redundant

stacking is likely to lead to model overfitting. Specifically, while the performance metrics during training are strong, the detection accuracy on the test set significantly decreases. To address this issue, we adjusted the stacking counts of the C2f structure in the backbone feature network to {1, 1, 1, 1}.

The simplified backbone design aligns with the Information Bottleneck Principle , which optimizes feature representation by eliminating redundant layers that propagate noise while retaining task-critical features. As demonstrated in Table 2, this architectural refinement improves the test set AP50 from 95.5% to 97.6% (↑2.1%), confirming its enhanced generalization capability. Specifically, for the YOLOv8x model with 480×480 input resolution (Fig. 2), the parameter size is reduced by 42 MB (33% reduction) through the adjustment of C2f layer stacking from {3,6,6,3} to {1,1,1,1}.

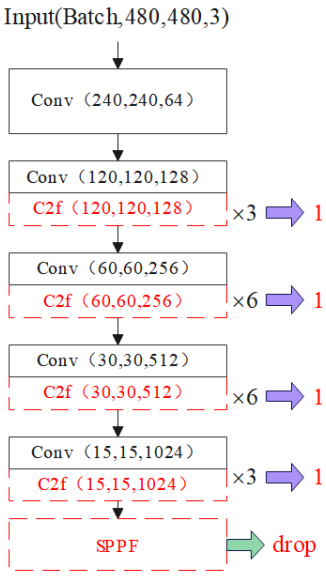


Fig. 2. Simplified backbone architecture: The original C2f stacking {3,6,6,3} (left) is reduced to {1,1,1,1} (right), with parameter size decreased by 33%

Additionally, although SPPF (Spatial Pyramid Pooling Fusion) effectively captures feature information through its multi-scale pooling operations, we chose to exclude SPPF in the crane detection task due to the relatively fixed size of cranes. This decision aims to simplify the model structure and reduce the number of parameters (by approximately 2.5 MB, thereby enhancing the model's operational efficiency.

Channel-wise LBP. The Local Binary Pattern (LBP) algorithm [17] is a widely used operator for describing local texture features in images, renowned for its significant grayscale invariance. However, the application of traditional LBP algorithms is limited because they only characterize textures within a small fixed-radius region (e.g., 3x3). The LBP algorithm can be summarized as follows:

$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c) \tag{1}$$

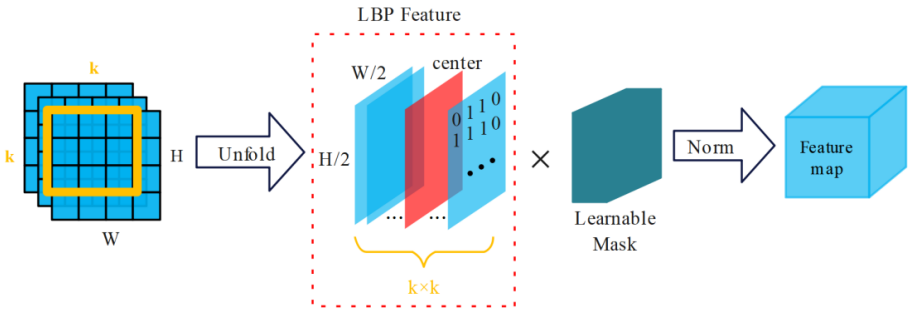


Fig. 3. The Channel-wise LBP module. Extracting channel LBP Features combined with learnable mark to obtain normalized output feature map

In the LBP algorithm, (x_c, y_c) represents the center point within a 3×3 window, while i_c and i_c denote the grayscale values of the center point and its neighboring pixels, respectively. The function $s(\cdot)$ is a binary function defined as follows:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & else \end{cases} \tag{2}$$

In contrast, feature maps generated by convolutional neural networks (CNNs) not only exhibit higher resolution but also contain a greater number of channels. Consequently, directly applying the LBP algorithm to convolutional operations significantly increases computational complexity [18]. This limitation restricts its use in real-time applications or resource-constrained environments.

To address these limitations, we propose a channel-wise LBP feature extraction method that integrates LBP and convolutional features with learnable mask parameters and normalization. This method efficiently processes multi-dimensional feature maps, capturing intricate texture patterns by analyzing inter-channel feature variations. It is particularly effective in transmission line scenarios, where the texture distinctions between cranes and background elements (e.g., utility poles and vegetation) are more pronounced, as illustrated in Fig. 3.

Given an input feature map I with dimensions (N, C, H, W), we utilize the unfold operation with a window size of k and a stride of s during feature extraction. This operation selects elements from the second and third dimensions of the feature map and concatenates them along an additional channel dimension using R_c, constructing the neighborhood N_0. The expression is as follows:

$$N_o = I.unfold([2,3], \text{kernel\_size} = k, \text{stride} = s) \tag{3a}$$

$$N_o = R_c(N_o) \tag{3b}$$

Drawing inspiration from the LBP algorithm, we first identify the central feature map $Center^{i,j}$ along the channel dimension of N_o. This center map is then compared element-wise with the feature maps of the remaining channels in N_o, generating a channel-wise binary map that reflects feature differences across channels. Next, we perform an element-wise multiplication between this set of binary values and a learnable mask feature map M, resulting in the computation of the value L for N_o across both the width and height dimensions. This process can be expressed as:

$$\begin{cases} L^{i,j} = M^{i,j} \ if \ N_o^{i,j} > Center^{i,j} \ \text{else} \ 0 \quad i,j \in W,H \\ Center = \delta(N_o, m) \end{cases} \tag{4}$$

$$\frac{\partial L}{\partial M} = \sum_{i,j} \mathrm{I}(N_o^{i,j} > Center^{i,j}) \cdot \nabla_\theta L \tag{5}$$

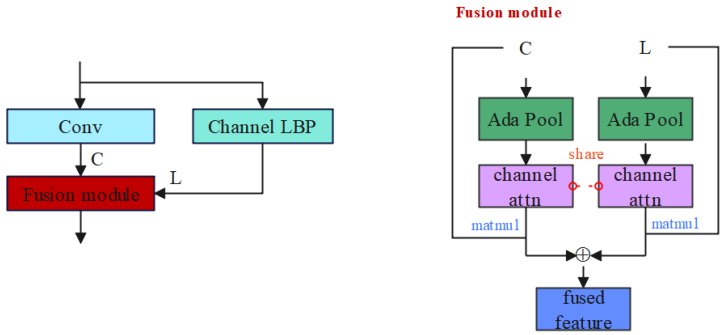
where $\frac{\partial L}{\partial M}$ denotes the gradient of the loss function with respect to the learnable mask. $\mathrm{I}(\cdot)$ is the indicator function. $\nabla_\theta L$ is the gradient of the overall loss function L with respect to the model parameter θ. The summation covers all spatial locations (i.e., height H and width W dimensions)

This approach not only captures fine-grained variations in features but also utilizes learnable parameters to adaptively concentrate on the most informative regions. This significantly enhances the model's ability to extract discriminative features, especially for tasks such as crane detection in complex environments.

After traversing the height H and width W dimensions, we obtain the sparse LBP feature map L_c. By summing L_c along the channel dimension using $sum(\cdot,2)$ and applying normalization $norm$ to ensure robustness against variations in scale and intensity, we derive the final output of the channel-wise LBP, denoted as F_o. This process can be expressed as:

$$F_o = norm(sum(L_c,2)) \tag{6}$$

Dual-branching Attention Mechanism Fusion (DAF) Strategy. To enhance the performance of YOLOv8, we introduce a Channel-wise Local Binary Pattern (LBP) module and propose a Dual-Branch Attention Mechanism Fusion Strategy (DAF), as illustrated in Fig. 4. The simplified backbone reduces redundant parameters (by 33%, Fig. 2), while the dual-branch attention compensates for potential information loss by adaptively fusing LBP texture features, achieving a balance between efficiency and accuracy (Table 2). The DAF module is utilized for downsampling operations within both the backbone feature network and the feature fusion network. It integrates convolutional features and LBP features by leveraging an attention mechanism to enable self-learning attention between the two types of features. This approach fully exploits the strengths of both feature types, thereby enhancing the model's ability to capture detailed texture information and spatial relationships.



(a) DAF outer layer (b) Fusion module internal structure

Fig. 4. Structure of the dual-branch attention fusion (DAF) module.

In the Fusion module, the convolutional feature branch C and the LBP feature branch L maintain consistent input dimensions and compute attention scores using shared weights. This weight-sharing mechanism fosters mutual supervision during the feature fusion process, thereby enhancing the model's generalization capability [19]. Furthermore, weight sharing allows the convolutional branch and the LBP branch to utilize shared computational logic and hardware resources during attention calculation, thereby, improving computational efficiency. This process can be expressed as follows:

$$DAF = \theta_{share}(C, L)) \tag{7}$$

Where θ_{share} represents the Fusion module of the shared weight attention mechanism, and the input-output feature graph maintains consistent.

3 Experiments

This section evaluates the performance of YOLOCrane using the CraneLine dataset [20], which features synthetic images of cranes set against transmission line backgrounds. The dataset comprises 4,739 training images and 300 validation images, each containing 11 cranes of varying sizes and positions, along with 100 real-world test images. This configuration ensures a thorough assessment of the model's accuracy, robustness, and generalization capabilities in both synthetic and real-world scenarios.

The experimental environment was configured as follows: Ubuntu 20.04 operating system, NVIDIA A800-160GB GPU, and the PyTorch 2.1 framework. During training, all models were trained for 300 epochs with a batch size of 16. The first 5 epochs were used for model warm-up, and the initial learning rate was set to 0.008. Both training and testing images were resized to 480×480 pixels, and the AdamW optimizer was employed. The evaluation metrics included mAP_{50}, precision (P), and recall (R) to comprehensively assess model accuracy and performance.

3.1 Performance of Channel-wise LBP

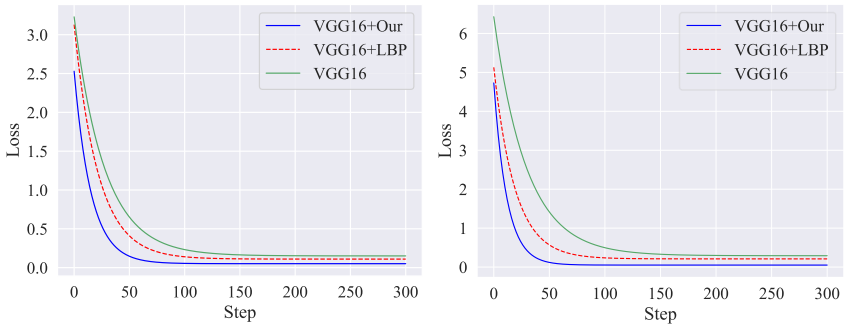
To validate the performance of Channel-wise LBP across different datasets, we selected VGG16 [21] due to its straightforward network architecture as the baseline model. Based on this, we conducted comparative experiments on the CIFAR-10 and CIFAR-100 datasets, comparing the LBP pooling algorithm proposed in [18] with our own method. All experiments maintained consistent training configuration parameters to ensure a fair comparison. This evaluation aims to demonstrate the effectiveness and generalization capability of our improved LBP algorithm in enhancing feature extraction and model performance across diverse datasets.

Table 1. Comparison of the performance metrics on CIFAR-10 and CIFAR100 datasets.

Datasets		method	Acc (Train)	Acc (Val)	Train-time
CIFAR10	VGG16	--	97.1	91.4	3.1h
		+ LBP	97.7	92.8	10.4h
		+Our	**98.5**	**93.1**	3.2h
CIFAR100		--	92.9	69.8	9.5h
		+ LBP	93.7	71.2	30.7h
		+Our	**95.4**	**73.6**	9.7h

As presented in Table 1, our method demonstrates higher accuracy on both datasets compared to the original LBP algorithm,with only a minimal increase in training time. The performance improvement is particularly notable on the more complex CIFAR-100 dataset, where training accuracy and validation accuracy increase by 2.5 percentage points and 3.8 percentage points, respectively. This clearly illustrates the effectiveness of our method on complex datasets and highlights its unique advantage in enhancing model generalization capabilities. These results underscore the robustness and adaptability of our improved LBP algorithm in diverse and challenging scenarios.

In Fig. 5, the blue curve represents our method, showing a rapid decline in loss during the early training phase. As training progresses, the loss continues to decrease, stabilizing at a low value after 50 epochs on both the CIFAR10 and CIFAR100 datasets. This demonstrates that our method efficiently accelerates training and achieves rapid convergence. The red curve corresponds to the combination of LBP and VGG16. While LBP provides some performance improvement for VGG16, the enhancement is limited. The loss decreases significantly in the early stages but flattens later, with a relatively higher final loss. The green curve represents the baseline VGG16, which exhibits a slower loss reduction and the highest final loss, indicating its limited performance on both CIFAR datasets.



(a) CIFAR10 (b) CIFAR100

Fig.5 Comparison of the training loss of the three algorithms on different data sets

Our method achieves the fastest loss reduction and the lowest final loss, highlighting its advantages in feature learning and generalization capability. In contrast, traditional LBP offers only marginal improvements, likely due to its fixed mask feature selection and computationally intensive single-channel feature map processing, which restrict further performance gains. These results underscore the superiority of our approach in enhancing model training efficiency and generalization across diverse datasets.

3.2 Ablation experiment

In this section, we conduct ablation studies to validate the effectiveness of the proposed methods. We select YOLOv8x as the baseline network and perform ablation experiments to evaluate the impact of the simplified backbone and the Dual-Branch Attention Fusion (DAF) module at various positions within the network architecture. Here, SB denotes the simplified backbone, DAF(B) represents the application of DAF in the Backbone, and DAF(N) indicates the use of DAF in the Neck. These experiments aim to systematically analyze the contributions of each component to the overall performance of the model.

Table 2. Ablation experiment of YOLOv8x on CraneLine dataset.

Baseline	SB	DAF (B)	DAF (N)	mAP_{50}	P	R	AP_{50}(test)	Params	FPS
YOLOv8x	--	--	--	83.4	90.9	72.1	95.5	130MB	37.81
	√	--	--	81.3	93.6	69.7	97.6	85.5MB	47.65
	√	√	--	82.6	94.1	69.5	97.9	87.3MB	47.61
	√	--	√	82.2	93.9	70.1	97.8	86.2MB	47.64
	√	√	√	85.8	94.5	71.8	98.1	88MB	47.53

As shown in Table 2, simplifying the backbone of YOLOv8x for the crane detection task results in a decrease in accuracy metrics on the training set. However, the simplified model demonstrates higher accuracy on the test set. Analyzing precision (P) and

recall reveals that the more complex model may learn specific noise or redundant information from the training set, performing well on similarly distributed data but being prone to overfitting, as indicated by decreased P and increased R. In contrast, the simplified model exhibits stronger generalization capabilities across different domains, leading to better performance on the test set. Additionally, the simplified model offers significant advantages in terms of parameter reduction and inference speed, making it more efficient and practical for real-world applications.

3.3 Comparison of YOLOCrane with SOTA Models

To ensure a fair comparison, we evaluated YOLOCrane against state-of-the-art (SOTA) object detectors using the same training configuration and at a similar scale. The results, presented in Table 3, demonstrate that the proposed algorithm outperforms other methods across all four evaluation metrics, showcasing its exceptional performance. This underscores the effectiveness of YOLOCrane in achieving superior accuracy, precision, recall, and mAP.

Table 3. Performance comparison between YOLOv8x and SOTA model with similar scale on CraneLine datastet.

Algorithms	AP_{50}	AP_{50-95}	AP_{50}(test)	FPS
YOLOv5x	83.1	59.8	95.1	38.19
TPH-YOLOv5x	83.4	60.6	96.3	21.42
YOLOv11x	83.7	61.1	96.0	29.54
RT-DETRx	84.1	60.3	96.7	37.2
YOLOCrane	**85.8**	**61.7**	**98.1**	**47.53**

To further validate the generalization capability of our method, we selected 4 representative images from the 2024 China Southern Power Grid AI Competition Crane Dataset and compared YOLOCrane with YOLOv11x [22], RT-DETRx [23], and TPH-YOLOv5x [24].

As shown in Fig. 6, the selected three test images feature cranes with small proportions and complex backgrounds, including columnar interferences such as utility poles. Comparative detection results demonstrate that our algorithm achieves the highest accuracy, successfully detecting cranes in all four test images.

4 Conclusion

This study proposes YOLOCrane, an enhanced YOLOv8-based algorithm specifically designed for crane detection in transmission line scenarios. YOLOCrane incorporates a simplified network to mitigate overfitting, a channel-wise LBP mechanism for robust texture extraction, and a dual-branch attention fusion module to improve feature fusion. Experimental results on the CraneLine dataset demonstrate the superiority of YOLOCrane over existing methods in terms of accuracy, robustness, and generalization.

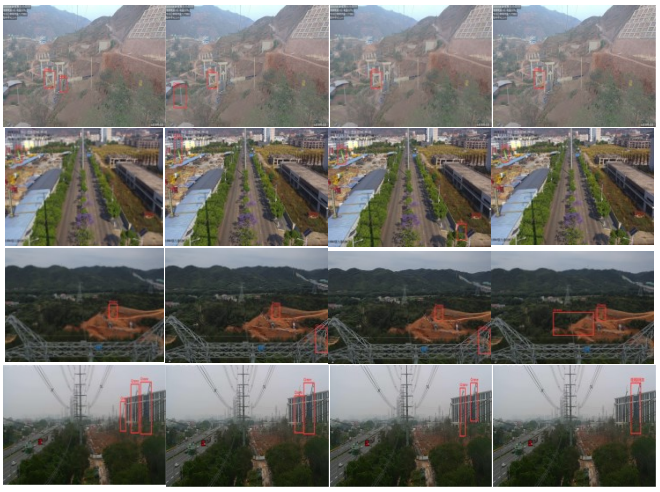


Fig. 6. Detection results of YOLOCrane versus baseline models on complex transmission line scenarios.

However, there are still limitations to be addressed in future work. Firstly, YOLOCrane's computational efficiency needs to be further optimized, especially for high-resolution images. Secondly, extending YOLOCrane to support multi-category detection would broaden its applicability. Lastly, advanced fusion techniques between LBP and neural network features could be explored to enhance the accuracy-efficiency trade-off. Despite these limitations, YOLOCrane offers significant potential for practical applications and further research in the field of object detection.

Acknowledgments. This study was funded by the National Key R & D Program of China, Project Number (Project No. : 2023YFC3006700); topic Five Number: 2023YFC3006705
Disclosure of Interests. The authors declare there is no conflict.

References

1. Sharma, K., Saini, L.M.: Power-line communications for smart grid: Progress, challenges, opportunities and status. Renew. Sustain. Energy Rev. 67, 704–751 (2017)
2. Zhu, H., Hwang, B.G., Ngo, J., Chan, A.P.C.: Applications of smart technologies in construction project management. J. Constr. Eng. Manag. 148(4), 04022010 (2022)
3. Sadeghi, S., Soltanmohammadlou, N., Rahnamayiezekavat, P.: A systematic review of scholarly works addressing crane safety requirements. Saf. Sci. 133, 105002 (2021)
4. Li, W.: Analysis of object detection performance based on Faster R-CNN. In: Journal of Physics: Conference Series, vol. 1827, 012085. IOP Publishing, Bristol (2021)
5. Diwan, T., Anirudh, G., Tembhurne, J.V.: Object detection using YOLO: Challenges, architectural successors, datasets and applications. Multimed. Tools Appl. 82(6), 9243–9275 (2023)

6. Ali, M.L., Zhang, Z.: The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. Computers 13(12), 336 (2024)
7. Chian, E.Y.T., Goh, Y.M., Tian, J., Guo, B.H.W.: Dynamic identification of crane load fall zone: A computer vision approach. Saf. Sci. 156, 105904 (2022)
8. Lu, Y., Qin, W., Zhou, C., Liu, Z.: Automated detection of dangerous work zone for crawler crane guided by UAV images via Swin Transformer. Autom. Constr. 147, 104744 (2023)
9. Sun, X., Lu, X., Wang, Y., et al.: Development and application of small object visual recognition algorithm in assisting safety management of tower cranes. Buildings 14(12), 3728 (2024)
10. Xia, L., Cao, S., Cheng, Y., Niu, L., Zhang, J., Bao, H.: Rotating object detection for cranes in transmission line scenarios. Electronics 12(24), 5046 (2023)
11. Vijayakumar, A., Vairavasundaram, S.: YOLO-based object detection models: A review and its applications. Multimed. Tools Appl. (2024)
12. Zhou, Y.: A YOLO-NL object detector for real-time detection. Expert Syst. Appl. 238, 122256 (2024)
13. Sohan, M., Sai Ram, T., Reddy, R., et al.: A review on YOLOv8 and its advancements. In: Proc. IEEE Int. Conf. DICI, pp. 529–545. Springer, Singapore (2024)
14. Khow, Z.J., Wang, C.Y., Liao, H.Y.M., et al.: Improved YOLOv8 model for a comprehensive approach to object detection and distance estimation. IEEE Access (2024)
15. Bochkovskiy, A., Farhadi, A.: YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934 (2020)
16. Saxe, A.M., Bansal, Y., Dapello, J., et al.: On the information bottleneck theory of deep learning. Journal of Statistical Mechanics: Theory and Experiment 2019(12), 124020 (2019)
17. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognit. 29(1), 51–59 (1996)
18. Ozdemir, C., Dogan, Y., Kaya, Y.: A new local pooling approach for convolutional neural network: Local binary pattern. Multimed. Tools Appl. 83, 34137–34151 (2024)
19. Pan, X.R., Ge, C.J., Lu, R., Song, S.J., Chen, G.F., Huang, Z.Y., Huang, G.: On the integration of self-attention and convolution. arXiv:2111.14556 (2021)
20. Wang, G.: CraneLine. Zenodo (2025). https://doi.org/10.5281/zenodo.14722538
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
22. Zhao, Y.A., Lv, W.Y., Xu, S.L., Wei, J.M., Wang, G.Z., Dang, Q.Q., Liu, Y., Chen, J.: DETRs beat YOLOs on real-time object detection. arXiv:2304.08069 (2023)
23. Khanam, R., Hussain, M.: YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725 (2024)
24. Zhu, X., Lyu, S., Wang, X., Zhao, Q.: TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. arXiv:2108.11539 (2021)