Sensors and Materials, Vol. 38, No. 4 (2026) 1959–1974
MYU Tokyo

S & M 4418

Enhanced Photovoltaic–You Only Look Once Model for Photovoltaic Panel Fault Detection Based on IR Thermal Imaging Sensors

Yi Jia and Quanming Zhao*

School of Electronic and Information Engineering, Hebei University of Technology, Tianjin 300401, China

(Received December 2, 2025; accepted February 27, 2026)

Keywords: IR thermal imaging sensor, multiscale, small targets, dynamic convolution, large receptive field

In response to the energy crisis, regions worldwide are continuously expanding their photovoltaic (PV) installation capacities. However, as the scale grows, the rapid identification of faults in PV panels has become a significant challenge. Although unmanned aerial vehicles equipped with IR thermal imaging sensors can quickly, cost-effectively, and noninvasively detect faults in PV panels, traditional detection methods still face issues with accuracy. This is because fault regions in IR images are often small, exhibit significant scale variations, and have unclear features. To address these issues, in this paper, we propose an IR thermal imaging-based PV panel fault detection model, PV–You Only Look Once (YOLO). The model first predicts a set of weights through input features and performs a weighted fusion of all expert convolution kernels to generate dynamic convolution kernels for specific inputs, thereby enhancing the detection ability for multiscale objects. Additionally, a multibranch dilated convolution structure is used to build convolution kernels with a large receptive field, which are then fused with the original spatial pyramidal pooling-fast module to further improve the detection accuracy of multiscale fault points. Finally, a multi-attention fusion mechanism is introduced, which enhances fault detection accuracy by parallelly fusing local structural attention, channel attention, and spatial attention. Experimental results show that PV–YOLO improves by 5.2 percentage points over the YOLOv8n model in Mean Average Precision at Intersection over Union 0.5, reaching 85.3%, while the recall rate increases by 4.2 percentage points to 80.5%. Compared with other mainstream algorithms, PV–YOLO achieves a better balance between detection accuracy and model complexity.

1. Introduction

With the worsening environmental pollution and escalating energy crisis, solar photovoltaic (PV) power generation has been widely adopted. According to data from the International Renewable Energy Agency (IRENA), the global cumulative installed capacity of solar PV power generation is expected to reach approximately 8519 GW by 2050, meeting more than 25% of

*Corresponding author: e-mail: qmzh@hebut.edu.cn
https://doi.org/10.18494/SAM6096

ISSN 0914-4935

global electricity demand.[1] Therefore, the timely detection of PV panel faults is crucial for ensuring the stable operation of PV systems and the reliability of the broader power grid.[2] Currently, there are two mainstream approaches for PV panel fault detection. The first relies on analyzing the electrical characteristics of PV panels and identifying faults using corresponding diagnostic algorithms.[3] The second approach employs unmanned aerial vehicle (UAV)-mounted IR thermal sensors to capture the image data of PV panels, followed by fault identification using image recognition algorithms.[4] With the continuous advancement of UAV technology, computational power, and recognition algorithms, UAV-based IR thermal imaging has become a routine method for PV panel fault detection owing to its flexibility and noninvasive nature.[5]

However, existing image recognition algorithms encounter two major challenges when applied to UAV-based IR thermal imaging scenarios: First, many models are highly complex, possessing large numbers of parameters and requiring substantial floating-point computations. This makes them difficult to deploy on resource-constrained devices and results in slow inference speed, which fails to meet real-time detection requirements.[6] Second, IR thermal images captured by sensors are easily affected by various types of noise, such as thermal noise and sensor noise.[7] Furthermore, the thermal distribution in IR images is typically smooth, and the contrast between hotspots and the background is often low, making it difficult for models to effectively distinguish fault regions from normal regions.[8] Collectively, these factors lead to suboptimal recognition accuracy in existing models. Therefore, enhancing detection accuracy for IR thermal images while maintaining manageable model complexity has become a key research focus.

With the rapid development of deep learning algorithms, current models for fault identification can generally be categorized into three types: single-stage object detection algorithms, two-stage object detection algorithms, and more recently, Transformer-based object recognition methods.[9] To enable deployment on computation-constrained devices while maintaining real-time detection performance, many researchers have opted for single-stage object detection algorithms such as You Only Look Once (YOLO)[10] and the single shot multibox detector (SSD).[11] For example, Zhou and Sun[12] enhanced the VGG16 model by incorporating texture features and applying data augmentation techniques. However, their method relies on visible-light images, which cannot accurately identify PV panel faults. Cheng[13] proposed adding additional detection heads to YOLOv3 to improve accuracy, but this approach substantially increases the computational cost. Hong *et al.*[14] applied partial convolution (PConv) to YOLOv10 to achieve a lightweight model. However, the baseline model used in their study already has high computational complexity, and the lightweighting effect provided by PConv is limited. Moreover, PConv tends to overlook fine-grained feature details, making it less suitable for IR thermal-sensor-based fault detection tasks.

In summary, to address the challenges of low recognition accuracy in IR images and the limited computational resources available during the UAV-based IR inspection of PV panels, in this study, we propose PV–YOLO, an algorithm specifically designed for such scenarios to enhance PV fault detection performance. The main contributions of this work are as follows.

- To address the low recognition accuracy caused by the small size and varying scales of PV fault regions in IR thermal images, we introduce a weighted fusion method that generates

dynamic weights on the basis of input features. By performing weighted fusion over multiple expert convolution kernels, the model produces convolution kernels that adapt to the characteristics of each input.

- To further enhance the model's ability to detect multiscale PV panel faults in IR thermal images, we employ a multibranch dilated convolution strategy to construct convolution kernels with a large receptive field. During inference, these kernels are equivalently merged into a single large convolution kernel, thereby improving feature representation without increasing computational overhead. This method is integrated with spatial pyramid pooling-fast (SPPF),[15] and the resulting module is named SPPF_UniRepLK.
- To improve the model's ability to handle interference between targets and background as well as the weak feature contrast commonly found in IR thermal images, in this study, a parallelized patch-aware (PPA) multiscale feature fusion module, which enhances feature representation through the combined use of local perception and channel/spatial attention mechanisms, is incorporated.

2. PV Panel Fault Detection Method Based on IR Thermal Imaging Sensors

To address the issues of small fault targets, large-scale variations, and the difficulty of feature extraction in thermal images during UAV-based PV panel fault detection, we improve the YOLOv8n[16] model from the perspective of enhancing detection accuracy and propose the PV–YOLO model. The overall network architecture of the PV–YOLO model is shown in Fig. 1.

As shown in Fig. 1, PV–YOLO introduces several enhancements to the original YOLOv8n model. First, a dynamic convolution kernel selection mechanism is incorporated into the convolutional layers of both the backbone and neck networks, forming the new DynamicConv module. Second, the original SPPF module is improved by adopting the UniRepLK large-kernel structure, resulting in the enhanced SPPF_UniRepLK module. Finally, a PPA module specifically designed for IR thermal image recognition is added after the SPPF module. Together, these three improvements significantly enhance the detection accuracy of the original model.

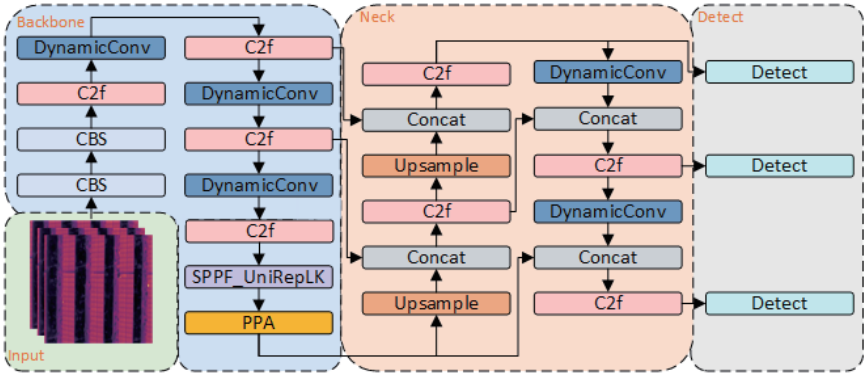


Fig. 1. (Color online) Overall network architecture of PV–YOLO.

2.1 Improved standard convolution: DynamicConv

When UAV-mounted IR thermal imaging sensors are used for PV panel fault detection, the relatively high flight altitude often results in fault regions appearing small and exhibiting considerable variation in scale in the thermal images. This significantly reduces the detection accuracy of conventional models. Traditional convolution kernels have fixed sizes, which limits their ability to effectively process targets of varying scales and heterogeneous shapes. Therefore, in this study, we introduce a novel convolutional layer design, DynamicConv, which integrates conditional computation with dynamic mechanisms. By dynamically selecting convolution kernels, the model can adaptively adjust the convolution operations in accordance with the input features, thereby enhancing computational flexibility and efficiency. This leads to improved detection accuracy for small fault targets and multiscale faults in PV panels.

The internal structure of DynamicConv is shown in Fig. 2. In this structure, the AdaptiveAvgPool2d layer performs global average pooling on each channel, extracting global semantic features from the input to generate expert weights. The Flatten layer reshapes the pooled features from $B \times C_{in} \times 1 \times 1$ to $B \times C_{in}$, enabling the features to be fed into the fully connected layer. The Linear layer serves as a fully connected layer that maps the channel-level global features from C_{in} dimensions to *num_experts* dimensions, thereby generating weights for each expert convolution kernel. The Dynamic Fusion module performs weighted fusion of multiple expert convolution kernels in accordance with the generated weights, producing an input-specific dynamic convolution kernel and completing the convolution operation. The formulation is

$$W^{(dyn)} = \sum_{i=1}^{N} \alpha_i W^{(i)}. \tag{1}$$

Here, $W^{(dyn)}$ denotes the dynamic convolution kernel, N represents the number of expert convolution kernels, and $W^{(i)}$ refers to the parameters of the i-th expert convolution kernel, which remain fixed. α_i denotes the dynamic weights, which are obtained through the AdaptiveAvgPool2d layer followed by a Sigmoid activation function.[17]

2.2 Enhanced SPPF: SPPF_UniRepLK

The primary function of the SPPF module is to extract information from features at different levels, enabling the model to better understand complex scenes, thereby improving detection

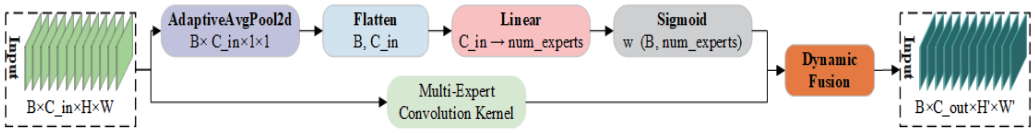


Fig. 2. (Color online) Diagram of the internal structure of the DynamicConv module.

accuracy, enhancing computational efficiency, and reducing computational overhead. Although the traditional SPPF module enhances the model's ability to recognize multiscale targets to some extent, its effectiveness remains limited in PV panel fault detection tasks—particularly when processing the IR thermal images of solar PV panels, where fault regions often exhibit significant scale variation. Therefore, an enhanced SPPF module, referred to as SPPF_UniRepLK, is adopted. The internal structure of the SPPF_UniRepLK module is illustrated in Fig. 3.

As shown in Fig. 3, the improved SPPF_UniRepLK differs from the original SPPF owing to the introduction of an additional UniRepKNetBlock module after the Concat operation. During training, this module integrates large-kernel depthwise convolutions with multibranch dilated convolutions, effectively enlarging the network's receptive field and thereby significantly enhancing the feature representation capability of SPPF. However, during inference, these operations can be unfolded into an equivalent dense convolution kernel, eliminating the computational overhead introduced by the multibranch structure and dilated convolutions, thus further improving the model's inference speed. The unfolding process is formulated as

$$K^{nd} = Transpose(K, I, stride = d), \tag{2}$$

where K represents the original dilated convolution kernel while K^{nd} denotes the converted equivalent nondilated convolution kernel. I is the unit convolution kernel of 1×1, which is used solely for kernel expansion without altering the numerical values. Transpose denotes the transposed convolution, in which $d - 1$ zero elements are inserted between the elements of K to convert the dilated convolution kernel into an equivalent large-kernel convolution. Here, d refers to the dilation rate of the original dilated convolution.

2.3 Enhancing IR detection performance through integration of the PPA module

Compared with visible-light images, IR images often lack sharp edges and fine details. When the contrast between the background and the target is low, targets can be easily confused with background noise. Moreover, IR images generally have lower resolution, and their details are less distinct than those in visible-light images. This makes it challenging for detection models to accurately capture target features, causing fine details to be overlooked. Consequently, when IR imaging sensors are used for PV panel fault detection, the recognition accuracy often becomes suboptimal. To address this issue, the PPA module is incorporated after the SPPF module to improve the detection accuracy of small IR targets. The internal structure of the PPA module is illustrated in Fig. 4.

Fig. 3. (Color online) Diagram of the internal structure of the SPPF_UniRepLK module.

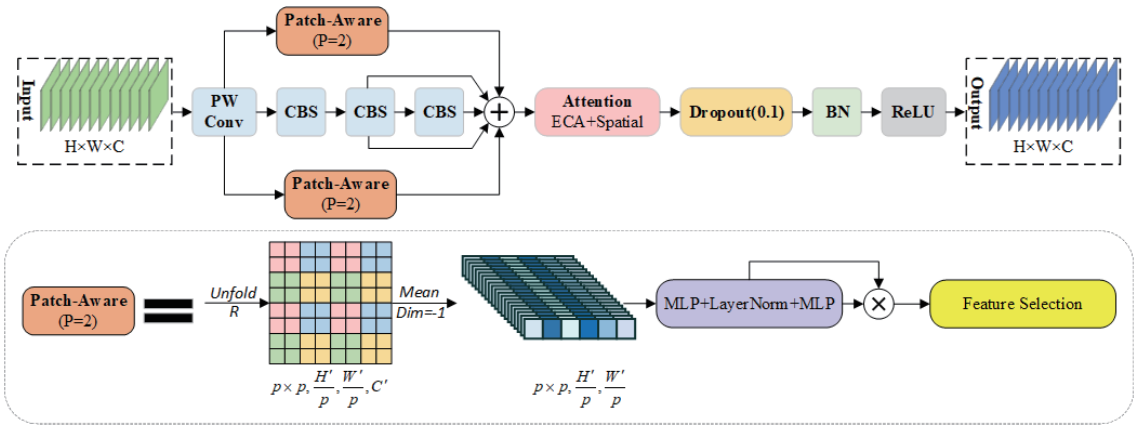


Fig. 4. (Color online) Diagram of the internal structure of the PPA module.

As shown in Fig. 4, the PPA module extracts multiscale local and global features through convolution and two Patch-Aware branches of different scales. The Patch-Aware branches generate structure-sensitive local attention using unfolding, local aggregation, and an MLP–Softmax mechanism, and subsequently fuse this attention with the backbone features. The fused features are further enhanced by channel attention and spatial attention, ultimately producing more discriminative feature representations. The formulation of the PPA module is

$$F_{PPA} = \delta(B(dropout(M_s(M_c(\tilde{F})) \otimes M_c(\tilde{F})))). \tag{3}$$

Here, $\tilde{F}$ represents the multibranch fused features generated from the convolution branch and the two Patch-Aware branches. M_c denotes the channel attention operation using the Efficient Channel Attention (ECA) module,[18] while M_s corresponds to the spatial attention operation. $\otimes$ denotes the element-wise multiplication operation. *dropout* refers to the dropout operation, which is employed to prevent overfitting. B represents the normalization operation, and δ denotes the ReLU activation function.

3. Experimental Setup and Model Evaluation

3.1 Dataset construction

Common PV panel faults can be categorized into five types: glass breakage (GB), hotspots, potential-induced degradation (PID), bypass diode failure, and shading. Since visible-light imaging relies on illumination, its performance degrades significantly on rainy or cloudy days and becomes completely ineffective at night. Moreover, visible-light images can only capture a limited range of fault types, typically only GB and shading—faults that are visually apparent. Consequently, in this study, a public dataset acquired through thermal IR sensors, comprising a total of 1267 images, is used. The dataset is publicly accessible at the following URL: https://

universe.roboflow.com/ryan-2h31z/pv-pics. Thermal images representing the five common types of PV panel defects are illustrated in Fig. 5.

To provide an in-depth understanding of the geometric characteristics of the training data, in this paper, the joint distribution of normalized width and height for all bounding boxes is presented. The color intensity represents the frequency of samples occurring within a specific size range. The distribution pattern of object scales in the dataset is illustrated in Fig. 6.

As illustrated in Fig. 6, the distribution of object scales within the dataset exhibits significant nonuniformity. The high concentration of samples toward the bottom-left corner of the coordinate system indicates a predominance of small objects, posing a severe challenge to the fine-grained feature extraction capabilities of the algorithm. Simultaneously, while small objects

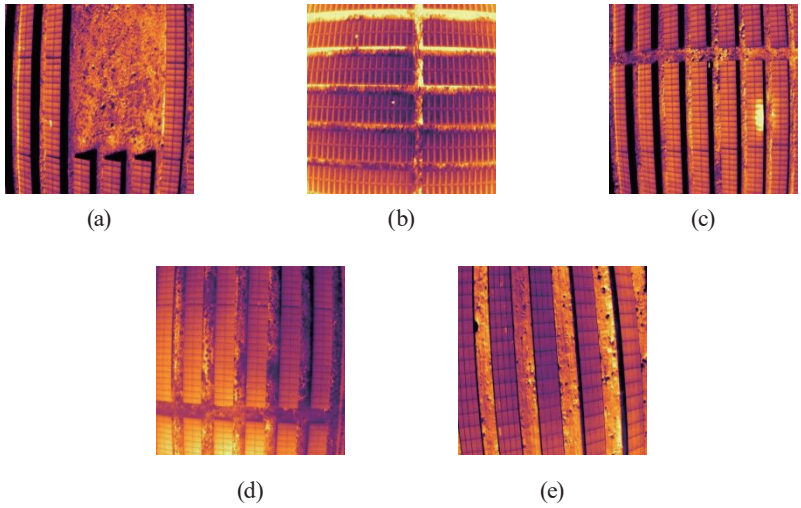


Fig. 5. (Color online) Thermal images of five common types of PV panel faults. (a) GB, (b) hotspots, (c) PID, (d) bypass diode failure, and (e) shading.

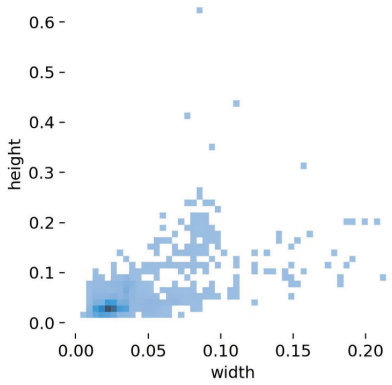


Fig. 6. (Color online) Distribution of object scales in the dataset.

are dominant, the distribution plot reveals a subset of large-scale samples characterized by high aspect ratios. This cross-scale "long-tail distribution" reflects the geometric diversity and complexity of the objects, necessitating robust multiscale feature fusion and adaptive scale-aware perception within the model.

3.2 Experimental platform configuration and model hyperparameter settings

The parameters of different experimental platforms have a significant impact on the performance of deep learning models. Moreover, given the computational demands of deep learning models—particularly their reliance on graphics processing unit (GPU)—it is crucial to select an appropriate experimental platform and ensure that all experiments are conducted under a consistent environment. The configuration of the experimental platform used in this study is presented in Table 1.

In addition to the configuration of the experimental platform, the hyperparameter settings of the model also have a significant impact on its performance. The hyperparameters used in the proposed model are presented in Table 2.

For the remaining models used in the comparative experiments, the hyperparameter settings vary significantly owing to differences in their architectural frameworks. Therefore, in all experiments of this study, the same hyperparameters as those listed in Table 2 are used for YOLO-series models, while the hyperparameters of the other models follow the official configurations provided by their respective authors or model documentation.

Table 1
Configuration of the experimental platform.

Component name	Component details
CPU	12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40 GHz
RAM	48 GB
GPU	RTX 3080 Ti(12 GB)
Operating system	Ubuntu 20.04, 64-bit Operating System
PyTorch deep learning framework	1.10.0
CUDA version	11.3

Table 2
Hyperparameter settings of the PV–YOLO model.

Training parameters	Parameter value
Input image resolution	640 × 640
Number of training epochs	250
Learning rate	0.01
Weight_decay	0.0005
Momentum	0.937
Optimizer	SGD

3.3 Experimental protocol design

On the basis of the established experimental platform and model hyperparameter settings, the following experiments were designed in this study: Experiment 1: Ablation studies on the proposed model to evaluate the performance improvements contributed by each modification, as well as the interaction effects among multiple modifications. Experiment 2: Comparison of the proposed model with existing mainstream models to validate its performance advantages. Experiment 3: Examination of visual detection results to intuitively demonstrate the recognition capability of the proposed model. Experiment 4: Analysis of the limitations of the improved model through its confusion matrix to gain insights for future enhancements.

3.4 Model performance evaluation metrics

The performance of the model is evaluated primarily from three aspects: detection accuracy, model size, and inference speed. The evaluation of detection accuracy primarily involves the following three metrics. *mAP*@0.5: This metric represents the mean average precision at an intersection over union (*IoU*) threshold of 0.5 and reflects the overall detection accuracy of the model.[19] *Recall*: *Recall* measures the model's ability to identify all relevant targets. A higher recall indicates fewer missed detections.[20] *P*: This represents precision, which is a measure of the reliability of the model's detection results, indicating the proportion of correct positive predictions.[21] The calculation formulas for *Precision*, *Recall*, and *mAP* are shown below.

$$P = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$mAP = \frac{\sum_{j=1}^{K} AP_j}{K} \tag{6}$$

In Eqs. (4) and (5), *TP* refers to cases where the model predicts that the PV panel contains fault type A, and the fault indeed exists, with the predicted bounding box correctly matching the ground-truth location. *FP* is the opposite of *TP*. It indicates that the model predicts fault type A in a region where the actual situation is one of the following: (i) the region is normal, or (ii) the region contains a different type of fault. *FN* denotes cases where the PV panel actually contains fault type A, but the model fails to detect it (missed detection). *TN* indicates that the model correctly predicts a region as normal when it is indeed normal. In Eq. (6), *mAP* represents the *mAP* across all fault categories, and *K* denotes the total number of fault types.

The core parameters used to evaluate model size and computational cost include the model size, the number of parameters, and the numbers of floating-point operations (FLOPs). The

model size refers to the storage size of the model's weight file, while the number of parameters and FLOPs do not have unified calculation formulas as they are influenced by various architectural components such as convolutional layers, fully connected layers, and pooling layers.

The primary metric used to evaluate the detection speed of a model is frames per second (*FPS*),[22] and its calculation is

$$FPS = \frac{1000}{T_P + T_I + T_N}, \tag{7}$$

where T_P denotes the preprocessing time of the model; T_I represents the inference time, which reflects the computational efficiency of the model during its core processing stage; and T_N refers to the nonmaximum suppression (NMS) time, which is the duration required to filter redundant bounding boxes and retain the most representative predictions.

4. Experimental Results and Analysis

4.1 Ablation study of the model

To evaluate the independent contributions of each module and their synergistic effects on detection performance, ablation experiments were conducted. The evaluation considers three dimensions: detection accuracy (*mAP*, *Precision*, *Recall*), model complexity (Parameters, GFLOPs, Size), and *FPS*. The results of the ablation study are presented in Table 3. In the table, "√" indicates that the corresponding module is included.

As shown in Table 3, Model A corresponds to the original YOLOv8 model, while Model B incorporates the DynamicConv module. The *mAP*@0.5 of Model B increases from 80.1 to 82.3%, an improvement of 2.2 percentage points. Moreover, the recall rises significantly from 76.3 to 81.8%, indicating that DynamicConv enhances the model's ability to capture target features and improves its detection sensitivity. However, this dynamic mechanism expands the activation range of features, causing certain background textures and noise regions to be

Table 3
Ablation study results.

Network Model	Dynamic Conv	PPA	SPPF_I	*mAP*@0.5 (%)	*Precision* (%)	*Recall* (%)	Parameters	GFLOPs	Size (MB)	*FPS* (f/s)
A				80.1	85.2	76.3	3006623	8.1	6.3	113.1
B	√			82.3	76.9	81.8	4723123	7.2	9.3	119.8
C		√		84.7	79.6	80.3	5177863	9.7	10.2	101.5
D			√	81.5	83.4	78.5	5256991	9.9	10.3	117.6
E	√	√		84.9	80.5	79.7	6898923	8.8	13.4	90.6
F	√		√	83.7	79.6	82	6978035	9.1	13.6	108.7
G		√	√	82.4	85.5	73.8	7428231	11.4	14.5	89.9
H	√	√	√	85.3	83.4	80.5	9149275	10.5	17.7	83.3

mistakenly identified as targets. As a result, the precision drops from 85.2 to 76.9%, reflecting an increase in false positives. Additionally, because DynamicConv generates convolution kernels adaptively on the basis of the input, its computational structure is more compatible with GPU acceleration. Consequently, despite the increased number of parameters, the *FPS* rises to 119.8, the highest speed among all models. Model C incorporates only the PPA module. With PPA applied independently, *mAP* improves from 80.1 to 84.7%, demonstrating that the multiscale local–global attention mechanism and enhanced spatial/channel modeling capabilities in PPA significantly boost detection performance. However, GFLOPs increase to 9.7 and *FPS* decreases to 101.5, indicating that the complex attention structure introduces substantial computational overhead. Model D incorporates only the SPPF_I module, replacing the traditional SPP structure with the UniRepLK large-kernel architecture. The results show a slight improvement in *mAP*@0.5 but a minor decrease in precision. Nevertheless, since large-kernel convolutions are more GPU-friendly, this module maintains a relatively high *FPS* despite the increased computational complexity.

In the case of multimodule combinations, Model E integrates both DynamicConv and PPA. Its *mAP*@0.5 reaches 84.9%, which is very close to that of Model C, indicating that the two modules exhibit a degree of complementarity. Compared with using PPA alone, recall decreases slightly, whereas precision improves substantially. As the number of parameters and computational load are increased in both modules, *FPS* drops to 90.6. Model F combines DynamicConv with SPPF_I, achieving a recall of 82.0%, the highest among all models, and *mAP*@0.5 of 83.7%, while maintaining relatively high *FPS*. This suggests that the strong recall capability of DynamicConv complements the acceleration characteristics of SPPF_I, enabling the model to achieve a favorable balance between inference speed and recall. Model G integrates PPA with SPPF_I. Although it achieves the highest precision among all models at 85.5%, its recall drops significantly to 73.8%. Additionally, it has the highest computational cost—11.4 GFLOPs—and its *FPS* decreases to 89.9. Model H is the model proposed in this study. With the integration of DynamicConv, PPA, and SPPF_I, it achieves the highest *mAP*@0.5 of 85.3%, marking a 5.2 percentage-point improvement over the baseline. Both precision and recall remain at high and balanced levels, reaching 83.4% and 80.5%, respectively. This improvement results from the integration of DynamicConv, PPA, and SPPF_I, which enabled the model to acquire stronger dynamic feature extraction in lower layers, richer local–global attention representation in middle and higher layers, and enhanced global contextual information through large-kernel pooling. Together, these modules form a complementary feature enhancement pathway. The synergistic interaction of these three modules significantly improves the integrity and discriminative capacity of feature representation, leading to optimal results in mAP, precision, and recall.

4.2 Comparative experiments of algorithms

To thoroughly validate the superior performance of the proposed PV–YOLO model, comparative experiments were conducted against several mainstream detection models. The comparison primarily focuses on two aspects: the detection accuracy represented by the

mAP@0.5 metric, and the model size represented by the Size parameter. The comparison results are presented in Table 4.

As shown in Table 4, the proposed PV–YOLO model is compared with several mainstream object detection models in terms of *mAP*@0.5 and model size. Overall, PV–YOLO achieves the highest detection accuracy while maintaining a relatively compact model size.

From the lightweight YOLO models, the *mAP*@0.5 values of YOLOv5n, YOLOv6n, YOLOv8n, YOLOv11n, and YOLOv13n are 80.8, 75.3, 80.1, 78.1, and 76.5%, respectively, with model sizes ranging from 3.7 to 9.9 MB. With a model size of 17.7 MB, PV–YOLO achieves *mAP*@0.5 of 85.3%, representing improvements of 4.5 percentage points over the best-performing YOLOv5n, and 5.2 and 7.2 percentage points over YOLOv8n and YOLOv11n, respectively. These results indicate that within the lightweight detection framework, PV–YOLO achieves a substantial accuracy improvement while sacrificing only a modest increase in model size.

Compared with other mainstream detectors, PV–YOLO also demonstrates strong overall performance. The Transformer-based RT-DETR achieves *mAP*@0.5 of 84.9%, slightly lower than PV–YOLO's 85.3%. However, its model size reaches 38.6 MB, which is approximately 2.2 times larger than that of PV–YOLO. Two-stage detectors such as Faster Regions with Convolutional Neural Networks (R-CNN) and one-stage detectors including SSD and RetinaNet achieve only 49–61% *mAP*@0.5, with model sizes exceeding 90 MB; RetinaNet is particularly large at 139.1 MB. These models struggle to balance accuracy and model compactness, making them less suitable for deployment. Although EfficientDet-D0 has a relatively small model size (15.1 MB), its *mAP*@0.5 is only 48.51%, indicating that its detection accuracy is significantly low.

Overall, the results show that among all compared models, PV–YOLO achieves the highest accuracy in terms of *mAP*@0.5 while maintaining a model size significantly smaller than larger models such as Real-Time Detection Transformer (RT-DETR), Faster R-CNN, SSD, and

Table 4
Results of comparative experiments.

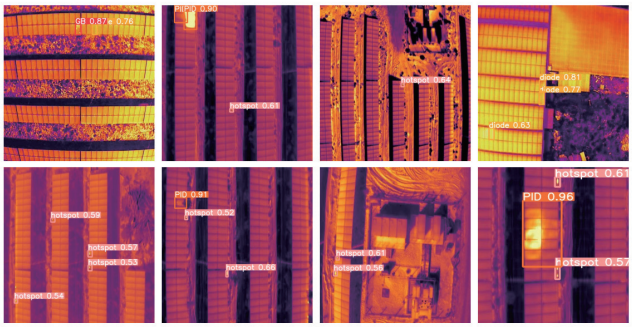
Model	*mAP*@0.5 (%)	Size (MB)
YOLOv5n	80.8	3.7
YOLOv6n	75.3	9.9
YOLOv7	61.11	71.4
YOLOv7-tiny	50.9	11.7
YOLOv8n	80.1	6.3
YOLOv9-c	79.4	98.1
YOLOv10n	73	5.5
YOLOv11n	78.1	5.3
YOLOv12n	71.9	4.2
YOLOv13n	76.5	5.2
RT-DETR	84.9	38.6
Faster R-CNN	61.35	108.2
SSD	60.67	91.6
Retinanet	49.53	139.1
Efficientdet-D0	48.51	15.1
PV–YOLO	85.3	17.7

RetinaNet, and only slightly larger than some extremely lightweight YOLO variants and EfficientDet-D0. This demonstrates that PV–YOLO achieves a superior balance between detection accuracy and model size, making it more suitable for deployment in real-world scenarios where both accuracy and computational efficiency are critical.

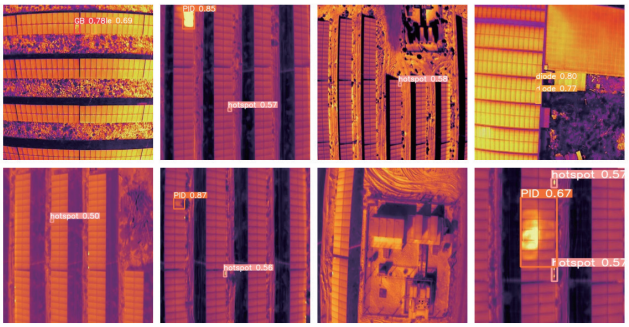
4.3 Detection performance and limitations of PV–YOLO

To visually demonstrate the superior detection accuracy of the proposed PV–YOLO model, a comparative visualization experiment was conducted. The detection outputs of PV–YOLO were compared with those of the original YOLOv8n model, and the results are presented in Fig. 7.

As illustrated in Fig. 7, PV–YOLO significantly outperforms YOLOv8n in terms of detection accuracy. In the detection of small-target faults such as “hotspots”, PV–YOLO achieves precise identification; conversely, the baseline YOLOv8n model exhibits lower confidence levels and a higher rate of missed detections. When large- and small-scale faults coexist on the same PV panel, PV–YOLO demonstrates superior multiscale detection capabilities, effectively preventing small-target leakage while maintaining high confidence for large targets. In contrast, the original YOLOv8n not only tends to overlook small targets in these scenarios but also suffers from a concomitant decline in detection confidence for large targets. In summary, the proposed PV–YOLO model exhibits enhanced robustness and superior performance when addressing small-



(a)



(b)

Fig. 7. (Color online) Results of visual detection using the proposed PV–YOLO model and the original YOLOv8n model. (a) PV–YOLO detection results, (b) YOLOv8 detection results.

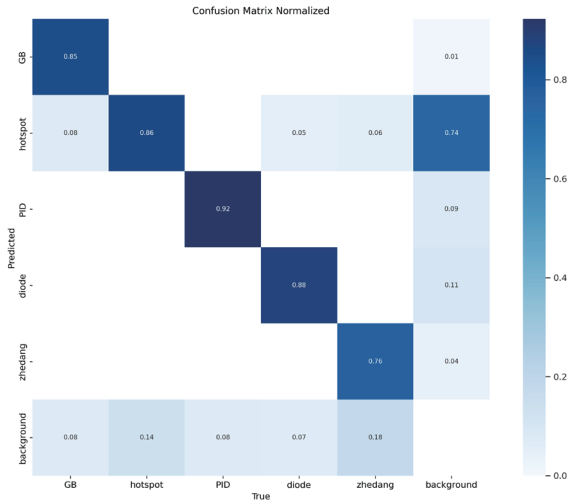


Fig. 8. (Color online) Confusion matrix of the PV–YOLO model.

target and multiscale fault detection tasks.

To identify which fault categories are more challenging for the PV–YOLO model to recognize and which faults are prone to misclassification, thereby providing guidance for future model improvements, the confusion matrix of the PV–YOLO model, shown in Fig. 8, is analyzed

As shown in the normalized confusion matrix in Fig. 8, the proposed model performs well across all fault categories, with most samples concentrated along the main diagonal. Among them, the PID and diode categories exhibit the highest recognition accuracy, indicating that the model effectively captures their discriminative features. Occlusion faults are more difficult to distinguish, with approximately 18% misclassified as background. Similarly, around 14% of hotspot samples are incorrectly labeled as background. These results suggest that the features of these two fault types partially overlap with those of normal regions in the feature space, indicating the need for further improvement through data augmentation or enhanced feature modeling. In addition, the model shows a noticeable "fault-biased" tendency during decision-making, frequently classifying uncertain background regions as faults to reduce the risk of missed detections, although this also leads to a higher false-positive rate. Overall, the proposed PV–YOLO model demonstrates strong discriminative ability for major fault categories; however, the decision boundaries between background and certain fault classes still require further refinement.

5. Conclusions

With the increasing penetration of PV power generation units into modern power systems, the commonly used UAV-based IR thermal imaging methods for PV panel fault detection still suffer from low detection accuracy, primarily owing to the small scale of fault regions, their varying shapes, and the inherent difficulty of interpreting IR images. To address these challenges, in this study, we proposed an enhanced PV–YOLO model for PV panel fault detection

using IR thermal imaging sensors. To evaluate the effectiveness of the structural enhancements and overall performance of the proposed model, a series of comparative experiments were conducted. The main conclusions are as follows.

(1) With the YOLOv8 framework as the basis, the original standard convolution layers were replaced with the DynamicConv module. This module enhances the model's ability to capture target features and improves detection sensitivity, resulting in higher *mAP*@0.5. However, because of its increased sensitivity to subtle textures, certain background textures and noise regions may also be mistakenly identified as targets, leading to a slight increase in false positive rates.

(2) To address the challenge of recognizing small IR targets, the PPA module is integrated into the network. This module employs a multibranch local–global attention mechanism, followed by channel attention and spatial attention after feature fusion, thereby enhancing feature representation from multiple dimensions. As a result, the model's accuracy in detecting small targets is significantly improved.

(3) The SPPF module in the model was optimized by replacing its original structure with the UniRepLK large-kernel architecture. Experimental results showed a slight improvement in detection accuracy. Moreover, since large-kernel convolutions are more GPU-friendly, the module maintains a relatively high *FPS* despite a minor increase in computational complexity.

Because of limitations in the experimental environment and conditions, the proposed model has not yet been deployed or tested on an actual UAV platform equipped with edge devices. For future work, we will consider implementing and validating the model in a real UAV system. In addition, the model occasionally misclassifies background regions as faults in certain scenarios. This is primarily attributed to the limited imaging quality of the IR thermal sensors and the high visual similarity between some fault patterns and background areas. Future improvements will focus on enhancing noise suppression and robustness of the model, as well as employing higher-performance IR imaging sensors to further improve system reliability.

References

1 W. Li, H. Liu, X. Hu, X. M. Hu, X. C. Lu, S. L. Tao, M. Qian, H. T. Yang, Y. C. Liu, M. X. Li, T. H. Li, and Q. H. Gao: Appl. Energy **402** (2026) 127025. https://doi.org/10.1016/j.apenergy.2025.127025
2 Q. P. Zheng, J. M. Ma, M. H. Liu, Y. C. Liu, Y. X. Li, and G. Shi: Sensors **22** (2012) 4617. https://doi.org/10.3390/s22124617
3 W. Chine, A. Mellit, V. Lughi, A. Malek, G. Sulligoi, and A. M. Pavan : Renewable Energy **90** (2016) 501. https://doi.org/10.1016/j.renene.2016.01.036
4 R. H. F. Alves, G. A. de Deus Junior, E. G. Marra, E. G. Marra, and P. R. Lemos: Renewable Energy **179** (2021) 502. https://doi.org/10.1016/j.renene.2021.07.070
5 M. Dihkan and E. Mus: Arabian J. Geosci. **14** (2021) 567. https://doi.org/10.1007/s12517-021-06947-1
6 J. W. Li, H. Tang, X. D. Li, H. D. Dou, and R. Li: Int. J. Wildland Fire **33** (2023) WF23044. https://doi.org/10.1071/WF23044
7 K. Awedat, G. Comert, M. Ayad, and A. Mrebit: Mach. Learn. Appl. **20** (2025) 100636. https://doi.org/10.1016/j.mlwa.2025.100636
8 A. Khatri, S. Khadka, N. Lamichhane, and R. Shrestha: Infrared Phys. Technol. **148** (2025) 105878. https://doi.org/10.1016/j.infrared.2025.105878
9 L. G. Xia, J. Chen, J. C. Luo, J. X. Zhang, D. Z. Yang, and Z. F. Shen: Remote Sens. **14** (2022) 4524. https://doi.org/10.3390/rs14184524

10 C. Zhao, X. Shu, X. Yan, X. Zou, and F. Zhu: Measurement **214** (2023) 112776. https://doi.org/10.1016/j.measurement.2023.112776
11 J. J. Ni, K. Shen, Y. Chen, and S. X. Yang: IEEE Trans. Instrum. Meas. **72** (2023) 1. https://doi.org/10.1109/TIM.2023.3244819
12 Y. J. Zhou and H. R. Sun: Sustainable Energy Technol. Assess. **53** (2022) 102476. https://doi.org/10.1016/j.seta.2022.102476
13 F. Cheng: IEEE Access **13** (2025) 185845. https://doi.org/10.1109/ACCESS.2025.3624734
14 Y. Hong, L. Wang, J. M. Su, Y. Li, S. K. Fang, W. Li, M. S. Li, and H. T. Wang: Digital Signal Process. **161** (2025) 105072. https://doi.org/10.1016/j.dsp.2025.105072
15 K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **37** (2015) 1904. https://doi.org/10.1109/TPAMI.2015.2389824
16 D. H. Wan, R. S. Lu, B. T. Hu, J. J. Yin, S. Y. Shen, T. Xu, and X. L. Lang: Adv. Eng. Inf. **62** (2024) 102709. https://doi.org/10.1016/j.aei.2024.102709
17 X. Qin, Q. H. Wang, X. Q. Zhao, X. S. Xia, L. Wang, Y. H. Zhang, C. He, D. L. Chen, and B. Jiang: Scr. Mater. **265** (2025) 116762. https://doi.org/10.1016/j.scriptamat.2025.116762
18 S. Y. Xu, Y. H. Cao, Z. H. Zhang, and M. J. Wang: Speech Commun. **166** (2025) 103154. https://doi.org/10.1016/j.specom.2024.103154
19 R. Ding, J. K. Yang, T. Y. Wang, C. H. Wang, X. Huang, S. H. Zhong, and R. Gu: Comput. Electron. Agric. **239** (2025) 111076. https://doi.org/10.1016/j.compag.2025.111076
20 R. Raushan, V. Singhal, and R. K. Jha: Autom. Constr. **170** (2025) 105887. https://doi.org/10.1016/j.autcon.2024.105887
21 T. Zoubek, R. Bumbálek, J. D. M. Ufitikirezi, M. Strob, M. Filip, F. Spalek, A. Hermanek, and P. Bartos: Crop Prot. **190** (2025) 107076. https://doi.org/10.1016/j.cropro.2024.107076
22 H. Zhang, M. Y. Liang, and Y. F. Wang: Sci. Rep. **15** (2025) 7558. https://doi.org/10.1038/s41598-025-88184-0