Sensors and Materials, Vol. 38, No. 5 (2026) 2525–2544
MYU Tokyo

S & M 4451

Simple Hotspot Inspection Network with Locating Matrix for Defect Detection of Solar Panel in Small-scale Solar Power Plants

Ming-Tsung Yeh* and Yen-Ting Lu

Department of Electrical Engineering, National Chin-Yi University of Technology,
57, Sec. 2, Zhongshan Rd., Taiping Dist, Taichung 411030, Taiwan

(Received December 30, 2024; accepted June 17, 2025)

Keywords: hotspot defect, YOLOv5s network, HSV color space, defect-locating matrix, solar panel defect detection

The development of renewable energy has resulted in a gradual increase in the application of solar energy. However, hotspots have been identified as a significant failure of solar panels after prolonged usage. This phenomenon has resulted in declining power generation efficiency and poses a potential risk of fire hazards or environmental contamination. Consequently, the maintenance and inspection of solar panels have become imperative. In addressing this challenge, the utilization of drones, in addition to conventional inspection methods, offers a novel solution for large-scale power plants. However, this approach may not be as economically viable for small-scale or residential solar installations. In this study, we propose a novel hotspot inspection network with a locating matrix method for efficiently detecting hotspots on solar panels in small-scale plants. The proposed method utilizes an embedded device combined with an infrared camera and a lightweight You Only Look Once (YOLO) v5s model as the basis for the defect detection network implemented. The defect-locating matrix method utilizes image processing techniques, such as hue, saturation, value (HSV) color space transformation and morphology, to convert the panel configuration of a field into a locating matrix. The center of gravity of the detected hotspot area is subsequently mapped to the locating matrix. The location of the defective panel is then labeled. The proposed approach enables real-time monitoring and analysis of solar panel status. The experimental findings demonstrate that the proposed defect detection network model reaches a *mAP* of 85.1% and an *F*1 *score* of 84.6%, and performs better than other YOLO models. A graphical user interface was designed to facilitate a more intuitive display of inspection results and enable accuracy monitoring.

1. Introduction

The issue of global warming is a grave problem owing to the accelerated development of industry and population growth, the unintentional destruction of the natural environment, and the increasing energy demand, which has resulted in a consistent annual increase in greenhouse

*Corresponding author: e-mail: mtyeh@ncut.edu.tw
https://doi.org/10.18494/SAM5532

ISSN 0914-4935
https://myukk.org/

gas emissions. To solve this problem, renewable energy is actively promoted as an alternative to traditional fossil energy and other highly polluting energy sources. Solar energy has emerged as a prominent and promising renewable energy source for promoting green energy.

The feasibility of implementing solar energy systems is contingent upon the presence of suitable construction conditions and available space. This encompasses both large-scale solar power plants and residential rooftop systems. However, note that solar panels may gradually degrade in performance as they age. This makes regular maintenance and repair particularly important. Hotspots are a phenomenon in solar energy systems where certain localized areas experience elevated temperatures. They may stem from internal or external factors such as defects in solar cells, wiring issues, surface contamination, damage, or shading. These factors can result in specific areas of solar panels being unable to receive sunlight correctly, and thus failing to generate electricity. As a result, the temperature in these areas increases, forming "hotspots," thereby impacting the system's overall efficiency.(1)

There are three types of hotspot defect on solar panels. The round type is easy to find, and the problem is caused by abnormal heat at the power cord junction and long-term leaf hotspots of photovoltaic (PV) panels. There are circular areas, approximately 3–6 cm in diameter, that are easy to remove. A mixture of bird droppings, dust, and rain mainly causes linear-type hotspots. There are 10 to 25 cm failure lines in length and less than 3 cm in width. This type of hotspot is challenging to remove. The panel has single internal defects or multiple failures in series and parallel to produce the square-type faults. The 10*10*N area showcases the square hotspots and cannot be eliminated (10*10 is the area of a single faulty battery module, and N refers to the number of faulty battery panels). The square-type and linear-type hotspots are difficult to inspect with a visible image, and the square type is the most common defect.(2) Figure 1 shows the square-type hotspots.

Conventional manual inspection techniques continue to be utilized; however, they are frequently inefficient and inadequate in terms of convenience and safety.(3) Inspectors often need to ascend to considerable heights, which increases labor costs and safety risks. While the utilization of drones has introduced a degree of convenience to the inspection of solar panels,(4) for smaller sites, drone inspection is often hindered by its high cost, which does not align with the actual demand. Many researchers have proposed methods for detecting hotspots on solar panels. A method combining a remote pilot aircraft system with a global navigation satellite

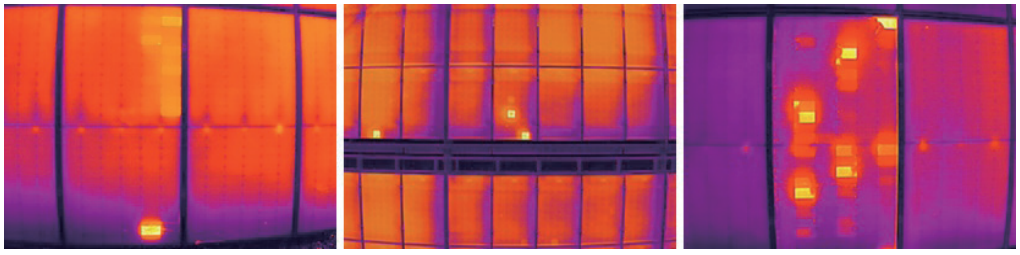

Fig. 1. (Color online) Square-type hotspots.

system was proposed by Addabbo *et al.* for the detection of hotspot anomalies.[5] This method is characterized by high efficiency, low cost, and safety, but weather conditions easily limit the detection results. Cao *et al.* proposed a mathematical modeling-based approach to accurately identify the effect of hotspots on the power generation efficiency of solar systems.[6] The method has been shown to accurately predict power loss with high adaptability by fully considering the effects of hotspots and environmental conditions. However, the method does not fully consider the spatial distribution and dynamic changes of hotspots, which may be oversimplified.

A number of the studies are more sophisticated, focusing on detecting hotspots for a single solar panel. Nakagawa *et al.* proposed a comparative analysis of the performance of infrared (IR) imaging and electroluminescence (EL) imaging in solar hotspot detection. Their detection method generates defect images by applying a voltage to solar panels and capturing the emitted light with an infrared camera. The presence of hotspots can result in the emission of light from the solar panel that is not uniform, which can lead to the identification of internal cracks or connection problems.[7] The advantages of EL imaging include high resolution and high sensitivity. However, the technique is not without its limitations, including the necessity of testing in dark environments and the requirement of specific equipment and operating conditions. Hajjdiab *et al.* proposed the application of a low-cost handheld thermal imager combined with image preprocessing techniques and using the Otsu threshold segmentation method to separate hotspots from solar panels. They developed an Android-based mobile application for the processing of thermal imaging and the analysis of defects. The system has been demonstrated to be portable and achieved efficient and accurate hotspot detection.[8] However, note that the resolution and sensitivity of the thermal imaging camera may constrain detection results. Implementing artificial neural networks in solar panel inspection represents a novel technological advancement, enabling the rapid and precise detection of hotspot defects. Alqahtani *et al.* proposed a solar hotspot detection system based on deep convolutional neural networks. The model exhibits several benefits, such as reduced inference time, high efficiency, and high accuracy.[9] Additional research might be necessary to reduce the neural network's computational demands, particularly on resource-constrained embedded devices.

In addressing the issues outlined above, in this paper, we propose a hotspot inspection network with a locating matrix method, which is a cost-effective and real-time solar defect detection system, and well-suited for implementation in small power plants. In light of the setup cost and field constraints, the system is installed in the embedded equipment on a single chip, equipped with an infrared camera. Concerning the inspection network model, it is necessary to consider the computing resource availability. Implementing the lightweight You Only Look Once (YOLO)-based network model serves the dual purpose of reducing the device's computational demands and ensuring the inspection's precision. The system will periodically capture thermal images of the field and then convert the layout of solar panels into a locating matrix by utilizing the hue, saturation, value (HSV) color space transformation and image morphological methods.[10] The defect detection system determines the location of any anomalous solar panels and records the findings in the locating matrix. The results thus obtained determine the necessity for maintenance or replacement of the damaged panels.

Compared with the employment of a drone for inspection, this system offers a substantial reduction in inspection costs while maintaining the precision of real-time monitoring, thereby enhancing its practicality. Conventionally, large-area inspections require a period of at least half a year for comprehensive evaluations. The proposed system can inspect the plant's solar panels in real time. Moreover, this system is expected to mitigate safety concerns typically associated with inspection operations, thereby paving the way for a more cost-effective and feasible solution for the future expansion and development of green energy.

2. Proposed Method

In this section, we demonstrate the proposed hotspot inspection system for small-scale solar power plants. We introduce the lightweight YOLO-based network and hotspot locating matrix method.

2.1 System framework

In this paper, a simple hotspot inspection method is proposed to detect hotspots of solar panels in small-scale solar power plants. The present approach utilizes the YOLOv5s lightweight module, which has been integrated into an embedded device, for the purpose of rapidly detecting and recognizing the precise locations of hotspot regions on solar panels. The system employs an embedded device, such as the Raspberry Pi, equipped with an IR camera to capture thermal imaging of the solar panel, thereby acquiring the thermal distribution state on the panel surface. Subsequently, these images are analyzed by a trained YOLOv5s network, which identifies the defective areas. The defect matrix locating method utilizes image processing techniques, such as HSV color space transformation and morphology, to convert the panel configuration of a field into a locating matrix. Then, the resulting image of the defective areas is used to map their location in the locating matrix. This enables the rapid identification of which panel has hotspot defects and requires maintenance by the location matrix. The system framework is illustrated in Fig. 2.

The training flowchart of the detection network is demonstrated in Fig. 3. The IR images of solar panels from multiple sites are obtained and subjected to image processing to enhance clarity and remove noise. After image preprocessing, data augmentation expands data sets and labels the defect region on the sample images. The implementation of the light YOLOv5s network module then follows for training and tuning its parameters. The optimized YOLOv5s detection network is then installed in the embedded device set at the various deployment sites to inspect hotspots.

2.2 Devices

In this study, we employed the Seek Thermal 302NP infrared camera, which possesses a primary resolution of 320 × 240 and a maximum frame rate of 9 Hz. The Raspberry Pi 4 features

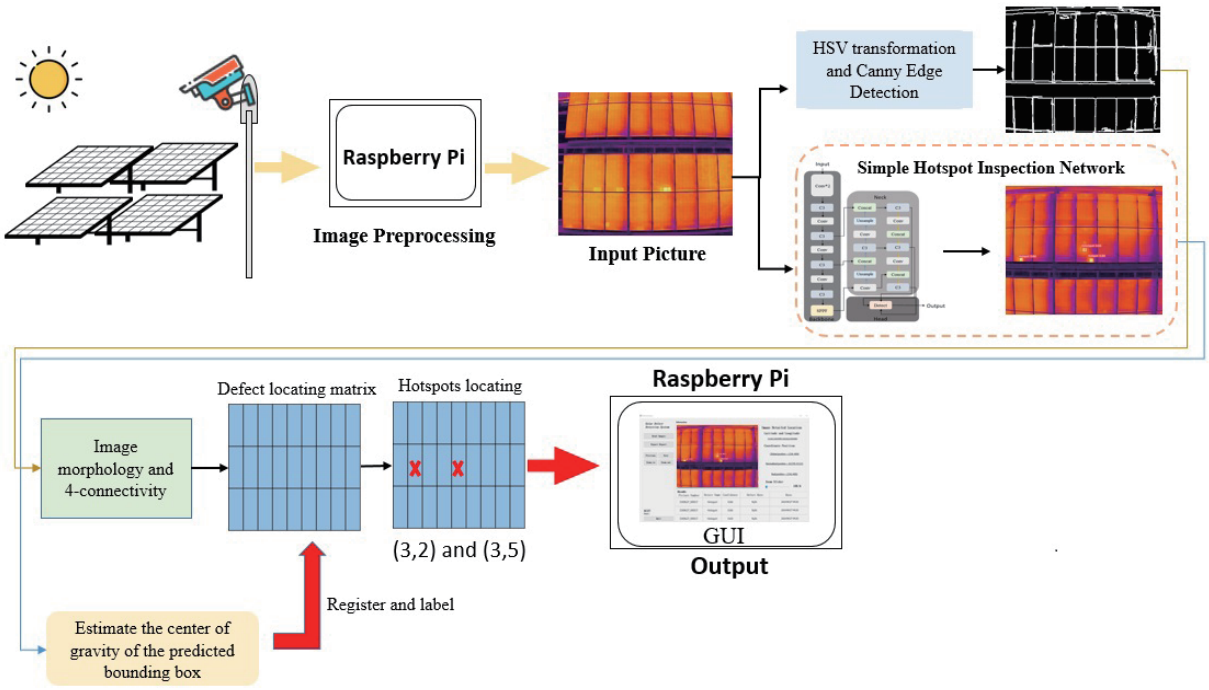

Fig. 2. (Color online) System framework.

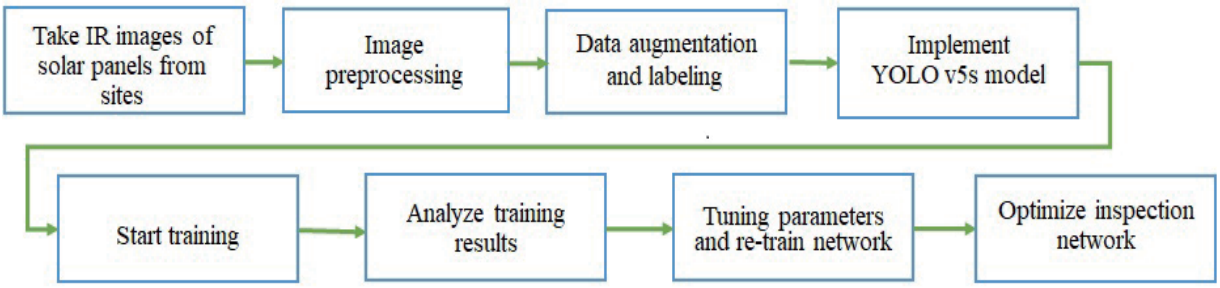

Fig. 3. (Color online) System training flowchart.

a quad-core Cortex-A72 processor clocked at 1.5 GHz and supports up to 8 GB of LPDDR4-3200 SDRAM. The IR camera is set up in the Raspberry Pi 4 embedded device to take thermal images. The implemented defect detection network then utilizes these images to inspect hotspots on the solar panels. As illustrated in Fig. 4, the Raspberry Pi 4 module is equipped with an IR camera and situated on site for data collection.

2.3 Simple hotspot inspection network with locating matrix

We propose a simple hotspot inspection network with a locating matrix to detect defects on solar panels in small-scale solar power plants. The inspection network must consider the

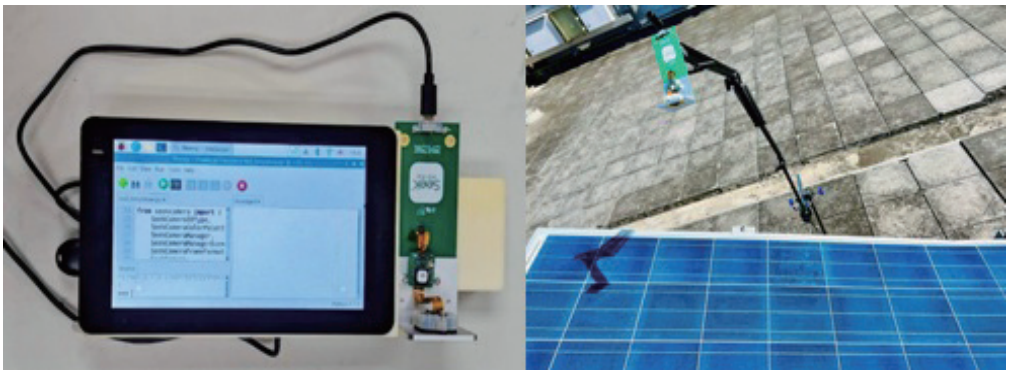

Fig. 4. (Color online) Raspberry Pi 4 module with 302NP IR camera.

computing resources available, given that the system is to be implemented in an embedded device with limited resources. Consequently, the detection network is based on the lightweight YOLOv5s model.

YOLOv5 is an advanced object detection network model with considerable popularity in computer vision. Its efficiency, accuracy, and lightweight design have contributed to its success. As illustrated in Fig. 5, the YOLOv5 architecture comprises three principal components. The backbone utilizes CSPDarknet53 as the primary network for feature extraction from the input image. The neck section employs the path aggregation network for feature fusion. Finally, the head component contains three detection layers of varying scales for object localization and classification.[11]

The YOLOv5 network comprises four versions: v5s, v5m, v5l, and v5x. Despite this, all four versions share the same backbone, neck, and head structures. The primary distinction among these versions lies in their module size, with the depth and width multiplier employed to achieve expansion. The depth multiplier corresponds to the depth of the model, which results in more layers being added to the network. The width multiplier increases the number of kernel filters in the layers, increasing channels for layer outputs. The YOLOv5 model employs these multipliers to facilitate the construction of scalable models. Table 1 provides a comparative analysis of the various versions. As demonstrated in Table 1, the v5s version exhibits the most compact model size and the highest recognition speed among all the versions. However, its accuracy is marginally lower, yet it remains adequate.

The convolutional block attention module (CBAM) attention mechanism has been incorporated within the YOLOv5 network, as illustrated in Fig. 6. CBAM is composed of two modules: channel-wise and spatial-wise modules. The efficacy of CBAM is demonstrated by its capability to suppress redundant information in features and enhance feature representation, thereby improving model accuracy.[12] The original YOLOv5 is designed with a complete *IoU* (*CIoU*) loss function. The efficient *IoU* (*EIoU*) is derived from the *CIoU* enhancement, which has been demonstrated to facilitate the refinement of the bounding box of the predicted objects, as illustrated in Eqs. (1)–(5).[13] The proposed detection network has a model-based design, with the YOLOv5s model forming the foundation. The detection system is required to employ the

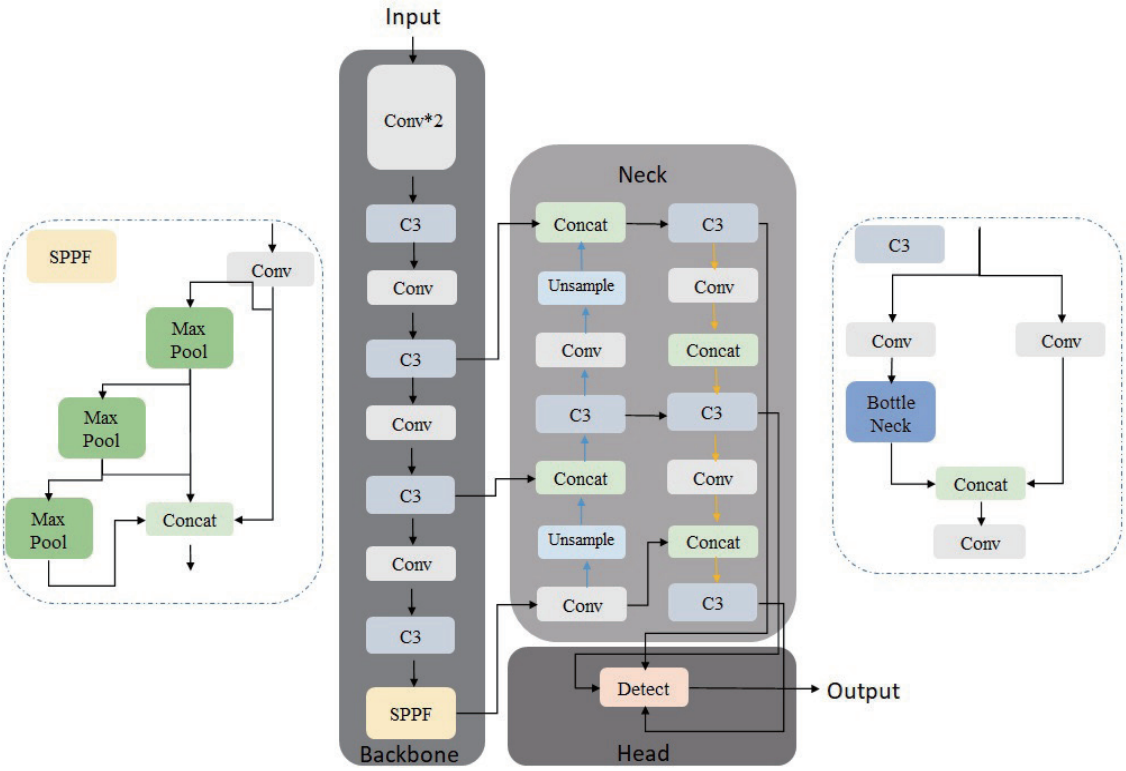

Fig. 5. (Color online) YOLOv5 network architecture.[13]

Table 1
Comparison of different versions of YOLOv5 model.[14]

YOLO version	Depth multiplier	Width multiplier	Speed CPU (ms)	*mAP*@50	Weights size (MB)
V5s	0.33	0.5	98	56.8	27.5
V5m	0.67	0.75	224	64.1	53.0
V5l	1.0	1.0	430	67.3	80.7
V5x	1.33	1.25	766	68.9	136.6

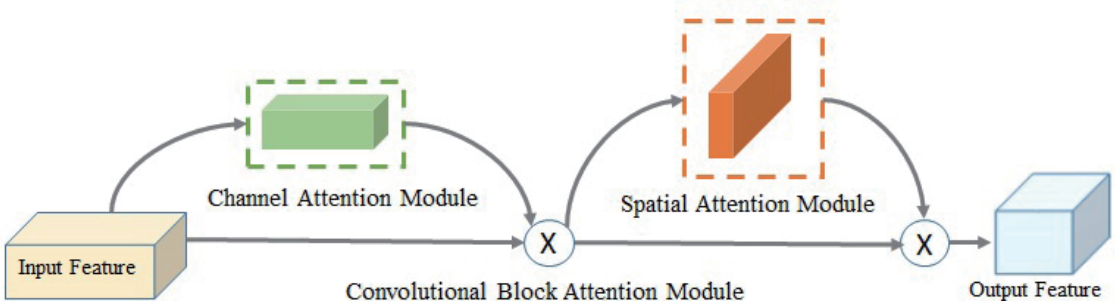

Fig. 6. (Color online) CBAM architecture.[12]

hotspot prediction bounding box to map to the locating matrix's detailed position. This necessitates a more accurate and proximate hotspot size bounding box to avoid erroneous correspondences. Consequently, the loss function of the detection network is incorporated into *EIoU*.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp} \tag{2}$$

$$L_{IoU} = 1 - IoU \tag{3}$$

$$L_{dis} = \frac{\rho^2\left(b, b^{gt}\right)}{c^2} \tag{4}$$

$$L_{asp} = \frac{\rho^2\left(w, w^{gt}\right)}{c_w^2} + \frac{\rho^2\left(h, h^{gt}\right)}{c_h^2} \tag{5}$$

Here, |A| and |B| are the predicted and ground truth boxes, respectively. L_{IoU}, L_{dis}, and L_{asp} are the overlap loss, the midpoint loss, and the width and height of bounding box loss, respectively. b and b^{gt} are the prediction and truth bounding boxes, respectively, w is the width of the prediction bounding box, and w^{gt} is the width of the truth bounding box.

After identifying hotspot defects on the panel using the YOLOv5s network, it is essential to designate the location in the field. In this study, we propose the defect-locating matrix method, which employs image processing techniques, color space transformation, object edge detection, and morphological processing to generate the locating matrix of the panel in the field. Subsequently, an estimation is conducted to ascertain the center of gravity of the hotspot bounding box, and this estimation is then employed to map it into the matrix. After this estimation and mapping, the registration and labeling of the hotspot defect are conducted.

It is a fact that video cameras generally capture images using the RGB color system. This type of imagery is susceptible to variations in illumination caused by shadows or changes in ambient lighting. The presence of noise in the images poses a significant challenge in segmenting objects of interest. Conversely, alternative color space systems, such as HSV or HSI, leverage different color spaces to present pixels to mitigate the impact of noise. The three-dimensional Cartesian coordinate system describes the RGB color space. It presents the object's color using the R, G, and B components in digital images, which reflect the amount of color reflected by the object under illumination and are also sensitive to illumination changes. The HSV color system, in contrast, utilizes cylindrical coordinates to present images that differ from those rendered by

the RGB system. This system is denoted by the hue (H), saturation (S), and value (V) components. The hue element is typically associated with color or chroma, encompassing variations such as red, green, and others. The saturation component, on the other hand, is defined as the ratio of colorfulness to brightness or chroma to lightness and is represented by gray-level intensities. In the absence of saturation, when the hue component is zero, the saturation component is also referred to as the brightness component. The value component is synonymous with "color lightness" and may be illustrated by light green or dark green examples.

The HSV color space is closer to the human visual perceptiveness than the RGB system. Equations (6)–(8) are applied to convert the RGB image to an HSV image.[15] In this paper, the infrared image in the field is used to generate the corresponding location matrix of the panels. As demonstrated in Fig. 7(a), the IR image displays the temperature distribution of the object itself, resulting in relatively simple coloration. Figures 7(b) to 7(d) illustrate the H, S, and V components following the HSV color space transformation application, thereby highlighting the more pronounced boundaries of the S and V components of the panels. However, if the Canny edge detection is performed directly with the RGB image, the image is full of noise, as is shown in Fig. 8(a), and the correct panel correspondence matrix cannot be segmented. Utilizing the intersection of HSV's S and V components to perform Canny edge detection results in an image shown in Fig. 8(b), delineating a distinct boundary for each panel. Subsequently, employing the methodologies of image morphology and 4-connectivity facilitates the generation of the corresponding position matrix within the field. Figure 9 presents a flowchart illustrating the process of identifying and locating hotspots in the matrix.

$$H=\begin{cases}0^\circ, & \text{if } \max(r,g,b)=\min(r,g,b)\\ 60^\circ\times\dfrac{g-b}{\max(r,g,b)-\min(r,g,b)}+0^\circ, & \text{if } \max(r,g,b)=r \text{ and } g\geq b\\ 60^\circ\times\dfrac{g-b}{\max(r,g,b)-\min(r,g,b)}+360^\circ, & \text{if } \max(r,g,b)=r \text{ and } g<b\\ 60^\circ\times\dfrac{b-r}{\max(r,g,b)-\min(r,g,b)}+120^\circ, & \text{if } \max(r,g,b)=g\\ 60^\circ\times\dfrac{b-r}{\max(r,g,b)-\min(r,g,b)}+240^\circ, & \text{if } \max(r,g,b)=b\end{cases} \tag{6}$$

$$S=\begin{cases}0, & \text{if } \max=0\\ \dfrac{\max(r,g,b)-\min(r,g,b)}{\max(r,g,b)}=1-\dfrac{\min(r,g,b)}{\max(r,g,b)}, & \text{otherwise}\end{cases} \tag{7}$$

$$V=\frac{\max(r,g,b)}{255} \tag{8}$$

(a) (b) (c) (d)

Fig. 7. (Color online) RGB IR image and HSV components. (a) RGB IR image and (b) hue, (c) saturation, and (d) value components.

(a) (b)

Fig. 8. (Color online) Comparison of images obtained with the Canny edge detection. Canny edge detections for (a) an RGB image and (b) the intersection of S and V components

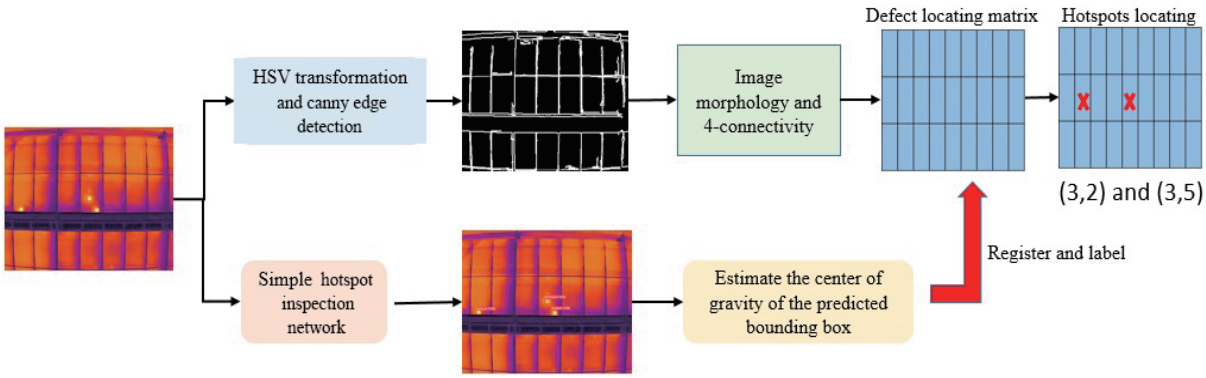

Fig. 9. (Color online) Flowchart of hotspot locating.

2.4 Data augmentation

The process of training a neural network to identify hotspots necessitates the utilization of numerous defective solar panel images. This is essential to mitigate the overfitting effect that enhances the accuracy and efficiency of the network's detection capabilities. Since this study

mainly focuses on detecting defects in small solar fields, it is not easy to obtain an open-access database. Therefore, it is necessary to take the images oneself to collect enough photos of solar hotspots. However, it is difficult to photograph and collect many pictures of defective panels, and the image augmentation of defective samples becomes an essential and indispensable step in the image preprocessing. This not only improves model training but also improves the model's adaptability to different scenarios. Standard data augmentation methods include rotation, image scaling, and property adjustment. These methods can effectively increase the size of the dataset and thus improve the performance of the network model. Different data augmentation methods can be implemented according to different task types.

In this study, we selected the following image augmentation methods to augment the original 1200 to 18000 images. The image property adjustment method that adjusts the brightness and contrast is a pragmatic approach to increasing the number of pictures while effectively suppressing sunlight's interference in hotspot detection. The rotation of the image has been demonstrated to enhance the diversity, thereby improving the accuracy of the detection model. Considering the potential impact of arboreal and architectural shading on the PV panel construction sites, the shading augmentation method is employed to simulate such scenarios. This approach expands the dataset and facilitates the model's adaptation to the generalizability in the real environment. As illustrated in Fig. 10, the images have been augmented.

3. Experimental Results and Discussion

In this section, we present the experimental results and analysis of network training. The evaluation metrics for testing and validating the network are then presented. Additionally, we applied different YOLO models to assess their performance, aiming to compare and determine the most suitable network model for this research.

3.1 Evaluation metrics

We utilized four evaluation metrics to evaluate our proposed network model's performance during training: precision, recall, *F*1 *score*, and *mAP*. All four metrics apply the confusion matrix, shown in Table 2, to estimate each value. Equations (9)–(13) represent each metric.

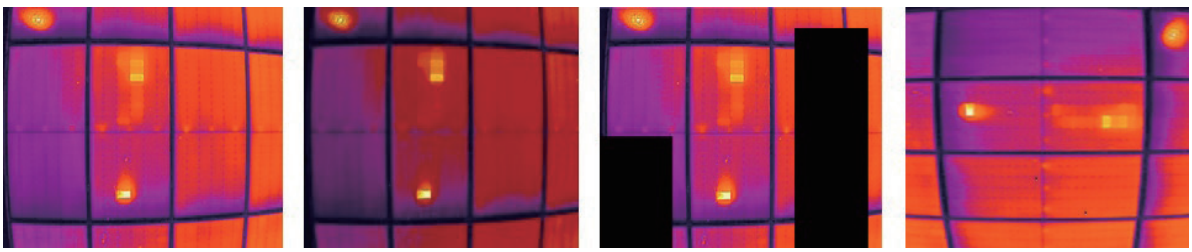

Fig. 10. (Color online) Image augmentation. (a) Original image, (b) contrast enhancement, (c) image shading, and (d) image rotation.

Table 2
Confusion matrix.

	Actual correct category	Actual incorrect category
Predicted correct result	True positive (TP)	False positive (FP)
Predicted incorrect result	False negative (FN)	True negative (TN)

$$Precision = \frac{tp}{tp + fp} \tag{9}$$

Here, *tp* represents true positive: the actual result is positive, and the prediction is also positive. *fp* represents false positive: the exact result is negative, but the prediction is positive.

$$Recall = \frac{tp}{tp + fn} \tag{10}$$

fn represents a false negative: the actual result is positive, but the prediction is negative.

The precision and recall metrics represent the model's capability using different definitions. However, when the sample data is not uniformly categorized, both metrics may not attain consistently high values. Consequently, the *F*1 *score* defined in Eq. (11) can serve as an alternative metric that provides a fair representation of the overall performance of the classification model by harmonizing and calculating the mean value of the two metrics. If the precision and recall values are similar, the *F*1 *score* will be high, indicating that the model is well balanced. However, if either value is low, the *F*1 *score* will also decrease, indicating that the model may be biased towards a particular type of error.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{11}$$

AP is the key metric by which the accuracy of each class is measured, which is usually defined as the area under the precision-recall curve. This can be calculated using Eq. (12). The value ranges from 0 to 1, where a higher value of *AP* corresponds to a higher level of accuracy. However, in practical applications, most object detections encompass multiple categories. Therefore, the mean average precision (*mAP*) is generally employed as the primary metric for evaluating the overall performance of the model, as illustrated in Eq. (13). *mAP* is determined by calculating the mean of the *AP* values across all categories, thereby indicating the model's accuracy and stability across different categories. In general, an elevated *mAP* value indicates that the model can detect different types of object with greater precision and exhibits superior recognition capability in various scenarios. Furthermore, the calculation of *mAP* yields different results when applied to different *IoU*s. The prevailing evaluation standard is *mAP*_0.5, which is defined as $IoU \geq 0.5$. In this paper, *mAP* is selected as the primary comparison standard to assess

the impact of varying model architectures, feature extraction methods, and training strategies, with the objective of identifying the most appropriate model for enhancing detection accuracy and performance.[16]

$$AP = \int_0^1 P(R)\,dR \tag{12}$$

$$mAP = \frac{1}{N}\sum_{k=1}^{N} AP_k \tag{13}$$

3.2 Training results

The simple hotspot inspection network is based on the YOLOv5s network model. Before the training stage, several critical hyperparameters must be configured, which are listed in Table 3. The loss function is characterized by two metrics: bounding box loss and object loss values. These metrics are utilized to estimate the training effect. "Box loss" denotes the discrepancy between an object detection model's predicted bounding box and the ground truth. As illustrated in Fig. 11, the box loss curve for the training and validation stages is presented. The object loss is used to show the overall confidence loss value, as demonstrated in Fig. 12. The results of three evaluation metrics are displayed in Fig. 13. The initial training epoch was set to 300; however, with the incorporation of the patience early stop command, the training was concluded at 295

Table 3
Training parameters.

Parameter	Value
Image size	640 × 640
Batch size	16
Epochs	300
Optimizer	AdamW
Learning rate	0.01
Patience	20

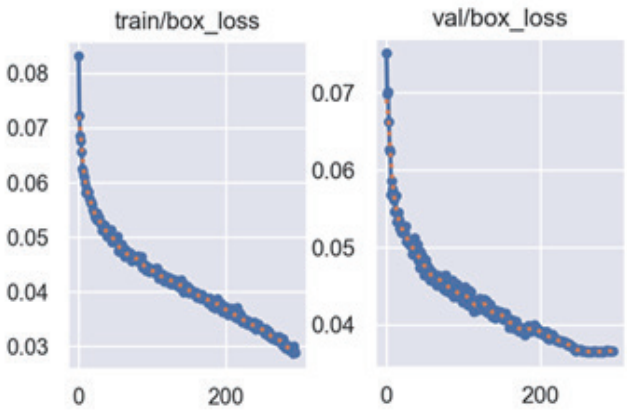

Fig. 11. (Color online) Box loss graph for YOLOv5s.

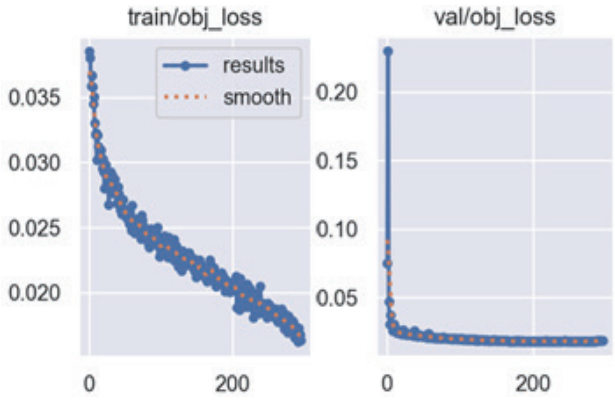

Fig. 12. (Color online) Object loss graph for YOLOv5s.

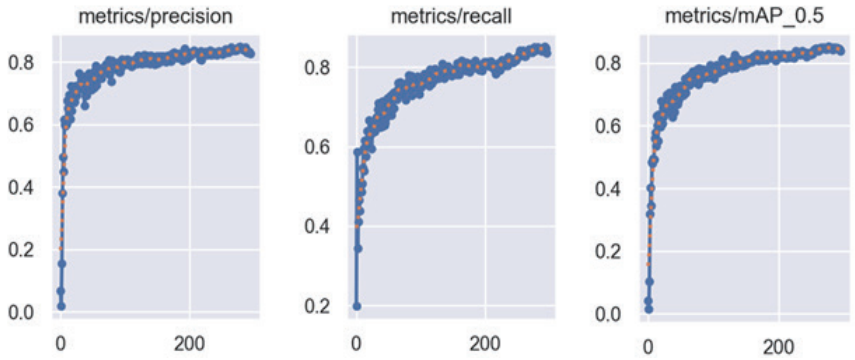

Fig. 13. (Color online) Evaluation metrics results for YOLOv5s.

epochs. As demonstrated by the curvature of the two loss curves and the three accuracy metrics, all loss values undergo a gradual decrease, reaching a low point, while the three evaluation metrics demonstrate a gradual increase. As shown in Table 4, the final validation loss values, namely, box loss and object loss values, are low at 0.037 and 0.018, respectively. In contrast, *mAP*_0.5 achieves a value of 0.852, indicating that the proposed detection network model exhibits higher accuracy and superior performance.

3.3 Prediction results

The prediction results are demonstrated in Fig. 14; the findings indicate the effectiveness of the simple hotspot detection network module proposed in this study for detecting panel defects in small-scale PV plants. As the figure shows, the detection network can accurately identify defective areas with a confidence level exceeding 60%. Then, applying the locating matrix to mapping the correct position of a defect panel facilitates the direct output of the location of panels with hotspots. This capability enables maintenance personnel to expedite the identification of defective panels for repair and to monitor the status of the PV plant in real time.

Table 4
Final training metrics results.

Metrics	Value
Train/box_loss	0.028
Train/obj_loss	0.016
Val/box_loss	0.037
Val/obj_loss	0.018
Metrics/precision	0.848
Metrics/recall	0.85
*F*1 *Score*	0.849
Metrics/*mAP*_0.5	0.852

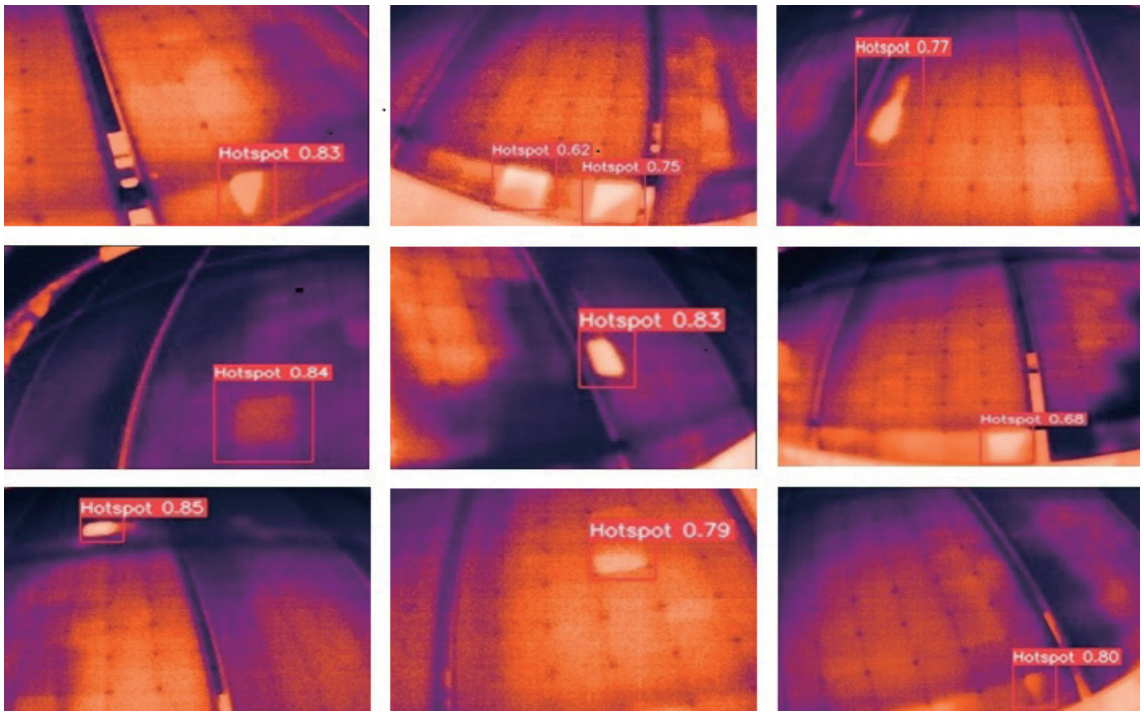

Fig. 14. (Color online) Prediction results.

3.4 Comparison and discussion

We conducted a comparative analysis of the network performance and efficacy of various lightweight YOLO models, including v8-nano and v9-tiny, in conjunction with the v5s model implemented in the proposed hotspot inspection network. These three models are derived from the related YOLO family, which are stable, lightweight, and suitable for implementation in embedded devices with constrained computational resources.[17,18] The architecture of YOLOv8 is based on an improved cross stage partial network, which replaces the original C3 block with a C2F block. This change reduces the computation amount while maintaining a high accuracy

rate. Furthermore, it introduces the decoupled head, which separates the classification and regression tasks. The classification loss utilizes the varifocal loss to address the target detection's positive and negative sample imbalance. In contrast, the regression loss incorporates *CIoU* and distribution focal loss (DFL). The amalgamation of these two components enhances the detection accuracy and stability of the model, thereby addressing the limitations of the conventional *IoU* loss in terms of localization accuracy. The YOLOv8-nano model has been modified to reduce its size by decreasing the number of channels and depth. The depth multiplier is set to 0.33, the width multiplier is 0.25, and the total parameter size is 3.2 M. The first layer of the standard version of the channel is 64, whereas in the nano version, it is only 16.[19]

YOLOv9 adopts GELAN as the backbone module to augment the comprehensive feature extraction capacity by implementing more efficient feature aggregation. The incorporation of the neck module with RTMNeck facilitates enhanced feature extraction. Additionally, the head module utilizes a decoupled head with DFL to discretely execute classification and regression tasks, thereby improving detection accuracy. Furthermore, YOLOv9 introduces programmable gradient information, incorporating an auxiliary reversible branch to enhance gradient flow during training. This enables the model to learn from an additional auxiliary channel, thereby optimizing the model's performance. In the YOLOv9-tiny version, the backbone module employs a simplified version of GELAN, which primarily reduces the number of channels and the number of stacked layers of blocks. This reduction in parameters and computation is significant, although it does result in a slight sacrifice in feature acquisition capability. However, the inference speed is enhanced. The YOLOv9-tiny neck module streamlines the RTMNeck configuration by diminishing the number of convolutional layers and channels. This approach results in a more lightweight overall model.[20]

These three lightweight models demonstrate efficacy in inference speed and the capability to substantially reduce model parameters and computational costs while maintaining high accuracy. Notably, their architecture renders them well-suited for remote solar panel defect detection and real-time applications. The v8-nano and v9-tiny versions are two recent developments designed to be more lightweight than the v5s version. As demonstrated in Table 5, the benchmark comparison results indicate that the two recent versions exhibit smaller parameter sizes than the v5s version and higher inference speeds. However, the v5s version demonstrates the optimal *mAP* accuracy.[21,22]

The two models compared are retrained with the same datasets in this paper. The training precision, recall, *F*1 *score*, and *mAP* plots of different YOLO models are presented in Figs. 15–18. The training results demonstrate that YOLOv5s exhibits superior performance across all evaluation metrics. After the training stage, the test dataset is used to estimate the network performance of each model. Table 6 demonstrates the comparison results for each evaluation

Table 5
Benchmark comparison of different YOLO network models.[23–25]

Model	Size (pixels)	*mAP*val (50–95)	Params (M)	FLOPs @640 (B)
YOLOv5s	640	43.0	9.1	24.0
YOLOv8n	640	37.3	3.2	8.7
YOLOv9t	640	38.3	2.0	7.7

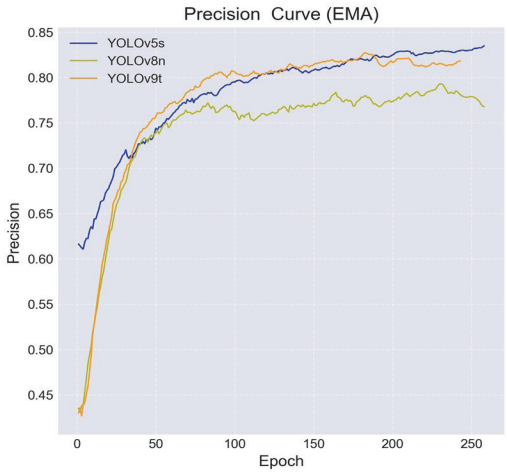

Fig. 15. (Color online) Precision curve.

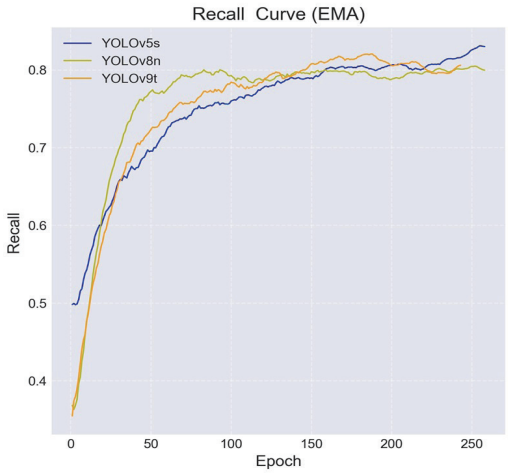

Fig. 16. (Color online) Recall curve.

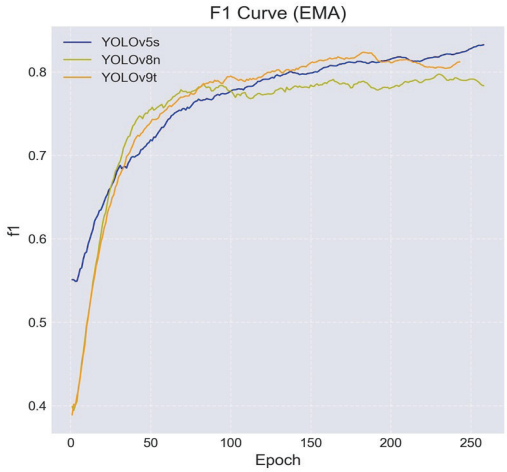

Fig. 17. (Color online) *F*1 *score* curve.

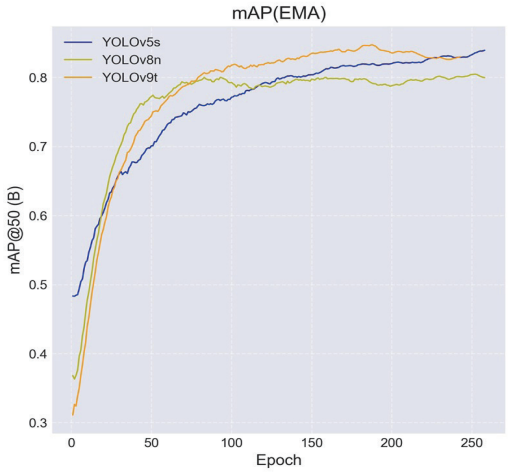

Fig. 18. (Color online) *mAP*_0.5 curve.

Table 6
Comparison of testing results in different YOLO models.

Model type	Test *mAP*_0.5	Average precision	Average recall	*F*1	*FPS*
YOLOv5s	0.851	0.845	0.846	0.846	66.8
YOLOv8n	0.81	0.797	0.796	0.796	52.8
YOLOv9t	0.845	0.828	0.815	0.821	71.4

metric, including AP, average recall, *F*1 *score*, *mAP*, and frames per second (*FPS*) which is the number of images processed every second by the network module. As shown in Table 6, the *mAP* accuracy value is the highest for the YOLOv5s model. However, the YOLOv8-nano and YOLOv9-tiny exhibit fast inference and a more compact module. The proposed simple hotspot

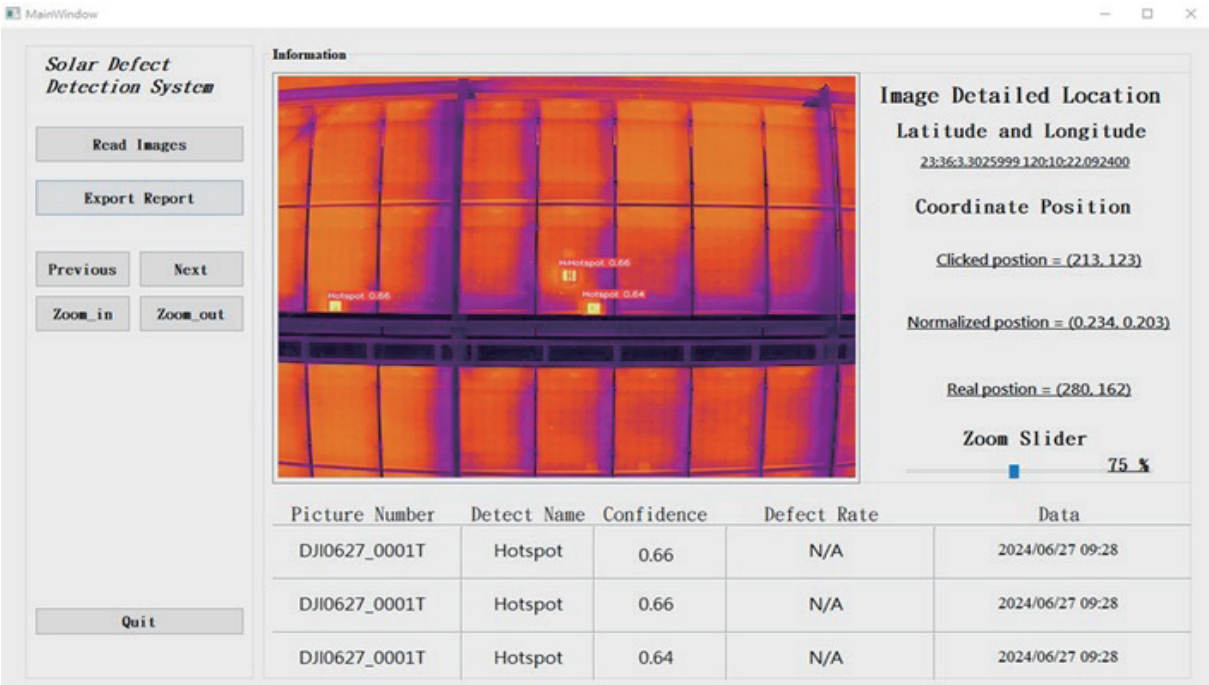

Fig. 19. (Color online) GUI of inspection system.

inspection network was designed with a trade-off among accuracy, inference speed, and model size in mind. It was based on the higher stability and accuracy of the YOLOv5s network.

3.5 Design of operation interface

In this paper, we propose a simple hotspot inspection network to monitor panel defects in small-scale or residential solar power systems. As demonstrated in Fig. 19, the system has been designed with a graphical user interface (GUI) that displays panels with hotspot defects and their respective positions in the field. This approach enables the presentation of real-time inspection results more intuitively and clearly, thus facilitating a more comprehensive understanding of the field's current status for maintainers and users.

4. Conclusions

The rapid development of renewable energy has prompted numerous countries to adopt solar power generation, particularly in home-oriented and miniaturized settings, especially in low-latitude areas with prolonged sunshine hours. However, prolonged operation of solar power generation systems often results in the occurrence of hotspots, which can potentially lead to a decline in system efficiency or equipment damage. The necessity of hotspot detection in PV system maintenance is therefore paramount. In this study, we proposed a simple hotspot

inspection network that utilizes the lightweight YOLOv5s network module in conjunction with a designed defect-locating matrix method to detect hotspots on the panels of small-scale PV plants. This system is particularly well-suited for embedded devices with limited resources owing to its high performance and lightweight characteristics, enhancing maintenance efficiency. Experimental results demonstrate the system's capability to delineate the inspected panels' hotspot area accurately. The comparison of testing results under the same operation environment shows the better performance of YOLOv5s, with a 6.02% increase in average precision, a 6.28% increase in average recall and *F*1 *score*, and a 5.06% increase in *mAP* accuracy compared with other YOLO models. Furthermore, the defect-locating matrix method can accurately determine the location of the defective panel, facilitating the real-time monitoring of the PV plant's status and a more intuitive display of results in the designed GUI.

References

1 T. Tajwar, O. H. Mobin, F. R. Khan, S. F. Hossain, M. Islam, and M. M. Rahman: Proc. 2021 IEEE 12th Energy Conversion Congress & Exposition - Asia (IEEE, 2021) 1542–1547. https://doi.org/10.1109/ECCE-Asia49820.2021.9478998

2 T. Suna, H. Xinga, S. Caoa, Y. Zhangb, S. Fana, and P. Liua: Energy Rep. **8** (2022) 1219. https://doi.org/10.1016/j.egyr.2022.08.130

3 Y. Higuchi and T. Babasaki: Proc. 2018 7th Int. Conf. Renewable Energy Research and Applications (IEEE, 2018) 391–396. https://doi.org/10.1109/ICRERA.2018.8566833

4 W. Sha, C. Dai, and L. Jiang: Proc. 2019 Int. Conf. Intelligent Computing, Automation and Systems (IEEE, 2019) 668–671. https://doi.org/10.1109/ICICAS48597.2019.00145

5 P. Addabbo, A. Angrisano, M. Bernardi, Graziano Gagliarde, A. Mennella, M. Nisi, and S. Ullo: Proc. 2017 IEEE Int. Workshop on Metrology for AeroSpace (IEEE, 2017) 345–350. https://doi.org/10.1109/MetroAeroSpace.2017.7999594

6 S. Cao, Y. Su, B. Zhao, and X. Yao: Proc. 2020 Chinese Control And Decision Conf. (IEEE, 2020) 1576–1579. https://doi.org/10.1109/CCDC49329.2020.9164863

7 K. Nakagawa, A. Kaligambe, and G. Fujita: Proc. 2023 4th Int. Conf. High Voltage Engineering and Power Systems (IEEE, 2023) 98–102. https://doi.org/10.1109/ICHVEPS58902.2023.10257489

8 H. Hajjdiab, G. E. Mohamed, S. Alzaabi, M. Alhalabi, A. Gad, M. Yaghi, and M. Alkhedher: Proc. 2023 10th Int. Conf. Future Internet of Things and Cloud (IEEE, 2023) 410–415. https://doi.org/10.1109/FiCloud58648.2023.00066

9 T. Alqahtani, A. Almutared, and A. Alzahrani: Proc. 2023 IEEE Int. Future Energy Electronics Conf. (IEEE, 2023) 327–330. https://doi.org/10.1109/IFEEC58486.2023.10458570

10 J. Chen, Y. Li, and Q. Ling: Proc. 2020 Chinese Control And Decision Conf. (IEEE, 2020) 4651–4655. https://doi.org/10.1109/CCDC49329.2020.9164255

11 Y. Lei, X. Wang, R. Wang, F. Yang, and A. An: Proc. 2023 China Automation Congress (IEEE, 2023) 5595–5600. https://doi.org/10.1109/CAC59555.2023.10451945

12 J. Ren, Y. Deng, Y. Hu, and S. Ding: Proc. 2023 3rd Int. Conf. Electronic Information Engineering and Computer Communication (IEEE, 2023) 704–708. https://doi.org/10.1109/EIECC60864.2023.10456687

13 D. Zhu, L. Dai, and P. Du: Proc. 2023 IEEE 5th Int. Conf. Civil Aviation Safety and Information Technology (IEEE, 2023) 824–829. https://doi.org/10.1109/ICCASIT58768.2023.10351744

14 Pretrained-checkpoints: https://github.com/ultralytics/yolov5#pretrained-checkpoints (accessed December 2024).

15 S. P. Pathak and S. A. Patil: IEEE Access **11** (2023) 72848. https://doi.org/10.1109/ACCESS.2023.3293756

16 M. R. Alvarez, C. Q. Vega, and L. Wong: Proc. 2023 Int. Conf. Electrical, Computer and Energy Technologies (IEEE, 2023) 1–6. https://doi.org/10.1109/ICECET58911.2023.10389215

17 U. Obu, Y. Ambekar, H. Dhote, S. Wadbudhe, S. Khandelwal, and S. Dongre: Proc. 2023 3rd Int. Conf. Pervasive Computing and Social Networking (IEEE, 2023) 528–533. https://doi.org/10.1109/ICPCSN58827.2023.00092

18 R. Liang, Y. Chen, S. Li, and H. Yang: Proc. 2023 3rd Int. Conf. Consumer Electronics and Computer Engineering (IEEE, 2023) 244–248. https://doi.org/10.1109/ICCECE58074.2023.10135213
19 M. Zhang and L. Yin: IEEE Access **10** (2022) 80804. https://doi.org/10.1109/ACCESS.2022.3195901
20 J. Anandakrishnan, A. K. Sangaiah, N. K. Son, S. Kumari, M. L. Arif, and M. A. Abd Rahman: Proc 2024 IEEE Int. Conf. Smart Internet of Things (IEEE, 2024) 16–21. https://doi.org/10.1109/SmartIoT62235.2024.00012
21 M. Razaghi, H. E. Komleh, F. Dehghani, and Z. Shahidi: Proc. 2024 13th Iranian/3rd Int. Machine Vision and Image Processing Conf. (IEEE, 2024) 1–5. https://doi.org/10.1109/MVIP62238.2024.10491172
22 Q. Wang, Z. Liao, and M. Xu: Proc. 2023 IEEE Int. Conf. Electrical, Automation and Computer Engineering (IEEE, 2023) 1412–1417. https://doi.org/10.1109/ICEACE60673.2023.10442092
23 Performance-metrics: https://docs.ultralytics.com/zh/models/yolov5/#performance-metrics (accessed December 2024).
24 Performance-metrics: https://docs.ultralytics.com/zh/models/yolov8/#performance-metrics (accessed December 2024).
25 Performance-metrics: https://docs.ultralytics.com/zh/models/yolov9/#performance-metrics (accessed December 2024).