 https://doi.org/10.24191/jeesr.v29i1.015

Instance Segmentation-Based Hotspot Detection in Solar Photovoltaic Arrays: A Comparative Study of YOLOv11 and YOLOv8

Nur Raihah Khiruddin, Kamarulazhar Daud*, Muhammad Khusairi Osman, Mohd Zulhamdy Ab Hamid, Ahmad Puad Ismail, Ahmad Asri Abd. Samat, Fadzil Ahmad, Lyly Nyl Ismail, Mohd Shawal Jadin and Sabhan Kanata

***Abstract*—Hotspot defects in solar photovoltaic arrays significantly impact energy efficiency and pose safety risks, necessitating automated detection solutions to replace time-intensive manual inspections. This study presents a comparative analysis of YOLOv8 and YOLOv11 instance segmentation models for hotspot detection in thermal imagery. A dataset of 766 manually annotated thermal images from solar farms in Kelantan, Malaysia, captured via drone-based inspections, was used for model training and evaluation. Multiple variants of both YOLO architectures were systematically compared across precision, recall, F1-score, mean Average Precision (mAP), and computational efficiency metrics. Results demonstrate that YOLOv11x achieves superior performance with the highest mask mAP@50 of 0.987, balanced precision of 0.959, and recall of 0.970, significantly outperforming YOLOv8 models. While YOLOv8 models generally process faster, YOLOv11x provides optimal accuracy-speed balance at 30.4 FPS, making it suitable for real-time monitoring applications. The study concludes that YOLOv11x is the most effective model for automated hotspot detection in solar panel monitoring systems, offering superior detection accuracy essential for preventing system failures and ensuring operational safety.**

***Index Terms*—Computer Vision, Hotspot Detection, Instance Segmentation, Solar Photovoltaic, Thermal Imaging, YOLOv8, YOLOv11.**

I. INTRODUCTION

With the rapid development of society, the global demand for renewable energy continues to grow [1]. Solar panels play a critical role in the field of clean energy, with their core technology based on advanced principles such as photoelectric or photovoltaic effects. This technology allows for the continuous and efficient direct transformation of abundant solar radiation into practical electrical energy [2,15]. Solar panels are supposed to generate power for at least 20 years, allowing individuals and businesses to decrease or even remove their energy expenses but, the continuous price war has caused some manufacturers to make shortcuts in order to stay in business. Consequently, some solar panels may begin to degrade before their intended lifespan is reached [3,20].

Some of the solar panel defects are hotspots, micro cracks, broken glass, dust build-up, cracked cells, snail trails, etc. [4,14]. Moreover, due to the installation of photovoltaic panels outdoors, they are easily covered by pollutants such as dust, debris, and bird droppings, which can cause local solar cells to overheat and form hotspots that affect power generation efficiency [5,17]. Any flaws, such as fractures, mismatches, or improperly soldered connections, might contribute to hotspots and increased resistance in the long term [6,13,21]. These defects not only impact the performance of solar cells but also can reduce battery life and pose potential safety risks [7,18]. Fig. 1 shows an example of the original thermal image of a solar panel containing a hotspot defect, which appears as a bright yellow-orange region indicating localised overheating.

In the field of surface defect detection and classification of solar panels, traditional methods often rely on manual inspection techniques, including visual inspection, the use of measuring tools, light reflection tests, electrical performance tests, infrared temperature measurements, and sound detection [2]. These methods can be time-consuming, costly, and subject to human error. As solar installations grow in size and complexity, there is a pressing need for more efficient and automated solutions that can overcome these limitations. Many existing solar panel inspection systems can detect areas but cannot correctly determine if these areas contain hotspots. This lack of proper classification may cause unnecessary maintenance or missed hotspots, resulting in increased costs and reduced system efficiency [1,4,14]. Without proper assessment of these key performance metrics across different model variants, it becomes difficult to understand which architecture provides superior hotspot detection capabilities. The absence of detailed comparative analysis between these two latest YOLO versions

This manuscript is submitted on December 9, 2025, revised on May 8, 2026, accepted on June 15, 2026, and published on October 31, 2026.

Nur Raihah Khiruddin, Kamarulazhar Daud, Muhammad Khusairi Osman, Mohd Zulhamdy Ab Hamid, Muhamad Asri Ismail, Ahmad Puad Ismail, and Fadzil Ahmad@Mohamad are with Electrical Engineering Studies, Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500 Permatang Pauh, Pulau Pinang, MALAYSIA. (e-mail: kamarul395@uitm.edu.my).

Mohd Shawal Jadin is with Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, MALAYSIA. (e-mail: mohdshawal@umpsa.edu.my).

Sabhan Kanata is with Faculty of Engineering & Computer Science, Universitas Muhammadiyyah Semarang, Central Java, 50273, INDONESIA. (e-mail: sabhankanata@unimus.ac.id).

*Corresponding author
Email address: kamarul395@uitm.edu.my

1985-5389/© 2026 The Authors. Published by UiTM Press. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

creates uncertainty for practitioners who need to select the most effective model for their specific solar panel monitoring applications.

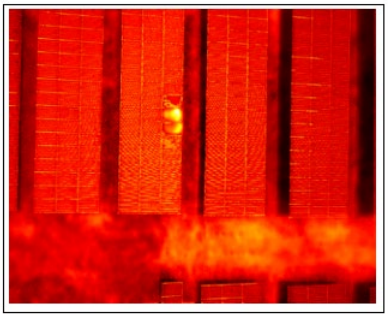


Fig. 1. Thermal image of solar panel showing hotspot defect.

This study explores instance segmentation using YOLO models, a modern approach in computer vision. The advantages of instance segmentation include its ability to scale across large datasets, automate the defect detection process, and significantly reduce human intervention, making it an ideal solution for inspecting solar farms. The research aims to develop instance segmentation models using YOLOv11 and YOLOv8 for hotspot detection in solar panels, classify solar panels areas into hotspot and non-hotspot regions using automated detection methods and evaluate the performance of both models in terms of precision, recall and computational efficiency. The scope of this project investigates and develops instance segmentation using YOLOv11 and YOLOv8 for the detection of hotspot defects in solar panels through preparing and preprocessing image data for training, implementing both model architectures to detect hotspot defects, and evaluating their performance using a dataset of 766 manually annotated thermal images from solar farms in Malaysia.

This study is important because it aims to make detecting and classifying hotspot defects in solar panels easier and more accurate. By conducting a comprehensive comparison between these two latest YOLO architectures, this study helps practitioners choose the most suitable model for their specific solar panel monitoring needs. Finding defects early and accurately can reduce repair time, prevent further damage, and help the panels operate more efficiently over a longer period. This is especially important as solar energy becomes a larger part of the world's energy supply.

II. Research Methodology

This study employs a systematic approach to develop and evaluate instance segmentation models for hotspot detection in solar photovoltaic panels. As shown in Fig. 2, the methodology follows a structured workflow that encompasses data collection, image annotation, data processing, model training, and performance evaluation phases. The research utilises both YOLOv8 and YOLOv11 architectures across multiple model variants to provide comprehensive performance comparison for hotspot detection applications.

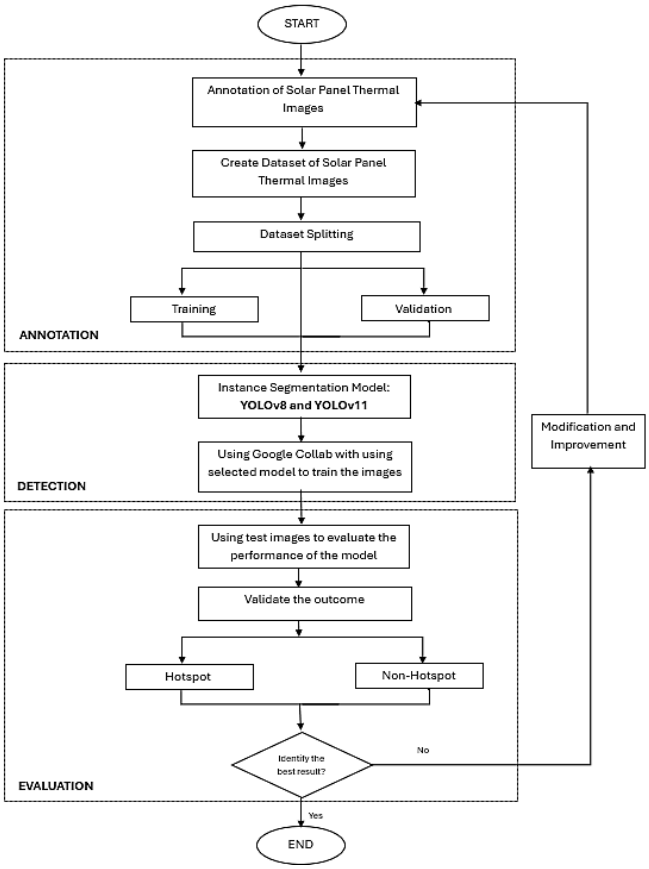


Fig. 2. Flowchart of research.

A. Image and Data Collection

The image data used in this study was obtained from previous research in the field [22]. The datasets were collected at solar farm located at Kelantan, Malaysia. It consists of 888 thermal images of solar panels captured using a drone equipped with a thermal camera. Using drones allows for efficient scanning of large solar farms and helps detect temperature variations across the panels. Thermal imaging enables the identification of hotspots, which appear as localised areas of high temperature and indicate potential defects such as partial shading, faulty soldering, and micro-cracks in solar cells. The dataset includes images taken from different angles to ensure complete coverage. These thermal images were used to train the model and serve as the primary resource for detecting and classifying hotspots, allowing the system to effectively identify anomalies in solar panels.

B. Image Annotation Process

Images of the solar panel thermal data were uploaded to Roboflow as the first step in the annotation process. The collected images are annotated using both manual and AI-assisted tools within the Roboflow platform. A total of 766 images has been annotated with high accuracy and detail for instance segmentation tasks. Each thermal image is carefully analysed to identify and label hotspot regions using polygon-based annotation tools that provide precise boundary marking of defect areas. This annotated dataset becomes the foundation for training the YOLOv8 and YOLOv11 models, ensuring they

learn to accurately recognise and detect hotspot defects in solar panels based on real-world data.

Roboflow was used to generate ground truth masks for instance segmentation, focusing specifically on hotspot regions as the primary target for detection. The polygon-based annotation tool allows for precise boundary definition of hotspot areas, creating accurate segmentation masks that outline the exact shape and extent of each hotspot defect. In this dataset, hotspots are treated as individual instances that require detection and segmentation as demonstrated in Fig. 3. Accurate and detailed labelling using polygon-based tools is essential to provide high-quality training data for model development.

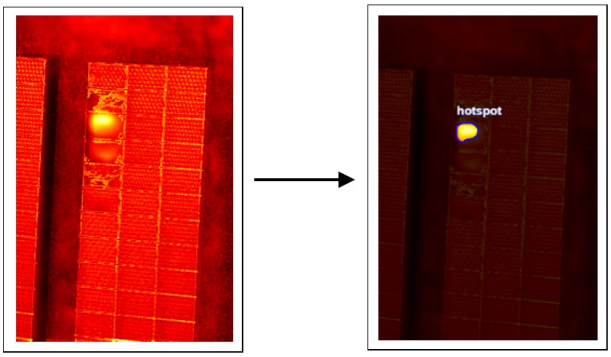


Fig. 3. Image annotation process showing (a) original thermal image of solar panel with hotspot defect and (b) polygon based annotated image with hotspot region labeled for instance segmentation training.

C. Data Processing

In machine learning, splitting the dataset is an important step to make sure the model is trained and tested properly. This process involves dividing the dataset into separate parts, where each part has a specific purpose in the training process. The main goals of splitting the data are to prevent overfitting, improve the model's ability to work with new data, and to measure its performance more accurately.

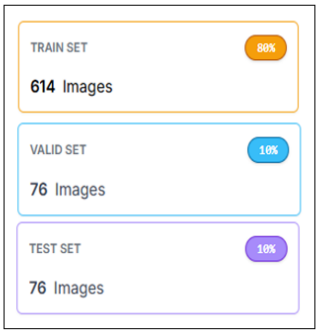


Fig. 4. Dataset split configuration showing distribution of 766 thermal images into training (614 images, 80%), validation (76 images, 10%), and test (76 images, 10%) sets.

For this study, a total of 766 annotated thermal images were divided into three separate sets, following standard practice. As shown in Fig. 4, 614 images (80% of the dataset) were used for training the model. This set helps the model learn patterns and adjust its parameters. The validation set includes 76 images (10%) and is used during training to fine-tune settings and to check if the model is starting to overfit. The last 76 images (10%) form the test set, which is only used at the end to evaluate the model's final performance on new, unseen data.

After the annotation process, the dataset is prepared using Roboflow's export feature to create a format that works with both YOLOv8 and YOLOv11 models. The annotation files are converted into TXT format, which is supported by both YOLO versions as demonstrated in Fig. 5. In this step, the TXT format is chosen for the annotations, and YAML configuration files are also included to describe the dataset structure and the class names used. The export process also creates segmentation masks and the necessary metadata for training the models using instance segmentation.

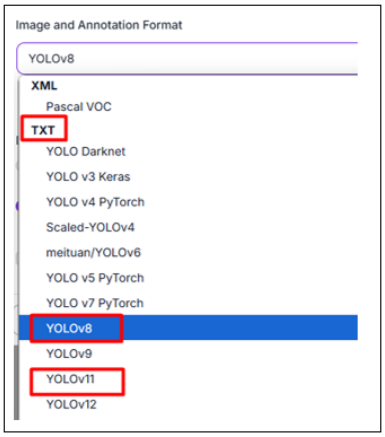


Fig. 5. Roboflow export interface shows TXT annotation format selection for YOLOv8 and YOLOv11 model compatibility.

D. Model Training

The YOLOv8 and YOLOv11 models were trained in a computing environment equipped with a Tesla T4 GPU with 15,095 MiB of memory, Python 3.11.12, PyTorch 2.6.0, and CUDA 12.4. The training was carried out on the Google Collaboratory platform to make use of cloud-based GPU resources, which allowed for efficient and faster model development. The dataset used for training consisted of 766 manually annotated thermal images of solar panels, which included hotspot defects captured under various operating conditions. The dataset was prepared through a consistent and structured process to ensure both models were trained under the same conditions. Pre-trained weights were applied to both YOLOv8 and YOLOv11 models to speed up the training

process. The training configuration employed standardised hyperparameters across all model variants to ensure fair comparison, as detailed in Table I. For both YOLOv8 and YOLOv11 model families, the training was conducted over 200 epochs to thoroughly adapt the models to the characteristics of the solar panel hotspot detection environment. All thermal image inputs were resized to a resolution of 640x640 pixels. This resolution was chosen to provide a good balance between processing speed and detection accuracy, which is important for identifying hotspot areas in thermal images of solar panels. The models were configured to detect a single class (hotspot) using the AdamW optimiser with a learning rate of 0.002. A batch size of 8 was used during training to make efficient use of GPU memory while keeping the training process stable. The dataset split followed an 80-10-10 distribution for training, validation, and test sets respectively, ensuring proper model development and evaluation procedures.

TABLE I. HYPERPARAMETER SETTINGS FOR TRAINING PROCESS

Hyperparameter	Description
Input image	[640, 640]
No. of class	1 (hotspot)
Optimiser	AdamW
Learning rate	0.002
Batch size	8
Total epoch	200
Data split	80% (training), 10% (validation), 10% (test)

Data augmentation techniques were applied in order to improve the model's ability to handle different conditions. These included horizontal flipping, blur effects, and changes in image intensity. These techniques helped simulate real-world situations in thermal imaging, such as changes in camera angle or lighting conditions.

E. Performance Evaluation Metrics

To evaluate the model's performance in classifying panels as hotspot or non-hotspot, precision, recall, and F1 score were used. Precision measures how accurately hotspot panels are identified, without falsely classifying any non-hotspot panel. Recall evaluates the model's ability to correctly identify all true hotspots panel and minimise missed detections. The F1 score is a balanced metric, as it evaluates both precision and recall, therefore providing an overall view of the entire model performance [8,19].The precision, recall and f1-score are calculated as follows [9].

$$Precision, P = \frac{TP}{TP + FP} \quad (1)$$

$$Recall, R = \frac{TP}{TP + FN} \quad (2)$$

$$f1\ score = \frac{2TP}{2TP + FP + FN} \quad (3)$$

The mean Average Precision (mAP) is an essential metric for evaluating the model's ability to detect hotspots in solar panels[10]. This metric assesses the precision and recalls of the predicted hotspots compared to the actual hotspots across different confidence thresholds. By calculating precision and recall at various confidence levels, mAP provides a comprehensive evaluation of the model's overall performance. A higher mAP value indicates superior hotspot detection capabilities, demonstrating that the model can effectively differentiate between actual hotspots and false predictions.

$$mAP = \frac{1}{|TP| + |FP|} \sum_{i=1}^{N} Precision\ (i)\ \times Recall\ (i) \quad (4)$$

In addition to accuracy metrics, the model's complexity and computational demand were evaluated by assessing the number of convolutional layers, total parameters, and Giga Floating Point Operations (GFLOPs) [11,16]. The computational characteristics of each model were examined to better understand their resource usage and architectural complexity. The number of parameters refers to the total count of trainable elements in the model. This directly impacts memory usage and storage requirements. GFLOPs represent the number of calculations the model performs during inference, which helps to measure its processing speed and efficiency. The number of layers in each model reflects the depth of the network architecture. This affects how well the model can extract useful features from thermal images and plays an important role in overall detection performance.

$$Parameter_{Model} = Total\ trainable\ weights\ and\ biases \quad (5)$$

$$GFLOPs = \frac{Total\ floating - point\ operations}{10^9} per\ image \quad (6)$$

$$Layers_{convolutional} = Total\ convolutional\ layer \quad (7)$$

III. RESULTS AND DISCUSSION

This section presents the comprehensive results obtained from implementing instance segmentation techniques for hotspot detection in solar panels using YOLOv8 and YOLOv11 models. The study evaluates the performance of YOLOv8 and YOLOv11 models to determine their effectiveness for this task. The results compare the different model variants to identify which ones perform best in detecting hotspots.

A total of ten models were tested, consisting of five YOLOv8 models and five YOLOv11 models. Each model comes in

different sizes, ranging from small (nano) to very large (extra-large). Smaller models generally process faster but may have lower accuracy, while larger models may offer higher accuracy but require more processing time. All models were trained and tested using the same procedures to ensure fair and consistent comparisons. The study measured the accuracy of each model detects hotspots and the speed of processes images. The analysis also included the computational requirements of each model.

Fig. 6 demonstrates the comparison of instance segmentation results between YOLOv8 and YOLOv11 models, where both models successfully detect and classify the hotspot region. These examples illustrate the effectiveness of both model architectures in identifying hotspot defects from thermal infrared images of solar panels.

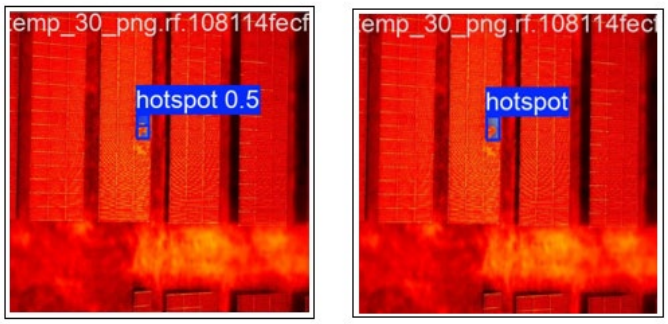


Fig. 6. Comparison of instance segmentation results between (a) YOLOv8 and (b) YOLOv11 models showing hotspot detection on thermal solar panel image.

The rest of this chapter is organised into five subsections as Evaluation Metrics: Precision, Recall, F1-score and mAP@50, Evaluation of Parameters, GFLOPs, and Layers used in YOLOv8 and YOLOv11, Training Time Analysis, Evaluation of Image Processing Speeds, and Discussion of Findings.

A. Precision and Recall Analysis for YOLOv8 and YOLOv11

The evaluation results show that YOLOv8 model variants have different performance strengths. As shown in Table 2, YOLOv8s achieved the highest precision of 0.958 for both box and mask predictions, making it highly accurate in identifying hotspots with fewer false alarms. In contrast, YOLOv8n recorded the highest recall with a mask recall of 0.999, meaning it detected nearly all actual hotspots in the dataset. This shows a clear trade-off between precision and recall, YOLOv8s focuses on accuracy while YOLOv8n prioritises complete detection. The medium model YOLOv8m performed well with mask precision of 0.956, close to YOLOv8s, but had lower recall at 0.874.

The larger models YOLOv8l and YOLOv8x did not always outperform smaller ones. YOLOv8l showed balanced but moderate performance with precision and recall around 0.840, while YOLOv8x achieved good recall of 0.920 but lower precision of 0.870. These results in Table II suggest that increasing model size does not guarantee better performance for hotspot detection.

TABLE II. PRECISION AND RECALL PERFORMANCE COMPARISON OF YOLOV8 MODEL VARIANTS

Model	Box		Mask	
	Precision	Recall	Precision	Recall
YOLOv8n	0.857	**0.959**	0.893	**0.999**
YOLOv8s	**0.958**	0.916	**0.958**	0.916
YOLOv8m	0.912	0.834	0.956	0.874
YOLOv8l	0.839	0.840	0.839	0.840
YOLOv8x	0.870	0.920	0.870	0.920

The performance analysis of YOLOv11 series shows clear differences among model variants. As shown in Table III, YOLOv11x achieved the most outstanding results with mask precision of 0.959 and mask recall of 0.970, representing the best-balanced performance in the series. The model also shows strong box precision of 0.897 and box recall of 0.911, meaning it correctly identifies hotspot locations while maintaining good detection coverage. This balanced performance suggests YOLOv11x provides optimal trade-off between accuracy and completeness in hotspot detection. In contrast, YOLOv11n achieved the highest box recall of 0.960 and competitive mask recall of 0.920, making it most effective at detecting all possible hotspots, though with lower precision scores.

The performance comparison between YOLOv8 and YOLOv11 models shows clear differences in their hotspot detection abilities. As shown in Tables II and III, YOLOv11x demonstrates superior balanced performance with mask precision of 0.959 and recall of 0.970, significantly outperforming the best YOLOv8 model (YOLOv8s), which achieved 0.958 precision but only 0.916 recall. While YOLOv8n shows exceptional recall at 0.999 of real hotspots, its lower precision of 0.893 creates more false alarms compared to YOLOv11x's 0.959 precision. This comparison shows that YOLOv11x provides the best overall performance by combining high accuracy with strong detection coverage.

TABLE III. PRECISION AND RECALL PERFORMANCE COMPARISON OF YOLOV11 MODEL VARIANTS

Model	Box		Mask	
	Precision	Recall	Precision	Recall
YOLOv11n	0.813	**0.960**	0.883	0.920
YOLOv11s	0.906	0.880	0.906	0.880
YOLOv11m	0.880	0.876	0.944	0.880
YOLOv11l	**0.923**	0.953	0.884	0.913
YOLOv11x	0.897	0.911	**0.959**	**0.970**

B. F1-score Analysis for YOLOv8 and YOLOv11

As shown in Table IV, YOLOv8s achieves the best F1-scores with 0.937 for both box and mask predictions, making it the most balanced model in the YOLOv8 series. This high score comes from its strong precision of 0.958 and good recall of 0.916, showing that it can find hotspots accurately without making too many mistakes. YOLOv8n follows with F1-scores

of 0.905 for box and 0.943 for mask predictions. Even though YOLOv8n has the highest recall in the series, its lower precision slightly reduces its overall balance score.

TABLE IV. F1-SCORE PERFORMANCE RESULTS FOR YOLOV8 MODELS

Model	Box (F1-score)	Mask (F1-score)
YOLOv8n	0.905	**0.943**
YOLOv8s	**0.937**	0.937
YOLOv8m	0.871	0.913
YOLOv8l	0.839	0.839
YOLOv8x	0.894	0.894

The F1-score analysis of YOLOv11 shows significant improvements compared to YOLOv8. As shown in Table V, YOLOv11x achieves the highest mask F1-score of 0.964, making it the best performer for balanced mask prediction across all models tested. This excellent score comes from its high precision of 0.959 combined with a high recall of 0.970. YOLOv11l shows the highest box F1-score of 0.938, indicating strong balanced performance for box detection, while achieving a mask F1-score of 0.898.

TABLE V. F1-SCORE PERFORMANCE RESULTS FOR YOLOV11 MODELS

Model	Box (F1-score)	Mask (F1-score)
YOLOv11n	0.880	0.901
YOLOv11s	0.893	0.893
YOLOv11m	0.878	0.911
YOLOv11l	**0.938**	0.898
YOLOv11x	0.904	**0.964**

C. Mean Average Precision (mAP@50) Results for YOLOv8 and YOLOv11

The mAP@50 results reveal an interesting distribution of performance across the model variants. As shown in Table VI, for the initial object detection task, the YOLOv8s model achieved the highest Box mAP@50 score of 0.943. However, for the more detailed task of instance segmentation, the YOLOv8n model recorded the highest Mask mAP@50 score of 0.965. This outcome is particularly significant, as it shows that the smallest and lightest model in the series produced the most accurate segmentation masks overall.

TABLE VI. MAP@50 PERFORMANCE RESULTS FOR YOLOV8 MODELS

Model	Box (mAP@50)	Mask (mAP@50)
YOLOv8n	0.929	**0.965**
YOLOv8s	**0.943**	0.928
YOLOv8m	0.924	0.958
YOLOv8l	0.912	0.927
YOLOv8x	0.925	0.943

The results from the YOLOv11 series show a significantly improved pattern of performance. As presented in Table VII, for the initial object detection task, the YOLOv11l model achieved the highest Box mAP@50 score of 0.961. In the more detailed task of instance segmentation, the YOLOv11x model recorded the highest Mask mAP@50 score of 0.987, representing the best segmentation performance across all models tested. This strong result is mainly due to its superior combination of high mask precision (0.959) and high mask recall (0.970), which significantly boosts its average precision score.

TABLE VII. MAP@50 PERFORMANCE RESULTS FOR YOLOV11 MODELS

Model	Box (mAP@50)	Mask (mAP@50)
YOLOv11n	0.921	0.933
YOLOv11s	0.932	0.934
YOLOv11m	0.928	0.953
YOLOv11l	**0.961**	0.923
YOLOv11x	0.939	**0.987**

D. Evaluation of Parameter, GFLOPs and Layers used in YOLOv8 and YOLOv11

According to Table VIII, the analysis of computational resource use between YOLOv11 and YOLOv8 models shows clear differences between the two designs for hotspot detection in solar panels. The YOLOv11 models show different parameter needs, ranging from 2.83 million parameters for the smallest model (YOLOv11n) to 62.00 million parameters for the largest model (YOLOv11x). The YOLOv8 models follow a similar pattern but use slightly more parameters, starting with 3.26 million for YOLOv8n and reaching 71.72 million for YOLOv8x. These parameters directly affect memory usage and processing needs, allowing users to choose suitable models based on available computer resources and required detection accuracy.

TABLE VIII. F1-SCORE PERFORMANCE RESULTS FOR YOLOV11 MODELS

Model		No. of Parameters	GFLOPs	Layers
YOLOv8	YOLOv8n	3,258,259	12.0	85
	YOLOv8s	11,779,987	42.4	85
	YOLOv8m	27,222,963	110.0	105
	YOLOv8l	45,912,659	220.1	125
	YOLOv8x	71,721,619	343.7	125
YOLOv11	YOLOv11n	2,834,763	10.2	113
	YOLOv11s	10,067,203	35.3	113
	YOLOv11m	22,336,083	123.0	138
	YOLOv11l	27,585,363	141.9	203
	YOLOv11x	62,003,283	318.5	203

The computational complexity, measured in GFLOPs (Giga Floating Point Operations), changes significantly across

different model sizes. The YOLOv11 models require between 10.2 GFLOPs for the smallest type and 318.5 GFLOPs for the largest type. The small YOLOv11 models work well for systems with limited resources or applications needing fast processing. In contrast, the larger YOLOv11 models need substantial computer resources but provide better accuracy for complex detection tasks. The YOLOv8 models show similar computational patterns, with needs ranging from 12.0 GFLOPs for the smallest model to 343.7 GFLOPs for the largest type. This shows the natural trade-off between processing speed and detection accuracy.

The design complexity, shown by the number of layers, reflects the design approach for different model types. The YOLOv11 models use different layer counts depending on their size setup. The nano and small YOLOv11 models both use 113 layers each, designed for efficient processing with minimal computational needs. The medium YOLOv11 model uses 138 layers, providing a balanced approach between speed and accuracy. The large and extra-large YOLOv11 models use 203 layers, with the extra-large type making up for this through increased parameter density per layer. The YOLOv8 models follow a similar design approach, with small models using 85 layers, medium models using 105 layers, and large models using 125 layers. The resource use efficiency shows how well each design balances computational needs with detection performance. The YOLOv11 models show better parameter efficiency, achieving good results with reduced computational needs compared to YOLOv8 models.

According to Fig. 7, the performance characteristics of different YOLOv8 and YOLOv11 model sizes applied in hotspot detection for solar panels are clearly shown. The findings provide practical guidance for model selection based on specific solar panel monitoring needs. For continuous monitoring systems with limited computational resources, the smaller types of both designs offer efficient training and processing abilities. For medium-scale solar installations requiring balanced accuracy and computational efficiency, the medium-sized models provide good performance. For large-scale solar farms demanding maximum detection accuracy, the largest models deliver superior results despite increased computational requirements and training time.

E. Training Time Analysis for YOLOv8 and YOLOv11

Both YOLOv8 and YOLOv11 models were trained using the same setup to ensure fair comparison. All models went through 200 training epochs, which means the training process was repeated 200 times. The choice of 200 was based on several important reasons for solar panel hotspot detection tasks. This number of epochs provides enough training time for the models to learn complex hotspot patterns while avoiding overfitting problems that could reduce performance on new solar panel images. The 200-epoch training period follows standard practices in computer vision research, especially for YOLO-based detection tasks, where most models achieve stable performance within this period.

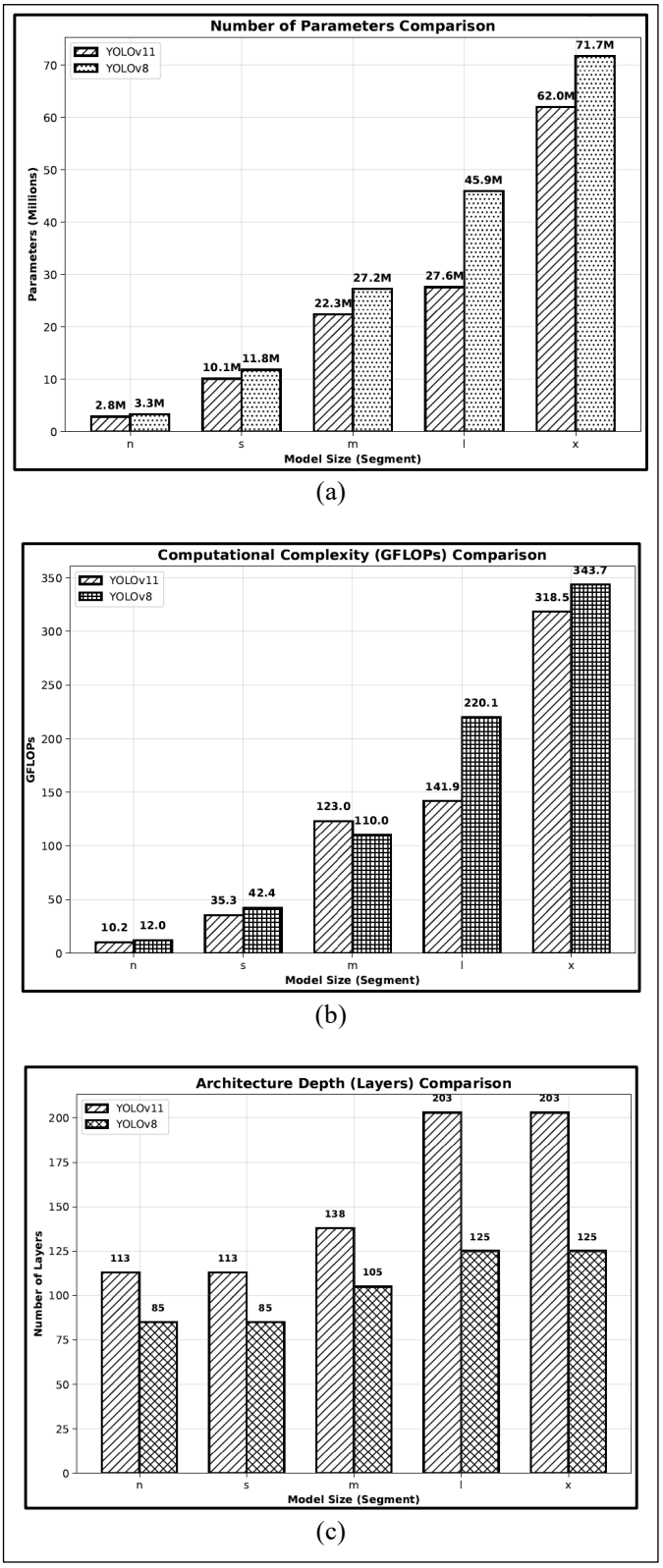


Fig. 7. Performance characteristics of different YOLOv8 and YOLOv11 model sizes applied in hotspot detection for solar panels. Subfigure (a) displays the total number of parameters in each model size, (b) shows the GFLOPs, and (c) illustrates the number of layers used in the architecture.

This approach is supported by recent research in radar image detection applications, where Yang et al. [12] successfully used 200 epochs for training YOLO models on Range-Doppler imagery, demonstrating that this training duration allows models to achieve optimal performance while maintaining practical training efficiency. Their study confirmed that 200 epochs provides sufficient time for model convergence without excessive computational overhead, making it suitable for academic research environments. The training was conducted on a standardised platform using a Tesla T4 GPU with enough memory capacity, allowing all models to train under the same conditions and making the comparison between different model types reliable and meaningful.

As shown in Table IX, the training time results show clear pattern between both model families, with model size directly affecting training duration. YOLOv8 models showed training times ranging from 1.006 hours for YOLOv8n to 3.906 hours for YOLOv8x. The small models (YOLOv8n and YOLOv8s) completed training efficiently in approximately 1 hour, while medium models required around 1.7 hours, large models needed about 2.5 hours, and extra-large models took nearly 4 hours. YOLOv11 models demonstrated similar scaling patterns but with generally better training efficiency. YOLOv11n achieved the fastest training time across all tested models at 0.897 hours, making it faster than YOLOv8n which required 1.006 hours. The YOLOv11 series completed training in times ranging from 0.897 hours to 3.570 hours, with YOLOv11x showing better time efficiency at 3.570 compared to YOLOv8x which took 3.906 hours.

TABLE IX. TRAINING TIME COMPARISON FOR YOLOV8 AND YOLOV11 MODELS OVER 200 EPOCHS

Model		Epochs Trained	Training Hours (hours)
YOLOv8	YOLOv8n	200	1.006
	YOLOv8s	200	1.054
	YOLOv8m	200	1.694
	YOLOv8l	200	2.508
	YOLOv8x	200	3.906
YOLOv11	YOLOv11n	200	0.897
	YOLOv11s	200	1.071
	YOLOv11m	200	1.812
	YOLOv11l	200	2.169
	YOLOv11x	200	3.570

When comparing both architectures, YOLOv11 shows consistent training efficiency advantages across most model sizes. According to Fig. 8, the results demonstrate that both architectures scale predictably with model complexity, where smaller models complete training in approximately 1-hour, medium models require around 1.7 to 1.8 hours, large models need about 2.2 to 2.5 hours, and extra-large models take 3.6 to 3.9 hours. This scaling pattern helps researchers choose the right model size based on their available time and computational resources. For researchers with limited time or computational resources, the nano variants of both architectures offer excellent training efficiency, with YOLOv11n providing the fastest option overall.

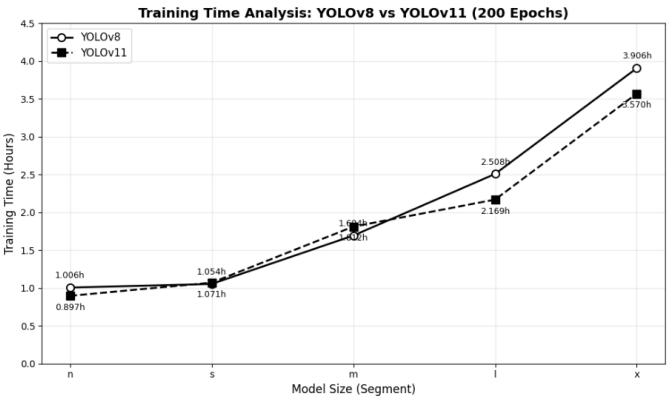


Fig. 8. Training Time Comparison for YOLOv8 and YOLOv11 models over 200 epochs

F. Evaluation of Image Processing Speeds

The evaluation of image processing speeds is an important part in checking how practical YOLO models are for solar panel hotspot detection applications. Processing speed shows how fast models can analyse thermal images and find hotspot problems. This performance measure is important for real-world solar panel monitoring systems, where fast hotspot detection helps maintenance teams fix problems before they become serious operational failures or safety risks.

The processing speed results shown in Table X show clear performance differences between model types with the measurements of inference speed and frame per second (FPS). YOLOv8n achieves the fastest processing speed at 4.2 milliseconds per image, equal to 238.1 FPS, making it the best choice for real-time monitoring systems that need very fast processing. YOLOv8s follows with a good performance at 7.2 milliseconds and 138.9 FPS, still providing excellent speed for most real-time applications. As model size increases, processing speeds become slower while keeping useful performance levels. The medium-sized models YOLOv8m and YOLOv11m achieve 65.8 and 57.1 FPS respectively, while the large models YOLOv8l and YOLOv11l both maintain about 42 FPS, providing a good balance between processing speed and detection accuracy.

The comparison between YOLOv8 and YOLOv11 models shows that YOLOv8 models generally process faster than their YOLOv11 counterparts across all model sizes. YOLOv11n processes at 8.0 milliseconds (125.0 FPS) compared to YOLOv8n's 4.2 milliseconds (238.1 FPS), showing that YOLOv8n is almost twice as fast. Similarly, YOLOv11s takes 9.1 milliseconds (109.9 FPS) while YOLOv8s takes only 7.2 milliseconds (138.9 FPS). For the largest models, YOLOv11x takes 32.9 milliseconds (30.4 FPS) compared to YOLOv8x at 33.8 milliseconds (29.6 FPS), showing that YOLOv11x is slightly faster than YOLOv8x while delivering better accuracy performance. This indicates that while YOLOv11 models

generally require more processing time due to their improved architecture, they provide better detection performance, and in the case of the largest models, YOLOv11x achieves both better speed and accuracy than its YOLOv8 counterpart.

G.Discussion on Findings

After comprehensive evaluation of both YOLOv8 and YOLOv11 across all performance metrics, YOLOv11x is determined as the most suitable YOLO model for practical deployment in solar panel monitoring systems. The selection of YOLOv11x as the most suitable model is based on its superior overall performance across critical metrics essential for practical solar panel monitoring deployment. YOLOv11x achieved the highest mask mAP@50 score of 0.987, significantly outperforming all other models including the best YOLOv8 model (YOLOv8n with 0.965). This represents the best performance segmentation across all the models tested.

TABLE X. PROCESSING SPEED PERFORMANCE SHOWING INFERENCE TIME AND FRAME PER SECOND FOR YOLOV8 AND YOLOV11 MODELS TRAINING

	Model	Inference Speed (ms)	FPS
YOLOv8	YOLOv8n	4.2	238.1
	YOLOv8s	7.2	138.9
	YOLOv8m	15.2	65.8
	YOLOv8l	23.6	42.4
	YOLOv8x	33.8	29.6
YOLOv11	YOLOv11n	8.0	125.0
	YOLOv11s	9.1	109.9
	YOLOv11m	17.5	57.1
	YOLOv11l	23.8	42.0
	YOLOv11x	32.9	30.4

The results show significant improvement over existing research in this field. YOLOv11x achieved a mask mAP@50 of 98.7%, which is better than previously reported YOLO-based hotspot detection methods. Analysis with existing literature shows that recent studies achieved mAP@50 values 72.9% for hotspot detection [13]. Even when compared to YOLOv8n, which achieved 96.5% mAP@50 in this study, YOLOv11x still demonstrates superior performance. YOLOv11x also shows good balanced performance with mask precision of 0.959 and mask recall of 0.970, providing good accuracy and completeness needed for reliable hotspot detection in real applications.

YOLOv11x offers significant practical advantages for solar panel monitoring systems despite requiring more computational resources. Its superior detection accuracy ensures that hotspots are identified with minimal false alarms while maintaining comprehensive coverage, which is critical for preventing system failures and safety hazards. The model's processing speed of 30.4 FPS is still adequate for real-time monitoring applications, as speeds above 30 FPS are sufficient for effective solar panel surveillance. Additionally, YOLOv11x demonstrates better parameter efficiency compared to YOLOv8x, using 62.00 million parameters versus 71.72 million parameters while achieving superior performance.

Therefore, YOLOv11x is conclusively identified as the optimal choice due to its exceptional detection accuracy (highest mAP@50 of 0.987), superior balanced performance (precision 0.959, recall 0.970), and adequate processing speed (30.4 FPS) for real-world solar panel monitoring applications where detection accuracy is paramount for system safety and reliability.

IV. CONCLUSION

In conclusion, this study successfully achieved its objectives by developing and comparing instance segmentation models for hotspot detection in solar photovoltaic panels using YOLOv8 and YOLOv11 architectures. The comprehensive evaluation of ten model variants across precision, recall, F1-score, mAP@50, and computational efficiency metrics revealed that YOLOv11x delivers the best overall performance with the highest mask mAP@50 of 0.987 and exceptional balanced performance with precision of 0.959 and recall of 0.970. While YOLOv8 demonstrated faster processing speeds, YOLOv11x provides the optimal combination of detection accuracy and processing efficiency at 30.4 FPS, making it highly suitable for real-world solar panel monitoring applications. The findings demonstrate that modern YOLO architecture can effectively automate hotspot detection, reducing reliance on manual inspection methods and enabling early detection of defects before they become critical operational failures. This research contributes valuable insights for practitioners selecting appropriate models for solar panel monitoring systems, with YOLOv11x being the recommended choice for applications prioritizing detection accuracy and system reliability. For future work, this research can be extended to develop real-time monitoring systems that provide continuous hotspot detection and automated alerts for solar farm maintenance.

TABLE XI. PERFORMANCE COMPARISON WITH EXISTING YOLO-BASED METHOD FOR HOTSPOT DETECTION

Model	mAP@50 (%)
YOLOv11x	98.7
YOLOv8n	96.5
YOLOv5n [13]	72.9

ACKNOWLEDGMENT

The authors wish to thank Universiti Teknologi MARA (UiTM), Cawangan Pulau Pinang, Malaysia, as well as Electrical Engineering Studies, Universiti Teknologi MARA Cawangan Pulau Pinang, Permatang Pauh Campus, 13500 Seberang Perai, Pulau Pinang, Malaysia, for all the support and funding given during this research. The authors extend their heartfelt gratitude to the Universiti Malaysia Pahang Sultan Abdullah (UMPSA) and Universitas Muhammadiyah Semarang (UNIMUS) for generously providing financial support through the Matching Grant under Grant No. RDU242729/UIC241531.

REFERENCES

[1] B. Zheng and Y. Yang, "Surface Defect Detection Technology for Solar Panels Based on Machine Vision," in *IEEE Information Technology, Networking, Electronic and Automation Control Conference, ITNEC*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1831–1835. doi: 10.1109/ITNEC60942.2024.10733164.

[2] Z. Liang, M. Xu, Y. Su, H. Chen, and G. Jin, "A Survey of Solar Panel Surface Defect Detection Methods Based on Improved VGG-16 Model," in *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI)*, IEEE, May 2024, pp. 820–827. doi: 10.1109/ICETCI61221.2024.10594221.

[3] A. M. Assaf, H. Haron, H. N. Abdull Hamed, F. A. Ghaleb, S. N. Qasem, and A. M. Albarrak, "A Review on Neural Network Based Models for Short Term Solar Irradiance Forecasting," *Applied Sciences*, vol. 13, no. 14, p. 8332, Jul. 2023, doi: 10.3390/app13148332.

[4] A. Bhutta, "10 Common Solar Panel Defects and How to Avoid Them," *Aztech Solar*, Nov. 23, 2025. https://www.aztechsolar.com.au/10-common-solar-panel-defects-and-how-to-avoid-them/.

[5] H. Hu, Y. Li, J. Wang, and Y. Lu, "Aerial Photovoltaic Panel Infrared Image Defect Detection Method Based on Improved YOLOv8," in *2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP)*, IEEE, Apr. 2024, pp. 1777–1780. doi: 10.1109/ICSP62122.2024.10743577.

[6] V. Raja, V. Rakesh, A. Kumar, and S. Siva Sankari, "Comparative Analysis on Solar Panel Defect Detection Using Deep Learning Approaches," in *2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, IEEE, Dec. 2023, pp. 1–6. doi: 10.1109/ICDSAAI59313.2023.10452666.

[7] Z. Zhang, S. Liu, and J. Xiong, "MLCA-YOLO: improved yolov8 for solar cell defect detection," in *2024 6th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI)*, IEEE, Jul. 2024, pp. 24–27. doi: 10.1109/IoTAAI62601.2024.10692802.

[8] I. K. Tasawar, A. K. Tanzeem, M. M. Rahman, T. Ahmed, M. Islam, and S. F. Zarin, "Fault Diagnosis of PV Modules Using Deep Convolutional Neural Networks," in *ICEPE 2022 - International Conference on Energy and Power Engineering: Green Energy for All*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICEPE56629.2022.10044899.

[9] H. Tella, M. Mohandes, B. Liu, S. Rehman, and A. Al-Shaikhi, "Deep Learning System for Defect Classification of Solar Panel Cells", doi: 10.1109/cicn.2022.77.

[10] Q. Zheng, J. Ma, M. Liu, Y. Liu, Y. Li, and G. Shi, "Lightweight Hot-Spot Fault Detection Model of Photovoltaic Panels in UAV Remote-Sensing Image," *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124617.

[11] R. Sapkota and M. Karkee, "COMPARING YOLOV11 AND YOLOV8 FOR INSTANCE SEGMENTATION OF OCCLUDED AND NON-OCCLUDED IMMATURE GREEN FRUITS IN COMPLEX ORCHARD ENVIRONMENT A TECHNICAL NOTE," 2025. [Online]. Available: https://docs.ultralytics.com/models/yolo11/

[12] J. Yang, M. Dong, C. Li, and F. Nie, "YOLO-RDFEA: Object Detection in RD Imagery with Improved YOLOv8 Based on Feature Enhancement and Attention Mechanisms," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3485499.

[13] L. Zhang, H. Chen, L. Zhang, J. Qin, and Q. Dong, "DFL-PVNet: A Lightweight Photovoltaic Hotspot Detection Network Based on Dynamic Feature Fusion," in *2024 5th International Conference on Intelligent Computing and Human-Computer Interaction, ICHCI 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 622–629. doi: 10.1109/ICHCI63580.2024.10807972.

[14] A. Jha, Y. Rawat, and S. Vyas, "PV-S3: Advancing Automatic Photovoltaic Defect Detection using Semi-Supervised Semantic Segmentation of Electroluminescence Images," Apr. 2024, [Online]. Available:http://arxiv.org/abs/2404.13693.

[15] P. A. A. Pramana and R. Dalimi, "Hotspot Detection Method in Large Capacity Photovoltaic (PV) Farm," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Dec. 2020. doi: 10.1088/1757-899X/982/1/012019.

[16] H. Al Mahdi, P. G. Leahy, M. Alghoul, and A. P. Morrison, "A Review of Photovoltaic Module Failure and Degradation Mechanisms: Causes and Detection Techniques," *Solar*, vol. 4, no. 1, pp. 43–82, Jan. 2024, doi: 10.3390/solar4010003.

[17] T. Sun, H. Xing, S. Cao, Y. Zhang, S. Fan, and P. Liu, "A novel detection method for hot spots of photovoltaic (PV) panels using improved anchors and prediction heads of YOLOv5 network," *Energy Reports*, vol. 8, pp. 1219–1229, Nov.2022, doi: 10.1016/j.egyr.2022.08.130.

[18] K. Gu and Y. Chen, "YOLOv3-MSSA based hot spot defect detection for photovoltaic power stations," *Journal of Measurements in Engineering*, vol.12, no. 1, pp. 23–39, 2024, doi: 10.21595/jme.2023.23418.

[19] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," Jan.2020,[Online].Available: http://arxiv.org/abs/2001.05566.

[20] M. Knott, D. Odion, S. Sontakke, A. Karwa, and T. Defraeye, "Weakly supervised image segmentation for defect-based grading of fresh produce,"Nov.2024,[Online]. Available: http://arxiv.org/abs/2411.16219.

[21] R. Sapkota, D. Ahmed, and M. Karkee, "Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments," *Artificial Intelligence in Agriculture*, vol. 13, pp. 84–99, Sep. 2024, doi: 10.1016/j.aiia.2024.07.001.

[22] M. Z. A. Hamid, K. Daud, Z. H. C. Soh, M. K. Osman, I. S. Isa, & M. S. Jadin, "Deep Learning-Driven Thermal Imaging Hotspot Detection in Solar Photovoltaic Arrays Using YOLOv10," *ESTEEM Academic Journal*, vol. 20, no. September, pp. 77–90, Sep. 2024, doi: 10.24191/esteem.v20iSeptember.1861.g1773.