Design of a Photovoltaic Panel Defect Detection System Based on the YOLO Algorithm

Jingjing Liu, Yihuan Zhang, Zikang Shao, Xinyu Zhang, Jingyi Yang, Dongyang Zhang, Boyang Zhang, Ruian Yan*

School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China

**Corresponding Author*

Abstract: With the rapid expansion of the scale of photovoltaic power plants, the industry operation and maintenance links put forward higher requirements for the efficiency, real-time and intelligent level of defect detection. However, traditional inspection methods mostly rely on manual observation or simple monitoring methods, which have problems such as low detection efficiency, limited real-time analysis capabilities, and difficulty in structured management of defect information. To this end, this paper designs and implements an intelligent monitoring system for photovoltaic modules, which integrates defect detection, target tracking, multi-modal analysis and digital twin visual management functions to improve the efficiency and reliability of power station inspection and operation and maintenance decision-making. The system uses a front-end and back-end separation architecture: the front-end builds an interactive visual interface based on the Vue 3 framework and TypeScript language technology; the back-end uses the Flask framework to build RESTful services, which are responsible for data processing and business collaboration. In order to improve the security and engineering availability of the system, the system introduces the JWT authentication mechanism, and adopts the asynchronous task scheduling mechanism based on Celery to support time-consuming reasoning and batch processing tasks. At the same time, the whole system enhances maintainability and scalability through modular design. In terms of defect analysis, the system integrates YOLO series target detection algorithm model, ByteTrack tracking algorithm and Qwen-VL-Plus multi-modal model to realize the recognition, location and continuous tracking of targets such as bird droppings, cracks, stains and panel areas. The experimental results show that in the crack detection task, the model achieves a performance of mAP @ 0.5 of 0.773 and a recall rate of 0.827. In order to adapt to the edge deployment scenario, the system further combines optimization strategies such as TensorRT acceleration, ONNX Runtime reasoning, model quantization and image enhancement to significantly improve reasoning efficiency and operational stability. The system can provide effective technical support for intelligent monitoring and early fault warning of photovoltaic power plants.

Keywords: Photovoltaic Module Defect Detection; Intelligent Monitoring System; YOLO; ByteTrack; Multimodal Analysis; Edge Deployment

1. Introduction

The integration of artificial intelligence, computer vision, and software engineering technologies in recent years has opened a new technological path for the intelligent operation and maintenance of photovoltaic power stations. Breakthroughs in deep learning and Transformer architectures have accelerated the rapid evolution of GPT series models since 2018. ChatGPT and GPT-4 have led industry transformation and ushered in the era of multimodal large language models, providing technical support for the interpretability analysis of defect detection results. Computer vision has evolved from traditional methods to deep learning-based paradigms. AlexNet has propelled convolutional neural networks into the mainstream, and models integrating attention mechanisms and Transformers architectures have significantly improved accuracy in photovoltaic panel defect detection. With

continuous advances in related technologies, application scenarios continue to expand. This system is based on the PyTorch framework, integrating the YOLOv8 object detection algorithm and the Qwen-VL multimodal large language model for defect detection. The backend is deployed on a lightweight hardware platform, whose low power consumption and high adaptability enable core functions such as real- time image scheduling and edge AI inference. Multimodal AI detection technology enables the integration of visual perception and semantic understanding into photovoltaic power station operation and maintenance scenarios. Through this integration, traditional operation and maintenance approaches are gradually evolving toward more intelligent, automated, and predictive forms. These capabilities help address several key challenges faced by the industry and also provide important technical support for the sustainable development of the new energy sector.

2. Front-end Design and Implementation

2.1 Front-End Technology Selection

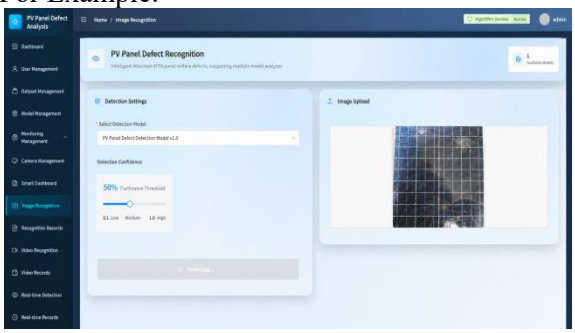
The front-end technology stack is built around the Vue 3.5 core framework, with TypeScript and ElementPlus as supporting tools to ensure development quality and efficiency [1-3]. Tools such as Pinia, VueRouter, and Axios, which are compatible with the Vue ecosystem, are selected for state management, routing, and data requests, respectively [4-6]. With the support of these technologies, the frontend development process follows a type-safe and component-based structure. The system is organized in a modular manner so that different functions can be developed and maintained independently, which helps improve code maintainability and team collaboration efficiency. Additionally, ECharts, flv.js, and Axios are introduced to meet the needs of data visualization, real-time video playback, and network requests, respectively, covering a complete range of business scenarios [7]. Vite is used as the project build tool to support faster project startup and efficient hot module replacement during development. The selected technology stack remains relatively lightweight and consistent, which helps maintain system stability while supporting continuous project iteration.

2.2 Front-end Page Design

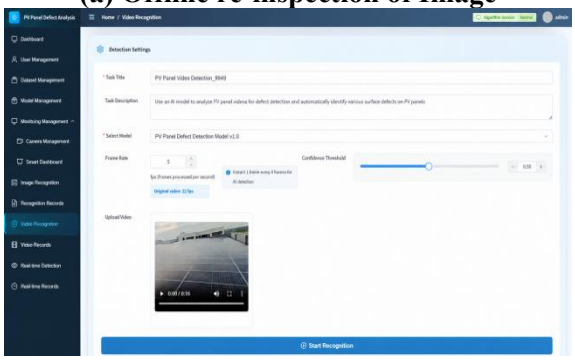
The front-end page module includes several functional components in the system workflow, such as data overview, model management, multi-form detection, analysis and decision-making, park visualization, and drone inspection [8]. The multi-modal detection module mainly supports three typical inspection scenarios: static image detection, offline video analysis, and real-time streaming media processing, which correspond to different stages of photovoltaic inspection tasks [9].Each detection mode allows users to adjust parameters such as model selection and confidence threshold. The interface also provides visual annotations of detected defects and supports exporting the detection results, which makes it convenient to review and record inspection outcomes. For real-time inspection, the system accepts streaming media input and performs continuous detection of photovoltaic modules. During operation, defect locations can be tracked dynamically so that operators can observe the detection results directly. Functions related to real-time detection and drone inspection also rely on streaming media protocols to display detection results and movement trajectories in real time, which is consistent with practical photovoltaic inspection scenarios. Figure 1 shows examples of detection results under four representative scenarios. Figure 1(a) illustrates the configuration interface used for offline detection. When the preset “Photovoltaic Panel Risk Identification Model v1.0” is selected, the improved YOLOv11n model weights are automatically loaded. At the same time, the interface displays a default confidence threshold of 50% and allows the detection process to start with a single click. Figure 1(b) presents an online spot-checking scenario on a production line. Running on a GTX 4060 GPU, the algorithm reaches a processing speed of 118 FPS and continuously detects broken-finger defects on moving photovoltaic panels. Figure 1(c) Device management: a unified camera interface retrieves and connects all online devices for real- time monitoring; camera ID 1 is shown as online, confirming the capacity for stable long- term access to multiple streams. Figure 1(d) Drone- based high- altitude inspection: the cropped oblique view focuses on the panels, and the algorithm localizes cracks with confidence scores (0.75, 0.58) while displaying the anomaly count (10), accuracy (95.4%), a one- minute trend curve, and real- time alarms. These results

validate the improved algorithm's high engineering deployment flexibility and strong multi- scenario generalization capability.

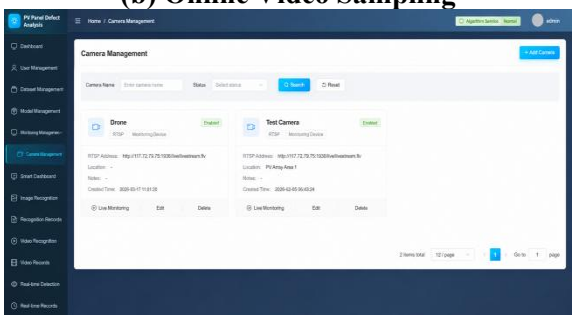
For Example:



(a) Offline re-inspection of Image



(b) Online Video Sampling



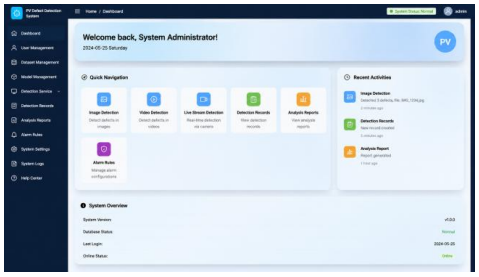
(c) Fixed surveillance camera

(d) Unmanned Aerial Vehicle Inspection

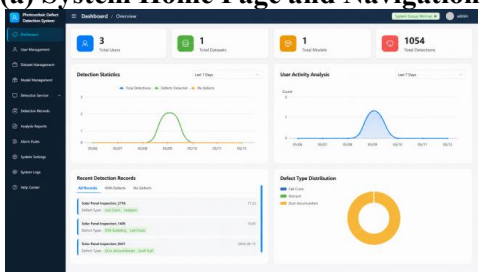
Figure 1. YOLO Photovoltaic Panel Multi-Morphology Detection Set Diagram

Each module is deeply tailored to the scenario-specific needs of photovoltaic panel defect detection and park energy management, with functions such as model management, analysis reports, and supporting progress monitoring and traceability viewing. The visual display screen achieves 3D modeling through 3D rendering, intuitively presenting the spatial layout and status of photovoltaic panels, and integrates the display of power station information, real-time operational indicators, performance analysis data, and alarm events. At the same time, all core modules provide operation interfaces for result export and record saving, combined with Qwen-VL to generate semantic analysis of detection results, providing defect causes, risk levels, and maintenance suggestions. While ensuring functional coverage across the entire business process, the operation path is simplified, balancing functional completeness and ease of use. Figure 2 presents the system interfaces across four representative scenarios. Figure 2(a) displays the offline image detection risk identification configuration interface, where users can select the preset "Photovoltaic Panel Risk Identification Model v1.0" to automatically load the improved YOLOv11n weight file; the interface shows the batch detection confidence threshold in real time (50% in this example) and provides a one-click intelligent detection start function. In the online production-line inspection (e.g. Figure 2 Assembly diagram of each module of the photovoltaic panel detection system(b)); only the video display area is shown, capturing a broken busbar defect stably detected on continuously moving photovoltaic panels. The device management function is illustrated in Figure 2(c): through a unified camera management interface, the system can retrieve and connect all online devices while supporting real-time monitoring, with camera ID 1 marked as "online," thereby validating the system's capability of stably accessing multiple surveillance video streams over long periods. Figure 2(d) shows the drone-based high-altitude inspection interface, captured from an oblique view with part of the background sky cropped to focus on the photovoltaic panels; the algorithm automatically identifies and localizes crack defects, annotating them with confidence scores such as 0.75, 0.58, and 0.68, and simultaneously displays the total anomaly count (10), the detection accuracy (95.4%), an anomaly trend curve over the past minute, and real-time defect alarm records, thereby realizing intelligent real-time monitoring and operation and maintenance control of photovoltaic module defects.

For Example:

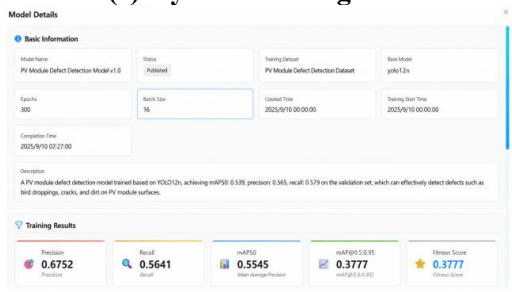


(a) System Home Page and Navigation



(b) Analysis of Testing Data Trends

(c) Hybrid 3D Large Screen



(d) Algorithm Model Management and Performance Metric

Figure 2. Assembly Diagram of Each Module of the Photovoltaic Panel Detection System

3. Backend Design and Implementation

3.1 Backend Functional Module Division

As illustrated in Figure 3, the backend system for photovoltaic panel defect detection is organized into six core modules: user authentication and permission management, dataset management, model training and management, defect detection service, detection record management, and intelligent analysis report generation. These modules together support the full workflow of an intelligent photovoltaic inspection system, from user registration to final defect analysis [10]. Specifically, the user authentication module implements fine-grained role-based access control mechanisms. The dataset module supports data upload, format verification, and automatic dataset partitioning, with an approximate ratio of 8:1:1 for the training, validation, and test sets. The model training module is built upon the YOLO series algorithms, which have demonstrated strong performance in real-time object detection and are widely applied in photovoltaic defect detection research [11]. It provides functions including model training, evaluation, and model publishing. The defect detection module supports multiple detection scenarios, including images, videos, and real-time streaming data, and integrates ByteTrack to enable object tracking within visual detection pipelines [12]. Detection results are persistently stored in the detection record module, which supports multi-dimensional querying and retrieval.

The intelligent analysis report module utilizes the Qwen-VL-Plus large model to conduct defect cause analysis, risk assessment, and maintenance recommendation generation. Through standardized interfaces, all modules interact with each other, thereby forming a complete and closed business workflow.

3.2 Backend Technology Selection

As shown in Figure 3, the backend technology stack is built around the Flask 3.0.x core web framework, complemented by SQLite3 database and JWT authentication mechanism, balancing development efficiency and system security. Tools such as Ultralytics YOLO series algorithms, ByteTrack, and Qwen-VL-Plus, which are suitable for AI detection scenarios, are selected for model inference, object tracking, and multimodal analysis [10,11]. This approach constructs a lightweight, high- performance, and modular service system, effectively enhancing the efficiency of algorithm implementation and detection accuracy. Additionally, the introduction of local file systems and Celery asynchronous queues ensures concurrent task processing, making the overall technology stack lightweight and efficient, suitable for small and medium-sized AI application deployment scenarios [13]. This achieves a strong ecological adaptability of the overall technology stack, low resource consumption, and balances system operational performance and deployment convenience.

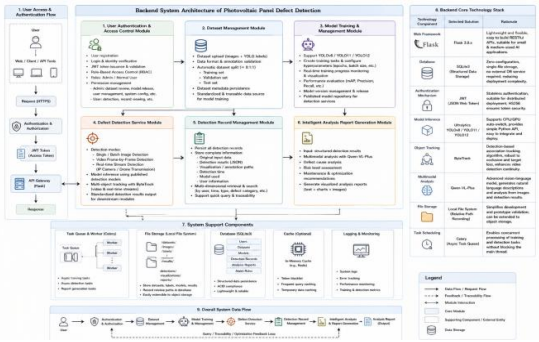


Figure 3. Overall Architecture and Data Flow of the Backend System

3.3 Construction and Implementation of Database

As shown in Figure 4, the database uses SQLite as its core storage engine, follows five design principles, and balances standardization with lightweight deployment requirements. Business aspects such as users, datasets, and models are designed with six core data tables, including users, datasets, models, and others, connected through foreign keys to build atraceable, structured, and highly adaptable storagesystem. This effectively improves data retrieval efficiency and enhances the ability for bidirectional traceability across the entire process. At the same time, field types that fit the business scenarios are precisely configured for each data table, with expansion space reserved to meet iterative needs, ensuring functional coverage across all business scenarios. SQLite, as a lightweight database system, provides constraint mechanisms that help ensure data uniqueness, integrity, and consistency [14]. These features make it suitable for managing system data in a stable and reliable way. The overall database design remains lightweight and portable, requiring no additional deployment dependencies. This approach effectively balances the system's storage needs while maintaining adaptability for different deployment scenarios. For Example:

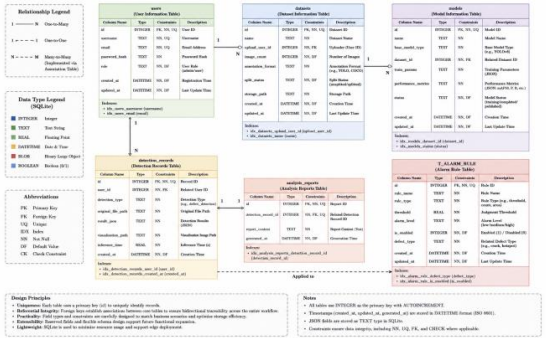


Figure 4. Database Structure Diagram

4. Research and Implementation of Photovoltaic Panel Defect Detection Algorithm Based on YOLO

4.1 Algorithm Architecture Design

This system adopts the YOLO family of single-stage object detection algorithms as the core visual perception engine, building a multimodal defect detection framework that supports images, videos, and real-time streams. Compared with two-stage detection methods such as Faster R-CNN, the YOLO series can maintain relatively high detection accuracy while providing faster inference speed, which satisfies the real-time inspection requirements of photovoltaic power plants [15,16].

As shown in Figure 5, the overall detection process follows a standard object detection pipeline. The input layer receives images with a resolution of 640 × 640, and letterbox padding is used to maintain the original aspect ratio and avoid feature distortion. The backbone network employs a cross-stage partial (csp) local connection structure, which reduces computational redundancy while retaining rich features. The neck network achieves bidirectional feature fusion through a path aggregation network (PANet) structure, enhancing the detection capability of defects at different scales [17]. The detection head outputs three scale feature maps for classification and bounding box regression. Finally, non-maximum suppression is used to remove redundant boxes, obtaining defect categories, confidence scores, and positional coordinates.

For Example:

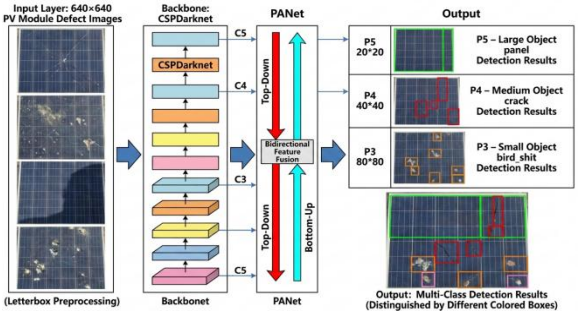


Figure 5. Flowchart of the YOLO Photovoltaic Panel Defect Detection Algorithm

4.2 Construction of a Dedicated Dataset for Photovoltaic Panel Defects

Constructing a dataset tailored to the actual scenarios of photovoltaic power stations is fundamental to enhancing defect detection

capabilities [18]. Addressing the issue of the lack of photovoltaic panel defect samples in public datasets, this study collected data covering various lighting conditions such as strong light, backlight, and cloudy weather, as well as different shooting angles such as front view, top view, and oblique view. After screening and annotation, a dataset containing 3276 images was ultimately constructed, covering four types of defects: bird droppings, cracks, soiling, and panel areas. The dataset samples and category distribution are illustrated in Figure 6.

For Example:

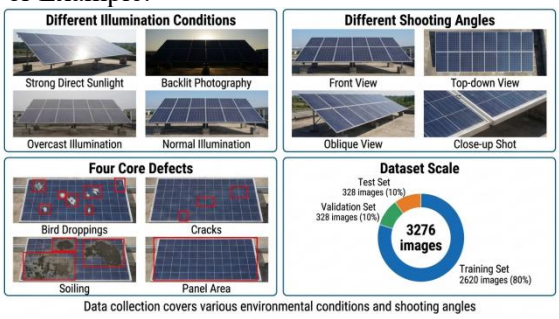


Figure 6. Diversity Demonstration of Photovoltaic Panel Defect Dataset

Data annotation follows the YOLO standard format and is manually conducted using the LabelImg tool. Each image corresponds to a text file that records the defect category and the relative coordinates of the bounding box's center point and width and height. The annotation results are cross-checked by two annotators, and difficult samples are arbitrated by a third party to ensure quality. The dataset is divided into a training set, a validation set, and a test set in an 8:1:1 ratio, and various categories are balanced. An example of the annotation format and the resulting dataset split is shown in Figure 7.

For Example:

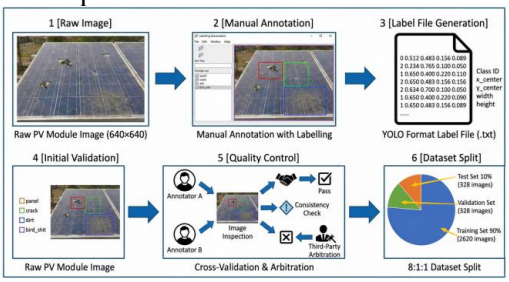


Figure 7. YOLO Data Annotation Flowchart

To meet the needs of model training, a multi-dimensional data augmentation strategy is adopted. Geometric transformations include random flipping, rotation, and scaling to simulate different shooting perspectives; color jittering perturbs within the HSV color gamut to adapt to changes in illumination; mosaic enhancement increases small target samples by splicing four images [17]; MixUp integrates two images and their labels to enhance robustness to occlusion and noise [19]. To address the issue of limited bird droppings samples, we combine oversampling and loss function tuning to augment minority class samples at the data layer, and guide the model to focus on difficult-to-classify samples through focal loss parameters [20].

4.3 Model Training Strategy and Hyperparameter Optimization

The model training is based on the PyTorch framework and is completed on an NVIDIA RTX3090 GPU server. Given the small data size (3276 images), a transfer learning strategy is adopted [21]: the backbone network is initialized with pre-trained weights from ImageNet, and the first 20 layers are frozen for the first 10 epochs, with only the Neck and detection head being trained, allowing the model to quickly adapt to photovoltaic defect features. Subsequently, all parameters are unfrozen, and end-to-end fine-tuning is performed at a lower learning rate. The training input size is 640 × 640, with a batch size of 16. The SGD optimizer (momentum=0.937) and cosine annealing learning rate strategy are used, with the learning rate decaying from 0.01 to 0.0001 and a weight decay of 0.0005. Training is performed for 50–100 epochs, and an early stopping mechanism is set to prevent overfitting. As shown in Figure 8, the loss function combines classification loss, localization loss (CIoU) [22], and confidence loss, summing them by weight to form the total loss, guiding the model to achieve balanced optimization between classification and localization tasks.

For Example:

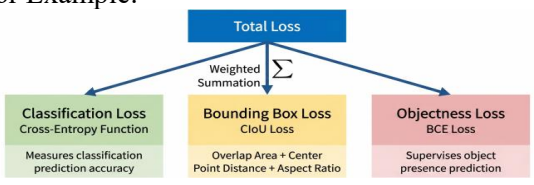


Figure 8. YOLO Loss Function Structure

4.4 Multi-scene Detection Mode and Algorithm Performance Evaluation

After format conversion and image resizing, images are sent into the YOLO network for inference. The system can also support comparative testing among different models,

including YOLOv8, YOLOv11, and YOLOv12. Video detection technology analyzes the detected video. The system extracts key frames to reduce redundant computations. It also applies the ByteTrack algorithm for cross-frame object tracking. The process enables the system to form a continuous spatiotemporal trajectory for defect detection. The real-time detection is mainly used for online monitoring. The system adopts a multi-threaded pipeline architecture. It combines TensorRT INT8 quantization with an adaptive frame-rate adjustment mechanism which can improve detection speed and maintain stable system performance.

This experiment evaluated the algorithm's behaviour on an independent test set using metrics including mAP@0.5, mAP@0.5:0.95, precision, recall, and FPS. The results reveal that YOLOv12n receives the best overall performance, with an mAP@0.5 of 0.539 approximately 5% higher than YOLOv8n while the inference speed remains stable. From the detection results, the model performs well in identifying panel areas and crack defects. However, there is still room to improve about detecting smudges and bird droppings. This is mainly because these defects have indistinct edges and relatively fewer training samples are collected. In the aspect of real-time performance, the detection time for a single image should be controlled around 100 ms, video running speed needs to exceed 10 FPS, and the delay for real-time streaming detection should preferably remain under 500 ms. Compared to traditional human-operated inspection, this system significantly improves detection efficiency at the same time delivering more consistent data results.

5. Overall System Design

As shown in Figure 9, this system is specially developed for photovoltaic defect detection scenarios. It adopts a four-layer technical architecture consisting of a presentation layer, an application layer, an algorithm layer, and a support layer. These four layers have independent functions and work in efficient coordination to complete the overall detection tasks. The presentation layer serves for user interaction and is built on the Vue.js framework with TypeScript and mainstream modern front-end technologies [1,2]. Vue Router is used for page route management and jump control, Axios realizes data transmission and interaction between the front end and back end, and Pinia is responsible for handling complex data state management in the project [6]. In addition, Element Plus and ECharts component libraries are integrated into the system. They help build a simple and user-friendly operation interface and intuitively present system data in diverse visualized chart forms. The application layer is developed based on the Flask framework and mainly includes modules for user interaction, business processing, database access, and entity management. The frontend communicates with the back end through Web APIs to support data transmission and function invocation. To avoid request blocking during image analysis, asynchronous task processing and a message queue mechanism are introduced to improve system response efficiency. In addition, the system integrates database management, log recording, permission verification, and file upload functions to ensure stable operation and data security. The algorithm layer is implemented in Python and combines technologies such as PyTorch, YOLO, OpenCV, and multimodal large models [18]. This layer is responsible for image preprocessing, model training, and defect analysis, and it also provides standardized algorithm APIs for photovoltaic defect detection and decision support. The support layer covers data storage (SQLite), computing resources (CPU/GPU), and operating system compatibility, along with digital twin and 3D rendering technologies. Real-time feeds from drones and cameras are used to create three-dimensional visual monitoring and situational awareness for the PV plant. HTTP/HTTPS, JSON, JWT, and secure transmission protocols connect the layers. The result is a system with the scalability, reliability, and maintainability needed for the complex demands of photovoltaic defect detection.

For Example:

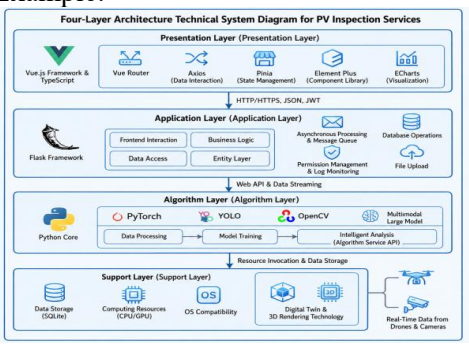


Figure 9. Architecture Diagram of Photovoltaic Detection System

6. Summary and Expectations

For the existing technical pain points in the photovoltaic station operation and maintenance, this study is based on AI vision technology. It designs an online defect detection system. It also implements the system for photovoltaic panels.

The actual tested data reveals that the system still has certain capability limitations under complex working situations. In the interference scenarios like strong backlight, water stain coverage, surface dust accmulation, the system gains recognition accuracy and targets subtle photovoltaic panel defects. It's accuracy will be reduced obviously. Limited by the sample data size of this study, the system poorly identifies low-frequency atypical defects.

Frame deformation and junction box failure fall into this category. So that it still leaves much room for improvement.

At the same time, the real-time detection frame rate and data processing efficiency of the system in low-configuration edge devices have not fully met the stringent requirements of large-scale field applications. Subsequently, by expanding the special defect sample database and optimizing the learning algorithm of small samples, the recognition accuracy of special defects can be improved, and the shortage of scene detection ability can be supplemented. Further improve the comprehensiveness and accuracy of system inspection.

Subsequent research will continue to promote iterative optimization of the system, focusing on improving improving the model detection accuracy in complex scenarios, improving the end-edge cloud cooperative operation efficiency, and improving the interface adaptability between the system and the industry platform. The purpose of this research is to continuously meet the actual operation and maintenance needs of photovoltaic power stations, build an efficient and intelligent photovoltaic panel defect detection technology system, and provide technical support for intelligent upgrade of photovoltaic operation and maintenance.

References

[1] Vue.js Team, Vue. js v3.5 Documentation, 2024. [Online]. Available: https:// vuejs.org/.

[2] Microsoft, TypeScript 5.5 Handbook, 2024. [Online]. Available: https:// www. typescriptlang.org/.

[3] Element Plus Team, Element Plus 2.x Documentation, 2024. [Online]. Available: https:// element-plus.org/.

[4] E. You, Pinia: The intuitive store for Vue.js, 2024. [Online]. Available: https:// pinia.vuejs.org/.

[5] E. You, Vue Router 4, 2024. [Online]. Available: https:// router.vuejs.org/.

[6] Axios Contributors, Axios HTTP Client, 2024. [Online]. Available: https:// axios-http.com/.

[7] Apache ECharts Team, ECharts 5.5 Documentation, 2024. [Online]. Available: https:// echarts.apache.org/.

[8] X. Sun et al., "A Web-Based Visualization System for Smart Factory Monitoring Using ECharts,"2024 IEEE 7th International Conference on Computer and Communication Systems (ICCCS), 2024, pp. 123-128.

[9] Bilibili Inc., flv. js: HTML5 Flash Video (FLV) Player, GitHub repository, 2024. [Online]. Available:https:// github.com/ bilibili/ flv. js.

[10] M. Deitsch, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess, "Automatic classification of defective photovoltaic module cells in electroluminescence images," Solar Energy, vol. 185, pp. 455–468, 2019, doi: 10.1016/ j.solener.2019.02.067.

[11] J. Redmon and A. Farhadi,"YOLOv3: An incremental improvement," arXiv preprint arXiv: 1804.02767, 2018, doi: 10.48550/arXiv.1804.02767.

[12] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," in Proc. European Conf. Computer Vision (ECCV), Cham, Switzerland: Springer, 2022, pp. 1–21, doi: 10.1007/978-3-031-20047-2_17.

[13] N. Dragoni, S. Giallorenzo, A. Lluch-Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina," Microservices: Yesterday, today, and tomorrow," in Present and Ulterior Software Engineering, Cham, Switzerland: Springer, 2017, pp. 195–216, doi: 10.1007/978-3-319-67425-4_12.

[14] D. R. Hipp, "SQLite," Proceedings of the VLDB Endowment, vol. 3, no. 1–2, pp. 1723–1724, 2010, doi: 10.14778/1920841.1920887.

[15] Redmon J, Divvala S, Girshick R, et al. You

Only Look Once: Unified, Real-Time Object Detection//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016: 779-788.

[16]Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks//Advances in Neural Information Processing Systems. 2015, 28: 91-99.

[17]Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: 2004.10934, 2020.

[18]Zhang, T., Cui, B. W., & Cui, X. M. (2026). Photovoltaic panel defect detection algorithm based on improved YOLOv11n. Electronic Measurement Technology, 1–12. Retrieved from https://link.cnki.net/urlid/11.2175.TN.20260130.1931.012.

[19]Zhang H, Cisse M, Dauphin Y N, Lopez-Paz D. mixup: Beyond Empirical Risk Minimization// International Conference on Learning Representations (ICLR). 2018.

[20]Lin T Y, Goyal P, Girshick R, He K, Dollár P. Focal Loss for Dense Object Detection//Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2017: 2999-3007.

[21]Pan S J, Yang Q. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.

[22]Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression// Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 12993-13000.

[23]Khanam R, Asghar T, Hussain M. Comparative Performance Evaluation of YOLOv5, YOLOv8, and YOLOv11 for Solar Panel Defect Detection. Solar, 2025, 5(1): 6.

[24]Zhang Y, Sun P, Jiang Y, et al. ByteTrack: Multi-Object Tracking by Associating Every Detection Box//European Conference on Computer Vision. Cham: Springer, 2022: 1-21.

