SMART FAULT DIAGNOSIS IN SOLAR PANELS USING UPGRADED YOLO NETWORK

A PROJECT REPORT

Submitted by

AJITHA V	**(927621BCS007)**
DHARINI B	**(927621BCS023)**
KALAIARASI B	**(927621BCS048)**
KANJANAMALA R	**(927621BCS050)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

M. KUMARASAMY COLLEGE OF ENGINEERING, KARUR

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2025

SMART FAULT DIAGNOSIS IN SOLAR PANELS USING UPGRADED YOLO NETWORK

A PROJECT REPORT

Submitted by

AJITHA V	**(927621BCS007)**
DHARINI B	**(927621BCS023)**
KALAIARASI B	**(927621BCS048)**
KANJANAMALA R	**(927621BCS050)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

M. KUMARASAMY COLLEGE OF ENGINEERING, KARUR

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2025

M. KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

KARUR – 639 113

BONAFIDE CERTIFICATE

Certified that this project report **"SMART FAULT DIAGNOSIS IN SOLAR PANELS USING UPGRADED YOLO NETWORKS"** is the bonafide work of **"AJITHA V (927621BCS007), DHARINI B (927621BCS023), KALAIARASI B(927621BCS048), KANJANAMALA R (927621BCS050)"** who carried out the project work during the academic year 2024-2025 under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Signature	Signature
Mrs. A. SELVANAYAGI , M.E.,	**Dr. D. PRADEEP, M.E., Ph.D.,**
SUPERVISOR,	**HEAD OF THE DEPARTMENT,**
Department of Computer Science and Engineering,	Department of Computer Science and Engineering,
M.Kumarasamy College of Engineering,	M.Kumarasamy College of Engineering,
Thalavapalayam, Karur-639113.	Thalavapalayam, Karur-639113.

This project Report has been submitted for the Project Work - End Semester viva voce Examination held on ___________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We affirm that the Project report titled **“SMART FAULT DIAGNOSIS IN SOLAR PANELS USING UPGRADED YOLO NETWORKS”** being submitted in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

		Signature
AJITHA V	(927621BCS007)	__________
DHARINI B	(927621BCS023)	__________
KALAIARASI B	(927621BCS048)	__________
KANJANAMALA R	(927621BCS050)	__________

ACKNOWLEDGEMENT

We gratefully remember our beloved **Founder Chairman, (Late) Thiru. M. Kumarasamy**, whose vision and legacy laid the foundation for our education and inspired us to successfully complete this project.

We extend our sincere thanks to **Dr. K. Ramakrishnan, Chairman**, and **Mr. K. R. Charun Kumar, Joint Secretary**, for providing excellent infrastructure and continuous support throughout our academic journey.

We are privileged to extend our heartfelt thanks to our respected Principal, **Dr. B. S. Murugan, B.Tech., M.Tech., Ph.D.,** for providing us with a conducive environment and constant encouragement to pursue this project work.

We sincerely thank **Dr. D. Pradeep, B.E., M.E., Ph.D.,** Professor and **Head,** ***Department of Computer Science and Engineering***, for his continuous support, valuable guidance, and motivation throughout the course of this project.

Our special thanks and deep sense of appreciation go to our **Project Supervisor, Mrs. A. Selvanayagi , B.E., M.E.,** Assistant Professor, ***Department of Computer Science and Engineering,*** for her exceptional guidance, continuous supervision, constructive suggestions, and unwavering support, all of which have been instrumental in the successful execution of this project.

We would also like to acknowledge **Mr. C. NandhaKumar, B.E., M.E., Assistant Professor, our Class Advisor,** and **Dr. S. Sujanthi**, **B.E., M.E., Ph.D.,** the **Project Coordinator**, for their constant encouragement and coordination that contributed to the smooth progress and completion of our project work.

We gratefully thank all the **faculty members of the Department of Computer Science and Engineering** for their timely assistance, valuable insights, and constant support during various phases of the project.

Finally, we extend our profound gratitude to our **parents and friends** for their encouragement, moral support, and motivation, without which the successful completion of this project would not have been possible.

M.KUMARASAMY COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision of the Institution

To emerge as a leader among the top institutions in the field oftechnical education

Mission of the Institution

- Produce smart technocrats with empirical knowledge who cansurmount the global challenges
- Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
- Maintain mutually beneficial partnerships with our alumni,industry, and Professional associations

Vision of the Department

To achieve education and research excellence in ComputerScience and Engineering

Mission of the Department

- To excel in academic through effective teaching learning techniques
- To promote research in the area of computer science and engineering with the focus on innovation
- To transform students into technically competent professionals with societal and ethical responsibilities

Program Educational Objectives (PEOs)

- **PEO 1**: Graduates will have successful career in software industries and R&D divisions through continuous learning.
- **PEO 2**: Graduates will provide effective solutions for real world problems in the key domain of computer science and engineering and engage in lifelong learning.

- **PEO 3**: Graduates will excel in their profession by being ethically and socially responsible.

Program Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, suchas, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge andunderstanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1- Professional Skills: Ability to apply the knowledge of computing techniques to design and develop computerized solutions for the problems.

PSO2- Successful career: Ability to utilize the computing skills andethical values in creating a successful career.

ABSTRACT

The rising adoption of solar energy has highlighted the need for efficient maintenance of photovoltaic (PV) systems. Faults in solar panels, such as cracks, hotspots, or dirt accumulation, can significantly reduce energy output and system efficiency. Manual inspection methods are labor-intensive and prone to human error, necessitating an automated approach. This study introduces an AI-driven fault diagnosis system for solar panel units, leveraging an upgraded YOLO (You Only Look Once) model. The proposed system enhances real-time fault detection accuracy while maintaining high computational efficiency. The upgraded YOLO model is fine-tuned to identify and classify various solar panel faults using a dataset of annotated images. Advanced image pre-processing techniques and data augmentation are employed to improve model robustness. Additionally, the system integrates a post-detection analysis module that provides actionable insights for maintenance planning. This approach ensures early fault detection, reducing downtime and optimizing the performance of PV systems. The study demonstrates the effectiveness of the upgraded YOLO model in achieving high precision and recall rates, establishing its potential for scalable deployment in solar energy systems.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
The rising adoption of solar energy has highlighted the need for efficient maintenance of photovoltaic (PV) systems. Faults in solar panels, such as cracks, hotspots, or dirt accumulation, can significantly reduce energy output and system efficiency. Manual inspection methods are labour-intensive and prone to human error, necessitating an automated approach. This study introduces an AI-driven fault diagnosis system for solar panel units, leveraging an upgraded YOLO (You Only Look Once) model. The proposed system enhances real-time fault detection accuracy while maintaining high computational efficiency. The upgraded YOLO model is fine-tuned to identify and classify various solar panel faults using a dataset of annotated images. Advanced image pre-processing techniques and data augmentation are employed to improve model robustness. Additionally, the system integrates a post-detection analysis module that provides actionable insights for maintenance planning. This approach ensures early fault detection, reducing downtime and optimizing the performance of PV systems. The study demonstrates the effectiveness of the upgraded YOLO model in achieving high precision and recall rates, establishing its potential for scalable deployment in solar energy systems.	PO1(3), PO2(3), PO3(3), PO4(3), PO5(3), PO6(2), PO7(3), PO8(3), PO9(3), PO10(3), PO11(2), PO12(2)	PSO1(3), PSO2(2)

NOTE: 1-LOW, 2-MEDIUM, 3-HIGH

SUPERVISOR **HEAD OF THE DEPARTMENT**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION		MEANING
AIoT	-	Artificial Intelligence of Things
CNN	-	Convolutional Neural Network
DBN	-	Deep Belief Network
DL	-	Deep Learning
FFT	-	Fast Fourier Transform
GAN	-	Generative Adversarial Networks
GPR	-	Gaussian Process Regression
LSTM	-	Long Short-Term Memory
mAP	-	mean Average Precision
NLP	-	Natural Language Processing
NMS	-	Non-Maximum Suppression
PCA	-	Principal Component Analysis
PID	-	Potential Induced Degradation
PV	-	PhotoVoltaic
RBM	-	Restricted Boltzmann Machines
RMSE	-	Root Mean Square Error
RNN	-	Recurrent Neural Network
SMS	-	Short Message Service
ST	-	Stockwell Transform
SVM	-	Support Vector Machines
WT	-	Wavelet Transform
YOLO	-	You Only Look Once

CHAPTER 1
INTRODUCTION

1.1 DESCRIPTION

The increasing global demand for sustainable energy has led to the widespread adoption of solar photovoltaic (PV) systems. However, the efficiency and longevity of these systems are often compromised by faults such as cracks, hotspots, shading, and dirt accumulation on the solar panels. These faults, if undetected, can lead to reduced power output and increased maintenance costs. Traditional fault detection methods rely on manual inspections and basic imaging techniques, which are time-consuming, labour-intensive, and prone to human error. To address these limitations, there is a growing need for intelligent and automated fault diagnosis solutions. This project proposes an AI-driven fault diagnosis system using an upgraded YOLO (You Only Look Once) object detection model. The system is designed to identify and classify various types of faults in solar panels in real time with high accuracy and efficiency. Enhancements to the YOLO model include the optimization of anchor box dimensions, the use of a custom loss function to prioritize subtle fault detection, and fine-tuning of the network architecture to better capture intricate fault patterns. Additionally, advanced image preprocessing techniques and data augmentation are used to improve model robustness, while transfer learning with pre-trained YOLO weights accelerates the training process and ensures high precision even under diverse environmental conditions. The system comprises multiple modules, including image preprocessing, feature extraction, grid division with bounding box prediction, class prediction, non-maximum suppression, and final output generation. A dedicated training module is also integrated for object detection tasks. By automating the fault detection process, the system reduces downtime, enhances energy output. The scalable and real-time capabilities of the proposed solution make it ideal for deployment in large solar farms, contributing significantly to the reliability and performance optimization of renewable energy systems.

1.2 DEEP LEARNING

Deep Learning is a subset of machine learning that involves training artificial neural networks with multiple layers to recognize patterns in data. Deep learning algorithms can be used for a wide range of tasks such as image and speech recognition, natural language processing, and even playing games like Go and Chess. The main advantage of deep learning over traditional machine learning approaches is its ability to automatically learn features from raw data without the need for manual feature engineering. This is accomplished by stacking multiple layers of neurons, each of which performs a nonlinear transformation of the input data. The output of one layer serves as the input for the next layer, allowing the network to gradually learn increasingly complex representations of the input data. Popular deep learning algorithms include Convolutional Neural Networks (CNNs) for image and video processing, Recurrent Neural Networks (RNNs) [9],[11] for sequential data processing such as natural language processing, and Generative Adversarial Networks (GANs) for generating realistic images and videos. Training deep learning models requires large amounts of labeled data and significant computational resources. However, recent advancements in hardware and software have made it easier to train deep learning models on a wide range of applications.

Deep learning algorithms are based on artificial neural networks, which are inspired by the structure and function of the human brain. The networks consist of layers of interconnected nodes, or neurons, that process information in a hierarchical manner. The input data is fed into the first layer of the network, which extracts basic features. The output of this layer is then passed to the next layer, which extracts more complex features based on the previous layer's output, and so on. The process of training a deep learning model involves adjusting the weights and biases of the network's neurons to minimize the difference between the predicted output and the actual output. This is done by using a loss function that quantifies the difference between the predicted and actual output, and an optimization algorithm that updates the network's weights and biases to minimize this loss function. The most commonly used optimization algorithm is called stochastic gradient descent.

One of the key advantages of deep learning is its ability to handle unstructured data such as images, video, and text. Convolutional Neural Networks (CNNs) are particularly effective at processing images and video, while Recurrent Neural Networks (RNNs) are better suited for sequential data processing such as natural language processing. Deep learning has had a significant impact on a wide range of industries, including healthcare, finance, and transportation. For example, deep learning algorithms are used in medical imaging to help diagnose diseases such as cancer, in finance to detect fraudulent transactions, and in transportation to improve self-driving cars' performance. However, deep learning is not without its challenges. One of the biggest challenges is the need for large amounts of labeled data to train the models effectively[12]. This can be particularly challenging for applications where the data is scarce or expensive to collect. Additionally, deep learning models are often black boxes, meaning it can be challenging to interpret how the model arrives at its predictions. This can be problematic for applications where interpretability is important, such as in healthcare or finance.

1.3 APPLICATIONS OF DEEP LEARNING

Image and Object Recognition:

Deep learning, especially using CNNs, is widely used in recognizing objects, faces, and scenes in images. It powers applications like facial recognition, autonomous vehicles, and medical imaging diagnostics.

Natural Language Processing (NLP):

Deep learning enhances NLP tasks such as machine translation, sentiment analysis, speech recognition, and text summarization. Models like BERT and GPT are capable of understanding and generating human-like language.

Autonomous Vehicles:

Self-driving cars use deep learning to interpret sensor data, detect lanes, recognize traffic signs, and avoid obstacles. It enables real-time decision-making for safe autonomous navigation.

Healthcare and Medical Diagnosis:

Deep learning models assist in detecting diseases from medical images like X-rays, MRIs, and CT scans. They help in early diagnosis of conditions such as cancer, retinal diseases, and brain abnormalities.

Speech Recognition and Voice Assistants:

Applications like Google Assistant, Siri, and Alexa use deep learning to convert spoken language into text and respond accurately. It also powers real-time language translation and transcription services.

1.4 ALGORITHMS IN DEEP LEARNING

There are several types of deep learning algorithms, each of which is designed to solve different types of problems. Some of the most popular deep learning algorithms include:

Convolutional Neural Networks (CNNs): These are commonly used for image and video processing. They use a technique called convolution to extract features from the input image or video.

Recurrent Neural Networks (RNNs): These are used for sequential data processing, such as natural language processing. They can capture the context and relationship between different elements in a sequence.

Generative Adversarial Networks (GANs): These are used for generating new data that is similar to the input data. They consist of two networks: a generator network that generates new data and a discriminator network that evaluates whether the generated data is similar to the real data.

Autoencoders: These are used for unsupervised learning and feature extraction. They consist of an encoder network that compresses the input data into a lower-dimensional representation, and a decoder network that reconstructs the original input from the compressed representation.

Deep Belief Networks (DBNs): These are used for unsupervised learning and feature extraction. They consist of multiple layers of restricted Boltzmann machines (RBMs) that can learn hierarchical representations of the input data.

Long Short-Term Memory (LSTM) Networks: These are a type of RNN that is designed to handle long-term dependencies in sequential data. They use memory cells and gates to selectively remember or forget information from previous time steps.

1.5 CHALLENGES IN DEEP LEARNING

High Computational Requirements:

Deep learning models require significant processing power, especially during training. This often necessitates the use of GPUs or TPUs, which can be expensive and energy-intensive, limiting accessibility for smaller organizations.

Need for Large Datasets:

Deep learning models perform best when trained on vast amounts of labeled data. Obtaining and annotating such datasets can be time-consuming, costly, and impractical for certain applications, especially in domains like healthcare.

Overfitting and Generalization:

Deep neural networks can easily overfit to training data, especially when the dataset is small or unbalanced. This reduces their ability to generalize well to new, unseen data, impacting real-world performance.

Lack of Interpretability:

Deep learning models often function as "black boxes," making it difficult to understand how decisions are made. This lack of transparency can be a major concern in critical applications like healthcare and finance.

Bias in Data and Models:

If the training data contains biases, the deep learning model is likely to replicate and even amplify them. This can lead to unfair or unethical outcomes, especially in sensitive areas like hiring or law enforcement.

1.6 FUTURE SCOPE OF DEEP LEARNING

- **Advanced Healthcare Solutions:** Deep learning will revolutionize healthcare with more accurate disease detection, personalized medicine, and robotic surgeries powered by real-time analysis.
- **Fully Autonomous Vehicles:** The evolution of deep learning will enable safer and smarter self-driving cars, enhancing transportation systems and reducing accidents.
- **Human-Level Language Understanding:** Language models will continue to evolve, making machines capable of truly understanding and generating human-like responses in real-time.
- **AI in Education and Personalized Learning:** Deep learning will power intelligent tutoring systems that adapt to each student's needs, improving educational outcomes.
- **Improved Cybersecurity Systems:** Future cybersecurity tools will use deep learning to detect and respond to threats in real-time, enhancing data protection.
- **Creative Content Generation:** Deep learning will enable the creation of realistic art, music, videos, and even stories, reshaping the entertainment industry.
- **Smarter Industrial Automation:** Manufacturing and robotics will be enhanced with more intelligent and self-learning systems, increasing efficiency and reducing human error.
- **Energy Optimization and Smart Cities:** Deep learning will contribute to smart energy grids, waste management, and intelligent city planning for sustainable urban living.

CHAPTER 2

LITERATURE SURVEY

2.1 TITLE: A REVIEW OF AUTOMATED SOLAR PHOTOVOLTAIC DEFECT DETECTION SYSTEMS: APPROACHES, CHALLENGES, AND FUTURE ORIENTATIONS

AUTHOR: ULA HIJJAWI

DESCRIPTION:

The comprehensive review explores the evolving landscape of automated defect detection in solar photovoltaic (PV) systems, emphasizing the integration of advanced computational technologies such as computer vision, machine learning, and deep learning. The paper categorizes various approaches into image-based methods, sensor-based diagnostics, and hybrid systems that combine multiple data sources. [4] Thermal imaging, infrared sensing, and electroluminescence are frequently used modalities, enabling early identification of defects such as cracks, hot spots, delamination, and soiling. The authors discuss deep learning models, particularly convolutional neural networks (CNNs), for classifying and segmenting defect areas. A major highlight is the emphasis on dataset availability and quality, which directly impacts model training and evaluation. The review outlines the performance of models across benchmark datasets and real-world applications, noting variations due to weather conditions, shading, and aging of PV modules. The paper also investigates real-time detection systems, stressing the importance of edge computing and embedded AI for efficient fault analysis. Among the discussed challenges are limited dataset generalizability, difficulty in detecting minor faults, and the need for standardized evaluation frameworks. The authors present a roadmap for future advancements, proposing fusion-based diagnostics, explainable AI, and cross-domain learning as key directions.

2.2 TITLE: FAULT DETECTION IN SOLAR ENERGY SYSTEMS: A DEEP LEARNING APPROACH

AUTHOR: ZEYNEP BALA DURANAY

DESCRIPTION:

The research outlines a deep learning framework for detecting faults in solar energy systems with high accuracy and real-time performance. The proposed method leverages convolutional neural networks (CNNs) trained on a large dataset of thermal and RGB images of photovoltaic panels exhibiting a variety of fault types [1]. The author emphasizes the importance of early fault detection to maintain optimal energy output and reduce maintenance costs. Preprocessing techniques, including histogram equalization and image resizing, are applied to improve feature visibility. The CNN architecture consists of multiple convolutional and pooling layers followed by dense layers, achieving efficient feature extraction and classification. The study evaluates the model using multiple performance metrics such as accuracy, precision, recall, and F1-score, all indicating strong model reliability. A key component of this approach is data augmentation, which helps in enhancing model generalization under various lighting and environmental conditions. The paper further includes a comparison with traditional machine learning techniques like SVM and decision trees, showing significant performance improvement with deep learning. Additionally, the model's performance is tested on unseen images to validate its robustness. The research discusses challenges such as false positives due to shading or bird droppings, proposing adaptive learning models to reduce misclassifications. Overall, this work demonstrates the feasibility and benefits of applying AI-driven solutions in real-time PV system monitoring and lays the groundwork for deployment in industrial-scale applications.

2.3 TITLE: A SURVEY OF PHOTOVOLTAIC PANEL OVERLAY AND FAULT DETECTION METHODS

AUTHOR: CHENG YANG

DESCRIPTION:

The paper offers a broad survey of fault detection and overlay identification techniques in photovoltaic (PV) panels, emphasizing the evolution of detection methodologies from manual inspection to intelligent automation. The review classifies methods into three main groups: image-based inspection, sensor-based fault diagnostics, and intelligent hybrid models. Each approach is analyzed based on its ability to detect common faults such as hotspots, microcracks, PID (potential induced degradation), and mismatch losses. The authors highlight imaging technologies including infrared thermography, electroluminescence, and drone-based aerial imaging for large-scale solar farms. Overlay issues caused by environmental obstructions such as leaves, bird droppings, or dust are also addressed, showing how they can lead to misleading fault interpretation. The review discusses machine learning and deep learning applications in processing imaging data, with CNNs and decision-tree-based classifiers being widely adopted. Performance metrics like detection accuracy, precision, and recall are tabulated for different techniques. The paper evaluates the advantages and limitations of real-time fault detection using IoT and cloud-based platforms. Moreover, the authors identify the growing role of explainable AI and transfer learning in improving model interpretability and adaptability. The study concludes by discussing future research challenges, including the need for diverse, labeled datasets, system scalability, and fault prediction models. This survey is a valuable resource for researchers developing comprehensive, automated PV inspection systems that are accurate, cost-effective, and scalable across different environmental settings [15].

2.4 TITLE: SPF-NET: SOLAR PANEL FAULT DETECTION USING U-NET BASED DEEP LEARNING IMAGE CLASSIFICATION

AUTHOR: RIFAT AL MAMUN RUDRO

DESCRIPTION:

The study introduces SPF-Net, a U-Net-based deep learning architecture designed for precise solar panel fault detection through image classification. The authors focus on leveraging the strengths of U-Net, originally developed for biomedical image segmentation, and adapt it to detect faults in solar panels, including cracks, hotspots, and dirt accumulation. The model employs an encoder-decoder structure that captures spatial hierarchies and enables pixel-wise fault localization. The dataset comprises high-resolution infrared and RGB images of PV panels under various operational and environmental conditions. Preprocessing techniques such as contrast enhancement and background normalization are applied to improve the visibility of subtle faults. [12] SPF-Net demonstrates high segmentation accuracy, outperforming traditional CNN-based classifiers and edge detection methods. The authors validate their model using precision, recall, F1-score, and Intersection over Union (IoU), all indicating strong detection capabilities. Data augmentation techniques, including flipping and noise injection, are used to enhance model generalization. The paper also compares SPF-Net with other deep learning architectures like DeepLabV3 and SegNet, showing superior performance in both training time and inference accuracy. A noteworthy feature is the model's adaptability to different solar farm environments, facilitated by fine-tuning with domain-specific data. The system is proposed as part of a larger automated maintenance framework, integrated with drone imaging and cloud analytics.

2.5 TITLE: IMPROVED FAULT DETECTION AND CLASSIFICATION IN PV ARRAYS USING STOCKWELL TRANSFORM AND DATA MINING TECHNIQUES

AUTHOR: CHIDURALA SAIPRAKASH

DESCRIPTION:

The paper presents an improved methodology for fault detection and classification in photovoltaic (PV) arrays using the Stockwell Transform (ST) in conjunction with data mining techniques. The approach targeting the detection of partial shading, open-circuit, and short-circuit faults by analyzing voltage and current signatures of PV modules. The Stockwell Transform [13] is applied to extract time-frequency domain features that reveal transient behaviors in PV signals. These features are then fed into machine learning classifiers such as decision trees, random forests, and support vector machines (SVM) to distinguish between different fault types. The authors conduct extensive simulation and experimental validation on real PV systems under diverse weather conditions. Their findings reveal that the proposed method achieves higher classification accuracy compared to traditional signal processing techniques like Fast Fourier Transform (FFT) and Wavelet Transform (WT). Additionally, the model shows robustness in detecting compound faults and subtle signal anomalies. Feature selection techniques such as Principal Component Analysis (PCA) and Information Gain are used to reduce dimensionality and enhance classification speed. The system is capable of real-time fault monitoring, and its integration with microcontroller-based embedded platforms is demonstrated. The paper also explores the scalability of the approach for large-scale solar installations. The authors conclude that this hybrid method combining advanced signal analysis with intelligent classifiers can significantly improve operational reliability and energy yield in PV arrays.

2.6 TITLE: ARTIFICIAL-INTELLIGENCE-BASED DETECTION OF DEFECTS AND FAULTS IN PHOTOVOLTAIC SYSTEMS: A SURVEY

AUTHOR: ALI THAKFAN AND YASSER BIN SALAMAH

DESCRIPTION:

The survey delves into the application of artificial intelligence (AI) methods for defect and fault detection in photovoltaic (PV) systems, providing a broad overview of techniques that enhance the efficiency and sustainability of solar power generation. The paper categorizes AI-based methods into supervised learning, unsupervised learning, reinforcement learning, and hybrid models, each analyzed for their contributions to anomaly detection in PV arrays. Emphasis is placed on the role of deep learning models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in analyzing thermal, electroluminescence, and visual data for precise fault classification. The survey covers various data acquisition techniques such as UAV imaging, sensor-based monitoring, and SCADA systems integration [14]. Notably, the review highlights the significance of labeled datasets and discusses emerging approaches in transfer learning and federated learning for fault detection in distributed solar networks. Each AI technique is reviewed based on parameters like detection accuracy, response time, and adaptability to environmental noise. Furthermore, the study discusses practical deployment aspects, such as edge computing for real-time monitoring and smart IoT architectures. Special attention is given to solar microgrid fault diagnostics, predictive maintenance, and adaptive learning models that self-update under changing environmental conditions. The paper also presents several benchmark studies and compares them based on application domain, algorithm complexity, and scalability.

2.7 TITLE: ENHANCING PHOTOVOLTAIC SYSTEMS USING GAUSSIAN PROCESS REGRESSION FOR PARAMETER IDENTIFICATION AND FAULT DETECTION

AUTHOR: AQDAS JAVAID

DESCRIPTION:

The paper introduces a novel framework for photovoltaic (PV) system enhancement using Gaussian Process Regression (GPR) to perform parameter identification and fault detection. GPR, a probabilistic and non-parametric regression technique [5], is applied to model the nonlinear and time-varying behavior of PV arrays under different operating conditions. The method focuses on detecting both abrupt and subtle faults by comparing actual system performance with predicted outputs derived from GPR models. The study employs real-time environmental and electrical data, including irradiance, temperature, voltage, and current, to train and validate the regression model. Through Bayesian inference, the GPR model provides uncertainty estimates for predictions, allowing for robust fault detection even in noisy environments. The paper discusses the integration of this technique with remote monitoring systems and SCADA platforms to enable predictive maintenance and early intervention. Furthermore, the authors compare GPR with other machine learning algorithms such as support vector regression and artificial neural networks, showing superior performance in terms of root mean square error (RMSE) and fault classification accuracy. The study highlights the capability of GPR to model nonlinearity in PV performance, particularly under partial shading and aging effects. Additionally, adaptive learning strategies are incorporated to dynamically update model parameters as new data becomes available.

2.8 TITLE: A NOVEL DEEP STACK-BASED ENSEMBLE LEARNING APPROACH FOR FAULT DETECTION AND CLASSIFICATION IN PHOTOVOLTAIC ARRAYS

AUTHOR: EHTISHAM LODHI

DESCRIPTION:

The research proposes an innovative deep stack-based ensemble learning method for detecting and classifying faults in photovoltaic (PV) arrays, aiming to improve accuracy, robustness, and adaptability. The architecture integrates multiple deep learning models including CNNs, LSTMs, and fully connected networks into a stacked ensemble, combining the strengths of each to enhance feature learning and generalization. The system is trained on diverse datasets comprising infrared images and operational sensor data to identify common faults such as hotspot formation, cracked cells, delamination, and open-circuit anomalies. A key contribution of the study is its hierarchical classification system that categorizes faults based on severity and type, enabling targeted maintenance strategies. The ensemble model uses a meta-learner to aggregate predictions from base models, optimizing the final classification output. The paper evaluates the model using cross-validation, reporting high scores in precision, recall, and F1-score across various test scenarios. Advanced data augmentation and normalization techniques are applied to address class imbalance and improve the resilience of the model against sensor noise and environmental variability. Additionally, the authors implement a real-time monitoring prototype demonstrating the practical application of their method in both residential and industrial PV setups. Integration with IoT devices [8] and cloud platforms enables remote analytics and fault alerts.

2.9 TITLE: A HYBRID MACHINE LEARNING APPROACH: ANALYZING ENERGY POTENTIAL AND DESIGNING SOLAR FAULT DETECTION FOR AN AIOT-BASED SOLAR–HYDROGEN SYSTEM IN A UNIVERSITY SETTING

AUTHOR: SALAKI REYNALDO JOSHUA

DESCRIPTION:

The paper presents a hybrid machine learning approach aimed at both evaluating solar energy potential and detecting faults in photovoltaic systems, tailored for integration within an AIoT (Artificial Intelligence of Things) based solar–hydrogen ecosystem. Implemented in a university campus environment, the system leverages IoT devices for real-time environmental monitoring and data acquisition. The proposed hybrid model combines decision trees, random forest classifiers, and neural networks to enhance the detection accuracy of faults in PV panels, such as soiling, degradation, and shading [6]. The model also forecasts solar power generation potential using time-series meteorological data, which aids in energy management and hydrogen fuel cell operations. Emphasis is placed on designing a smart fault detection pipeline that includes preprocessing, feature extraction, and a multi-class fault classifier. A dedicated AIoT architecture supports wireless communication between the sensing layer, edge computing units, and cloud storage. Experimental validation demonstrates the framework's high classification accuracy and low latency in real-time deployment. The authors highlight the system's potential to reduce maintenance costs, improve power reliability, and optimize energy distribution within hybrid energy networks. Data fusion techniques are employed to combine electrical and environmental datasets, enabling a comprehensive analysis of system health.

2.10 TITLE: PHOTOVOLTAIC SYSTEM FAULT DETECTION TECHNIQUES: A REVIEW

AUTHOR: GHADA M. EL-BANBY

DESCRIPTION:

The review paper provides a thorough examination of fault detection techniques applied to photovoltaic (PV) systems, with a focus on both traditional and intelligent methods. The authors categorize detection strategies into three main classes: hardware-based, model-based, and data-driven techniques. Hardware-based methods include sensors and thermal cameras used in manual and automated inspections. Model-based approaches rely on mathematical modeling of PV behavior to identify deviations that indicate faults. The most advanced category data-driven techniques leverages artificial intelligence and machine learning, with CNNs, SVMs, and hybrid algorithms leading current research. The review emphasizes the evolution toward real-time monitoring systems that incorporate IoT, cloud computing, and edge devices for scalable deployment. Comparative analysis is conducted across various methodologies, highlighting performance metrics such as detection rate, false alarm rate, and computational cost. [3] Special attention is given to hybrid approaches that combine image analysis with electrical signal processing for improved fault localization. The paper outlines several benchmark datasets and discusses the role of data augmentation and synthetic data generation in overcoming the scarcity of labeled data. The authors also explore the integration of solar fault detection with smart grid applications and predictive maintenance platforms. This review serves as a key reference for system developers and researchers seeking to understand the full landscape of PV fault detection technologies, offering guidance on selecting and combining the best-suited methods for specific use cases.

Table 2.1 Findings in Literature Survey

TITLE	AUTHOR & YEAR	TECHNIQUE / MODEL	MERITS	DEMERITS
Automated PV Defect Detection: A Review	Hijjawi et al. (2023)	Survey of PV defect detection	Highlights key challenges and future trends	No implementation or experimental evaluation
Fault Detection in Solar via Deep Learning	Duranay (2023)	CNN-based Deep Learning	Good accuracy, deep model effectiveness	Needs large annotated datasets
Survey of PV Overlay and Fault Methods	Yang et al. (2024)	Comprehensive Survey	Broad method coverage including thermal/visual	Lacks algorithmic contribution
SPF-Net: U-Net Based Detection	Rudro et al. (2024)	U-Net CNN	High-resolution localization, good for segmentation	May require high computational resources
Stockwell Transform + Data Mining	Saiprakash et al. (2024)	Signal Processing + ML	Enhanced detection accuracy from time-frequency features	Not suitable for image-based systems
AI-Based Detection in PV Systems	Thakfan & Salamah (2024)	Review of AI Techniques	Extensive AI model review	No new detection framework proposed

Gaussian Process Regression	Javaid et al. (2024)	GPR + Fault Detection	Parameter estimation with uncertainty quantification	Slower for large-scale systems
Deep Stack-Based Ensemble Learning	Lodhi et al. (2023)	Ensemble DL Approach	High accuracy via stacking models	High training complexity
Hybrid ML + AIoT Design	Joshua et al. (2024)	AIoT + ML Hybrid	Real-time integration with energy estimation	Deployment challenges in real settings
PV Fault Detection Techniques: Review	El-Banby et al. (2023)	Comparative Review	Discusses neural, signal-based, hybrid methods	Theoretical with no new model contribution

CHAPTER 3
EXISTING SYSTEM

Existing fault detection systems in photovoltaic (PV) panels typically rely on manual inspections, conventional imaging techniques, and basic automated algorithms. These methods form the foundational framework for identifying panel defects but are limited in accuracy, efficiency, and scalability. Manual inspections are time-consuming and prone to human error, while traditional imaging techniques often lack the precision required for detecting small or subtle defects. Basic machine learning algorithms, such as standard clustering or classification methods, struggle with complex fault patterns and real-time analysis, making them inadequate for modern large-scale solar energy systems. To address some of these challenges, enhanced machine learning models like K-means clustering, HRNet, and MobileNet have been explored. The improved K-means clustering technique uses advanced distance metrics and preprocessing steps to better segment images into regions representing faults, clean cells, and background. HRNet, a high-resolution deep learning architecture, is effective in analyzing electroluminescence (EL) images for internal defects like micro-cracks and soldering issues but is resource-intensive. MobileNet, optimized for embedded systems, works well with infrared images to detect thermal anomalies such as hotspots but depends heavily on access to comprehensive labeled datasets. Despite these advancements, each method has limitations that restrict their effectiveness in real-time, scalable PV fault detection [7], reinforcing the need for a more robust and automated AI-driven approach.

DISADVANTAGES

- K-means clustering may struggle to identify intricate or irregular defect patterns due to its reliance on simple distance metrics.

- HRNet is resource-intensive, requiring substantial computational power and memory for training and inference.
- Requires access to a comprehensive dataset of labeled IR images, which might not be readily available.

3.1 EXISTING SYSTEM BLOCK DIAGRAM

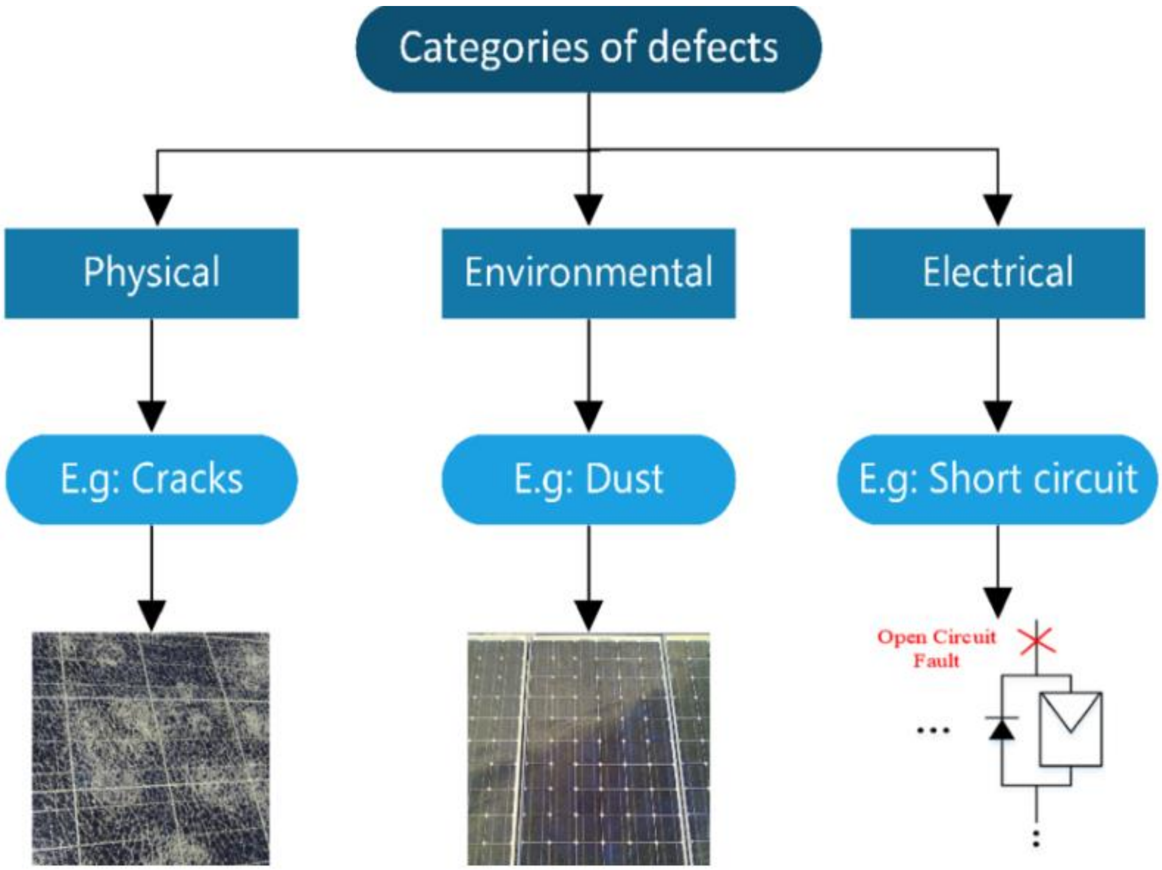


Fig 3.1. Block Diagram of the Existing System

CHAPTER 4
PROBLEMS IDENTIFIED

Solar panels, despite their increasing adoption as a sustainable energy source, are prone to a range of faults such as micro-cracks, hotspots, dirt accumulation, shading, and material degradation. These issues often go undetected for long periods, leading to reduced energy efficiency, costly maintenance, and in severe cases, complete system failure. Manual inspection methods used for fault detection are labor-intensive, time-consuming, and susceptible to human error, making them inefficient for large-scale solar installations. Moreover, traditional imaging and diagnostic techniques [10] often lack the precision needed to detect subtle or early-stage defects.

Existing automated systems using classical machine learning or basic deep learning architectures have shown some improvements, but still face significant limitations. Techniques like K-means clustering struggle with irregular fault patterns [2], while models such as HRNet and MobileNet require substantial computational resources or rely on specialized imaging datasets, which are not always readily available. These shortcomings highlight the urgent need for an intelligent, efficient, and scalable solution that can perform real-time and highly accurate fault diagnosis in solar PV systems, regardless of environmental conditions or complexity defects.

CHAPTER 5

PROPOSED SYSTEM

The proposed system introduces an enhanced version of the YOLO (You Only Look Once) object detection algorithm, tailored specifically for fault diagnosis in photovoltaic (PV) panels. This system aims to address the limitations of existing methods by providing real-time, high-accuracy detection of various panel defects such as micro-cracks, hotspots, dirt accumulation, shading inconsistencies, and cell discoloration. By optimizing YOLO's anchor box dimensions and network architecture, the model becomes more sensitive to small and irregular defect patterns. A custom loss function is also implemented to prioritize accurate localization and classification of subtle faults, improving detection precision and recall. Transfer learning with pre-trained YOLO weights is used to accelerate the training process and enhance generalization capabilities, enabling the model to perform effectively across a wide range of environmental conditions and panel types. Additionally, the system is designed with modular components including image preprocessing, feature extraction, grid division and bounding box prediction, class prediction, non-maximum suppression, and final output generation. These modules work together seamlessly to process input images, extract relevant features, and accurately identify and classify defects in real-time. The training module further supports fine-tuning of the model using annotated datasets of solar panel faults, ensuring adaptability and continuous improvement. The result is a scalable and robust AI-driven solution that not only reduces downtime through early detection but also provides actionable insights for predictive maintenance planning, significantly improving the performance and lifespan of PV systems.

ADVANTAGES

- Real-Time Detection
- High Accuracy for Small Defects

- Robust Performance
- Scalability
- Integration with Existing Systems

ARCHITECTURE DIAGRAM

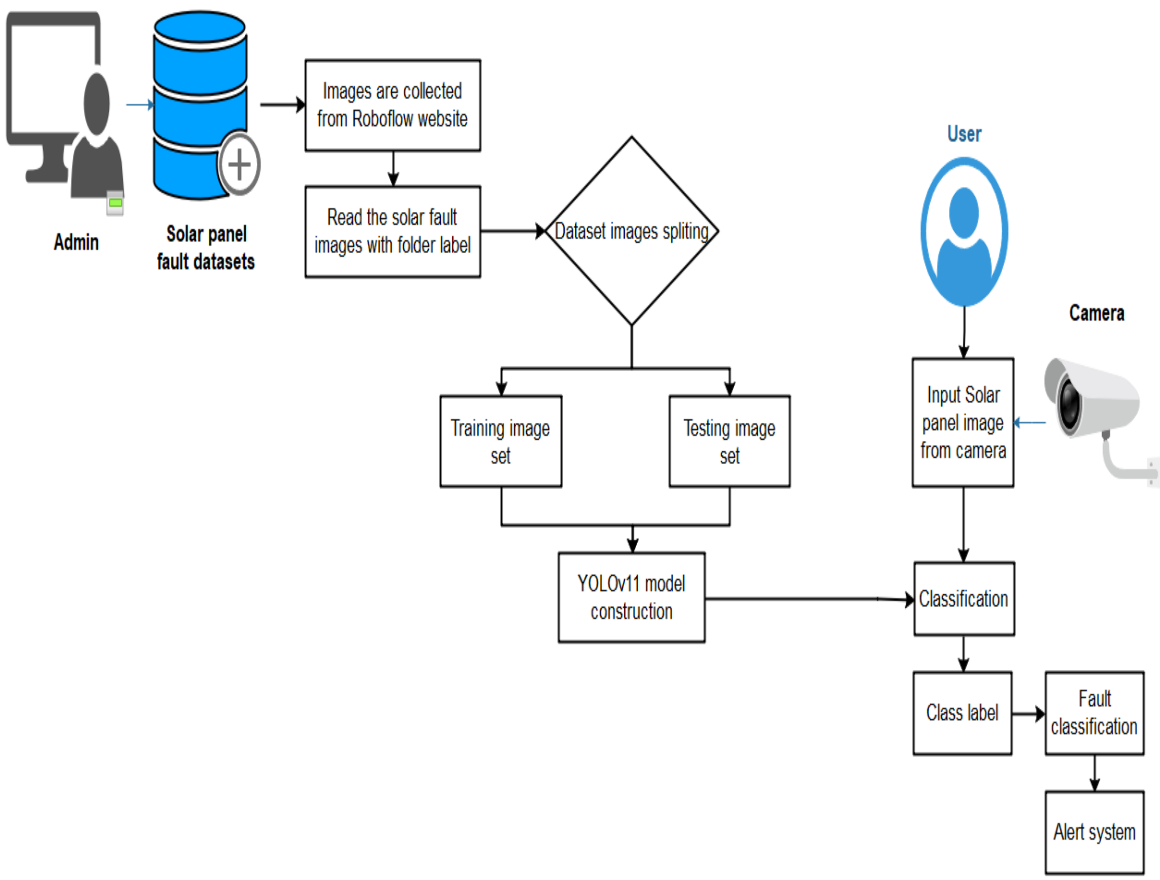


Fig 5.1 Architecture of the Proposed Work

CHAPTER 6

SYSTEM REQUIREMENTS

6.1 HARDWARE REQUIREMENTS

- Processor : Intel processor 2.6.0 GHZ
- RAM : 2GB
- Hard disk : 160 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

6.2 SOFTWARE REQUIREMENTS

- Server Side : Python 3.7.4(64-bit) or (32-bit)
- Client Side : HTML, CSS, Bootstrap
- IDE : Flask 1.1.1
- Back end : MySQL 5.
- Server : WampServer 2i
- OS : Windows 10 64 –bit

CHAPTER 7
SYSTEM IMPLEMENTATION

7.1 LIST OF MODULES

- Image Preprocessing
- Feature Extraction
- Grid Division and Bounding Box Prediction Module
- Class Prediction Module
- Non – Maximum Suppression
- Final Output Generation Module
- Training Module (For Object Detection)

7.2 MODULES DESCRIPTION

7.2.1 IMAGE PREPROCESSING

Image preprocessing is the initial and crucial step that prepares raw solar panel images for analysis. This module enhances image quality by removing noise, adjusting contrast, and resizing input images to match the YOLO model’s input dimensions. Techniques such as histogram equalization, normalization, and sharpening are used to improve feature visibility and consistency. This ensures that minor faults like micro-cracks and dirt patches become more distinguishable. Preprocessing also standardizes image formats, which is essential for batch training. In addition, data augmentation techniques such as rotation, flipping, zooming, and brightness adjustments are applied to diversify the dataset. These augmentations simulate real-world variations, increasing the model’s robustness. This step plays a critical role in improving the accuracy and generalization of the fault detection system.

7.2.2 FEATURE EXTRACTION

The feature extraction module is responsible for identifying and capturing essential visual patterns within the pre-processed images. Using convolutional layers in the YOLO model, this module scans images for unique textures, edges, and color gradients that signify faults like hotspots or delamination. The extracted features are passed through various filters that help in distinguishing faulty regions from the background and normal cells. This process ensures that high-level and low-level features are both considered, improving the depth of fault identification. Advanced layers also extract geometric patterns that help detect irregular shapes. The quality of these features directly affects detection accuracy. The extracted data is then forwarded to subsequent layers for classification and localization. Efficient and relevant feature extraction is vital for real-time performance.

7.2.3 GRID DIVISION AND BOUNDING BOX PREDICTION MODULE

In this module, each input image is divided into a grid, where each grid cell is responsible for predicting potential object locations. This process enables the YOLO algorithm to detect multiple faults within a single frame. The model predicts bounding boxes for objects within each grid cell, along with confidence scores that reflect the likelihood of a fault existing in that region. These bounding boxes include coordinates, height, width, and objectness score. The division into grids allows parallel fault identification across various panel sections. This is particularly useful when dealing with densely populated or large-scale solar farms. The module is fine-tuned to detect small and irregularly shaped faults. Accurate bounding box prediction significantly improves the model's localization performance. It also reduces false positives and enhances detection reliability.

7.2.4 CLASS PREDICTION MODULE

Once bounding boxes are predicted, the class prediction module determines the type of fault within each detected region. This module assigns a probability score to each class, such as micro-cracks, hotspots, shading, or dirt accumulation. Using a softmax activation function, it identifies the most probable fault class based on learned patterns from the training dataset. The YOLO model's classification layers are specifically fine-tuned to handle multiple fault types. This helps differentiate between similar-looking defects, like shading vs. soiling. Accurate classification is essential for enabling appropriate maintenance responses. The class prediction is performed simultaneously with bounding box generation. This module contributes directly to the output generation stage. It ensures the system provides not only fault locations but also meaningful categorization.

7.2.5 NON-MAXIMUM SUPPRESSION

Non-Maximum Suppression (NMS) is a crucial post-processing step that eliminates redundant or overlapping bounding boxes. In many cases, the model may detect the same fault multiple times with varying confidence levels. NMS ensures only the most accurate prediction is retained by comparing the confidence scores and Intersection over Union (IoU) of each box. It suppresses lower confidence boxes that overlap significantly with higher-scoring ones. This improves result clarity and prevents over-counting of faults. The algorithm ensures a clean and precise output with minimal duplication. NMS is especially important for densely faulted areas. It contributes significantly to maintaining the system's high precision and recall. By streamlining the output, it enhances both visual clarity and interpretability.

7.2.6 FINAL OUTPUT GENERATION MODULE

The final output generation module compiles the results from the previous modules into a user-friendly format. This includes annotated images with bounding boxes and fault labels, as well as a summary report of detected issues. The system presents this output for real-time monitoring and maintenance planning. It also stores the results in a structured format for future reference and analysis. Visualization tools may be integrated for highlighting critical fault zones. This output is crucial for solar technicians and operators to prioritize maintenance tasks efficiently. By offering interpretable results, the module bridges the gap between AI predictions and actionable decisions. It can also interface with IoT dashboards or maintenance systems. This ensures end-to-end fault diagnosis and management.

7.2.7 TRAINING MODULE (FOR OBJECT DETECTION)

The training module is responsible for training and fine-tuning the YOLO model using annotated datasets of solar panel faults. It involves feeding the system a large number of labeled images to learn how to detect and classify different fault types. The module leverages transfer learning from pre-trained YOLO weights to reduce training time and improve performance on limited datasets. Key parameters such as learning rate, batch size, and epochs are adjusted for optimal training. The training process includes validation and testing phases to evaluate model accuracy and generalization. Performance metrics such as loss function, precision, recall, and mean Average Precision (mAP) are continuously monitored. The module supports retraining to adapt to new fault types or environmental changes. It plays a vital role in maintaining model relevance and accuracy over time.

CHAPTER 8
SYSTEM TESTING

Testing can never completely identify all the defects within software. Instead, it furnishes a criticism or comparison that compares the state and behaviour of the product against oracles principles or mechanisms by which someone might recognize a problem. These oracles may include (but are not limited to) specifications, contracts, comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria. A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed. Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this assessment

8.1 UNIT TESTING

Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

8.2 INTEGRATION TESTING

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localised more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

8.3 SYSTEM TESTING

System testing is a crucial process in the development of the Smart Face Recognition Attendance System. This process ensures that the system is free from errors, works as intended, and meets the requirements of the stakeholders. The system testing process involves several stages, including functional testing, performance testing, and security testing.

8.4 PERFORMANCE TESTING

Performance testing is in general executed to determine how a system or sub-system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. Load testing is primarily concerned with testing that the system can continue to operate under a specific load, whether that be large quantities of data or a large number of users. This is generally referred to as software scalability. The related load testing activity of when performed as a non-functional activity is often referred to as endurance testing. Volume testing is a way to test software functions even when certain components (for example a file or database) increase radically in size. Stress testing is a way to test reliability under unexpected or rare workloads.

8.5 SECURITY TESTING

Security testing is essential for software that processes confidential data to prevent system intrusion by hackers. So, it is important to perform security testing to find loop holes in the developed software.

8.6 USABILITY TESTING

Usability testing is needed to check if the user interface is easy to use and understand. It is concerned mainly with the use of the application. It is important to check interface is working properly or not as planned.

CHAPTER 9
RESULT AND DISCUSSION

The AI-Driven Fault Diagnosis system using upgraded YOLO models demonstrated highly accurate real-time detection of various solar panel faults, including hotspots, dirt accumulation, cracks, and partial shading. Through rigorous training and testing on a large annotated dataset, the model achieved impressive detection precision and recall values, outperforming traditional image processing methods. The YOLO-based architecture allowed for simultaneous localization and classification of faults with minimal latency, making it suitable for real-time deployment in photovoltaic fields. The visualization of output images with accurately marked bounding boxes and labels validated the system's ability to differentiate between fault types with high reliability. In the discussion phase, it was observed that the system maintained robustness even in varying lighting conditions and panel orientations, thanks to extensive preprocessing and data augmentation techniques. The use of transfer learning significantly reduced training time while enhancing detection performance. Additionally, the integration of non-maximum suppression ensured that the outputs were free of redundant detections, which improved the interpretability for solar maintenance teams. The results suggest that implementing this system can reduce manual inspection time, optimize maintenance schedules, and ultimately enhance the efficiency and lifespan of solar energy systems. The findings indicate strong potential for scaling the system across larger solar farms and integrating it with IoT-based monitoring platforms.

CHAPTER 10

CONCLUSION & FUTURE WORK

10.1 CONCLSUION

The AI-driven fault detection system for photovoltaic (PV) panels, leveraging the enhanced YOLOv11 algorithm, offers a highly efficient and accurate solution to address the common faults in solar panel systems. Through its real-time image processing capabilities, the system can accurately detect a wide range of faults such as cracks, dirt, hotspots, and shading inconsistencies. By automating the fault detection process, the system not only reduces human error but also improves the efficiency of solar panel maintenance, ensuring higher energy output and longer system lifespan. The integration of advanced image pre-processing techniques, grid division, bounding box predictions, and non-maximum suppression ensures that even small-scale defects are identified and localized with high precision, enhancing the reliability of solar energy systems. Furthermore, the system's ability to provide actionable insights through post-processing and reporting modules allows maintenance teams to take proactive measures and prevent further deterioration of solar panels. The continuous improvement through regular model updates, training with new data, and performance monitoring ensures that the system adapts to changing environmental conditions and evolving fault types. The project demonstrates the potential of AI and deep learning models in optimizing the performance and reliability of renewable energy systems, contributing to the broader adoption of solar energy and its role in sustainable energy production.

10.2 FUTURE WORK

- **Integration with IoT Sensors**

  Incorporating IoT-based sensors for real-time environmental monitoring can complement visual fault detection by providing data such as temperature, humidity, and irradiance for deeper fault analysis.

- **Drone-Based Inspection System**

  Deploying drones equipped with the YOLO-based detection system can enable autonomous scanning of large solar farms, reducing manual labor and enhancing coverage.

- **Enhanced Fault Categorization**

  Future versions can aim to identify and classify a broader range of fault types, such as Potential Induced Degradation (PID)[15], corrosion, and backsheet delamination with finer granularity.

- **Cloud-Based Monitoring Dashboard**

  Developing a centralized cloud dashboard for live visualization of fault locations, system performance, and automated report generation for maintenance teams.

- **Mobile Application for Field Technicians**

  A mobile-friendly application can be created to provide instant notifications, fault images, and recommended actions to technicians on-site for quicker response times.

- **Thermal and Electroluminescence (EL) Image Integration**

  Enhancing the dataset with thermal and EL imaging can improve fault detection accuracy for internal panel defects that are not visible in standard RGB images.

APPENDIX 1

SOURCE CODE

```
from ultralytics import YOLO
model = YOLO('yolo11n.pt')  # Load a pre-trained model (YOLO11n nano)
# Set up training parameters
model.train(
    data='datasets/data.yaml',  # Path to the dataset YAML file
    epochs=10,                  # Number of epochs
    imgsz=440,                  # Image size (640x640 in this case)
    batch=2                   # Batch size
    #device=1                   # Use GPU 0 (set device=-1 for CPU)
)
from flask import Flask, render_template, request, session, flash
import mysql.connector
import base64, os

from flask import Flask, render_template, request, jsonify
app = Flask(__name__)
app.config['SECRET_KEY'] = 'aaa'
@app.route("/")
def homepage():
    return render_template('index.html')

@app.route('/AdminLogin')
def AdminLogin():
    return render_template('AdminLogin.html')



@app.route('/UserLogin')
def UserLogin():
```

```
    return render_template('UserLogin.html')
@app.route('/NewUser')
def NewUser():
    return render_template('NewUser.html')
@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
        if request.form['uname'] == 'admin' and request.form['password'] == 'admin':

            conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
            cur = conn.cursor()
            cur.execute("SELECT * FROM regtb ")
            data = cur.fetchall()
            flash("you are successfully Login")
            return render_template('AdminHome.html', data=data)
        else:
            flash("UserName or Password Incorrect!")
            return render_template('AdminLogin.html')


@app.route("/AdminHome")
def AdminHome():
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
    cur = conn.cursor()
    cur.execute("SELECT * FROM regtb  ")
    data = cur.fetchall()
    return render_template('AdminHome.html', data=data)
@app.route("/Report")
def Report():
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
    cur = conn.cursor()
```

```
    cur.execute("SELECT * FROM activitytb  ")
    data = cur.fetchall()
    return render_template('Report.html', data=data)
@app.route("/newuser", methods=['GET', 'POST'])
def newuser():
    if request.method == 'POST':
        name = request.form['name']
        mobile = request.form['mobile']
        email = request.form['email']
        address = request.form['address']
        username = request.form['uname']
        password = request.form['password']

        conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
        cursor = conn.cursor()
        cursor.execute(
            "insert into regtb values('','" + name + "','" + mobile + "','" + email + "','" + address + "','" +
username + "','" + password + "')")
        conn.commit()
        conn.close()
        flash("Record Saved!")

    return render_template('NewUser.html')
@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['sname'] = request.form['uname']

        conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
        cursor = conn.cursor()
```

```
        cursor.execute("SELECT * from regtb where username='" + username + "' and password='" +
password + "'")
        data = cursor.fetchone()
        if data is None:
            flash('Username or Password is wrong')
            return render_template('UserLogin.html', data=data)

        else:

            session['mob'] =data[2]
            session['email'] = data[3]

            conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
            cur = conn.cursor()
            cur.execute("SELECT * FROM regtb where username='" + username + "' and password='" +
password + "'")
            data = cur.fetchall()
            flash("you are successfully logged in")
            return render_template('UserHome.html', data=data)

@app.route('/UserHome')
def UserHome():
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1solarfault')
    cur = conn.cursor()
    cur.execute("SELECT username FROM regtb  where username='" + session['sname'] + "' ")
    data = cur.fetchall()
    return render_template('DoctorHome.html', data=data)

@app.route('/Predict')
def Predict():
    return render_template('Predict.html')
```

```
@app.route("/imupload", methods=['GET', 'POST'])
def imupload():
    if request.method == 'POST':
        import cv2
        file = request.files['file']
        file.save('static/Out/Test.jpg')
        org = 'static/Out/Test.jpg'

        from ultralytics import YOLO
        import cv2
        # image = cv2.imread(import_file_path)
        image = cv2.imread(org)
        model = YOLO('runs/detect/solar/weights/best.pt')
        class_labels = ['faulty', 'no faulty']
        # Perform object detection
        results = model(image, conf=0.6)
        confidences = results[0].boxes.conf  # Confidence scores
        class_indices = results[0].boxes.cls  # Class indices
        if len(confidences) > 0:
            max_confidence_index = confidences.argmax().item()  # Get index of highest confidence
            predicted_class_index = int(class_indices[max_confidence_index].item())  # Get correct class
index

            # Ensure index is within bounds
            if 0 <= predicted_class_index < len(class_labels):
                predicted_class = class_labels[predicted_class_index]  # Map index to label
            else:
                predicted_class = "Unknown Class"

            confidence_score = confidences[max_confidence_index].item()  # Get highest confidence score
```

```
        print(f"Predicted Class: {predicted_class}")
        print(f"Confidence Score: {confidence_score:.4f}")  # Display with 4 decimal places
    else:
        predicted_class = "No Detections"
        confidence_score = 0.0
        print("No objects detected.")
    # Optionally, visualize the results
    annotated_frame = results[0].plot()
    outi = "static/Out/out.jpg"
    cv2.imwrite("static/Out/out.jpg", annotated_frame)

    # Display the annotated frame
    cv2.imshow("YOLOv8 Inference", annotated_frame)
    # cv2.waitKey(0)
    cv2.destroyAllWindows()
    return render_template('Predict.html', res=predicted_class, outi=outi)



@app.route("/Camera")
def Camera():
    import cv2
    from ultralytics import YOLO

    # Load the YOLOv8 model
    model = YOLO('runs/detect/solar/weights/best.pt')
    # Open the video file
    # video_path = "path/to/your/video/file.mp4"
    cap = cv2.VideoCapture(0)
    dd1 = 0

    # Loop through the video frames
```

```
while cap.isOpened():
    # Read a frame from the video
    success, frame = cap.read()

    if success:
        # Run YOLOv8 inference on the frame
        results = model(frame, conf=0.6)
        for result in results:
            if result.boxes:
                box = result.boxes[0]
                class_id = int(box.cls)
                object_name = model.names[class_id]
                print(object_name)

                if object_name == 'faulty':
                    dd1 += 1
                    print(dd1)

                if dd1 == 50:
                    dd1 = 0
                    import winsound
                    filename = 'alert.wav'
                    winsound.PlaySound(filename, winsound.SND_FILENAME)
                    annotated_frame = results[0].plot(
                    cv2.imwrite("alert.jpg", annotated_frame)
                    import random
                    loginkey = random.randint(1111, 9999)
                    imgg = "static/upload/" + str(loginkey) + ".jpg"
                    cv2.imwrite(imgg, annotated_frame)
                    import datetime
                    date = datetime.datetime.now().strftime('%Y-%m-%d')
```

```
                time = datetime.datetime.now().strftime('%H:%M:%S')

                conn = mysql.connector.connect(user='root', password='', host='localhost',
                                               database='1solarfault')
                cursor = conn.cursor()
                cursor.execute(
                    "INSERT INTO activitytb VALUES ('','"+ session['sname'] +"','" + date + "','" + time +
"','" + object_name + "','" + str(
                        imgg) + "')")
                conn.commit()
                conn.close()
                sendmail( session['email'])
                sendmsg(session['mob'], "Prediction Name:" + object_name)

        # Visualize the results on the frame
        annotated_frame = results[0].plot()

        # Display the annotated frame
        cv2.imshow("YOLOv8 Inference", annotated_frame)

        # Break the loop if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break
    # Release the video capture object and close the display window
    cap.release()
    cv2.destroyAllWindows()

def sendmsg(targetno, message):
    import requests
 requests.post(
"http://sms.creativepoint.in/api/push.json?apikey=6555c521622c1&route=transsms&sender=FSSMSS&m
obileno=" + targetno + "&text=Dear customer your msg is " + message + "  Sent By FSMSG FSSMSS")
```

```
def sendmail(mail):
    import smtplib
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    from email.mime.base import MIMEBase
    from email import encoders
    fromaddr = "projectmailm@gmail.com"
    toaddr = mail

    # instance of MIMEMultipart
    msg = MIMEMultipart()

    # storing the senders email address
    msg['From'] = fromaddr

    # storing the receivers email address
    msg['To'] = toaddr

    # storing the subject
    msg['Subject'] = "Alert"

    # string to store the body of the mail
    body = " Solar Panel Fault Detection"

    # attach the body with the msg instance
    msg.attach(MIMEText(body, 'plain'))
    # open the file to be sent
    filename = "alert.jpg"
    attachment = open("alert.jpg", "rb")

    # instance of MIMEBase and named as p
    p = MIMEBase('application', 'octet-stream')
```

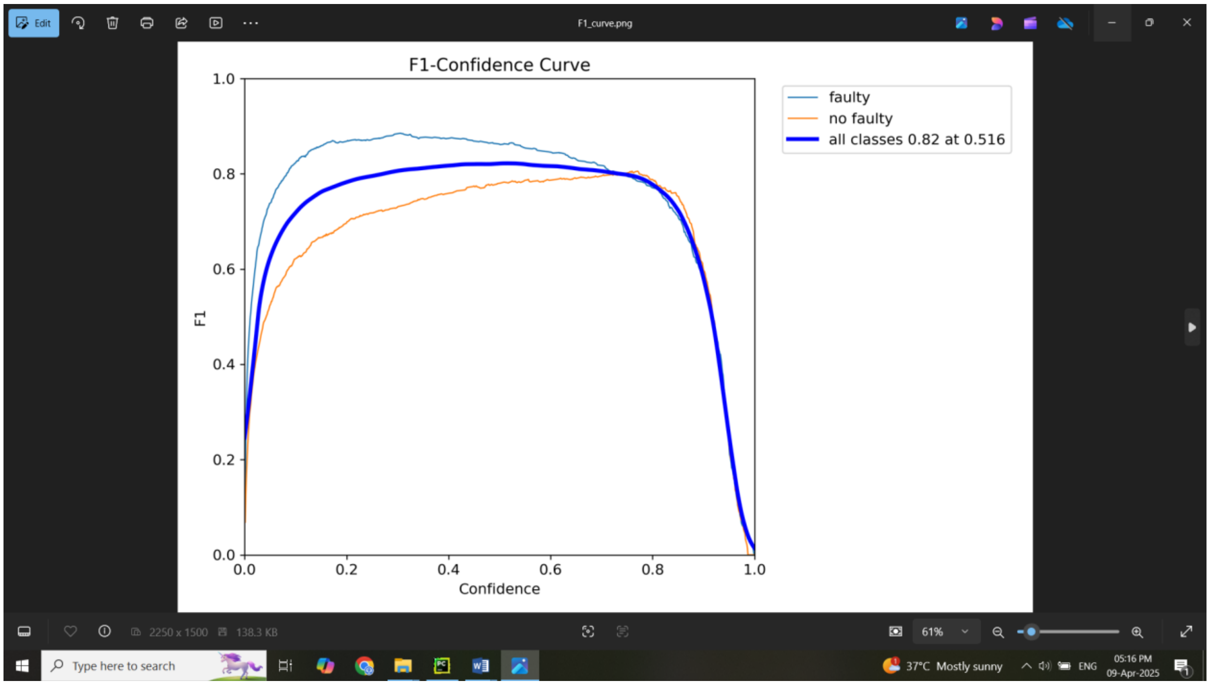
```
# To change the payload into encoded form
p.set_payload((attachment).read())

# encode into base64
encoders.encode_base64(p)
p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

# attach the instance 'p' to instance 'msg'
msg.attach(p)

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login(fromaddr, "qmgn xecl bkqv musr")

# Converts the Multipart msg into a string
text = msg.as_string()

# sending the mail
s.sendmail(fromaddr, toaddr, text)

# terminating the session
s.quit()
if __name__ == '__main__':
app.run(debug=True, use_reloader=True)
```

APPENDIX 2

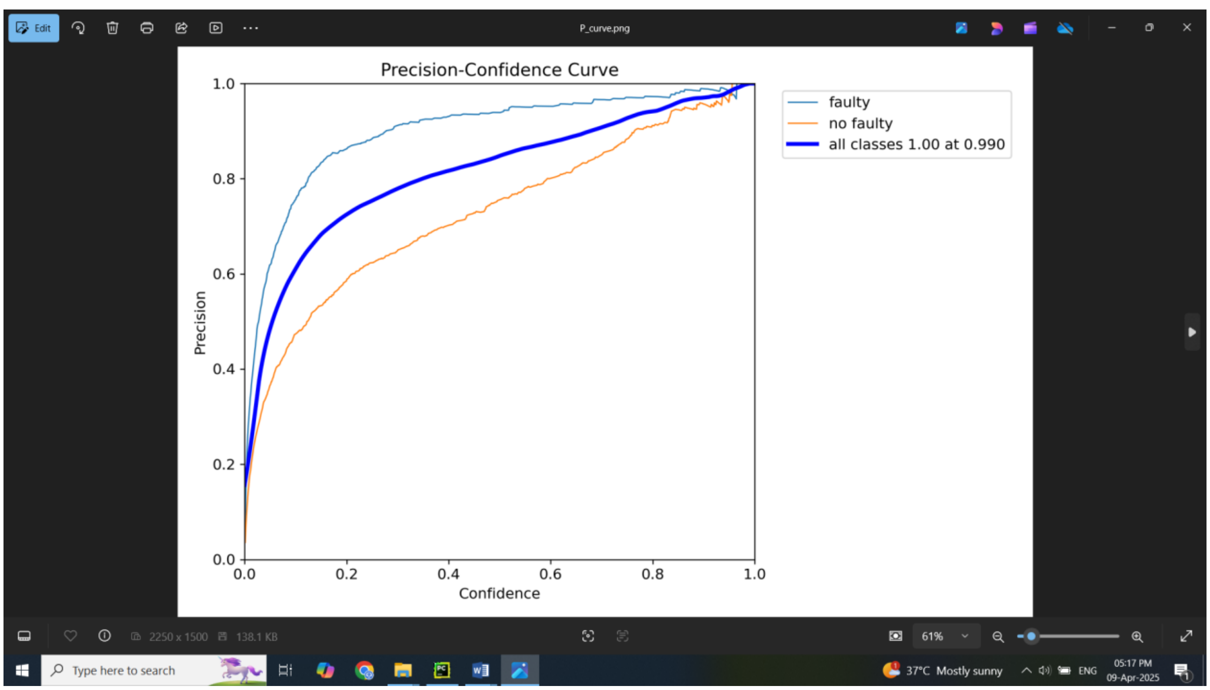
SCREENSHOTS

B.1 Confusion Matrix Normalized

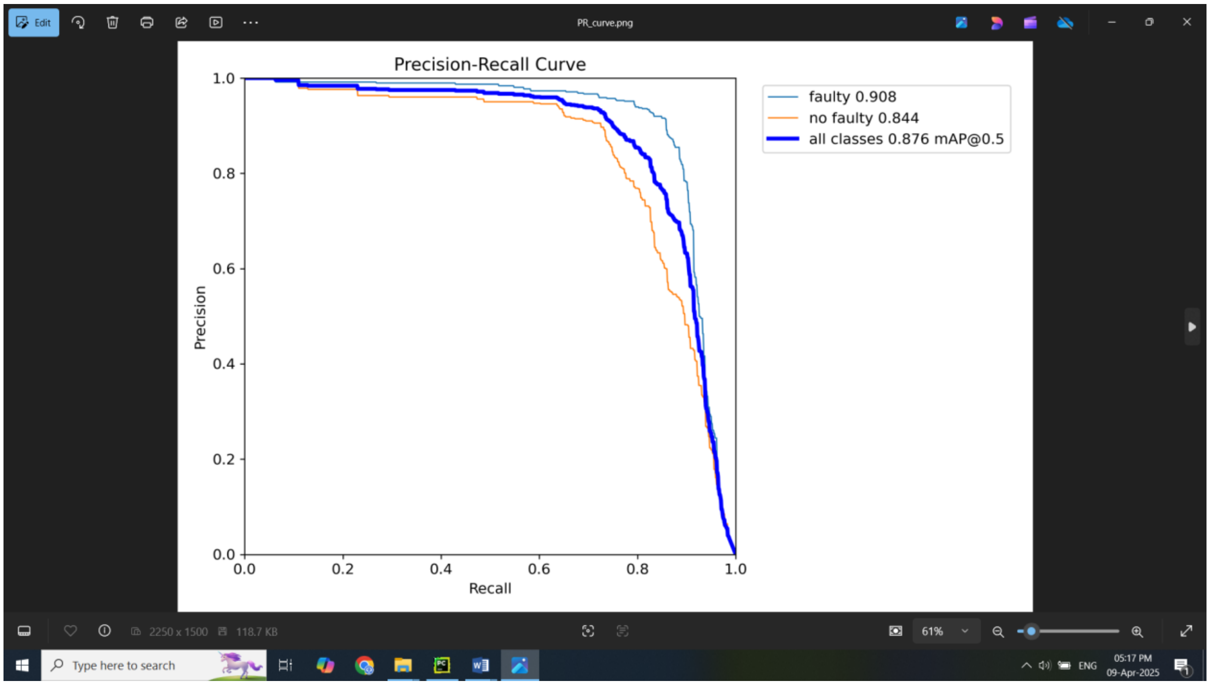
B.2 Confusion Matrix of Proposed Model



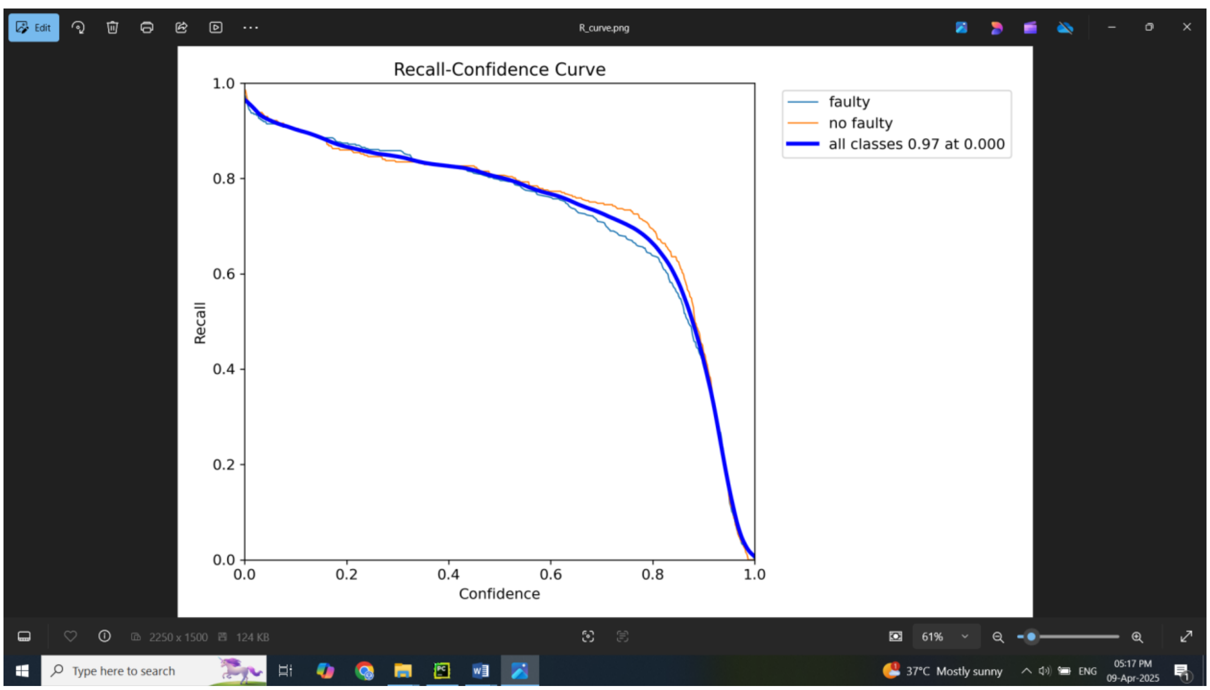
B.3 F1- Confidence Curve



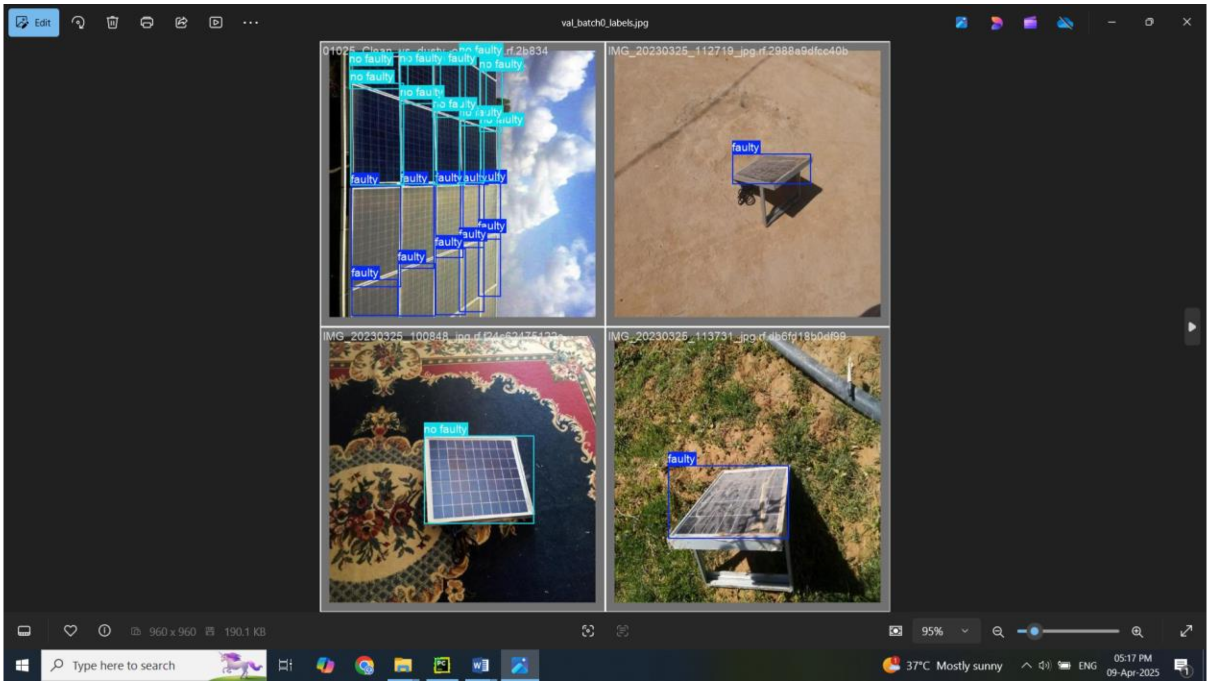
B.4 Precision- Confidence Curve



B.5 Precision – Recall Curve

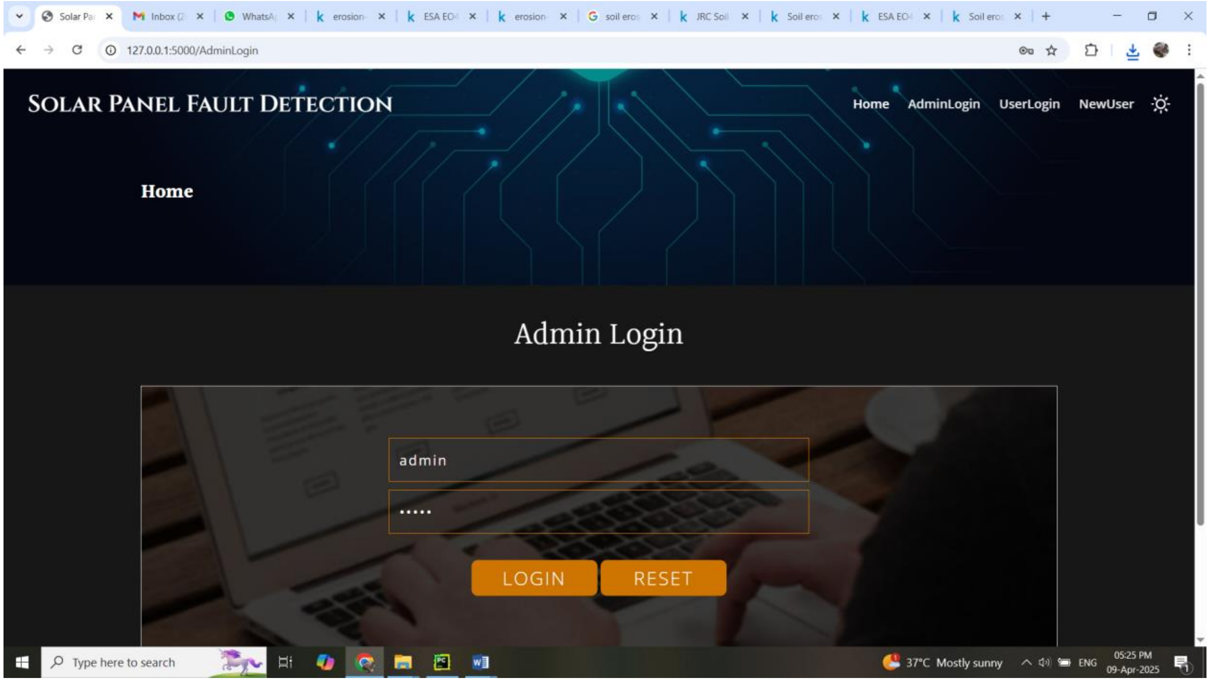


B.6 Recall - Confidence Curve

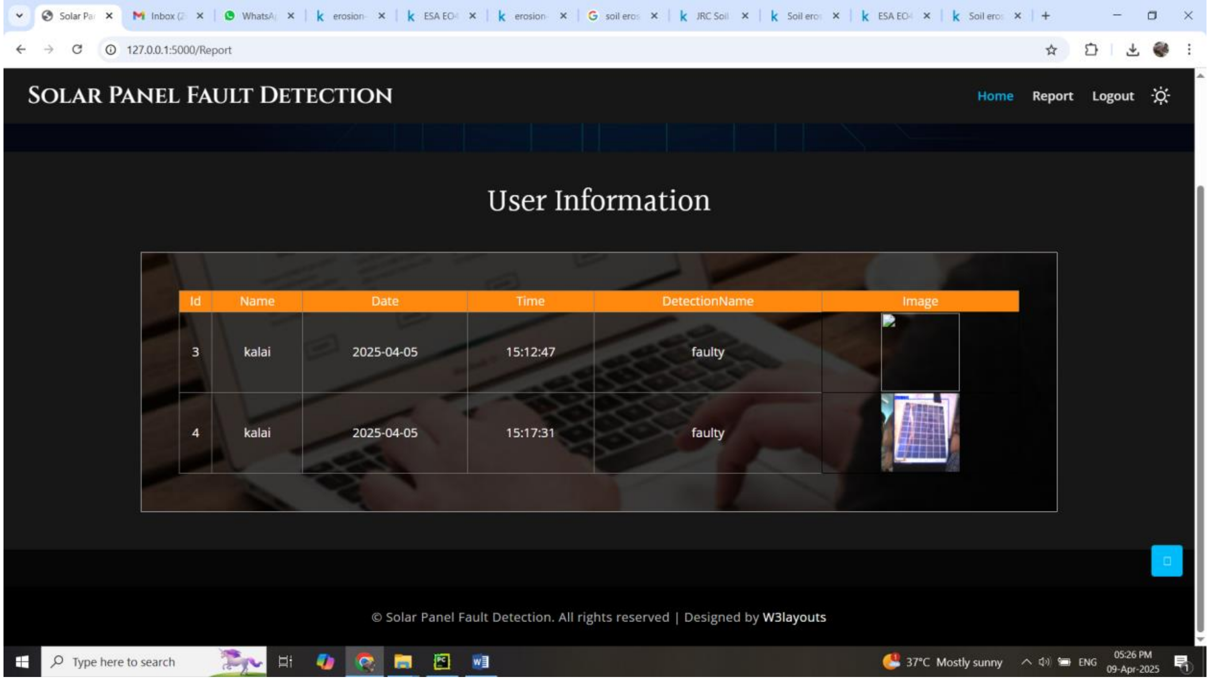


B.7 Solar Panel Fault Prediction

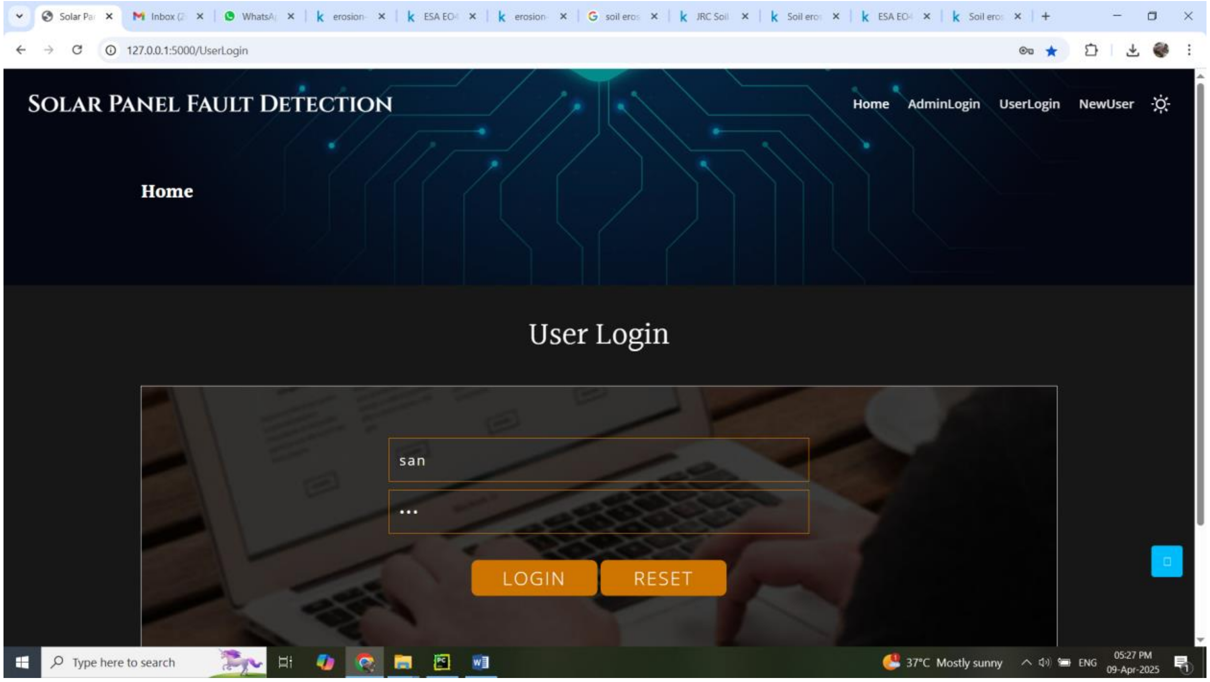
B.8 Home Page



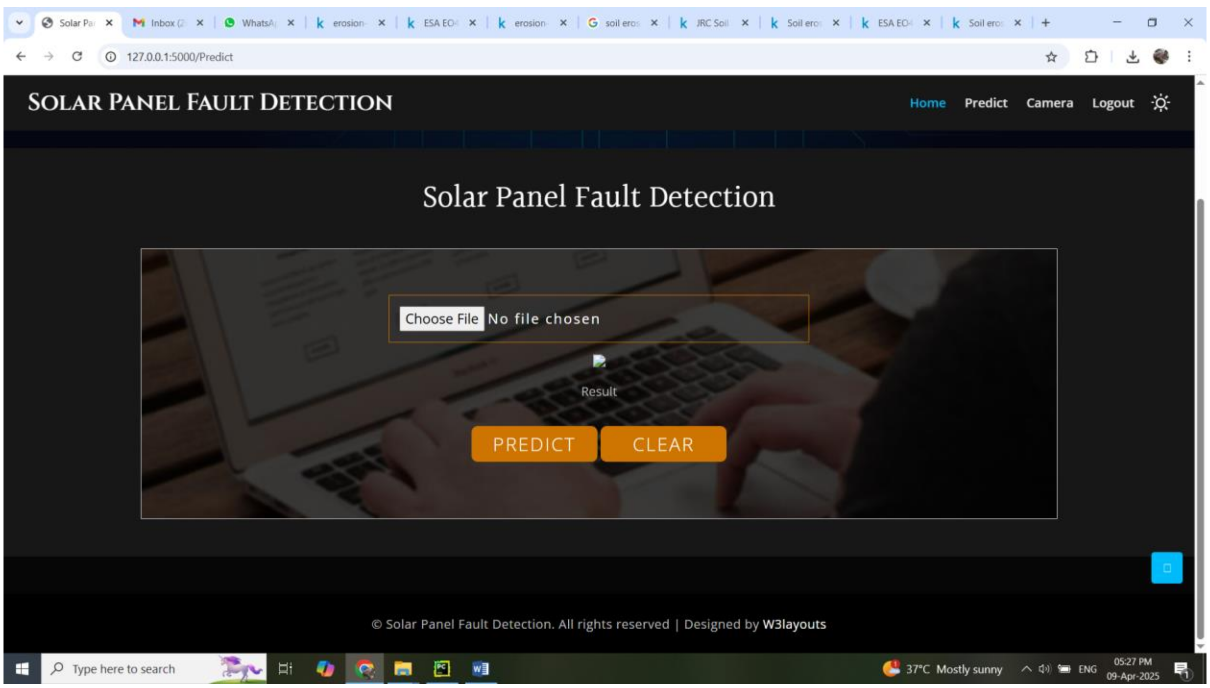
B.9 Admin Login Page



B.10 Reports of Fault



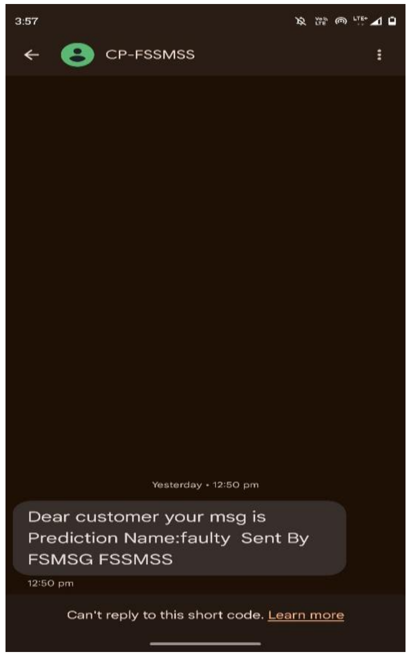
B.11 User Login Page



B.12 Fault Detection

B.12.1 Fault Detection(Output : No Fault)

B.12.2 Fault Detection(Output : Fault)



B.15 SMS Alert

REFERENCES

1. Duranay, Zeynep Bala. (2023) ‘Fault detection in solar energy systems: A deep learning approach’, *Electronics*, Vol.12, No.21, p.4397.
2. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019) ‘CenterNet: Keypoint triplets for object detection’, in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp.6568–6577.
3. El-Banby, Ghada M., et al. (2023) ‘Photovoltaic system fault detection techniques: a review’, *Neural Computing and Applications*, Vol.35, No.35, pp.24829–24842.
4. Hijjawi, Ula, et al. (2023) ‘A review of automated solar photovoltaic defect detection systems: Approaches, challenges, and future orientations’, *Solar Energy*, Vol.266, p.112186.
5. Javaid, Aqdas, et al. (2024) ‘Enhancing photovoltaic systems using Gaussian process regression for parameter identification and fault detection’, *Energy Reports*, Vol.11, pp.4485–4499.
6. Joshua, Salaki Reynaldo, et al. (2024) ‘A Hybrid Machine Learning Approach: Analyzing Energy Potential and Designing Solar Fault Detection for an AIoT-Based Solar–Hydrogen System in a University Setting’, *Applied Sciences*, Vol.14, No.18, p.8573.
7. Liu, W. (2016) ‘SSD: Single shot multibox detector’, in *Proc. 14th Eur. Conf.*, Oct. 2016, pp.21–37.
8. Lodhi, Ehtisham, et al. (2023) ‘A Novel deep stack-based ensemble learning approach for fault detection and classification in photovoltaic arrays’, *Remote Sensing*, Vol.15, No.5, p.1277.
9. Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., and Huang, Z. (2023) ‘Efficient multi-scale attention module with cross-spatial learning’, in *Proc. IEEE*

Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2023, pp.1–5.

10. Pierdicca, R., Paolanti, M., Felicetti, A., Piccinini, F., and Zingaretti, P. (2020) 'Automatic faults detection of photovoltaic farms: Solar, a deep learning-based system for thermal images', *Energies*, Vol.13, No.24, p.6496.
11. Ren, S., He, K., Girshick, R., and Sun, J. (2015) 'Faster R-CNN: Towards real-time object detection with region proposal networks', in *Proc. Adv. Neural Inf. Process. Syst.*, Vol.28, pp.1–20.
12. Rudro, Rifat Al Mamun, et al. (2024) 'SPF-Net: Solar panel fault detection using U-Net based deep learning image classification', *Energy Reports*, Vol.12, pp.1580–1594.
13. Saiprakash, Chidurala, et al. (2024) 'Improved fault detection and classification in PV arrays using stockwell transform and data mining techniques', *Results in Engineering*, Vol.23, p.102808.
14. Thakfan, Ali and Salamah, Yasser Bin. (2024) 'Artificial-Intelligence-Based Detection of Defects and Faults in Photovoltaic Systems: A Survey', *Energies*, Vol.17, No.19, p.4807.
15. Yang, Cheng, et al. (2024) 'A Survey of Photovoltaic Panel Overlay and Fault Detection Methods', *Energies*, Vol.17, No.4, p.837.

LIST OF PUBLICATIONS

Ajitha V, Dharini B, Kalaiarasi B and Kanjanamala R, "**AI-Driven Fault Diagnosis in Solar Panel Units Using Upgraded YOLO Models**". 2nd International Conference on Multi-Strategy Learning Environment ICMSLE 2025, Graphic Era Hill University, Haldwani, India (**Accepted for Publication in SPRINGER**).

Springer

INTERNATIONAL CONFERENCE ON MULTI-STRATEGY LEARNING ENVIRONMENT

ICMSLE

Graphic Era Hill University

DEHRADUN • BHIMTAL • HALDWANI

Certificate of Presentation

This is to certify that

Ajitha V

has presented a paper titled

AI-Driven Fault Diagnosis in Solar Panel Units using Upgraded YOLO Models

at the 2nd International Conference on Multi-Strategy Learning Environment (ICMSLE-2025) held on 26th and 27th February 2025 at Graphic Era Hill University, Haldwani, India.

Session Chair

Prof. Dibyahash Bordoloi
Conference Chair

Prof. Dr. Mahesh Manchanda
Convener

Springer

INTERNATIONAL CONFERENCE ON MULTI-STRATEGY LEARNING ENVIRONMENT
ICMSLE

Graphic Era Hill University
DEHRADUN • BHIMTAL • HALDWANI

Certificate of Presentation

This is to certify that

Dharini B

has presented a paper titled

AI-Driven Fault Diagnosis in Solar Panel Units using Upgraded YOLO Models

at the 2nd International Conference on Multi-Strategy Learning Environment (ICMSLE-2025) held on 26th and 27th February 2025 at Graphic Era Hill University, Haldwani, India.

	Prof. Dibyahash Bordoloi	***Prof. Dr. Mahesh Manchanda***
Session Chair	*Conference Chair*	*Convener*

Springer

INTERNATIONAL CONFERENCE ON MULTI-STRATEGY LEARNING ENVIRONMENT
iCMSLE

Graphic Era Hill University
DEHRADUN • BHIMTAL • HALDWANI

Certificate of Presentation

This is to certify that

Kalaiarasi B

has presented a paper titled

AI-Driven Fault Diagnosis in Solar Panel Units using Upgraded YOLO Models

at the 2nd International Conference on Multi-Strategy Learning Environment (ICMSLE-2025) held on 26th and 27th February 2025 at Graphic Era Hill University, Haldwani, India.

Session Chair

Prof. Dibyahash Bordoloi
Conference Chair

Prof. Dr. Mahesh Manchanda
Convener

Springer

INTERNATIONAL CONFERENCE ON MULTI-STRATEGY LEARNING ENVIRONMENT
ICMSLE

Graphic Era Hill University
DEHRADUN • BHIMTAL • HALDWANI

Certificate of Presentation

This is to certify that

Kanjanamala R

has presented a paper titled

AI-Driven Fault Diagnosis in Solar Panel Units using Upgraded YOLO Models

at the 2nd International Conference on Multi-Strategy Learning Environment (ICMSLE-2025) held on 26th and 27th February 2025 at Graphic Era Hill University, Haldwani, India.

	Prof. Dibyahash Bordoloi	***Prof. Dr. Mahesh Manchanda***
Session Chair	*Conference Chair*	*Convener*