Advances in Energy Science and Technology
Vol. 1, No. 1|1 June 2025
AEST

Integrating digital twin for optimizing autonomous aerial monitoring of photovoltaic systems

Mohammad Kolahi[a] ,Sayyed Majid Esmailifar[b] , Mohammadreza Aghaei[c,d*], AmirMohammad Moradi Sizkouhi[e]

[a]*Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran*

[b]*Department of Aerospace Engineering, Amirkabir University of Technology, Tehran, Iran*

[c]*Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology (NTNU), Ålesund, Norway*

[d]*Department of Sustainable Systems Engineering (INATECH), Albert Ludwigs University of Freiburg, Freiburg, Germany*

[e]*Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada*

Keywords

Photovoltaics (PV)
Aerial robot
Artificial intelligence (AI)
Digital twin (DT)
Autonomous aerial monitoring (AAM)

Article Info

DOI: 10.22060/AEST.2025.5748

Received 6 Jan 2025

Accepted 1 Feb 2025

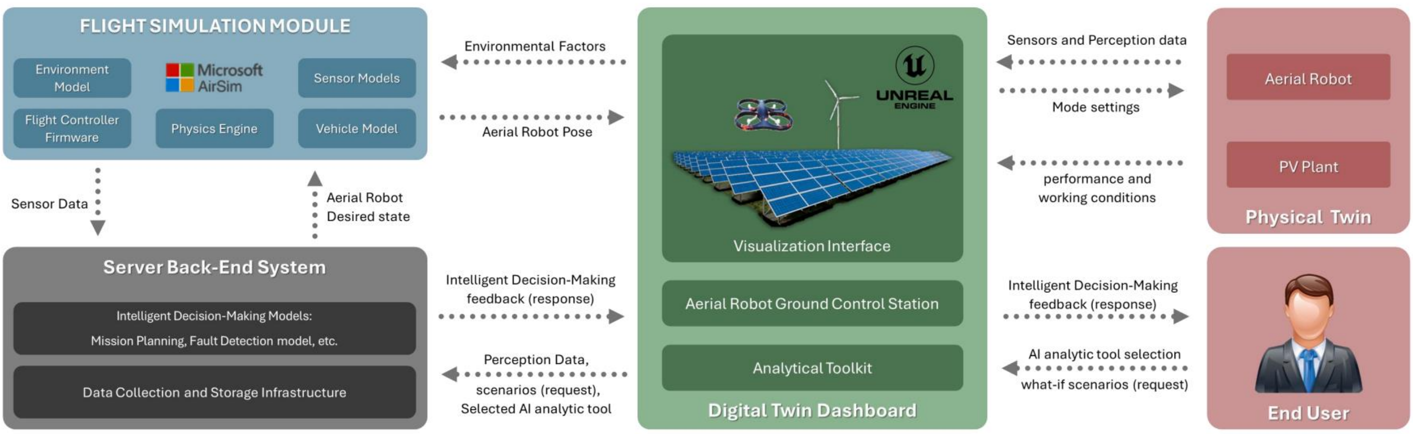
Published 1 June 2025

* **Corresponding author:**
Aghaei@studysolar.uni-freiburg.de

Abstract

In this research, we integrate digital twin (DT) technology through a platform designed for simulating and managing the autonomous aerial monitoring procedure of photovoltaic (PV) power plants, known as Digital-PV. This innovative platform enables users to test various scenarios and configurations of PV power plants, allowing for an evaluation of their impact on the autonomous aerial monitoring process. By doing so, it reduces the risks associated with real-world experiments and helps pinpoint the most effective strategies for improving PV system monitoring. Digital-PV also provides a virtual environment for conducting tests of autonomous flights and missions, covering aspects such as boundary detection, path planning, and fault detection. It includes features for generating data that inform the development of data-driven monitoring and inspection models. The development process of Digital-PV involved creating a digital twin of a utility-scale PV plant within Unreal Engine, simulating aerial robot flight with AirSim, and expanding of application programming interface (APIs) to enable our platform to adapt to different scenarios for evaluating smart monitoring models and collecting datasets. Moreover, throughout the study, a dataset of synthetic aerial images was collected from Digital-PV, which was subsequently used to train an end-to-end segmentation model aimed at detecting bird droppings on PV panels. Ultimately, this platform evaluated a range of intelligent aerial monitoring models, providing valuable insights into their capabilities and potential effectiveness in real-world applications.

Graphical Abstract



1. Introduction

In recent years, the global installation of large-scale photovoltaic (PV) power plants has surged. The International Renewable Energy Agency (IRENA) projects that by 2050, PV systems will supply over one-third of the world's electricity demand [1,2]. To ensure energy performance and reliability, efficient monitoring strategies are essential as the number and size of PV plants continue to grow [3].

Over the past two decades, the fields of artificial intelligence (AI), machine learning (ML), and deep learning (DL) have increasingly been leveraged to enhance the modeling of photovoltaic (PV) plants and streamline defect detection and decision-making processes within these plants [4–6]. Meanwhile, the rise of aerial robots in the last decade has emerged as a practical solution for monitoring and inspecting various sites, including PV plants [7,8]. Utilizing aerial robots for this purpose offers a quick, precise, and convenient method for inspections [9].

The aerial imagery captured during inspections by these aerial robots, when analyzed using advanced AI technology, serves as an effective tool for monitoring PV power plants. The images collected can be processed either offline or online, yielding valuable insights that enhance the efficiency and performance of the plants. This innovative approach has transformed traditional monitoring and inspection processes, increasing their speed and accuracy while cutting costs. It has also paved the way for the concept of autonomous aerial monitoring (AAM) for PV plants, marking a significant advancement in the field.

A great deal of progress has been made recently in the field of AAM of PV power plants. Most of these studies have primarily focused on automatically detecting faults in different types of aerial PV imagery, including visual [10], Infrared (IR) [11,12], and Electroluminescence (EL) [13] images. These studies have investigated a diverse range of faults for detection, including hot spots [14], bird drops [15], Shading and soiling [16], snail trails, glass breakage, burn marks, delamination, and discoloration [17,18], etc.

some studies have focused on enhancing time and battery efficiency during aerial PV monitoring by investigating the monitoring path, planning the optimal path, and identifying the optimal waypoints [19–21]. The traveling salesman algorithm [19] and density clustering, boustrophedon planning, and Bezier curves [20] were utilized to minimize battery usage. Similarly, a joint approach of Bezier curves and particle swarm optimization (PSO) considered flight attitude, path length, and gimbal limitations for path planning optimization [21].

Accurately defining the plant boundaries as a prerequisite for monitoring path planning is an essential task in the AAM of PV plants. A U-Net architecture was employed in [22,23] for effective boundary extraction followed by the use of different algorithms to determine the best flight path, with the exact cellular decomposition boustrophedon and grid-based wavefront coverage algorithms yielding the best results. Also, the Fully Convolutional Network (FCN) was utilized In [24,25], for semantic segmentation and extraction of the PV plant boundary.

One of the key challenges in employing AI techniques for monitoring PV plants is the difficulty in acquiring a large set of annotated data [26]. Public datasets are often limited [27,28], and conducting field flights for data collection can be costly, given the potential for damage to aerial robots or PV panels. Moreover, labeling data in a vast dataset can be a time-consuming and tedious task. To tackle these challenges, this paper introduces a framework that leverages digital twin (DT) technology.

The DT concept has gained considerable traction in the energy sector [29,30]. Our DT-based solution empowers users to simulate various scenarios and configurations of PV plants, evaluating their effects on these systems' monitoring processes. It also provides a virtual testing environment for autonomous flight missions, including boundary detection, path planning,

Advances in Energy Science and Technology
Vol. 1, No. 1|1 June 2025
AEST

and fault detection (FD). Additionally, this framework allows for the efficient augmentation of our dataset with labeled images, facilitating the training of intelligent monitoring models.

Our contribution, termed Digital-PV, includes several key elements: (i) the creation of a DT model for a utility-scale PV plant environment, (ii) expansion of an application programming interface (API) to enable our platform to adapt to different scenarios for evaluating smart monitoring models and collecting datasets, (iii) enhancement of detection capabilities by training the FD model using the synthetic data generated by Digital-PV, and (v) assessment of the performance of four monitoring models utilizing Digital-PV. With these contributions, Digital-PV aims to facilitate monitoring applications for PV systems, concentrating on automated aerial monitoring and the management and optimization of large-scale PV plant monitoring. This framework can be directly employed for on-site planning, data collection, and autonomous monitoring research of PV plants.

In the rest of the paper, the design and construction of Digital-PV's main components and data interactions are outlined in Section 3. Section 4 illustrates the platform's effectiveness in creating and assessing various intelligent models for aerial monitoring tasks. We achieve this by generating a dataset using Digital-PV, training a fault detection (FD) model with the collected data, and testing intelligent monitoring models that autonomously execute a range of tasks for the aerial monitoring of the virtual PV plant.

2. Development of Digital-PV

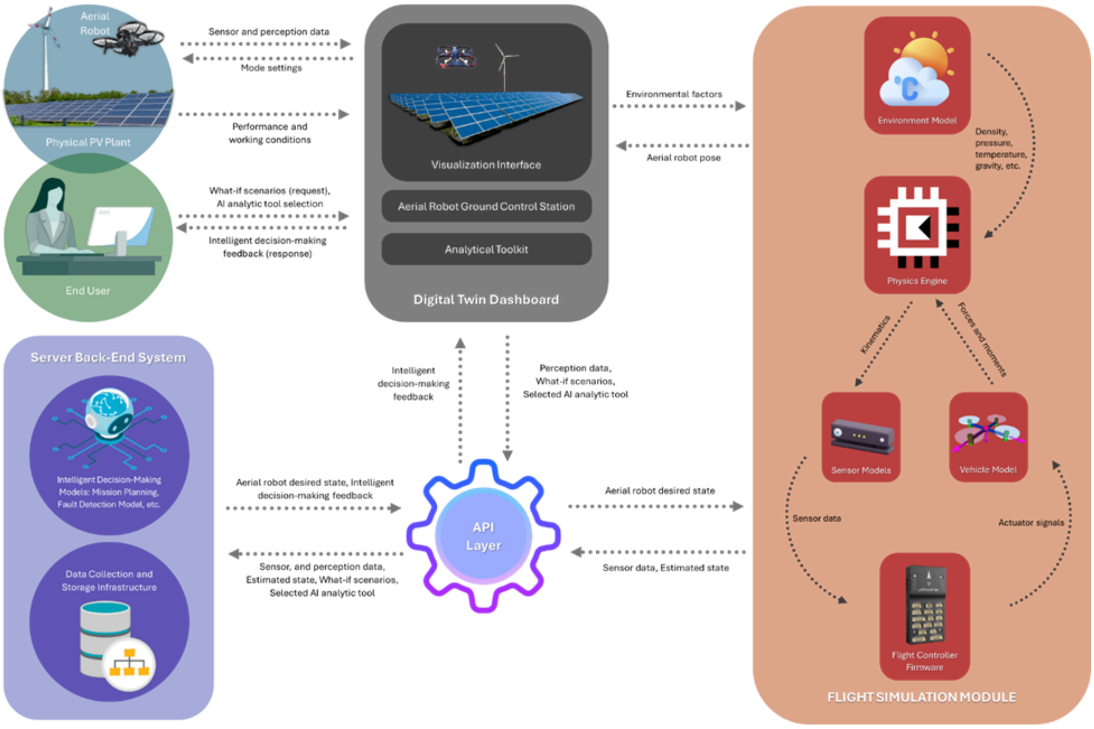


Fig. 1. Interaction and data flow diagram of Digital-PV components [31].

Digital-PV is made up of four key components: a DT dashboard that includes the visualization interface, aerial robot ground control station (GCS), and analytical toolkit; a flight simulation module; a server back-end system featuring intelligent decision-making models and data storage; and an API layer that enables communication between these components. Fig. 1 depicts how these components interact and the flow of data among them.

2.1. PV Plant's Virtual Model Development in Unreal Engine

Game engines have been suggested to create high-fidelity DTs of their physical counterparts by offering realistic graphics, lighting, fluid dynamics, and physics engines that simulate the real world. Using a game engine as the foundation of our DT-based platform enables the investigation of factors such as the sun's position, weather, air velocity, reflection, self-shading, etc. that might affect the quality of the monitoring process. In addition, high-quality visualization is crucial for developing and evaluating computer vision models. Therefore, we utilize Unreal Engine (UE), created by Epic Games for Digital-PV development. Using UE also makes our platform possible to import photogrammetry-created 3D models, including those captured with a drone or a phone, into the 3D environment of the PV plant, creating an incredibly realistic setting.

Fig. 2. An overall view of the virtual PV plant environment constructed for the 3D visual representation of the physical PV plant.

This study utilized a standard game environment workflow to develop a virtual PV plant. It began by modeling a flat terrain of 4 square kilometers using UE's landscape tools, incorporating minor variations for realism. To enhance the environment, pre-made assets and 3D models, such as trees, mountains, and rocks, were sourced from UE's Marketplace and Quixel. Additionally, vegetation was added in varying densities to create a natural landscape. A procedural sky system was implemented, and atmospheric and lighting settings were adjusted to replicate real-world conditions. Volumetric clouds were included to affect lighting and cast shadows on the panels, with customizable cloud density, height, and thickness for realism. To integrate plant components like PV panels and wind turbines, necessary assets, and 3D models were acquired from marketplaces such as TurboSquid and SketchFab. Also, the PV panel prototype's metal structure and glass texture were enhanced for realistic reflections. The static mesh of the PV panels and their structures was optimized for better performance by converting them to lower polygon, or "game-ready," assets to improve frame rate during simulations with numerous panels. Blueprints were also utilized to efficiently generate and arrange arrays of PV panels, allowing for a configurable and scalable environment. Fig. 2 shows the constructed environment. Finally, the built-in static mesh editor in UE was used to create custom meshes of bird droppings in various shapes and sizes, serving as fault characters on PV panels. This set includes five individual static meshes to simulate faults, aiding in dataset collection.

2.2. Aerial Robot Flight Simulation for performing AAM

To accurately simulate aerial robot dynamics and mimic real-world behavior, we use AirSim to import the aerial robot into the virtual PV plant environment. AirSim provides tools for a generic quadrotor model, a physics engine for drag, acceleration, and collisions, and an environment model that includes magnetic fields, gravity, and air pressure. Additionally, it includes various sensors like barometers, gyroscopes, accelerometers, magnetometers, GPS units, and vision sensors such as RGB and depth cameras. AirSim enables both software-in-the-loop (SITL) and hardware-in-the-loop (HITL) simulations for a more realistic experience. It also features customization options through a .json settings file, which we modified specifically for our platform. We set up a fixed "ExternalCamera" at the PV plant's "Reference Point" to capture vertical images that encompass the entire plant boundary. The "CaptureSettings" were fine-tuned to achieve a resolution of 480×640 with a 90-degree field of view. We adjusted the "MotionBlurAmount" to zero and increased the "AutoExposureSpeed" to 100 to ensure that we obtain high-quality images without any blur. Moreover, we made use of the "Recording" option for dataset collection, with detailed parameters outlined in the relevant section.

AirSim provides APIs for interacting with the vehicle during simulations, allowing control of the vehicle, image retrieval, state assessment, and more. In this research, we expanded its Python APIs to carry out a specific scenario involving four main tasks:

- Providing an image of the entire PV plant for boundary extraction.
- Generating a flight path for aerial monitoring using a path planning algorithm.
- Iterative imaging of the PV arrays and processing images while the aerial robot follows the generated path.
- Executing maneuver commands for detailed examination in case of fault or anomaly detection.

At the beginning of the simulation, the user's desired weather conditions and time of day are set, followed by the initialization and take-off of the aerial robot. Users can choose between two methods for capturing an image of the entire PV plant: a fixed camera positioned at a reference point for a satellite-like view, or using the aerial robot to take a picture after increasing altitude towards the reference point and stabilizing at that point. This image is then used for boundary detection. Once the plant boundary is detected, a convex curve is drawn around it and sent to the path planning algorithm to create waypoints for aerial monitoring.

With the waypoints established, the aerial robot heads to the first waypoint to begin monitoring. The gimbal on the robot's bottom camera adjusts to a 10-degree pitch angle for a better view of the panel surfaces. While following the planned path, the aerial robot continuously captures images of the panel surfaces with its camera. These images are fed into the fault detection (FD) unit, processed by an anomaly detection (AD) model to evaluate potential faults and output a "fault flag" to the decision-making unit. Each image also includes a location tag, enabling the robot to autonomously deviate from its path for further investigation of detected issues. Upon receiving a fault flag, the decision-making unit determines whether to continue or maneuver for a closer look. If a fault flag is activated, the decision-making unit instructs the aerial robot to descend to the fault location and capture an image of the potentially defective panel. This image is sent to the FD unit for further analysis by the module-level FD model. Fig. 3 illustrates the relationship and data flow between the aforementioned components.

After the investigation, the fault flag is reset to zero, and the aerial robot ascends back to its previous altitude to continue toward the next waypoint. This process repeats until reaching the last waypoint and the monitoring mission is complete. Upon reaching the last point of the path, the aerial robot is commanded to return home, and it returns to its take-off location to complete the simulation scenario after landing.

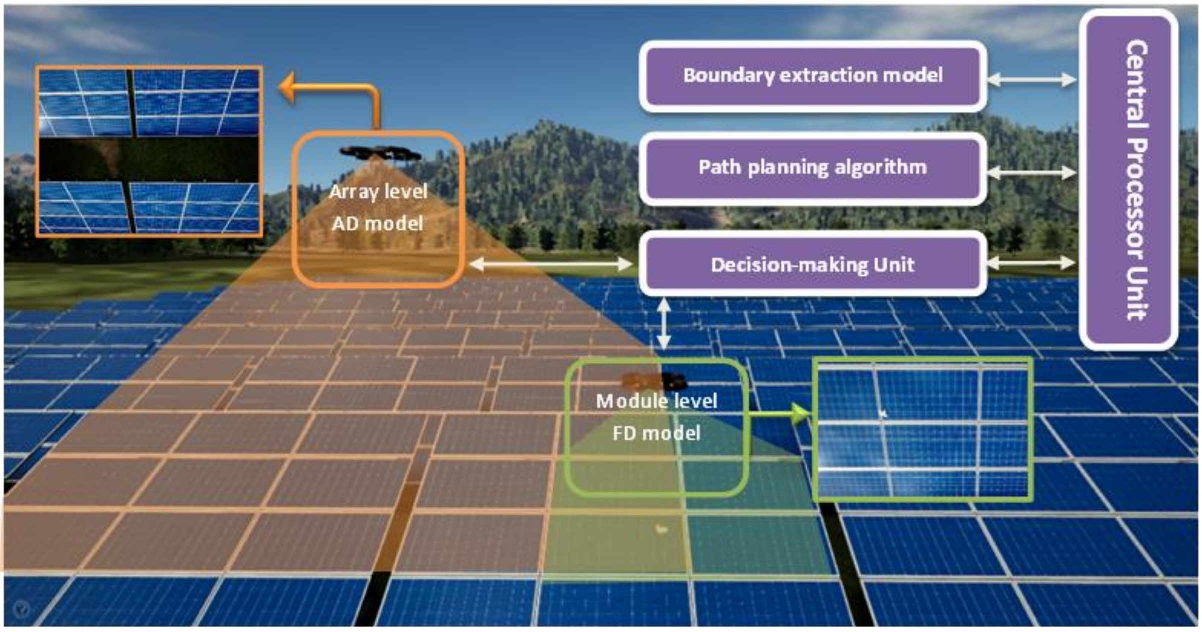


Fig. 3. Visual representation of the interlinked functions driving Digital-PV's mission scenario execution.

3. Generating Synthetic Training Data for Developing AI Models by Digital-PV

This section will focus on showcasing the capabilities of Digital-PV in generating datasets for monitoring model development. In the following section, we will explore the testing of intelligent monitoring models designed for autonomous aerial monitoring of PV plants.

Digital-PV enables the collection of diverse, realistic synthetic images from virtual PV environments, facilitating the training of AI models for various monitoring tasks, such as fault detection and boundary extraction. This study collected a dataset of PV panel images impacted by bird droppings. To simulate these faults, we created 3D models of bird droppings, featuring five unique static meshes strategically placed on panels with varying shapes, sizes, and orientations to represent different samples.

In addition, the "Recording" settings in the AirSim pre-simulation setting file were adjusted to enable recording images from the aerial robot's bottom camera as desired. RGB images and their corresponding segmentation masks were recorded every 100 milliseconds in compressed PNG format, with dimensions of 3x480x640. The "RecordOnMove" parameter was enabled to avoid duplicate recordings when the vehicle stopped. During several flights over the PV plant, the aerial robot collected images of defective panels under various conditions, creating a dataset that reflects real-world scenarios. Ultimately, the dataset comprised 2,469 samples, each paired with a binary segmentation mask that labeled the PV panels and their faults. Fig. 4 shows some sample images from the collected dataset.

Vol. 1, No. 1|1 June 2025
Advances in Energy Science and Technology
AEST

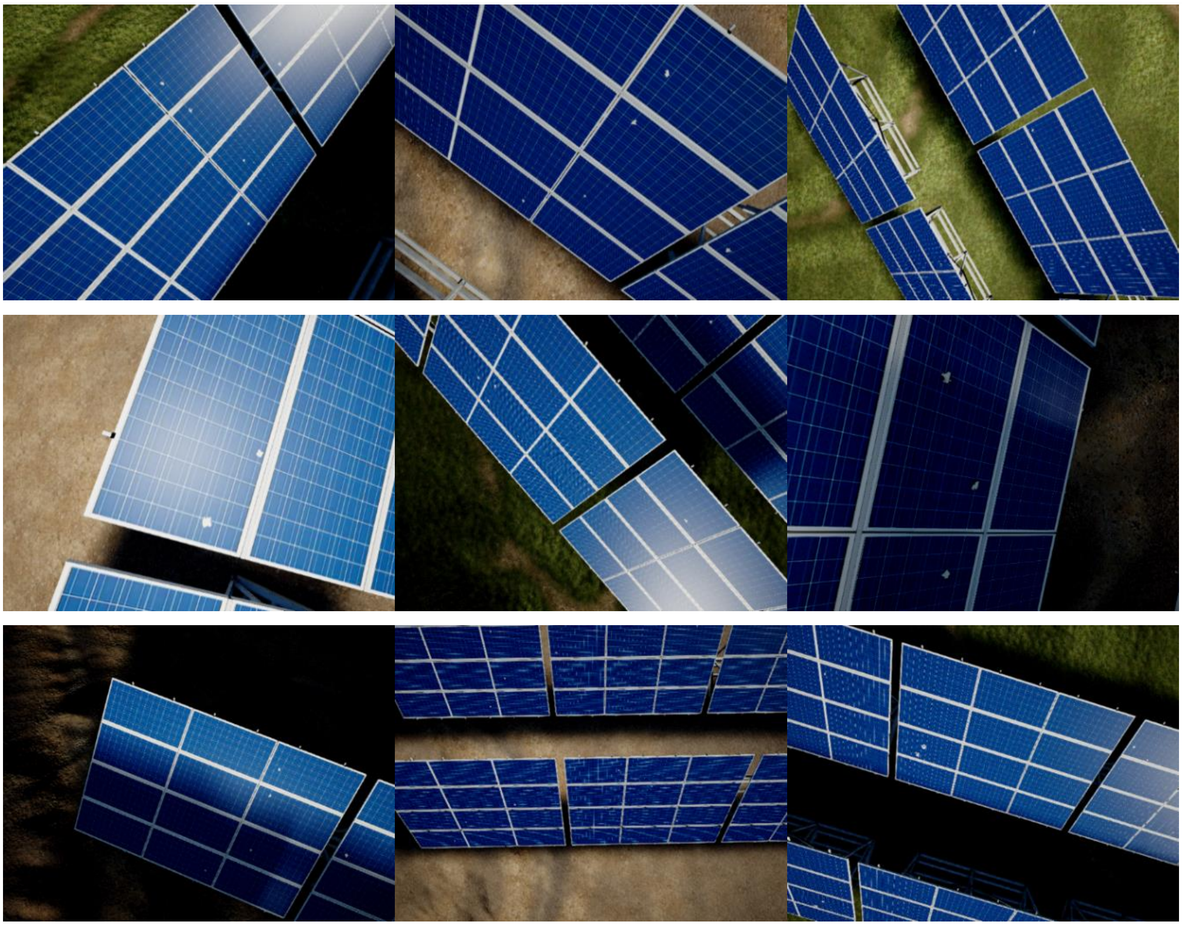


Fig. 4. Some sample images from the collected dataset. UE's movable lighting feature produced various lighting conditions. Panels were photographed from different angles, distances, and perspectives, capturing images that included one or more arrays. Additionally, two textures were applied to the PV plant's ground to simulate its location in desert and plain environments.

3.1. Development of an End-to-end Fault Detection Model

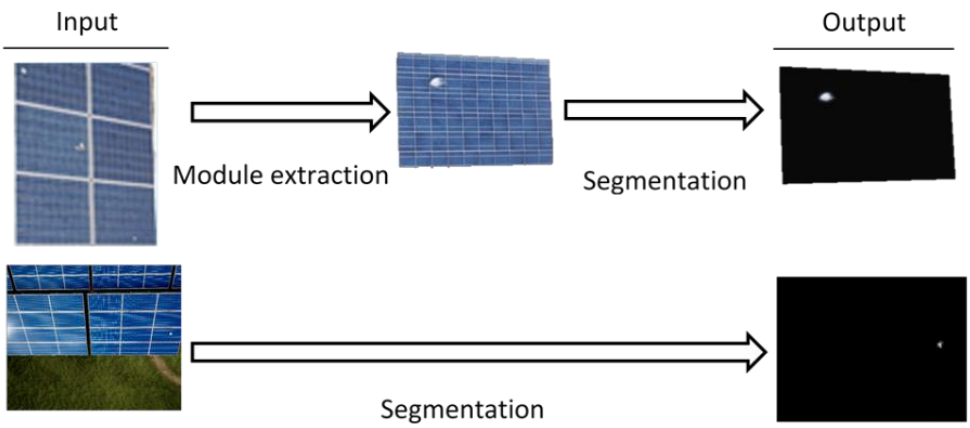


Fig. 5. Comparing end-to-end vs. multi-stage models for bird drop segmentation in aerial robot-captured aerial images. The multi-stage model has to be optimized for module extraction and fault segmentation separately under different criteria.

In this study, an end-to-end segmentation model has been proposed for the automatic FD of PV panels. In this approach, there is no need to separately extract modules from aerial images captured by the aerial robot, which includes multiple PV modules. Fig. 5 compares the end-to-end model with the multi-stage model for bird dropping segmentation in aerial photos captured by the aerial robot.

Table 1 outlines the layer types, outputs, and parameters for both the encoder and decoder sections. The model has a total of 16,281,921 parameters, with 8,646,657 trainable and 7,635,264 non-trainable.

Table 1. Layer types, outputs, and number of parameters produced per layer of FCN.

Layer (type)	Output Shape	Param
Input 1 (input layer)	(480, 640, 3)	0
Block1 conv1 (Conv2D)	(480, 640, 64)	1792
Block1 conv2 (Conv2D)	(480, 640, 64)	36928
Block1 pool (MaxPooling2D)	(240, 320, 64)	0
Block2 conv1 (Conv2D)	(240, 320, 128)	73856
Block2 conv2 (Conv2D)	(240, 320, 128)	147584
Block2 pool (MaxPooling2D)	(120, 160, 128)	0
Block3 conv1 (Conv2D)	(120, 160, 256)	295168
Block3 conv2 (Conv2D)	(120, 160, 256)	590080
Block3 conv3 (Conv2D)	(120, 160, 256)	590080
Block3 pool (MaxPooling2D)	(60, 80, 256)	0
Block4 conv1 (Conv2D)	(60, 80, 512)	1180160
Block4 conv2 (Conv2D)	(60, 80, 512)	2359808
Block4 conv3 (Conv2D)	(60, 80, 512)	2359808
Block4 pool (MaxPooling2D)	(30, 40, 512)	0
Block5 conv1 (Conv2D)	(30, 40, 512)	2359808
Block5 conv2 (Conv2D)	(30, 40, 512)	2359808
Block5 conv3 (Conv2D)	(30, 40, 512)	2359808
Conv2d transpose (Conv2DTranspose)	(60, 80, 256)	1179904
Conv2d transpose 1 (Conv2DTranspose)	(120, 160, 128)	295040
Conv2d transpose 2 (Conv2DTranspose)	(240, 320, 64)	73792
Conv2d transpose 3 (Conv2DTranspose)	(480, 640, 32)	18464
Conv2d transpose 4 (Conv2DTranspose)	(480, 640, 1)	33

The system employs an encoder-decoder-based FCN architecture, as proposed in [50], which is common in image segmentation. As illustrated in Fig. 6, the process starts by down sampling the input image and extracting its feature map using a modified VGG16 as the encoder unit of the network. To enhance training accuracy, transfer learning techniques are applied, utilizing pre-trained ImageNet weights for the weights of the first 14 layers. In the decoder, up sampling is carried out through deconvolution operations. This workflow is designed to map low-resolution features to full-resolution images for pixel-level segmentation.

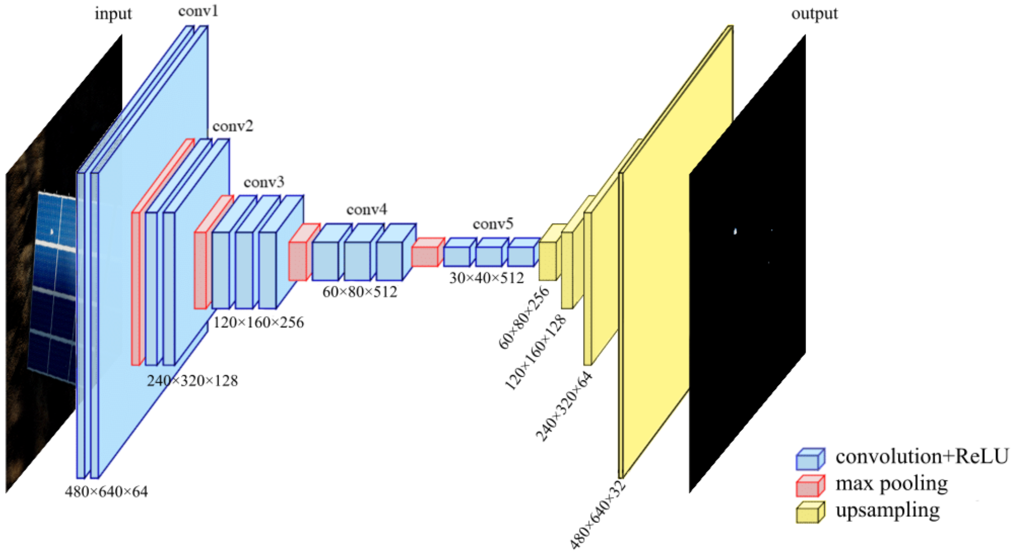


Fig. 6. The encoder-decoder architecture. It consists of 13 convolution and 4 up-sampling layers combined with pooling and non-linear layers. Each layer is accompanied by an activation map m_i^c.

3.2. Training and Test Results

For model training, the dataset images were divided into three groups: training (85%), validation (10%), and testing (5%). The training process consisted of 10 epochs with a batch size of 10, utilizing shuffled images to ensure diversity within each batch. A binary cross-entropy function, commonly employed for loss computation in image segmentation, was used to calculate the loss during training. This function compares the predicted value vector with the encoded target vector, as illustrated in Equation (1).

$$H_P(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\big(p(y_i)\big) + (1 - y_i)\log\big(1 - p(y_i)\big) \tag{1}$$

It evaluates each pixel, where y denotes the label of each pixel (1 for areas belonging to the fault and 0 for other areas), and p(y) represents the probability of each pixel being part of the target region (the fault area). The network's objective is to estimate the probability of each pixel belonging to the defect area. During the loss calculation, for all pixels in the fault region (where y = 1), the logarithm of p(y) is added to the total loss. Additionally, we utilized the Adam optimizer to update the network weights, using a learning rate of 0.001. The model parameters used in the training process are summarized in Table 2.

Table 2. Model training parameters.

Batch size	Epochs	Initial weight	Loss function	optimizer	Activation function	Last layer activation function
10	10	ImageNet	binary_crossentropy	adam	Relu	Sigmoid

Network performance is evaluated using pixel accuracy, which is calculated as shown in Equation (2). During training and validation, the system continuously updates accuracy and loss rates, achieving average accuracies of 98.31% for training and 97.93% for validation. The trained model accurately identifies birds on PV modules with a test accuracy of 95.2%. Additionally, precision calculated from Equation (3) is 94.44%, and recall calculated from Equation (4) is 94.89%.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Fig. 7 displays the predicted results and actual masks for three test images that remained unseen during training.

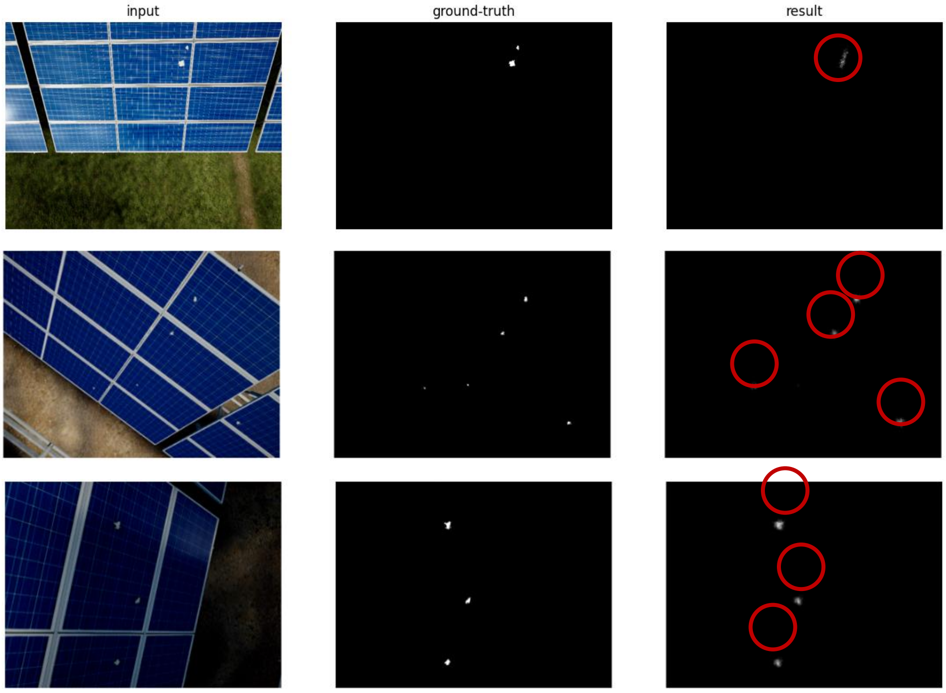


Fig. 7. Predicted results for bird droppings are shown with actual masks, highlighted by red circles indicating detected faults.

3.3. Assessment of Intelligent Monitoring Models' Performance Using Digital-PV

As previously mentioned, various intelligent aerial monitoring models can be assessed through Digital-PV to gain insights into their capabilities and potential performance in real-world scenarios. The performance of FD and boundary extraction models can be evaluated by Digital-PV using two key approaches. First, qualitative evaluations provide initial insights into the models, while quantitative analyses delve deeper into their effectiveness through various AI metrics like precision, F1-score, recall, accuracy, the Dice coefficient, and Intersection over Union (IoU). To evaluate flight trajectories, the aerial robot can be flown in the simulator in different patterns using the same commands as for the real vehicle, gathering and comparing the positions from both modes in local North-East-Down (NED) coordinates along with timestamps. In the rest of this section, we will select specific models for each task and evaluate their performance through our developed simulations, presenting the results thereafter.

3.3.1. Boundary Extraction Model Test

Boundary extraction is the first step for aerial monitoring of a PV plant. Accurate boundary extraction is essential for precise path planning. In [32], two methods were utilized for extracting the boundaries of a PV plant. The first approach leverages CIP techniques, while the second relies on an image segmentation model that employs deep learning techniques. Both methods were evaluated through separate simulations in Digital-PV. This simulation offers insights into the boundary detection system and its potential real-world performance.

To proceed, an image of the entire PV plant is captured and resized to fit the model's input layer according to the dimensions provided by the user. From this resized image, the boundary detection models produce a binary mask that indicates the PV plant area, assigning a value of one to pixels within the PV plant. The next step involves extracting points along the edge of the defined boundary from this output. Given that a PV plant may contain multiple groups of PV arrays, which results in several white regions in the output mask, a convex hull is used to encircle the detected plant area. This creates a unified boundary curve that encompasses all sections of the PV plant. Fig. 8 visually illustrates the steps taken for boundary extraction, utilizing both the CIP and DL-based models.

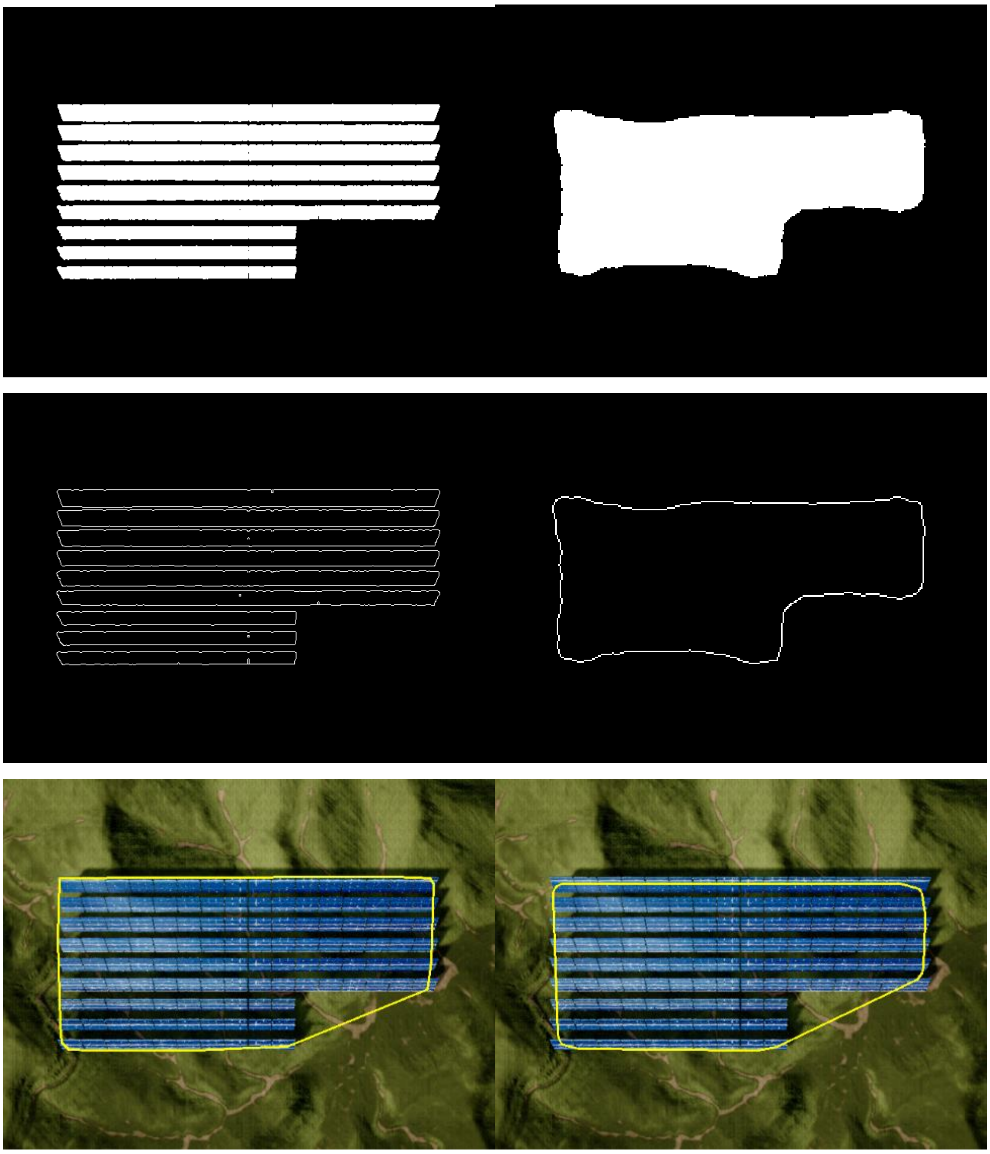


Fig. 8. Boundary detection and convex closure drawing steps by employing CIP (left) and DL-based (right) models.

3.3.2. Path Planning Model Test

Path planning plays a crucial role in enabling autonomous monitoring of PV plants using aerial robots. Digital-PV's versatility in simulating PV plants of various scales and shapes, along with adjustable flight dynamics and parameters, allows for testing and evaluating path planning algorithms under different conditions.

This section examines the performance of the algorithm presented in [25], which is designed for autonomous path planning. The algorithm starts by processing the plant boundary line obtained from the boundary extraction unit. After extracting the PV plant boundary (Fig. 9-left), the path planning model receives this input. Then, from the intersection of the boundary line with guidelines based on the plant's width, flight altitude, and the aerial robot's camera FoV, defining the ground coverage area, flight waypoints are generated (see Fig. 9-right and Fig. 10).

Vol. 1, No. 1|1 June 2025
Advances in Energy Science and Technology
AEST

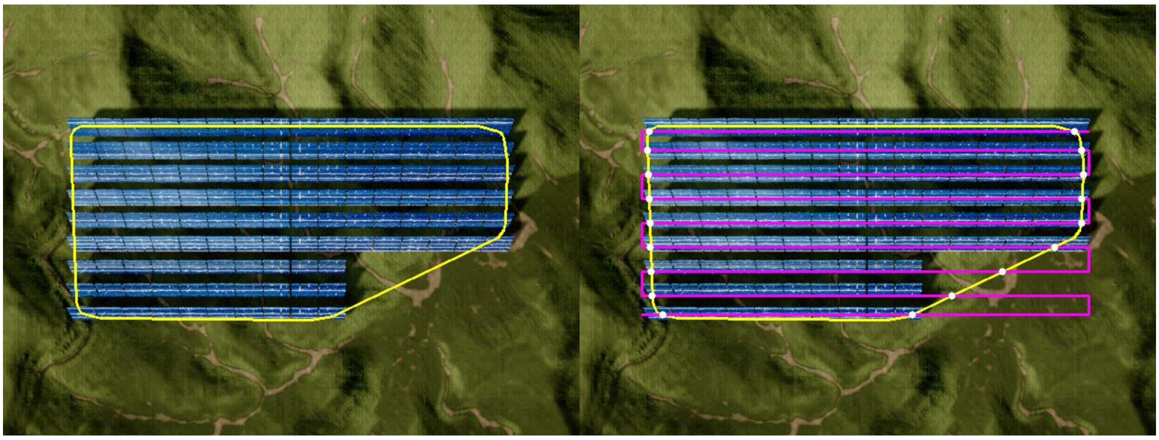


Fig. 9. Path planning input (left) and Path planning output (right).

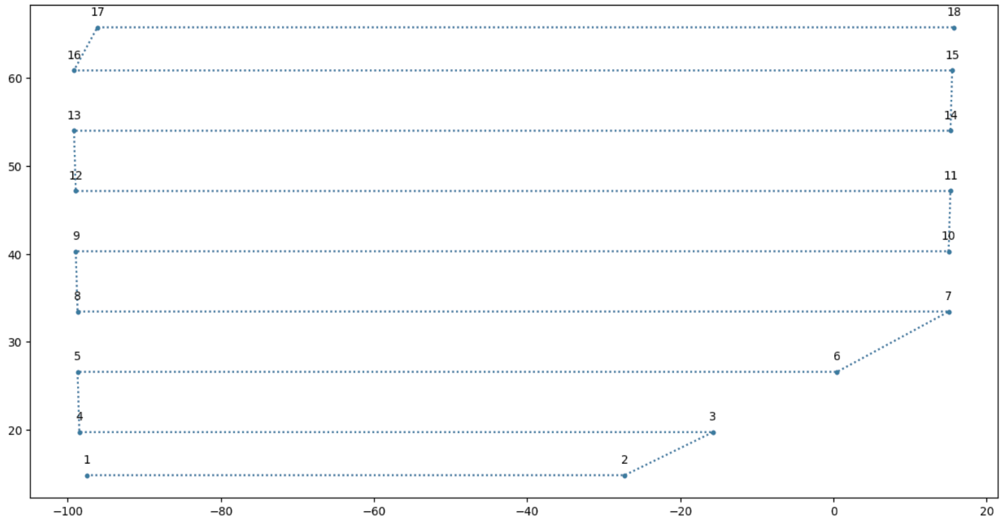


Fig. 10. Transformation of aerial robot's waypoints to NED coordinate system.

Finally, for the simulation of the aerial monitoring mission, waypoints are transmitted to the aerial robot via AirSim APIs, guiding it to the first waypoint and along the designated path. Fig. 11 illustrates the aerial robot's trajectory as it navigates through the specified waypoints in the developed environment.

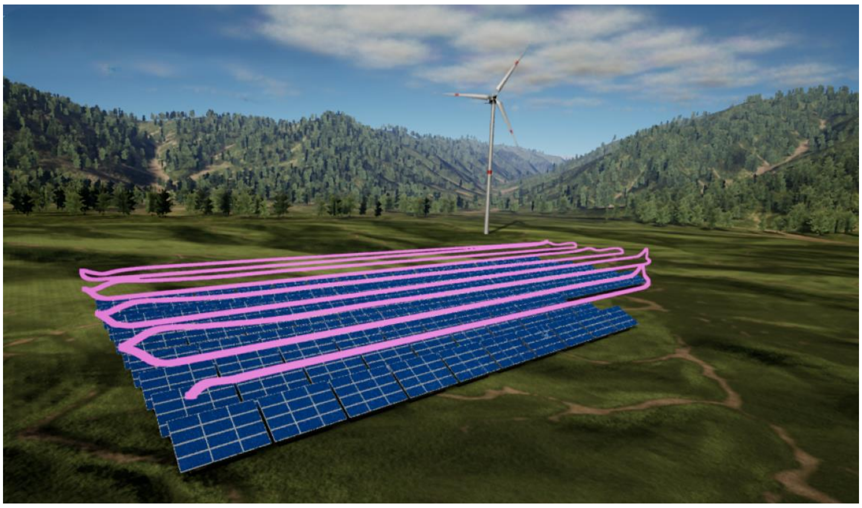


Fig. 11. Aerial robot's trajectory after passing through the given waypoints.

Advances in Energy Science and Technology
Vol. 1, No. 1|1 June 2025
AEST

3.3.3. Fault Detection Model Test

The anomaly or FD model at two levels can be assessed during a single simulation in Digital-PV. At the start of the virtual monitoring mission, multiple panel images from different arrays are captured simultaneously within the camera's FoV due to the higher aerial robot's flight altitude. The model that processes the images in this situation is called the "array-level AD model".

If the anomaly detection (AD) model identifies an issue at the array level, a maneuver command is issued, prompting the aerial robot to descend to a predetermined altitude. This adjustment places the aerial robot directly above the detected anomaly, allowing for the observation of individual modules or a single PV panel (Fig. 13). This close-up image is sent to the "module-level FD model" for further analysis before the aerial robot ascends back to its original altitude to continue its mission. Each model at these two levels can perform various tasks, such as classification and segmentation.

Given that the end-to-end segmentation model developed in Section 4.2 can detect faults from aerial images without needing to extract PV modules, we evaluated its performance at both the array and module levels during the monitoring simulation. To detect anomalies in the array-level images, the platform provides 3D NumPy arrays of dimensions 3×640×480 as input. The model's initial output is a 2D NumPy mapped to a numerical value within a specified range. If the output exceeds a certain threshold, the image is classified as faulty; if it falls below, it is considered healthy.(see Fig. 12)

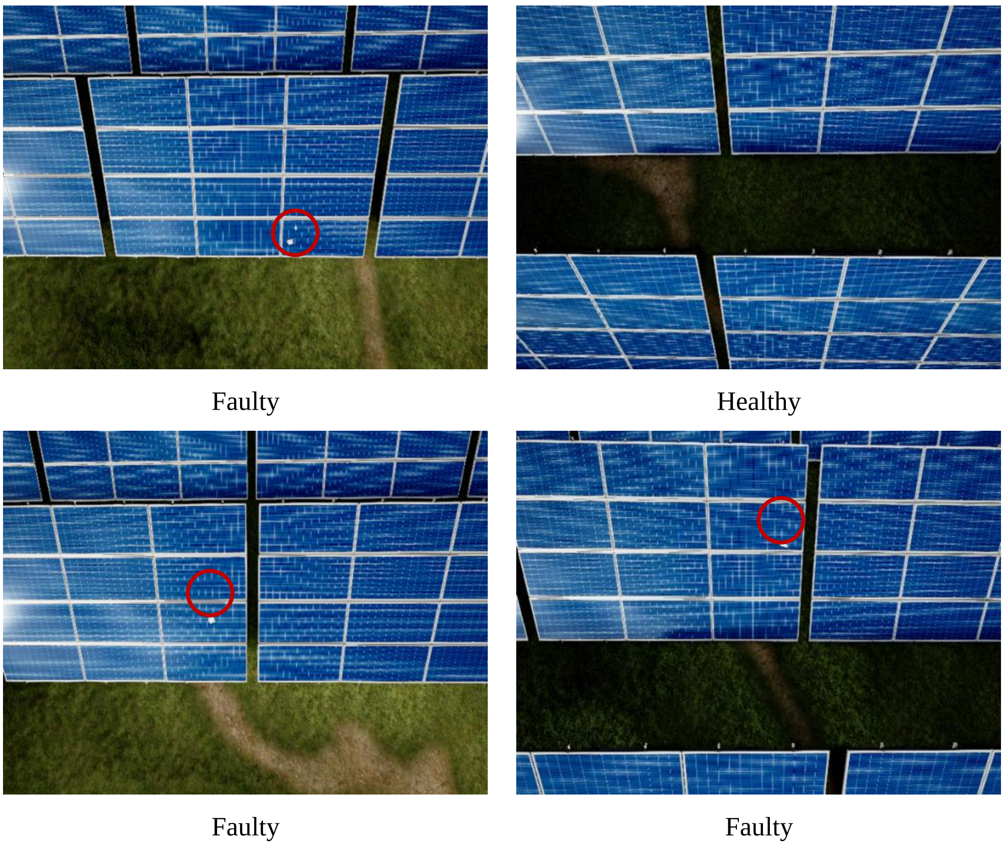


Fig. 12. Examples of the input images used for the AD model during monitoring, along with the corresponding predicted label for each image.

When a fault flag is raised, the aerial robot lowers its altitude to hover above the PV modules and capture a close-up image. This image is delivered to the module-level FD model as a 3D NumPy array with dimensions 3×640×480 for further analysis. Fig. 14 and Fig. 15 display the captured image of some defective panels identified during monitoring, along with their predicted segmentation masks

Fig. 13. Aerial robot hovering over the faulty panel for precise capture.

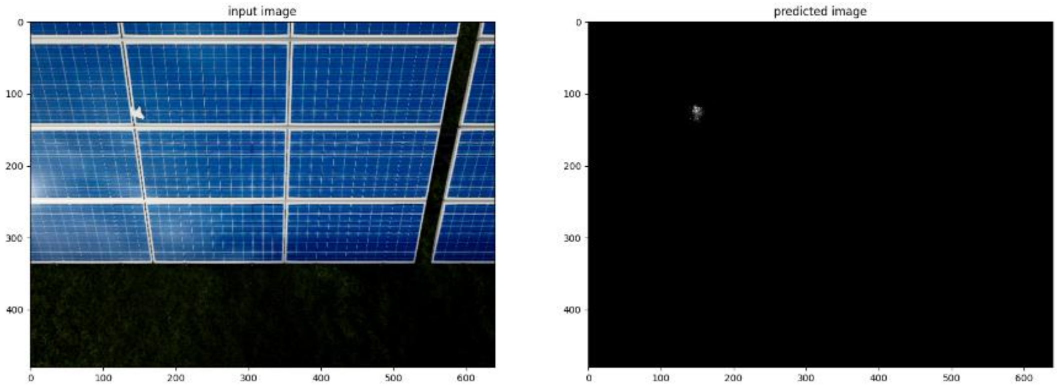


Fig. 14. Close-up taken image and corresponding segmentation mask.

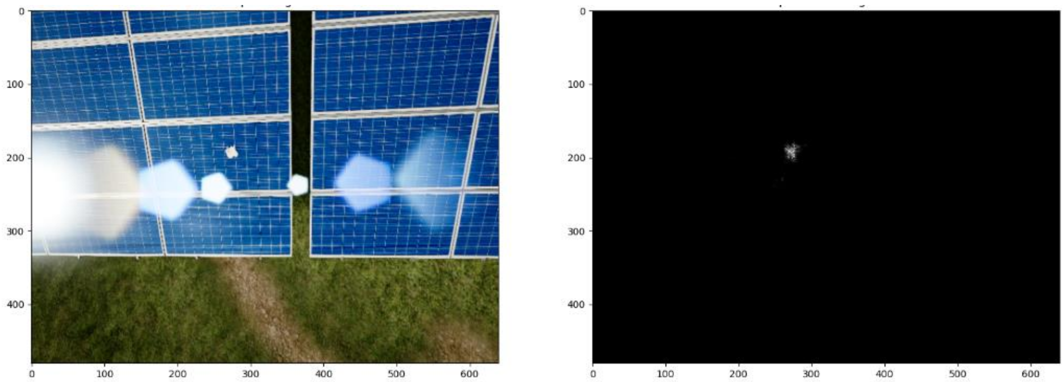


Fig. 15. Segmentation of bird droppings on **a** faulty PV panel. It can be seen that the assessed model works reasonably well in differentiating between sunlight reflections and bird droppings on PV panels.

4. Conclusion

In recent years, there has been a growing interest in deploying aerial robots and AI for the monitoring and management of PV plants. However, many studies continue to depend on real-world experiments, which often involve significant time and expense. Our research presents a DT-based platform designed for simulating, developing, and testing various monitoring models in the context of AAM of PV plants.

To accomplish this, we first created a virtual model of a utility-scale PV plant within UE. Following this, we implemented aerial robot flight simulations using AirSim, expanding its Python APIs to allow testing of different monitoring models in specified scenarios.

We collected a dataset of aerial images showing defective panels within this simulation environment and trained an end-to-end model to detect bird droppings on the PV panels. Furthermore, the developed environment enabled us to evaluate the performance of diverse monitoring models employed in the AAM of PV plants. Models such as plant boundary extraction, path planning algorithms, and fault detection approaches were assessed, yielding valuable insights into their capabilities and potential effectiveness in real-world applications.

Looking ahead, we propose several recommendations for future studies in this field. The platform facilitates the integration of various simulation and analytical tools into a cohesive DT, presenting opportunities to enhance functionality by combining different software applications and leveraging the unique strengths of each. For example, incorporating analytical tools such as MATLAB/Simulink or energy simulation and management applications into the DT framework, along with creating a customizable dashboard for monitoring key performance indicators (KPIs) and receiving alerts about potential issues, could significantly boost the Digital-PV's simulation and analytical capabilities. In addition, establishing a library of virtual PV plants globally, along with models of various fault types, would also be particularly advantageous.

On the other hand, enhancing the platform's simulation capabilities by developing APIs and utilizing AirSim’s features could allow for cooperative flight operations and the monitoring of PV power plants. This improvement would support the simulation of multi-agent systems (MAS) algorithms within Digital-PV, enabling researchers to explore and create swarm intelligence algorithms for inspection and monitoring tasks.

However, challenges remain in expanding the Digital-PV platform, including integration complexity, scalability, simulation accuracy, and user engagement. Addressing these issues through a modular architecture, optimized algorithms, and user-friendly interfaces complete with training resources will be essential for future success.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] IRENA, Future of Solar Photovoltaic: Deployment, Investment, Technology, Grid Integration and Socio-Economic Aspects (A Global Energy Transformation: Paper), Abu Dhabi, (2019).
[2] PV Magazine, BloombergNEF Says Global Solar Will Cross 200 GW Mark for First Time This Year, Expects Lower Panel Prices.
[3] M. Aghaei, M. Kolahi, A. Nedaei, N.V. Sridharan, A. Eskandari, A.K.V. De Oliveira, V. Sugumaran, R. Ruther, P. Parvin, S.M. Esmailifar, A Holistic Study on Failures and Diagnosis Techniques in Photovoltaic Modules, Components and Systems, in Proceedings of the 2023 International Conference on Future Energy Solutions (FES), IEEE, (2023), pp. 1–6.
[4] I.U. Khalil, A. Ul-Haq, Y. Mahmoud, M. Jalal, M. Aamir, M.U. Ahsan, K. Mehmood, Comparative Analysis of Photovoltaic Faults and Performance Evaluation of Its Detection Techniques, IEEE Access, 8, (2020), 26676–26700, doi:10.1109/ACCESS.2020.2970531.
[5] A.Y. Appiah, X. Zhang, B.B.K. Ayawli, F. Kyeremeh, Review and Performance Evaluation of Photovoltaic Array Fault Detection and Diagnosis Techniques, International Journal of Photoenergy, (2019).
[6] A. Mellit, G.M. Tina, S.A. Kalogirou, Fault Detection and Diagnosis Methods for Photovoltaic Systems: A Review, Renewable and Sustainable Energy Reviews, 91, (2018), 1–17, doi:10.1016/j.rser.2018.03.062.
[7] P. Nooralishahi, C. Ibarra-Castanedo, S. Deane, F. López, S. Pant, M. Genest, N.P. Avdelidis, X.P. Maldague, Drone-Based Non-Destructive Inspection of Industrial Sites: A Review and Case Studies, Drones, 5, (2021) ,106.

[8] C.-F. Jeffrey Kuo, S.-H. Chen, C.-Y. Huang, Automatic Detection, Classification and Localization of Defects in Large Photovoltaic Plants Using Unmanned Aerial Vehicles (UAV) Based Infrared (IR) and RGB Imaging, Energy Conversion and Management, 276, (2023), 116495.
[9] S.A. Rahaman, T. Urmee, D.A. Parlevliet, PV System Defects Identification Using Remotely Piloted Aircraft (RPA) Based Infrared (IR) Imaging: A Review, Solar Energy, 206, (2020), 579–595.
[10] N.V. Sridharan, V. Sugumaran, Deep Learning-Based Ensemble Model for Classification of Photovoltaic Module Visual Faults, Energy Sources, Part A: Recovery, Utilization and Environmental Effects, 44, (2022), 5287–5302, doi:10.1080/15567036.2022.2083729.
[11] L. Bommes, T. Pickel, C. Buerhop-Lutz, J. Hauch, C. Brabec, I.M.I.M. Peters, Computer Vision Tool for Detection, Mapping, and Fault Classification of Photovoltaics Modules in Aerial IR Videos, Progress in Photovoltaics: Research and Applications, 29, (2021), 1236–1251, doi:10.1002/pip.3448.
[12] D. Manno, G. Cipriani, G. Ciulla, V. Di Dio, S. Guarino, V. Lo Brano, Deep Learning Strategies for Automatic Fault Diagnosis in Photovoltaic Systems by Thermographic Images, Energy Conversion and Management, 241, (2021), doi:10.1016/j.enconman.2021.114315.
[13] W. Tang, Q. Yang, W. Yan, Deep Learning-Based Algorithm for Multi-Type Defects Detection in Solar Cells with Aerial EL Images for Photovoltaic Plants, CMES - Computer Modeling in Engineering and Sciences, 130, (2022), doi:10.32604/cmes.2022.018313.
[14] A.G. Imenes, N.S. Noori, O. Andreas, N. Uthaug, R. Kröni, F. Bianchi, A Deep Learning Approach for Automated Fault Detection on Solar Modules Using Image Composites, in Proceedings of the IEEE 48th Photovoltaic Specialists Conference (PVSC), (2021), pp. 1925–1930.
[15] A.M. Sizkouhi, M. Aghaei, S.M. Esmailifar, A Deep Convolutional Encoder-Decoder Architecture for Autonomous Fault Detection of PV Plants Using Multi-Copters, Solar Energy, 223, (2021), 217–228.
[16] E. Pedersen, S. Rao, S. Katoch, K. Jaskie, A. Spanias, C. Tepedelenlioglu, E. Kyriakides, PV Array Fault Detection Using Radial Basis Networks, in Proceedings of the 10th International Conference on Information, Intelligence, Systems and Applications, (2019).
[17] S.N. Venkatesh, V. Sugumaran, Machine Vision Based Fault Diagnosis of Photovoltaic Modules Using Lazy Learning Approach, Measurement, 191, (2022), 110786.
[18] N.V. Sridharan, V. Sugumaran, Visual Fault Detection in Photovoltaic Modules Using Decision Tree Algorithms with Deep Learning Features, Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, (2021), 1–17.
[19] E. Salahat, C.-A. Asselineau, J. Coventry, R. Mahony, Waypoint Planning for Autonomous Aerial Inspection of Large-Scale Solar Farms, in Proceedings of the IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 1, (2019), 763–769.
[20] Y. Ding, R. Cao, S. Liang, F. Qi, Density-Based Optimal UAV Path Planning for Photovoltaic Farm Inspection in Complex Topography, IEEE Xplore, (2020).
[21] X. Luo, X. Li, Q. Yang, F. Wu, Optimal Path Planning for UAV Based Inspection System of Large-Scale Photovoltaic Farm, IEEE Xplore, (2017).
[22] A. Pérez-González, Á. Jaramillo-Duque, J.B. Cano-Quintero, Automatic Boundary Extraction for Photovoltaic Plants Using the Deep Learning U-Net Model, Applied Sciences, 11, (2021), 6524.
[23] A. Pérez-González, N. Benítez-Montoya, Á. Jaramillo-Duque, J.B. Cano-Quintero, Coverage Path Planning with Semantic Segmentation for UAV in PV Plants, Applied Sciences, 11, (2021), 12093.
[24] A.M. Moradi Sizkouhi, S.M. Esmailifar, M. Aghaei, M. Karimkhani, RoboPV: An Integrated Software Package for Autonomous Aerial Monitoring of Large Scale PV Plants, Energy Conversion and Management, 254, (2022), 115217, doi:10.1016/j.enconman.2022.115217.
[25] A.M.M. Sizkouhi, S.M. Esmailifar, M. Aghaei, A.K.V. De Oliveira, R. Rüther, Autonomous Path Planning by Unmanned Aerial Vehicle (UAV) for Precise Monitoring of Large-Scale PV Plants, in Proceedings of the 2019 IEEE 46th Photovoltaic Specialists Conference (PVSC), (2019), pp. 1398–1402.
[26] A.K.V. de Oliveira, M. Aghaei, R. Rüther, Automatic Inspection of Photovoltaic Power Plants Using Aerial Infrared Thermography: A Review, Energies, 15, (2022), 2055.
[27] B. Li, C. Delpha, D. Diallo, A. Migan-Dubois, Application of Artificial Neural Networks to Photovoltaic Fault Detection and Diagnosis: A Review, Renewable and Sustainable Energy Reviews, 138, (2021), 110512.
[28] M. Aghaei, M. Kolahi, S.M. Esmailifar, A. Sizkouhi, A. Nedaei, M. Manni, A. Eskandari, G. Lobaccaro, Autonomous Intelligent Monitoring of Photovoltaic Systems: An In-Depth Multidisciplinary Review, Progress in Photovoltaics: Research and Applications, (2024).
[29] C. Ghenai, L.A. Husein, M. Al Nahlawi, A.K. Hamid, M. Bettayeb, Recent Trends of Digital Twin Technologies in the Energy Sector: A Comprehensive Review, Sustainable Energy Technologies and Assessments, 54, (2022), 102837.
[30] M. Aghaei, M. Kolahi, S.M. Esmailifar, A. Sizkouhi, A. Nedaei, M. Manni, A. Eskandari, G. Lobaccaro, Digital Twin Technology in Solar Energy, in Digital Twin Technology for the Energy Sector, Elsevier, (2025), pp. 191–212.
[31] M. Kolahi, S.M. Esmailifar, A.M.M. Sizkouhi, M. Aghaei, Digital-PV: A Digital Twin-Based Platform for Autonomous Aerial Monitoring of Large-Scale Photovoltaic Power Plants, Energy Conversion and Management, 321, (2024), 118963.
[32] A.M.M. Sizkouhi, M. Aghaei, S.M. Esmailifar, M.R. Mohammadi, F. Grimaccia, Automatic Boundary Extraction of Large-Scale Photovoltaic Plants Using a Fully Convolutional Network on Aerial Imagery, IEEE Journal of Photovoltaics, 10, (2020), 1061–1067.