Edge-platforms based decision-support approach for solar panels inspection with YOLOv8 deep neural-network

Hajar El Karch[1], Abdelkader Mezouari[2], Youssef Natij[2], Rachid El Gouri[1]
[1]Laboratory of Advanced Systems Engineering (LASE), National School of Applied Sciences (NSAS), University Ibn Tofail (UIT), Kenitra, Morocco
[2]Laboratory of Electronic Systems, Information Processing, Mechanics and Energetics (LESIME), Faculty of Sciences, University Ibn Tofail (UIT), Kenitra, Morocco

Article Info

Article history:

Received Feb 18, 2024
Revised Apr 8, 2024
Accepted Apr 13, 2024

Keywords:

Deep learning
Embedded edge platforms
Image processing
PV panels cleaning
Soiling detection
YOLOv8 network

ABSTRACT

This paper presents an innovative AI-based method for autonomous inspection, designed to enhance energy production efficiency by optimizing cleaning strategies for soiled photovoltaic panels, using advanced artificial intelligence algorithms to analyze panel conditions and environmental factors in real-time, allowing for targeted cleaning interventions. Based on the advanced YOLOv8 deep learning algorithm and computer vision approach, the proposed method offers distinct advantages in real-time detection and classification of various types of soiling and dust accumulation compared on solar panels to traditional methods, and underwent satisfactory testing across diverse scenarios. The NVIDIA Jetson Nano, the Raspberry Pi4 embedded devices, and the Raspberry Pi4 combined with NCS2 accelerator are used for implementing our approach. A comparison aims to provide a detailed exploration of the most suitable embedded platform for deploying our advanced system was discussed. This comparison considers processing speed and accuracy, energy consumption, and overall performance in executing the computationally intensive tasks. The results demonstrate that our model achieves high accuracy in detecting soiling and enhancing the model's detection speed. With an average precision of 99.5%, this approach ensures accurate fault identification, underscoring the effectiveness of computer vision using deep learning algorithms for detection tasks across a wide range of scenarios.

Corresponding Author:

Hajar EL Karch
Laboratory of Advanced Systems Engineering (LASE), National School of Applied Sciences (NSAS)
University Ibn Tofail (UIT)
Kenitra, Morocco
Email: Hajar.elkarch@uit.ac.ma

1. INTRODUCTION

In recent years, solar power has propelled the forefront of sustainable energy solutions. Therefore, research and development efforts must continue tracking the latest technologies [1]. Photovoltaic (PV) systems are one of the most promising technologies for exploiting the sun's free and unlimited energy. However, the performance of solar panels is susceptible and depends on many environmental factors, with the accumulation of soiling, dust, and other contaminants being a significant challenge [2]. As photovoltaic energy sources play an increasingly central role in mitigating environmental impact and meeting global energy demand, the efficiency and reliability of PV panels are of paramount importance.

Many studies [3]–[5] have focused on studying the impact of dust and soiling on solar PV panels and underscore that the significant impact of soiling on PV panel performance in solar power plants can not only affect the absorption of solar radiation but also causing performance reduction. Those studies have underscored the critical need for cleaning interventions to combat dust and soiling in PV installations to maintain optimal energy production. The importance of this inspection of photovoltaic panels lies in the fact that even small accumulations of soiling, dust, bird dropping, or other debris can significantly reduce the energy yield of photovoltaic panels. Traditional inspection and cleaning methods are often inefficiencies and time and resource-consuming, leading to sub-optimal performance and operational costs increasing.

A common traditional method involves periodic visual inspections of PV panels by technicians. These inspections often require physical access to the panels, especially for large-scale installations. Once soiling and deposits are detected, cleaning efforts may involve manual methods such as water spraying, wiping, or even brushing. Alternatively, automated cleaning systems, such as robotic cleaners or water-based cleaning systems, may be deployed based on predefined schedules or triggered by sensor data. However, scheduled cleaning routines based on traditional methods may result in unnecessary cleaning cycles, wasting resources, and increasing operational costs. The goal of our proposed method for a specific cleaning strategy is to optimize the energy, time, and cost of the operation by tailoring the cleaning process to the type of soiling encountered in the case of a soiled PV panel. The system aims to efficiently and effectively address different types of soiling.

While earlier studies have explored the application of deep learning algorithms in the realm of PV panel inspection and maintenance [6], [7]. One such study, conducted by [8], introduces a method based on convolutional neural networks (CNNs) to evaluate power loss in PV modules attributed to soiling, dust accumulation, and partial shading. By leveraging CNNs and image analysis techniques, the algorithm effectively estimates power loss for individual modules, accounting for the specific impacts of partial shading and soiling. Another study, undertaken by Tan *et al.* [9], focuses on providing more precise quantitative assessments of dust accumulation levels using denoising convolutional neural networks (DnCNN). However, these studies have notable limitations. While they successfully identify overall power loss, they lack the capability to differentiate between various types of soiling and determine the extent of soiling present, crucial factors for determining appropriate cleaning strategies.

In comparison, our research introduces a novel real-time methodology that integrates YOLOv8 deep learning-based object detection methods with image-processing techniques aiming to enhance detection accuracy and efficiency and provide efficient cleaning intervention Through synthesizing previous studies, we aim to contextualize our research within the broader landscape of solar panel dust and soiling detection for optimal cleaning methods proposition. The importance of deep learning lies in its ability to improve the efficiency of solar panel inspection, addressing the limitations of traditional manual cleaning methods. our models can be trained on various datasets to recognize patterns associated with different types and levels of soiling on solar panels. Automated detection identifies soiling promptly, ensuring that cleaning is carried out, when necessary, rather than according to a predetermined schedule. Our new method based on the YOLOv8 algorithm enables precise localization of soiling areas on solar panels and minimizes unnecessary cleaning, reducing water consumption and environmental impact.

The paper is structured into several sections to provide a comprehensive understanding of our research methodology and results. Firstly, the background study delves into existing cleaning methods and deployment systems for object detection systems, setting the stage for our approach. Following this, the research method section elaborates on our innovative methodology, based on the YOLOv8 neural network and image processing techniques for dust detection, along with insights into the training dataset crucial for accurate soiling classification and processing systems architecture used for implementation and deployment process are also discussed. Section 3 presents a discussion of the results showcasing the practicality and efficacy of our approach. The implementation system and evaluation performance of our model are also analyzed. Finally, the conclusion section synthesizes our research results and highlights future research aimed at developing an autonomous inspection robot incorporating our Real-time soiling detection and smart cleaning interventions system of photovoltaic modules.

2. BACKGROUND STUDY

The performance and electrical efficiency of PV systems are significantly affected by dust and soiling accumulation on their surfaces. This not only reduces the solar radiation reaching the PV surface but also leads to adhesion, corrosion, and a reduction in the overall lifespan of the solar panels. Therefore, regular cleaning of these cells is crucial. Researchers have explored various methods to clean and mitigate soiling and dust from PV installations [7].

2.1. Cleaning methods

This section delves into the current state of cleaning methods, and associated challenges, and discusses the issue of soiling, and dust on PV panels. An interesting paper by Cavieres *et al*. [8] with the overall objective of helping to improve the performance of solar photovoltaic systems, addresses the challenges associated with power generation from PV solar systems, emphasizing the low efficiency and the impact of environmental and installation factors. The paper reviews existing research on the causes and consequences of dust deposition on PV module surfaces, considering electrical, thermal, and optical characteristics. This research work summarizes current knowledge on dust mitigation techniques and provides a synthesis of past and present studies aiming to assist in the selection of suitable methods for dust cleaning.

A recent study by Gupta *et al.* [10], discusses the key points related to the exploration of cleaning technologies and various methods employed in self-cleaning. Moreover, it provides a comparative analysis of different cleaning technologies, including those incorporating self-cleaning approaches. Various cleaning strategies and methods have been discussed and compared [7], [10], [11]. Optimal cleaning methods vary based on parameters like PV dust composition, soiling types, system size, design, location, and water availability. The selection of a suitable cleaning method depends on multiple parameters, as many studies can demonstrate. Researchers in the paper [10] introduce a new cleaning methodology, considering economic aspects, strategy, and frequency. A comprehensive critical review by researchers [11] presents a recent investigation into the impact and effect of dust on photovoltaic systems and the advantages and disadvantages of different existing cleaning methods. The study thoroughly assesses the dust problem and the current cleaning methods, addressing associated challenges and prospects. The review highlights the most critical researchers' challenges to improve cleaning strategies and methods. One of the challenges is improving cleaning methods for PV systems, focusing on both technical and economic aspects, optimizing water consumption methods, and minimizing the power required for water pumping. Developing a censoring system that seamlessly integrates with an automatic cleaning robot requires a sophisticated combination of sensors to detect dust levels, assess efficiency losses, and initiate the cleaning process. In addition to technical considerations, the economic aspect of this challenge extends to developing efficient and automatic cleaning robots that can adapt to various PV technologies and environmental conditions. A comparison and classification of cleaning methods was conducted by Gupta *et al*. [10], this comparative analysis shows that some cleaning methods are manual, automatic, or preventive. The case study and many parameters need to be examined before the final choice of the cleaning method.

In this work, an AI-based support decision system is discussed for optimal cleaning strategies. The proposed system uses the improved YOLOv8 deep-learning network and a computer vision approach to assess the cleanliness status of solar panels and evaluate various soiling accumulation patterns to suggest the most effective and efficient methods for optimal cleaning. This optimization leads to increased PV energy efficiency, reduces maintenance costs, and improves the performance of energy systems.

2.2. Deployment systems for object detection

Embedded systems are being increasingly used for implementing deep learning architectures due to their high performance and computational capacity to act and process complex neural models in real-time without compromising efficiency and introducing significant latency. The integration of deep learning techniques into embedded systems, including platforms such as Jetson Nano, TPU, VPU, and Raspberry Pi, opens up a field of possibilities for advancing real-time processing capabilities on the device. Younis and Onsa *et al.* [12] in a previous study propose the development of a comprehensive environment facilitating the integration of deep learning algorithms into various embedded systems, coupled with a detailed performance analysis. Results from the experiments indicate that the proposed development environment successfully executes DL algorithms, demonstrating adaptability and update ability of libraries for more complex networks. The study suggests improving techniques for optimizing resource utilization in embedded systems for reduced latency and energy consumption and enhancing hardware design in embedded systems to execute DL networks effectively, including accelerators for computer vision tasks.

A recent benchmark analysis by [13] aims to analyze the performance of embedded system boards across different datasets in a CNN algorithm, with the ultimate goal of achieving high accuracy with minimal hardware requirements in deep learning applications. The study discussed the performance parameters including accuracy, GPU, CPU, and RAM consumption, and cost of NVIDIA Jetson Nano, NVIDIA Jetson TX_2, and Raspberry PI4 using a 2D CNN algorithm designed for classifying 13 fashion products with a dataset of 45K images. The Jetson TX_2 exhibited higher resource and power consumption due to its superior hardware capabilities, when considering the model's classification time, Jetson Nano and Jetson TX_2 were faster over GPU and increased with larger datasets. Raspberry PI results in considerably longer times via CPU classification [13].

Other research papers [14], [15] assess the performance of the Jetson Nano within a Dew Computing framework, employing a machine learning application to address real-time identification and counting of land vehicles in Quito, Ecuador. The study by Suzen *et al.* [14] includes an experiment that focuses on evaluating processing resources (CPU-GPU) and gauging the effectiveness of the platform when integrated with OpenDataCam for the specified application. The Second study by Valladares *et-al.* [15] introduces an edge AI approach for object detection using convolutional networks, specifically SSD MobileNet and SSD Inception V3, on the Jetson Nano embedded GPU platform. Booth of the research [13] and [15], underscores the suitability of the NVIDIA Jetson as a low-power embedded computing device for accelerating deep learning applications. Notably, experimental results of [15] indicate that, among the considered models, the SSD Inception V3 achieves the highest accuracy more than the SSD MobileNet. This paper contributes to this domain by focusing on three different embedded systems to explore and compare their performances through experiments to choose the optimal embedded system. The study validates the competitiveness of this proposed system in terms of performance and the support of complex deep-learning algorithm demands.

3. RESEARCH METHOD

This section outlines the methodology used in our research for identifying and classifying soiling and dust on solar panel surfaces and performing cleaning decisions based on the results of this inspection process. The main steps of our proposed system from image acquisition, networking, and processing data collection to training and our model evaluation are discussed. Our research methodology relies on leveraging the YOLOv8 algorithm, which has shown exceptional performance in object detection tasks. This section aims to provide a comprehensive understanding of our approach. It includes a concise overview of the YOLOv8 algorithm and its significant improvements used for soiling detection on PV images, along with the image processing techniques performed for dust extraction. Additionally, a detailed exposition of our experimental procedures, covering various aspects such as dataset acquisition, resources utilized for training and deployment, systems architectures used for implementation, and the mechanism employed for experimental testing is presented in this section.

3.1. YOLOv8 network

In this work, a YOLOv8 deep learning network for real-time object detection was used for soiling detection on PV panels. The last iteration of the YOLO algorithm was introduced in 2023 [16], [17] to enhance accuracy while preserving real-time performance, building upon the speed and accuracy of its predecessors [18]. This version of YOLOv8 introduces significant enhancements and novel convolutions to its architecture. Figure 1 represents the YOLOv8 architecture and the main key improvement lies in the replacement of C3 with C2f, where the core of the system underwent alterations [19].

One key improvement lies in the replacement of C3 with C2f, where the core of the system underwent alterations. This involved changing the initial 6x6 convolution in the stem to a more compact 3×3 convolution. Moreover, unlike in C3, where only the output from the final Bottleneck was used, C2f integrates the outputs from the Bottleneck, comprising two 3x3 convolutions with residual connections. Additionally, YOLOv8 eliminates convolutions #10 and #14 from the YOLOv5 configuration. The Bottleneck architecture in YOLOv8 closely resembles that of YOLOv5, except for a crucial modification in the first convolution's kernel size, which was increased from 1x1 to 3x3, aligning with the ResNet block identified in 2015 [20], [21]. Furthermore, YOLOv8 introduces anchor-free detection, wherein the model predicts the object's center directly instead of using predetermined anchor boxes, enhancing adaptability and effectiveness. This approach replaces the need for manually selecting anchor boxes, which can lead to suboptimal results, especially in earlier YOLO models. Instead, YOLOv8 utilizes a predefined set of boxes with fixed heights and widths, tiled across the image during detection, to locate object classes with the desired aspect ratio and scale. The network generates probabilities and properties for each tiled box, including background, Intersection over Union (IoU), and offsets, which are leveraged to adjust the anchor boxes. This approach allows for greater flexibility and efficiency in object detection tasks [22], [23].

3.2. OpenCV-based image processing

A comprehensive set of image processing was implemented to isolate regions affected by dust within the PV image. Image processing was facilitated using the computer vision library OpenCV [21]. The preprocessing phase involved various techniques: thresholding, binarization, edge detection, and morphological transformations as shown in Figure 2 aimed at enhancing image quality and achieving precise extraction of dust layers from the input image. The combination of these pre-processing techniques improves the quality of the input image, making it more suitable for the subsequent analysis stages and ultimately leading to more accurate extraction of the dust layers. Each technique played a distinct role in addressing different aspects of the image, collectively contributing to the effectiveness of the overall pre-processing process.

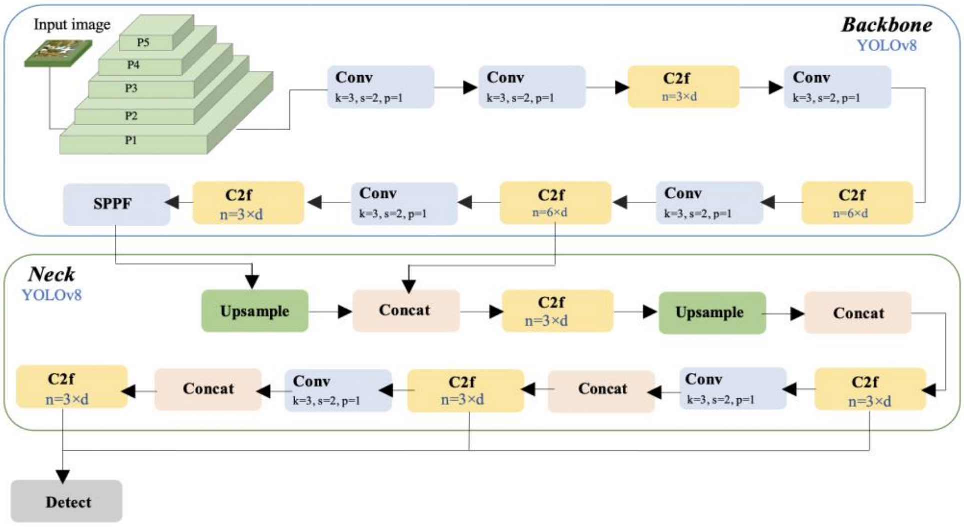


Figure 1. YOLOv8 architecture [19]

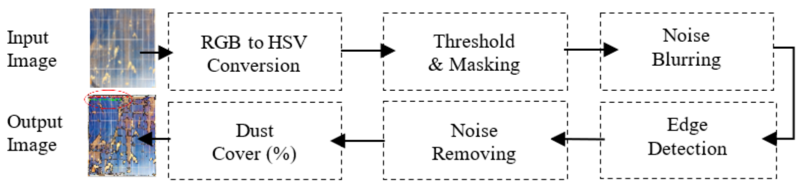


Figure 2. The Block diagram of image processing techniques performed with OpenCV

3.3. The proposed method

To provide a detailed overview of our system and the procedure used to figure out our approach throughout this study. Figure 3 presents our system architecture including a description of the main steps of the proposed system. The inspection process begins by searching for a QR code linked to each specific panel, even if the camera detects the QR code. The system proceeds to detect the edges of the related PV panel for eventual processing. This image is then transmitted to the main processing unit, the processing unit runs the processing algorithm based on the YOLOv8 network and image techniques using the Open Computer Vision library to identify soiling and dust deposits on the captured PV panel image.

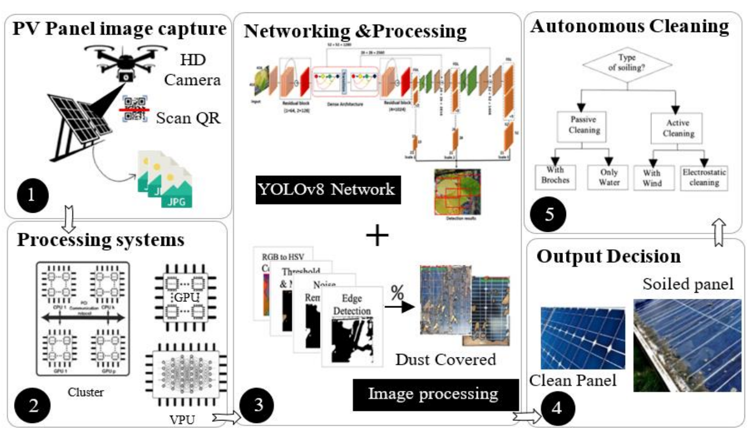


Figure 3. The proposed real-time soiling and detection system overview

The first part of the process involves using the YOLOv8 algorithm. This algorithm divides the input image into a grid and then proceeds to process each grid section individually. The algorithm begins by identifying regions of interest (ROIs) where potential soiling may be present. This is achieved through a CNN which analyzes these regions for signs of dirt or debris. Subsequently, bounding boxes are predicted around these identified areas using a regression algorithm. To refine the accuracy of the detection, a non-maximum suppression (NMS) algorithm is applied. This step prioritizes the bounding box with the highest confidence score, discarding any overlapping boxes that possess lower scores, according to a predefined threshold. This iterative process continues until all potential areas of soiling have been pinpointed and confirmed. Ultimately, this results in the comprehensive detection of soiled regions on the solar panel [24], [25].

The initial image undergoes a pre-processing stage in the second part of the process, this step focuses on the image to isolate the dust layer first, the image format is converted from RGB to HSV, potentially making dust pixels more distinct. Next, a threshold is applied based on specific HSV values to select pixels likely representing dust. To remove unwanted noise while keeping edges intact, a Gaussian filter is employed. Then, the Canny edge detection method identifies the boundaries of the dust region. Finally, morphological operations like erosion and dilation refine the dust area, eliminating small features along the border and preserving its overall shape and size. Lastly, the dust coverage percentage is calculated by dividing the number of non-zero pixels representing dust by the total number of pixels in the image.

Three embedded computing devices, Raspberry Pi 4, Raspberry Pi 4 combined with the edge accelerator device NCS2, and NVIDIA Jetson Nano were used for the implementation of our method. The results of the networking and processing step serve as a basis for making optimal decisions adapted to specified predefined threshold values, determining whether the PV panel is considered clean or not. However, the PV panel requires cleaning, so choose which type of cleaning intervention is deemed necessary. To ensure precise cleaning intervention and timing, the system incorporated algorithms and strategies based on a real-time data detection process to provide the appropriate and effective cleaning in removing specific types of soiling. This approach aimed to reduce water consumption and minimize energy losses and maintenance costs by managing the cleaning panels frequency to minimize both the inconvenience of excessive cleaning and the energy losses caused when panels are not cleaned.

The adaptive approach ensures that the cleaning process is adapted to the unique conditions of each situation, optimizing the effectiveness of the cleaning operation. If the panel is confirmed to be clean without significant soiling and dust accumulation, the system proceeds to move and search a new QR code and capture the next new PV image. However, if the dust and soiling percentage exceeds 20%, the YOLOv8 algorithm is performed to identify the type of soiling. Even if the soiling is detected the algorithm initiates a passive or active cleaning process depending on each identified soiling type. Many forms of soiling can impact the efficiency of PV panels as shown in Figure 4, tree leaves and branch dropping, Lichen deposits, and bird droppings are particularly recognized as one of the most severe types of soiling and Dust [15]. Our previous research has primarily concentrated on detecting this specific type of soiling using an earlier YOLOv5 version [20], [21]. In the current study, our aim lies in building a system specifically focused on the recognition of soiling caused by tree leaves dropping and dust detection.

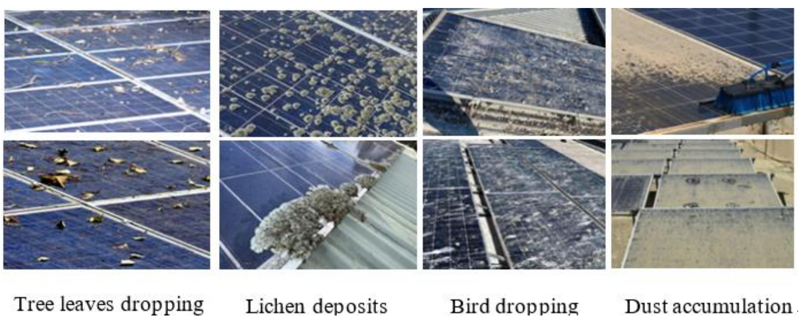


Figure 4. The proposed real-time soiling and detection system overview

To effectively address the issue of accumulated dust and other forms of soiling, our approach incorporates active cleaning measures tailored to the specific types of soiling encountered. Beyond the dust, instances of bird droppings, tree-dropped leaves, or branches can be identified and promptly addressed by our system. For bird droppings, a targeted water-cleaning method using brushes is recommended. This technique harnesses the combined power of water and mechanical scrubbing to effectively dislodge and remove the residue. The water acts to soften and dissolve the droppings, while the brushes ensure thorough removal, restoring optimal functionality to the panels. In contrast, when dealing with tree-dropped leaves and branches, a simpler yet equally effective strategy is employed. Using controlled airflow, the system facilitates the gentle dislodging and removal of debris from the solar panels. This natural cleaning process harnesses the power of wind to efficiently clear away leaves and branches without the need for manual intervention. Illustrated in Figure 5, our system's flow diagram delineates these sequential steps, ensuring a comprehensive and systematic approach to maintaining the cleanliness and efficiency of solar panel arrays.

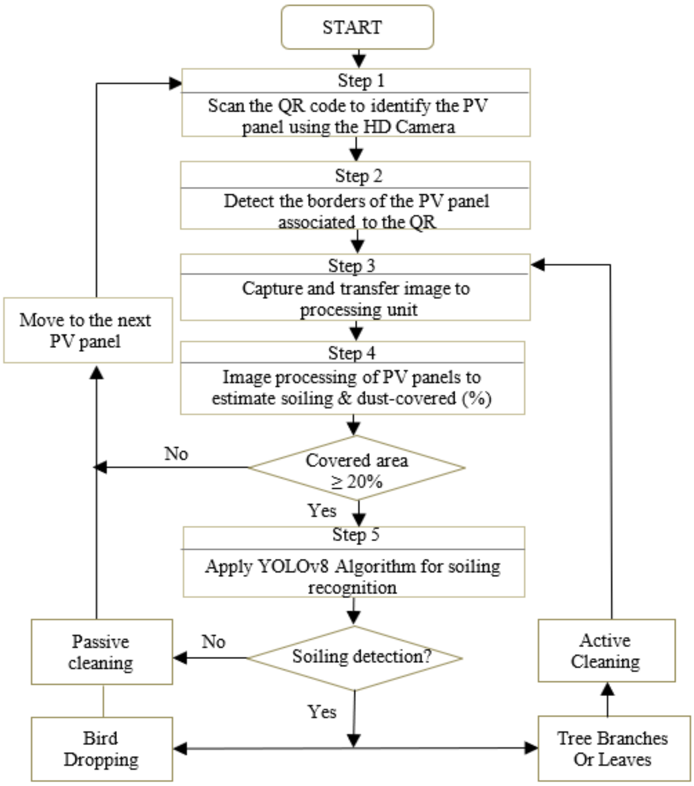


Figure 5. Flow diagram of the proposed system

3.4. Dataset and Resources for training and deployment

The workflow illustrated in Figure 6, outlines the proposed approach, offering a comprehensive overview of the entire process which involves data preparation, training YOLOv8 custom dataset, and resources for deployment stages. The process initiates with the preparation of the dataset which is a crucial step in machine learning, ensuring that the model in trained on relevant and diverse data. By meticulously selecting and preprocessing the dataset, the model can be trained to generalize effectively across various scenarios. The proposed approach offers a systematic and structured methodology for developing and deploying machine learning models.

In the initial data collection phase Figure 6(a), the objective is to collect a vast dataset encompassing various types of soiling that impact the efficiency and performance of solar PVs such as bird-dropping, lichen, and tree-falling leaves. These labels can be generated through either automated labeling tools or manual annotation processes. The dataset then was split, into training and validation subsets, allocating 80% of the data to the training set for model training and the remaining 20% to the validation set for performance validation. The majority allocation of the dataset for the training set is a strategic decision to provide the model with ample data for learning and achieving better performance when applied to real-world scenarios. Our YOLOv8 model is trained using the training set, which constitutes 80% of the dataset. Throughout this training phase, the model gains insights and refines its performance by learning from annotated images and optimizing its capabilities. The annotation was done by the Roboflow tool and exported directly in YOLOv8 annotation format. Following dataset preparation, the YOLOv8 model is trained using a custom dataset in Figure 6(b). This step involves learning the model used advanced algorithms and techniques to recognize and locate different types of soiling within PV panel images. This phase encompasses tasks such as feature engineering, model selection, and hyperparameter tuning to optimize the model's performance. Rigorous testing and validation procedures are also conducted to assess the model's accuracy and robustness. After successful training, the optimized model is deployed into three different processing systems in Figure 6(c) to extract and compare the performance of each architecture and choose the high-performance embedded device for our system. More detailed exploration of the dataset preparation and deployment phases. are discussed in the next subsection.

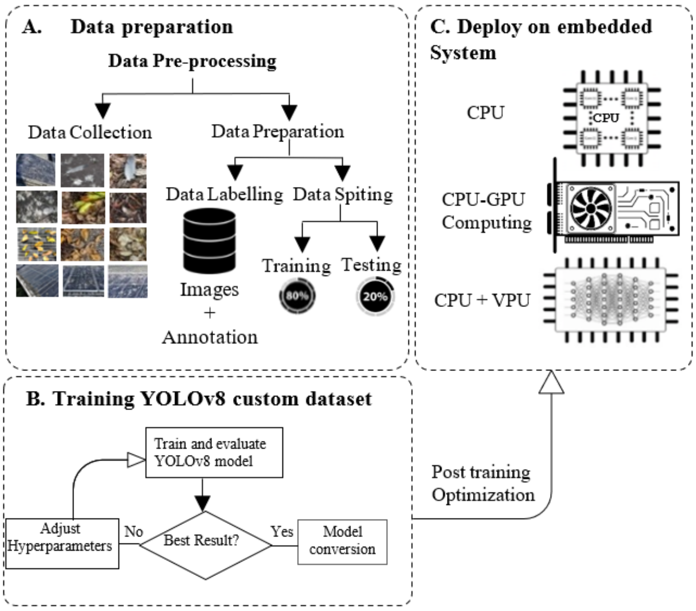


Figure 6. A general overview of the model training: (a) describes the preparation and collection of the dataset, (b) illustrate the training process to obtain the base models, and (c) present the deployment on the Jetson nano device

3.4.1. Dataset

A dataset compromising images of tree leaves was assembled for the real-time experiment application. The dataset includes a diverse sample of high-resolution images from real-life scenarios as well as sourced from Google images. The dataset comprises various leaf species encompassing tree leaves in different seasons and colors. By incorporating a wide range of scenarios, we aimed to create a robust dataset that reflects the natural variability and types of falling tree leaves fall. The platform Google Colab was used for training our dataset, leveraging a free GPU allowing for faster convergence and improved model performance. Over 5,000 images were used, and an augmentation process was applied to extend the dataset. These images were then reshaped and resized to meet the YOLO resolution which served as the input size for the pretrained model. Following this pre-processing step, the dataset underwent manual labeling to annotate the tree leaves in each image accurately. The dataset, comprising distinct classes, underwent meticulous annotation. Following annotation, each image was associated with an XML file specifying the bounding rectangles for specific classes. The dataset was partitioned for the model learning process as follows: 70% of the data was allocated for training, 15% for validation, and the remaining 15% for testing. This partitioning strategy ensures a balanced distribution of data for training, validation, and testing phases, contributing to robust model performance across various scenarios.

3.4.2. Processing systems for implementation

This paper delves into various computing architectures used in the implementation of YOLOv8 and explores different image-processing techniques. The deployment process of the YOLOv8 network on embedded systems requires careful consideration of hardware constraints, computational resources, and optimization techniques to ensure efficient and real-time object detection performances. This section focuses on practical implementation aimed at exploiting the potential of YOLOv8 in different detection scenarios. A NVIDIA Jetson nano-embedded device, Raspberry Pi 4, and Raspberry with VPU accelerator were used for the implementation [26], [27]. Each of these edge devices features an architecture. The Jetson Nano features CPU-GPU computing, offers substantial computational power, it is adept at running neural networks in parallel. His parallel processing capability proves satisfactory in applications covering image classification, object detection, and segmentation [19]. NVIDIA Jeston Nano is a widely adopted edge device family with accelerators for machine learning inferences. The GPUs emerge as ideal edge devices for high computational tasks it prove to be advantageous, particularly in real-time object detection, tracking, and advanced image analysis scenarios [28], [29].

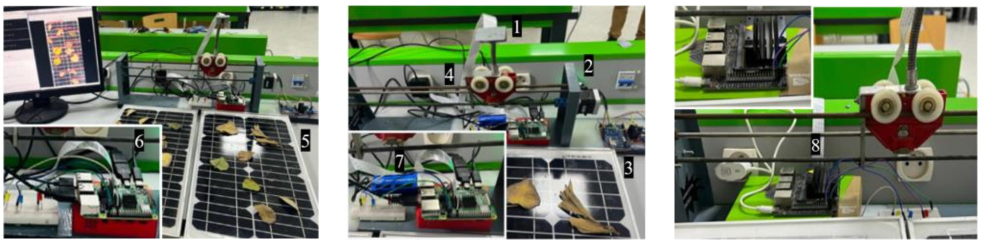
The Raspberry Pi 4B, leveraging the computational capabilities of the ARM architecture was used as a robust platform for real-time object detection and image analysis. A VPU accelerator Intel® neural compute stick 2 (NCS2) with a Raspberry Pi 4B enhanced the performance, allowing for accelerated visual data processing [27]. The combination of Raspberry Pi 4B and NCS 2 features a powerful processor and dedicated VPU. VPUs are optimized for parallel processing, and they can significantly speed up tasks that involve analyzing multiple visual elements simultaneously contributing to improved performance and efficiency in applications such as computer vision and image analysis [30].

The Raspberry Pi 4B, leveraging the computational capabilities of the ARM architecture was used as a robust platform for real-time object detection and image analysis. A VPU accelerator Intel® NCS2 with a Raspberry Pi 4B enhanced the performance, allowing for accelerated processing of visual data. The combination of Raspberry Pi 4B and NCS 2 features a powerful processor and dedicated VPU. VPUs are optimized for parallel processing, and they can significantly speed up tasks that involve analyzing multiple visual elements simultaneously contributing to improved performance and efficiency in applications such as computer vision and image analysis [30]. Figure 7 shows the experimental environment for implementation on the Raspberry Pi 4B platform Figure 7(a). The integration of the Raspberry Pi 4B with the NCS2 accelerator Figure 7(b). The implementation of Jetson Nano is shown in Figure 7(c).

To evaluate and validate our approach, a prototype has been developed. As shown in Figure 8 the prototype features a mechanism with an arm equipped with a CSI camera programmed to execute systematic movements for QR code scanning and image capturing associated with the solar panel. The camera movement is ensured by a stepper motor controlled by the output signal of an Arduino microcontroller.

Our exploration began with the base Raspberry Pi 4B 8 GB equipped with ARM Broadcom BCM2711 Cortex-A72 4-core processor running at 1.5GHz, 8 GB of RAM, and Debian 10 Buster 64-bit OS installed, establishing a resource-constrained benchmark. By harnessing the power of ONNX conversion, we unlocked significant speedups compared to native execution on the same platform. The integration of Intel's NCS2 delivered a leap in frame-per-second throughput, vividly demonstrating the benefits of offloading computations to dedicated AI hardware. To facilitate model execution on the Movidius NCS 2, OpenVINO toolkit, an open-source deep learning package. This toolkit features a high-level Python inference engine API, simplifying the deployment of pre-trained models on Intel hardware [30].

Finally, we ascended to the Jetson Nano Developer Kit 4GB equipped with ARM Cortex-A57 4-core processor running at 1.43 GHz, 4 GB of RAM, and Ubuntu 20.04 64bit OS installed, whose potent processor and GPU yielded the highest FPS (FPS)across all platforms except PC [26]. The operating system is deployed using the SDK Manager, a tool furnished by NVIDIA, which is based on Ubuntu. We have utilized version 32.6.1 for this purpose. To establish a comprehensive deep-learning library environment, we have installed CUDA, cuDnn, and TensorRT. This was achieved by installing the Jetpack package, specifically version 4.6. To squeeze even more performance from this beast, we delved into TensorRT, achieving further inference time reductions through model optimization and hardware-specific tuning.



(a) (b) (c)

Figure 7. Experimental environment for implementation on: (a) the Raspberry Pi 4B platform, (b) Raspberry Pi 4B with the NCS2 accelerator, and (c) on Jetson Nano

The main components of the mechanism are i) the CSI camera for image capturing, ii) the stepper motor that moves the camera on the axis; iii) the arduino microcontroller; iv) the axis for camera movement; v) PV panels, vi) the Raspberry Pi4; vii) the Intel NCS2; and viii) the NVIDIA Jetson Nano.

4. RESULTS AND DISCUSSION

Figure 8 displays the image processing results of extracting dust accumulation from PV panel images covered artificially with dust. Using a series of image-processing techniques based on the open computer vision library OpenCV-Python : Figure 8(a) HSV image, Figure 8(b) thresholded image for a range of dust colors, Figure 8(c) noise blurring, Figure 8(d) dust edge detection image, Figure 8(e) eroding, and Figure 8(f) dilating image. The results highlight a significant issue: a quarter (≈ 24%) of the photovoltaic (PV) panel surface is covered by dust, leading to a notable decrease in energy output, which underscores the necessity for effective cleaning intervention to reduce energy yield losses and optimize solar power generation efficiency. Addressing this issue through regular cleaning interventions is essential to maximize the efficiency and sustainability of solar power systems.

The proposed method's effectiveness in detecting soiling using a YOLOv8 framework was evaluated. Soiling was simulated manually by placing falling tree leaves on various photovoltaic panels. In Figure 9, images depict successful detection results achieved by YOLOv8 networks across more than three distinct scenarios, which highlights the effectiveness of our method in achieving accurate detection across a variety of scenes. The experimental results in Figure 9 underscore the reliability and effectiveness of the proposed method based on YOLOv8 with no missed detection.

The results show the YOLOv8 algorithm's capability to identify many classes of tree leaves on the photovoltaic panel images. The algorithm's accuracy in soiling detection showcased its proficiency, with a global detection rate surpassing 90%. We plan to enrich our training dataset to enhance further our model's capacity to identify a broader range of soiling types potentially impacting PV installation performance. This will involve integrating more diverse examples of soiling scenarios, including dust, bird droppings, and other environmental contaminants. By gathering and annotating additional data encompassing various forms of soiling, we aim to expose the model to a wider array of challenges and improve its detection capabilities.

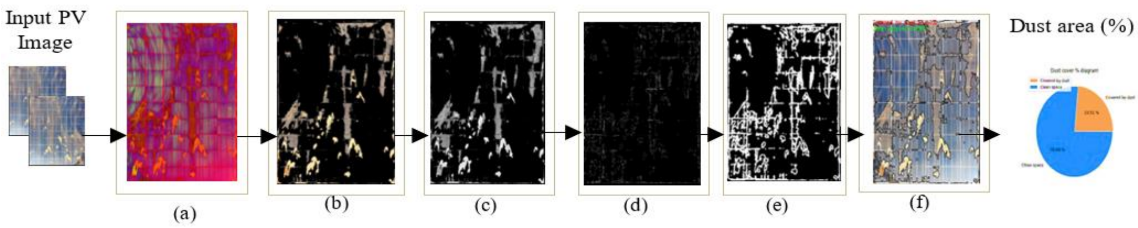


Figure 8. The block diagram of image processing techniques applied to the dusty panel: (a) HSV image, (b) thresholded image for a range of dust colors, (c) noise blurring, (d) dust edge detection image, (e) eroding, and (f) dilating image

Figure 9. YOLOv8 soiling detection results

Two performance metrics and evaluation indicators recall and precision [24], were discussed to assess the performance of the proposed YOLOv8 model, Table 1 shows the selected metrics for the trained model on our dataset. The table shows that the model achieved high precision (99.65%) and recall (99.49%) on its dataset. Precision refers to the proportion of true positives among the identified positives, while recall represents the proportion of true positives that were correctly identified. In simpler terms, the model accurately identified most of the soiling detected in the image. The mAP50 (mean average precision at intersection over union of 50%) is 23.48%, and the mAP50-95 (mean average precision at intersection over union between 50% and 95%) is even lower at 4.8%. These metrics indicate that while the model is good at identifying soiling, it struggles to locate it within the image precisely. In conclusion, the model shows promising results in terms of dust layer detection but needs improvement in precisely locating the dust regions within the image.

Table 1. Metrics of the proposed model

Model	Metrics of the model				
	Dataset	Precision	Recall	mAP50	mAP50-95
YOLOv8	Own	99.65%	99.49%	97.25%	23.48%

Table 2 presents the power consumption results for Raspberry Pi 4B and Jetson Nano computing platforms. The evaluation involves exploring diverse processing platforms to emphasize the strengths and limitations inherent in each system. The Raspberry Pi 4 has a lower power consumption than the Jetson Nano in both idle and execution states. As indicated in Table 3, it is evident that the Raspberry Pi 4 stands out as one.

The Jetson Nano showcases superior power capabilities even though with a considerably higher FPS count, the Jetson Nano becomes an excellent choice when processing speed is crucial, particularly for tasks like object recognition that require swift image processing. This is especially true for larger images, where the Jetson Nano excels. It is important to note that there is no singular best or ideal processing system; the selection depends on the main parameters and requirements of the system. In our case, where real-time processing is essential, we prioritize the average processing time, measured in milliseconds. The Jetson Nano, with its lower average processing time and higher FPS count, presents itself as an interesting option for constructing a high-performance power system. However, if working with smaller images and power consumption is not a critical factor, the Raspberry Pi with VPU remains the optimal choice.

A detailed analysis of the model classification over time indicates that the (Raspberry Pi 4 + VPU) configuration achieves faster results with lower power consumption, making it the preferred choice for our application system. This combination not only consumes less power but also boasts a shorter average processing time. In contrast, the Jetson Nano showcases superior power capabilities, albeit with higher consumption due to its advanced hardware features. A similar analysis of the model classification over time reiterates the efficiency of the (Raspberry Pi 4+VPU) configuration in terms of both speed and energy efficiency. The Jetson Nano emerges as the most powerful option for network processing, outperforming the Raspberry Pi 4B in power consumption and weight.

Table 2. Power consumption of the computing platforms

Raspberry PI 4B			Jetson Nano (ONNX Runtime)		
Idle			Execution		
Voltage(V)	Current (A)	Power (W)	Voltage (V)	Current (A)	Power (W)
5.4	0.35	1.89	5.4	0.66	3.56
5.4	0.78	4.2	5.4	1.9	10.2

Table 3. Technical specifications and results

Processing systems		Personnel computer	Raspberry Pi 4B	Raspberry Pi 4B + VPU	NVIDIA Jetson Nano
System features	CPU	Intel Core i7-8750H CPU with 12 cores	Quad-core Cortex-A72 (ARM v8) 1.5GHz	Intel Myriad X Vision Processing Unit	Quad-core ARM Cortex-A57
	GPU	Intel® Iris® Xe Graphics	GPU VideoCore VI		128-Core Maxwell GPU with CUDA Core
	Storage	2To SSD NVMe	64GB	64 GB + 4G free stockage space	16 GB eMMC 5.1 (Module) Not Include (Dev-Kit)
	Memory	16 GB	8GB RAM 64-bit	8GB+ 1GB of RAM	4 GB 64-bit LPDDR4
	Camera	12 MP, f/1.5, 26mm Dual-LED dual-tone Flash, HDR	2-Lane MIPI CSI-2 with DPHY 1.1 12.3 MP	2-Lane MIPI CSI-2 with DPHY 1.1 12.3 MP	2 lanes (3 × 4 or 4 × 2) MIPI CSI-2 DPHY 1.1 (1.5 Gbps)
Processing system consumption		180 W	3,56W	1,7W	10,2 W
System processing mAP /1frame		16 ms	42 ms	18 ms	12 ms
Frame per second		32	0,44	3	3,3

The outcomes of this study present favorable results, laying the groundwork for future research endeavors aimed at developing a robust robot for real-time inspecting photovoltaic modules. Furthermore, these findings have the potential to significantly enhance the accuracy and efficiency of real-time inspection conditions for PV panels. The YOLOv8 algorithm underwent testing on over 20 photovoltaic panels, exhibiting promising accuracy outcomes with a global soiling detection rate of 95%.

Leveraging YOLOv8 for real-time soiling detection on PV panels offers a promising solution with notable strengths in speed, accuracy, and scalability, particularly when integrated into high-performance systems like the Jetson Nano. YOLOv8 for real-time detection presents several strengths compared to other inspection techniques to detect multiple instances of soiling simultaneously enhancing its efficiency in large-scale PV panel arrays. By utilizing convolutional neural networks (CNNs) and advanced object detection techniques, YOLOv8 can efficiently scan images for the presence of soiling, accurately determining its location within the image. Additionally, through its classification capabilities, YOLOv8 can distinguish between different types of soiling, to provide the most effective cleaning strategies, and optimize energy production, for the related PV panels. However, careful consideration of its limitation lies in its reliance on large datasets for training, which may require significant computational resources and time for model development.

5. CONCLUSION

This study combines computer vision techniques with high-performance embedded edge platforms for dust and various types of soiling detection in photovoltaic (PV) panels. Traditional detection methods are often rigid, slow, and less accurate, prompting the need for more efficient and precise techniques. Experimental results demonstrate the effectiveness of the proposed approach in accurately detecting multiple classes of soiling present on solar panel surfaces. Through rigorous testing, it was observed that the proposed method, leveraging the YOLOv8 algorithm and image processing techniques, achieved high precision in detecting dust accumulation and soiling types of PV panels. The YOLOv8 model exhibited superior performance, with a mean average precision of 99.65% and a recall rate of 99.49%. This highlights YOLOv8 as the most advanced object detection system, offering a trade-off between accuracy and efficiency. Its effectiveness makes it well-suited for real-world applications in large-scale solar photovoltaic cell production.

The primary challenge faced by our system in this application revolves around ensuring a shorter average processing time. Towards the conclusion, the (Raspberry Pi 4 + Intel VPU) system stands out as a solution that not only consumes less power but also delivers superior performance, resulting in a reduced average processing time. On the other hand, the NVIDIA Jetson Nano demonstrates satisfactory performance with a higher frames per second count. However, during our exploration, certain limitations of the Jetson Nano board became apparent. The board poses challenges such as a complex environment setup, and it lacks support for some of the latest tool versions. Upon thoroughly comparing watts-FPS, it becomes evident that the Jetson Nano is the most optimal board model combination for executing YOLOv8. Notably, the YOLOv8 models displayed robust performance in terms of accuracy and achieved a satisfactory detection speed. In the realm of future research, several improvement avenues can be explored. One such avenue involves expanding the training image database to incorporate various types of soiling and defects, thereby enhancing the accuracy and efficiency of the detection and classification system. Additionally, there is an opportunity to identify a promising high-performance platform using Jetson AGX Orin tailored for embedded machine-learning systems. Our vision extends beyond the current capabilities of our system to develop an innovative cleaning robot to offer a versatile range of cleaning techniques.

REFERENCES

[1] T. Khatib and D. H. Muhsen, “Optimal sizing of standalone photovoltaic system using improved performance model and optimization algorithm,” *Sustainability (Switzerland)*, vol. 12, no. 6, p. 2233, Mar. 2020, doi: 10.3390/su12062233.

[2] T. Salamah *et al.*, “Effect of dust and methods of cleaning on the performance of solar PV module for different climate regions: Comprehensive review,” *Science of the Total Environment*, vol. 827, p. 154050, Jun. 2022, doi: 10.1016/j.scitotenv.2022.154050.

[3] A. Ullah, A. Amin, T. Haider, M. Saleem, and N. Z. Butt, “Investigation of soiling effects, dust chemistry and optimum cleaning schedule for PV modules in Lahore, Pakistan,” *Renewable Energy*, vol. 150, pp. 456–468, May 2020, doi: 10.1016/j.renene.2019.12.090.

[4] M. Memiche, C. Bouzian, A. Benzahia, and A. Moussi, “Effects of dust, soiling, aging, and weather conditions on photovoltaic system performances in a Saharan environment—Case study in Algeria,” *Global Energy Interconnection*, vol. 3, no. 1, pp. 60–67, Feb. 2020, doi: 10.1016/j.gloei.2020.03.004.

[5] H. Z. Al Garni, “The impact of soiling on PV module performance in Saudi Arabia,” *Energies*, vol. 15, no. 21, p. 8033, Oct. 2022, doi: 10.3390/en15218033.

[6] N. Khadka, A. Bista, B. Adhikari, A. Shrestha, D. Bista, and B. Adhikary, “Current practices of solar photovoltaic panel cleaning system and future prospects of machine learning implementation,” *IEEE Access*, vol. 8, pp. 135948–135962, 2020, doi: 10.1109/ACCESS.2020.3011553.

[7] M. Heinrich *et al.*, “Detection of cleaning interventions on photovoltaic modules with machine learning,” *Applied Energy*, vol. 263, p. 114642, Apr. 2020, doi: 10.1016/j.apenergy.2020.114642.

[8] R. Cavieres, R. Barraza, D. Estay, J. Bilbao, and P. Valdivia-Lefort, "Automatic soiling and partial shading assessment on PV modules through RGB images analysis," *Applied Energy*, vol. 306, p. 117964, Jan. 2022, doi: 10.1016/j.apenergy.2021.117964.
[9] Y. Tan, K. Liao, X. Bai, C. Deng, Z. Zhao, and B. Zhao, "Denoising convolutional neural networks based dust accumulation status evaluation of photovoltaic panel," in *Proceedings - IEEE International Conference on Energy Internet, ICEI 2019*, IEEE, May 2019, pp. 560–566. doi: 10.1109/ICEI.2019.00105.
[10] V. Gupta, M. Sharma, R. K. Pachauri, and K. N. D. Babu, "Comprehensive review on effect of dust on solar photovoltaic system and mitigation techniques," *Solar Energy*, vol. 191, pp. 596–622, Oct. 2019, doi: 10.1016/j.solener.2019.08.079.
[11] H. A. Kazem, M. T. Chaichan, A. H. A. Al-Waeli, and K. Sopian, "A review of dust accumulation and cleaning methods for solar photovoltaic systems," *Journal of Cleaner Production*, vol. 276, p. 123187, Dec. 2020, doi: 10.1016/j.jclepro.2020.123187.
[12] A. Younis and M. Onsa, "A brief summary of cleaning operations and their effect on the photovoltaic performance in Africa and the Middle East," *Energy Reports*, vol. 8, pp. 2334–2347, Nov. 2022, doi: 10.1016/j.egyr.2022.01.155.
[13] J. Gorospe, R. Mulero, O. Arbelaitz, J. Muguerza, and M. Á. Antón, "A generalization performance study using deep learning networks in embedded systems," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–29, Feb. 2021, doi: 10.3390/s21041031.
[14] A. A. Suzen, B. Duman, and B. Sen, "Benchmark analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," in *HORA 2020 - 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, IEEE, Jun. 2020, pp. 1–5. doi: 10.1109/HORA49412.2020.9152915.
[15] S. Valladares, M. Toscano, R. Tufiño, P. Morillo, and D. Vallejo-Huanga, "Performance evaluation of the Nvidia Jetson Nano Through a Real-Time Machine Learning Application," 2021, pp. 343–349. doi: 10.1007/978-3-030-68017-6_51.
[16] P. S. Mittapalli, M. R. N. Tagore, P. A. Reddy, G. B. Kande, and Y. M. Reddy, "Deep learning based real-time object detection on Jetson Nano embedded GPU," in *Lecture Notes in Electrical Engineering*, vol. 976 LNEE, 2023, pp. 511–521. doi: 10.1007/978-981-99-0412-9_46.
[17] E. Bisong, "Building machine learning and deep learning models on Google cloud platform," *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 2019, doi: 10.1007/978-1-4842-4470-8.
[18] C. H. Kang and S. Y. Kim, "Real-time object detection and segmentation technology: an analysis of the YOLO algorithm," *JMST Advances*, vol. 5, no. 2–3, pp. 69–76, Sep. 2023, doi: 10.1007/s42791-023-00049-7.
[19] R. H. Hasan, R. M. Hassoo, and I. S. Aboud, "Yolo versions architecture: review," *International Journal of Advances in Scientific Research and Engineering*, vol. 09, no. 11, pp. 73–92, 2023, doi: 10.31695/IJASRE.2023.9.11.7.
[20] E. karch Hajar, E. Rachid, B. Mohamed, A. Youssef, and M. Abdelkader, "Application of deep learning based detector YOLOv5 for soiling recognition in Photovoltaic Modules," in *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, IEEE, May 2023, pp. 1–6. doi: 10.1109/IRASET57153.2023.10152990.
[21] A. Khalfaoui, A. Badri, and I. El Mourabit, "An improved YOLOv5 for real-time human detection in infrared images," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 2, p. 1078, Nov. 2023, doi: 10.11591/ijeecs.v32.i2.pp1078-1085.
[22] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2021, doi: 10.1016/j.procs.2022.01.135.
[23] D. Nadia and C. Fathia, "Automatic detection of solar cell surface defects in electroluminescence images based on YOLOv8 algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 3, pp. 1392–1404, 2023, doi: 10.11591/IJEECS.V32.I3.PP1392-1404.
[24] N. Sharma, S. Baral, M. P. Paing, and R. Chawuthai, "Parking time violation tracking using YOLOv8 and tracking algorithms," *Sensors*, vol. 23, no. 13, p. 5843, Jun. 2023, doi: 10.3390/s23135843.
[25] Q. Li, Y. Shi, Q. Liu, and G. Liu, "Deep learning-based Pothole detection for intelligent transportation: A YOLOv5 approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 12, 2023, doi: 10.14569/ijacsa.2023.0141242.
[26] H. Ü. Dinkelbach, J. Vitay, F. Beuth, and F. H. Hamker, "Comparison of GPU- and CPU-implementations of mean-firing rate neural networks on parallel hardware," *Network: Computation in Neural Systems*, vol. 23, no. 4, pp. 212–236, 2012.
[27] C. Donegan, H. Yous, S. Sinha, and J. Byrne, "VPU specific CNNs through neural architecture search," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, Jan. 2021, pp. 9772–9779. doi: 10.1109/ICPR48806.2021.9412794.
[28] S. Mittal, "A survey on optimized implementation of deep learning models on the NVIDIA Jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428–442, Aug. 2019, doi: 10.1016/j.sysarc.2019.01.011.
[29] E. Assunção *et al.*, "Real-time weed control application using a jetson nano edge device and a spray mechanism," *Remote Sensing*, vol. 14, no. 17, p. 4217, Aug. 2022, doi: 10.3390/rs14174217.
[30] R. Dagli and S. Eken, "Deploying a smart queuing system on edge with Intel OpenVINO toolkit," *Soft Computing*, vol. 25, no. 15, pp. 10103–10115, Aug. 2021, doi: 10.1007/s00500-021-05891-2.

BIOGRAPHIES OF AUTHORS

Hajar El Karch is a Ph.D. student in Laboratory of Advanced Systems Engineering, National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco computer. She has a master's degree in embedded systems and microelectronic. Her research interests embedded systems, electronic instrumentation, renewable energy computer vision and artificial intelligence. She can be contacted at email: hajar.elkarch@uit.ac.ma.

Prof. Dr. Abdelkader Mezouari is a Professor in the High School of Technologies (HST) at Ibn Tofail University (ITU) in Kenitra, Morocco. His research interests in electrical engineering and renewable energy, the development of microelectronic energy management systems including Internet of things (IoT), computer vision and artificial intelligence. He authored or coauthored more than 30 publications in international journals and conferences. From 2016 to 2023. He can be contacted at email: abdelkader.mezouari1@uit.ac.ma.

Youssef Natij is a Ph.D. student in Computer Vision. His research centers on optimizing computer vision models for plant disease detection on low-computing devices, bringing powerful diagnostic tools to more accessible platforms. Youssef conducts his research at the Laboratory of Electrical Engineering and Energy Systems within the Faculty of Sciences at Ibn Tofail University in Kenitra, Morocco. He can be contacted at email: youssef.natij@uit.ac.ma.

Prof. Dr. Rachid El Gouri is a professor in the National School of Applied Sciences at Ibn Tofail University in Kenitra, Morocco. His research interests in electrical engineering and renewable energy include the modeling and design optimization of renewable energy systems, the development of microelectronic energy management systems and power electronic converters for renewable energy sources applications, and the development of sensors and electronic measurement systems and information security. He has more than 90 publications in international journals and conferences. He can be contacted at elgouri.rachid@yahoo.fr.