




# Edge Device for the Classification of Photovoltaic Faults Using Deep Neural Networks

André Biffe Di Renzo<sup>1,2</sup>  · Héber Renato Fadel de Morais<sup>1,3</sup> · André Eugenio Lazzaretti<sup>1</sup> · Lúcia Valéria Ramos de Arruda<sup>1</sup> · Heitor Silvério Lopes<sup>1</sup> · Cicero Martelli<sup>1</sup> · Jean Carlos Cardozo da Silva<sup>1</sup>

Received: 5 August 2023 / Revised: 25 June 2024 / Accepted: 19 July 2024  
© Brazilian Society for Automatics–SBA 2024

## Abstract

The use of photovoltaic panels for sustainable electricity generation is increasing worldwide. Hence, large solar power plants must be monitored to find defects quickly and easily, avoiding prolonged interruptions in electricity generation. The present study aims to analyse the incorporation of transfer learning in convolutional neural network models to classify defects in visible spectral images of solar panels. Deep learning with convolutional neural networks is known for their precise classification of images, but they need a significant volume of images and training time. Transfer learning is intended to help the training process become faster and more precise. In addition, a publicly available image dataset was constructed using 36,000 images containing three classes of defects and a class without defects to evaluate tested network models. In this study, 17 networks were tested as potential classification models. The best network exhibited an accuracy higher than 99%. This accuracy was obtained with the MobileNetV3 network, which was optimised with Nvidia Tensor RT to run on an edge device with low power consumption and low weight, enabling the real-time classification of the defects presented in this study and allowing the classification of an image in an average of 50 ms. This approach has yet to be explored in the literature, and this paper aims to contribute to this discussion. The presented work has the limitation of not making image segmentation, where the image obtained by the camera is directly classified. From experiments with a large dataset close to an in-field solar plant inspection, trained models successfully classified the defined classes. These findings help solar plant operation and maintenance teams make quick and accurate decisions about scheduled maintenance.

**Keywords** Photovoltaic panel · Failure classification · Convolutional neural networks · Embedded systems · Public image dataset

Héber Renato Fadel de Morais, André Eugenio Lazzaretti, Lúcia Valéria Ramos de Arruda, Heitor Silvério Lopes, Cicero Martelli and Jean Carlos Cardozo da Silva have equally contributed to this work.

✉ André Biffe Di Renzo  
andre.renzo@ifpr.edu.br

Héber Renato Fadel de Morais  
heber.morais@ifpr.edu.br

André Eugenio Lazzaretti  
lazzaretti@utfpr.edu.br

Lúcia Valéria Ramos de Arruda  
lvarruda@utfpr.edu.br

Heitor Silvério Lopes  
hslopes@utfpr.edu.br

Cicero Martelli  
cmartelli@utfpr.edu.br

Jean Carlos Cardozo da Silva  
jeanccs@utfpr.edu.br

## 1 Introduction

Following the growing energy demand, sustainable electricity generation is a recurring issue worldwide. In many countries, the primary electricity source is sustainable hydropower. However, they are adversely affected by droughts (Queiroz et al. 2019). Thus, other sustainable sources of electricity have been explored, such as photovoltaic (PV).

<sup>1</sup> CPGEI, Universidade Tecnológica Federal do Paraná - UTFPR, Av. Sete de Setembro 3165, Curitiba, Paraná 80230901, Brazil

<sup>2</sup> DSI/PROPLAN, Instituto Federal do Paraná - IFPR, Rua Emilio Bertolini 54, Curitiba, Paraná 82920030, Brazil

<sup>3</sup> Campus Jacarezinho, Instituto Federal do Paraná - IFPR, Avenida Doutor Tito, Jacarezinho, Paraná 86400000, Brazil

lectricity generation through photovoltaic panels ranks third in global production among sustainable sources (International Energy Agency 2022). Nonetheless, the higher the electricity demand, the larger the PV panel area required. Thus, operation and maintenance (O & M) costs also increase, which are critical in solar plants (Gallardo-Saavedra et al. 2018). Accordingly, new efficient inspection procedures must be employed to reduce costs, which are highly technical and economically essential (Victoria et al. 2021).

Computational algorithms are considered because many images can be captured during inspection tasks. Deep learning (DL) has been used for PV inspections, with images automating the inspection process (Yahya et al. 2022; Li et al. 2021). In DL, convolutional neural networks (CNNs) have been widely used to extract many highly discriminating features, achieving state-of-the-art results in object recognition and image classification (Alzubaidi et al. 2021).

In the literature, many research works developed their own CNNs to identify defects in PV using visual images (Li et al. 2021; Chen et al. 2020; Li et al. 2019; Espinosa et al. 2020; Cao et al. 2021). Their accuracy ranges from 70% to 98%, showing that different techniques can help increase accuracy.

However, building CNN models is not trivial. Several factors must be analysed to construct an efficient network regarding classification accuracy and the use of computational resources. To assist in this process, several authors consider using a previously known CNN model to address the problem (Li et al. 2021; Azis et al. 2022; Sizkouhi et al. 2022; Wang et al. 2022; Mehta et al. 2017).

Even with good results from using a known CNN, sometimes the used model requires many parameters, demanding considerable computational resources or leading to a model overfitting. The method known as transfer learning (TL) can solve this problem. It starts from models previously trained using a set of images (sometimes from other domains), and these models train only the last layers (as a classifier), maintaining the remaining trained network. Thus, the model adjusts to the new problem (Alzubaidi et al. 2021; Pan and Yang 2010; Ding et al. 2018; Venkatesh and Sugumaran 2021).

Furthermore, Li et al. (2020) used the edge device in an unmanned aerial vehicle (UAV) and developed algorithms to identify unknown defects through data augmentation (DA) and TL to train the network in identifying new defects. The proposed system's inference time was 62 ms, averaging 95% accuracy, whereas known CNN models presented an inference time from 660 ms to 1300 ms and an accuracy range from 59% to 85%.

The cited studies primarily used networks with many parameters, which in turn limit the application of edge devices and actual implementations of the models. Thus, this study will comprehensively compare several network models through TL to assess their performance in solving the prob-

lem under study. These comparisons aim to identify models not considered in the literature, which can facilitate the use of DL techniques to identify defects in PV panels in field environments, which will be the first contribution of this work. To use the TL in the networks chosen to be tested in this work, we used pre-trained networks with the ImageNet dataset, which is from another domain than photovoltaic panels. To make the TL, some layers of the CNN were unfrozen to train them, allowing this layer to learn about the specific problem presented in this work.

Another detail that hinders research and the practical implementation of methods for identifying defects in solar panels is the lack of access to real data (Gallardo-Saavedra et al. 2019). Most studies use few images, requiring techniques to increase the volume of data (Haitao et al. 2021; Ding et al. 2018). When such low volume of data is regarded, the generalisation of the CNN may be limited, given the wide variability of defect categories, viewing angles, and lighting conditions. Hence, datasets with a large volume of publicly available data must be generated to help find real data (Gaviria et al. 2022; Mehta et al. 2017; Zhang et al. 2021; Ozturk et al. 2021).

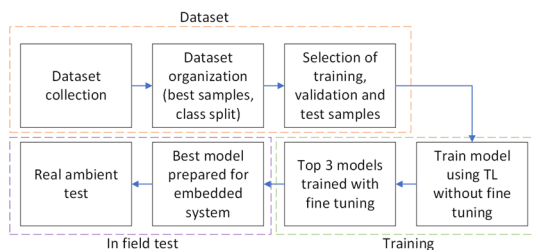
Considering the difficulty in finding datasets of visible spectrum images of PVs, we collected and organised a public image dataset using RGB images with four common PV defects, which is the second contribution of this paper. This dataset was treated using the TL method to assess the CNN model's performance. All tests performed in this study, including the codes, will also be made available to the scientific community in encouraging improvements to the model and presenting new applications in the real world.

Finally, tests done in a real environment with a high-performance, low-power edge device were also performed in this study. These tests aimed to demonstrate that DL can be applied to address the problem in question using embedded systems.

This paper is organised as follows: Section 2 presents the methods used in this study, including the construction of the dataset. Section 3 presents comparisons of different CNN models, as well as the results compared with the findings of other studies recently published in the literature. Section 4 presents the study's conclusions and future perspectives.

## 2 Methodology

Using the tools and techniques mentioned above, this section details the performed tasks to achieve the pretended contributions. Thus, three stages need to be followed. The first stage involves manually collecting and classifying images to create the dataset. Then, in the second step, CNN models are trained and tested using the dataset, including selecting network models to apply the TL technique and define the



**Fig. 1** Proposed method overview

training parameters. The last step consists of optimising the CNN model for the Nvidia processor and a real field test. For further clarification, Fig. 1 depicts the procedures used in this study. Each stage is detailed in the following subsections.

## 2.1 Dataset Creation

Because of the difficulty in finding datasets containing many images from the visual spectrum with common defects of PV panels, this work aimed to create a public dataset with identifiable defects in visible spectrum images and often mentioned in the literature. Two smartphones with a 12-MP camera and a DJI Phantom 3 Professional UAV with a 12-MP camera were used for image collection. Images were acquired through videos, with a resolution of  $1920 \times 1080$  pixels for smartphones and  $3840 \times 2160$  pixels for UAV.

After video frames were extracted, the smartphone-captured images were close to the solar panels, and no processing was necessary to extract the region of interest. Conversely, the extraction of the region of interest was required from the captured images with the UAV in each frame of the videos because a given image could contain several solar panels or points irrelevant to this study. Hence, the sliding window technique was chosen, using different window sizes because of the difference in flight heights and plant types. This approach created substantial diversity in the dataset, causing a natural variation in the collected images. Because of the sliding window, the dataset contained images ranging from  $800 \times 450$  pixels to  $1920 \times 1080$  pixels.

The collected images present in this study were gathered from solar power plants installed at the Federal University of Technology-Parana (Universidade Tecnológica Federal do Paraná-UTFPR)-Curitiba Campus and at the Federal Institute of Parana (Instituto Federal do Paraná-IFPR)-Jacarezinho Campus, Brazil. The images were collected on different days, prioritising sunny days with few clouds to improve image contrast and, consequently, the classification result.

Four power plants containing two types were monitored in the image capture: two comprise PV panels fixed on metallic structures, attached to a slab in one plant and a roof in the other, whereas in the other type of plant, PV panels are directly placed under a metal roof.



**Fig. 2** Sample images of each class: **1** shadow class, **2** obstruction class, **3** dirty class, and **4** clear class. The QR Code contains the address to access the public dataset

During the dataset creation process, we prioritised the acquiring of images containing primary defects that could impact solar panels' electricity generation efficiency. Among these defects, we selected those that can be more easily replicated, such as the obstruction of the solar panel by bush, dirt, and shading. In addition to these defects, a class with clear panels was also included, which is the reference for a panel without faults. Figure 2 shows a sample of each type in this dataset. Such classes will be detailed in the following subsections. In addition, this dataset can be accessed and used from the link<sup>1</sup> or using the QR code in Fig. 2. Comparatively, the datasets of visible spectrum images mentioned in the introduction section consider only one type of defect.

### 2.1.1 Shadow Class

Figure 2(1) shows images of PV panels with cast shadows from trees, cables, poles, or other objects. Some shadows in the collected images were natural, whereas others were created by placing objects on the panels to cause shading.

### 2.1.2 Obstruction Class

Figure 2(2) contains images with overlapping tree branches and shrubs atop solar panels, which cause a partial obstruction, possibly generating shadows and indicating the need to prune the vegetation where the panels are installed. We could not find any similar condition mimicking such obstruction in the environment where images are captured. Therefore, this class of images was collected by positioning natural and artificial bushes on top of the panels to simulate such condition.

<sup>1</sup> <https://github.com/abiffe/paper-pv-dnn>

### 2.1.3 Dirty Class

Figure 2(3) shows the class of images, wherein the accumulation of dust, sand, dirt, or bird droppings atop the panels interferes with electricity generation. These images were collected in actual and simulated conditions similar with previous classes.

### 2.1.4 Clear Class

Images were collected from panels without any irregularities present in previous classes, that is, with the ideal condition for electricity generation, as observed in the visible spectrum. A small sample of this class is shown in Fig. 2(4).

## 2.2 Image Split

After capturing the images, the most appropriate ones were chosen and separated into defined classes, selecting 36,000 images to compose the dataset. The number of images was determined so that each class constituted the same number of pictures, and thus balanced to improve the evaluation. For clarity, Table 1 shows the summary of the developed dataset. The table compiles each class's image size, and quantity, showing balance in the developed dataset.

Subsequently, training, validation, and test sets were constructed for the CNNs, containing 25,920; 6480; and 3600 raw images, respectively. The images were randomly selected for the collection, maintaining the same proportion of samples in each class. The images were organised in separate folders for the reproducibility of the available dataset.

The images were separated into three sets to evaluate the trained model more accurately. The training dataset was used to train the model. During the training, the validation dataset was used to evaluate the training process. If the results using the validation dataset differed from the results using the training dataset, the training parameters will be modified. Finally, when results were deemed satisfactory for the problem treated using the validation dataset, the test dataset is used to assess whether the validation accuracy results were relevant to a dataset different from the one used in training.

## 2.3 Transfer Learning

With the datasets created, the TL method facilitates and accelerates the CNN model training. TL was chosen because this approach uses prior knowledge acquired by a CNN during its training in a task (in this case, the ability to extract interesting features from images) and applies this knowledge in new situations (Alzubaidi et al. 2021; Pan and Yang 2010), with relevant results in the context proposed here, as demonstrated in (Ding et al. 2018; Gutoski et al. 2021).

In this study, 17 networks were chosen for training, testing, and comparison. Classification metrics were obtained using CNNs with different levels of complexity, and all models were previously trained using the well-known ImageNet dataset. The models used in this study were chosen based on studies found in the literature and on network size, input image size, and their availability in TensorFlow/Keras (Keras 2023), the selected development environment. The models were deliberately chosen for their diversity to assess how they treated the problem in question.

For the training, each set of training, validation, and test images was loaded with a batch size of 32. To evaluate the efficiency of these pre-trained models, a 2D global average pooling layer was added to the end of the network, a 0.2 dropout, and a dense layer the size of the number of classes of the problem in question (4 classes); more specifically, fine-tuning was not performed. Each of the chosen networks was trained for 20 epochs, at a learning rate of 0.001, with a loss function as a categorical cross-entropy and accuracy as a metric for model evaluation.

After the initial step, the three networks with the best results were subjected to new training, although with fine-tuning (FT). This approach was used to verify if, with FT, better results were accomplished. For the fine-tuning, 10% of the upper layers were unfrozen, and the networks were trained again using the same parameters. The threshold was set at 10%, based on tests to avoid overfitting.

Despite the vast dataset, DA was applied in all training sessions to increase model diversity and improve learning robustness towards using the model in real-world situations. Several DA tests were performed; the following methods produced results consistent with the dataset: horizontal and vertical mirroring, rotation with a 20% scaling with filling using the nearest algorithm, and increasing/decreasing the width and height of the image with 50% scaling. All DA operations were performed randomly. For the training, the images were resized according to the requirements of each model.

For all stages of the training process, including validation and testing, a computer with 32 GB of RAM, an Nvidia Geforce RTX 2060 GPU with 6 GB of RAM, and an Intel Core i7-10700 processor was used.

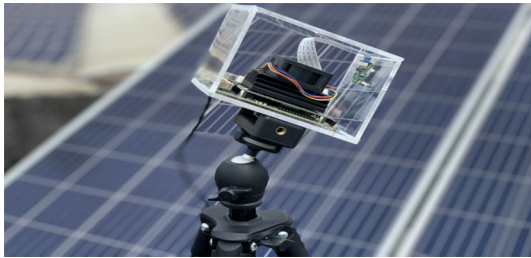
## 2.4 Tests in a Field Environment

After the tests with fine-tuning, the best-performing network was evaluated using a low-cost hardware for real-image classification. The test was performed in a real environment using the edge device that is constituted by an Nvidia Jetson Nano 4GB, with an 8-megapixel Raspberry Pi V2 camera to assess the inference time of the best network. The test was performed in two ways. Initially, the trained model was directly loaded into the TensorFlow and run on the Nvidia Jetson edge



**Table 1** Dataset constitution information with each class

Class	Examples			Acquisition condition	Image size	Image orientation
	Train	Validation	Test			
Clear	6480	1620	900	Sunny/partially cloudy	800 × 450 to 1920 × 1080	Vertical/horizontal
Dirty						
Obstruction						
Shadow						

**Fig. 3** Developed edge device with Nvidia Jetson Nano and RaspberryPiV2 Camera

device. Then, the model was optimised using the Nvidia TensorRT Python API, configured to use a 32-bit floating-point format. This optimisation aimed at compiling the model to run efficiently on Nvidia processors. For better visualisation, Fig. 3 shows the edge device in field test.

Parallel to the tests with and without the optimised model, other tests were performed using the Jetson power supply. In each test with the selected network, the device was initially powered through a USB port, which provided the maximum power of 5 W. Soon after, the same test was performed using a dedicated power source of up to 10 W.

### 3 Results

Applying the explained methodology, Table 2 outlines the training, validation, and test results, with their respective accuracy values, the major error class, and the number of total and trainable parameters of the network. In addition, the test results are ordered from best to worst in the test step to facilitate the identification of the best results in each stage, and these values are highlighted (italicised and underlined). Regarding the size of the input image, most networks used an image of  $224 \times 224$  pixels. The exceptions include Inception-ResNetV2, InceptionV3, and Xception networks pre-trained using  $299 \times 299$  pixels images.

By analysing Table 2, MobileNetV3Large, DenseNet201, and EfficientNetB0 networks exhibited the best results from training. However, during the training validation, the EfficientNetB0 network was substituted for MobileNet. This result highlights the importance of the validation dataset

because without it, this network could be mistakenly considered as one of the best.

Then, the networks were subjected to a test base to assess the behaviour with the data not used in training. In the test results, the top 3 validation stages remained unchanged, with a difference of more than 1.5% between the first and second networks.

Together with Table 2, the prediction confusion matrix for each network was generated using the test data to assess the models' false positive/negative rate, thus, identifying the class with the lowest accuracy of the model. This information is provided in the "major error class" column in Table 2. To clarify this process, Fig. 4 displays the confusion matrix of the MobileNetV3Large network, which provided the best results. In the test stage, 3600 images were used, with each class containing 900 images. The numbers diagonal to this confusion matrix should be as close as possible to 900; the further each number from the 900 value, the lower the accuracy will be in the corresponding class.

Overall, the clear class has the highest error, as outlined in Table 2. This confusion between classes may occur because many images of a panel without a defect visually analysed by a human can be easily confused with a defective panel and vice versa.

This threshold definition over analysed classes in this study was complex. Some studies on specific methods for threshold definition are available in the literature. To better visualise this error, Fig. 5 shows an example of an erroneous classification performed using MobileNetV3Large without FT. In this example, all images should have been classified as clear.

As expected, MobileNet outperformed the other networks because it can match larger networks, such as VGG16, as shown in the paper on MobileNet by Howard et al. (2017) and in the paper on MobileNetV3 by Howard et al. (2019), which highlights a considerable evolution. Accordingly, despite being smaller than others, this network can deliver excellent results (accuracy  $\geq 95\%$ ) for the problem addressed in this work.

From the results obtained, the MobileNetV3Small network result also demonstrates that smaller networks can match the performance of many parameter networks. The

**Table 2** Efficiency of models selected for the training, validation, testing stages, and parameters of each model

Net	Accuracy			Parameters		
	Training	Validation	Test	Major error class	Total	Trainable
MobileNetV3Large	<u>96.02</u>	<u>97.92</u>	<u>97.72</u>	Clear	3,000,196	3844
DenseNet201	<u>95.52</u>	<u>95.66</u>	<u>96.00</u>	Shadow	18,329,668	7684
MobileNet	94.43	<u>94.92</u>	<u>95.47</u>	Clear	2,263,108	5124
ResNet151V2	93.97	94.34	94.78	Clear	58,339,844	8196
VGG19	91.39	94.12	94.72	Dirty	20,026,436	2052
MobileNetV2-224	93.87	94.04	94.47	Clear	3,000,196	3844
ResNet50V2	<u>94.24</u>	94.46	94.33	Clear	23,572,996	8196
EfficientNetB0	95.25	93.33	94.17	Clear	64,107,931	10,244
Xception	93.26	94.34	93.97	Shadow	20,869,676	8196
EfficientNetV2L	89.43	92.87	93.00	Clear	21,810,980	8196
EfficientNetV2B0	94.83	92.45	92.53	Clear	117,751,972	5124
MobileNetV3Small	90.93	92.59	92.33	Clear	941,428	2308
InceptionV3	91.98	91	91.67	Clear	2,263,108	5124
VGG16	91.20	91.22	91.61	Clear	14,716,740	2052
DenseNet121	93.61	90.48	91.56	Clear	7,041,604	4100
EfficientNetB7	92.94	90.8	91.03	Clear	5,924,436	5124
InceptionResnetV2	83.61	79.20	78.36	Clear	3,232,964	4100

The data are ordered from best to worst accuracy

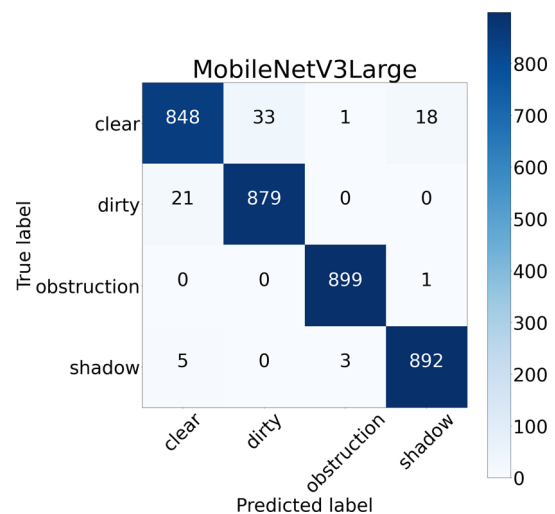
MobileNetV3Small was the smallest network used in this study. Nevertheless, its results were similar to those of many parameter networks and indicated the high potential of such network for applications requiring fast responses and low computational power.

The networks that used the largest input image performed similarly to those that used input images of  $224 \times 224$  pixels. This result shows that larger images do not provide better results. This is helpful because input image size is critical when using limited computational resources.

In finishing the initial tests without unfreezing the upper layers, the three best networks were fine-tuned. Table 3 outlines the parameters trained in each network and their performance in each training step.

The results showed that the difference between MobileNetV3Large and DenseNet201 narrowed after FT. Nevertheless, the training data analysis by epoch showed that both networks exhibited minor signs of overfitting. Therefore, for the problem in question, MobileNetV3Large showed better results, with more than three times lower trained parameters. In addition to the numerical results, the example images shown in Fig. 5 were correctly classified using MobileNetV3Large after FT, thus, proving the improvement of the network using this technique.

About the MobileNetV3Large structure, this network has a starting Conv2D layer with input of  $224^2 \times 3$  and after that many bottleneck layers that could have/not have Squeeze-And-Excite, nonlinearity, h-swish, ReLU, batch normalisation and stride. After this section of bottleneck lay-



**Fig. 4** Confusion matrix of the MobileNetV3Large network before fine-tuning

ers, it ends in a Conv2D  $1 \times 1$  with an input of  $7^2 \times 160$ . The next layer is a pool  $7 \times 7$  with input of  $7^2 \times 960$ . The penultimate layer is a Conv2D of  $1 \times 1$  with  $1^2 \times 960$  input. Finally, the last layer is a Conv2D of  $1 \times 1$  with input  $1^2 \times 1280$  with output of  $k$ , where  $k$  is the number of classes used to train the network. More in details of the evolution (versions) of the network and the structured are found in Howard et al. (2019).

Referring to the training process, this is a key issue when using CNN. Having many parameters, larger networks



**Fig. 5** Examples of images misclassified using MobileNetV3Large before fine-tuning. All these images were supposed to be classified as clear, but they were instead classified as follows: **1** obstruction, **2** shadow, and **3** dirty

require high computational power for training, which also affects the inference time when the model is used in real applications. In real applications, computers with low computational power may take a long time to process an image or fail to load the network model because of its size and the edge devices' RAM limitations.

These findings show that TL can be a helpful tool in addressing the problem in question. In addition, comparing several networks and assessing accuracy and network size help to select the best network. Moreover, smaller networks matched or outperformed larger networks in this problem. These results promote further research and new applications using smaller networks because, as shown in the literature, several studies have been based on large networks. Therefore, their results may be optimised using more efficient networks.

### 3.1 Tests with Embedded System

After the training and testing stages, the methodology's final process consists of a field test using the best-performed model. Thus, the MobileNetV3Large was selected for tests in real environment using the Nvidia Jetson Nano system. The code was developed through parallel processing and queued to ensure more image acquisition and processing fluidity. In this way, every frame is acquired independently from the processing, avoiding frame loss and waiting for process time. Initially, a test was directly performed using TensorFlow and OpenCV pre-processed images. In this test, the system required an average of 300 ms to infer an image

in a 10 W mode and 600 ms in a 5 W mode. Because this result was much lower than expected, based on the literature on MobileNetV3Large, optimisation was performed using TensorRT mentioned in Sect. 2. Accordingly, the system's average inference time to analyse an image was 50 ms in a 10 W mode and 90 ms in a 5 W mode. These results highlight the importance of the optimisation process.

Having positive results in the 5 W mode, the embedded system was tested in a real environment. This test involved simulating the three types of defects and the panels without fault to analyse the system's behaviour. This real test was recorded and can be accessed using the QR code in Fig. 2. The video clearly shows that the system could identify the predicted classes correctly and demonstrated fluidity identification because of the hardware and optimisation.

The main issue of the proposed system is with the direct image analysis. However, because the system has a GPU, it can run image segmentation algorithms efficiently. Nevertheless, this algorithm is a topic of a separate research found in many works on this theme (Yahya et al. 2022).

### 3.2 Comparison with Related Works

The results of the present study outperformed those of previously proposed networks mentioned in the introduction section, but some remarks must be addressed. Comparing the results discussed in the introduction section, superior or similar results were found using the networks applied in this study. More specifically, the results from the present study are outstanding and more promising than the survey conducted by Li et al. (2020), wherein classification tests in an edge device were reported and compared with those of other networks, in addition to the study's proposal.

The main difference between the present work and the compared research is the scope of the reported results with known and pre-trained networks. This wide-ranging comparison indicated that smaller networks, usually disregarded in most research, may have a high potential to solve PV panel analysis problems. In addition, this research used a commercially available edge tool (Nvidia Jetson Nano). It made the developed code, trained networks models, and dataset publicly available, facilitating its replication by researchers and solar power plant companies.

**Table 3** Efficiency of models in the training, validation, and testing stages of fine-tuning

Net	Accuracy			Major error class	Parameters
	Training	Validation	Test		Trainable
MobileNetV3Large	99.38	99.48	99.64	Dirty	801,204
DenseNet201	99.44	99.40	99.58	Dirty	2,887,748
MobileNet	97.83	97.44	97.56	Dirty	1,065,988

Collecting and organising the dataset in this study for community access must be emphasised because each study has datasets of different images, preventing direct comparisons among methods. Accordingly, the public dataset made available in this study enables subsequent comparisons between other methods, thus, facilitating the development of more efficient solutions to the problem in question. The publicly available dataset aims to inspire other research groups to publicise their data, contributing to the information presented herein.

## 4 Conclusion

This study used CNN with TL to classify defects observed in visible spectrum images of PV panels. This classification enables the optimisation of this method of electricity generation, helping to minimise sources of defects, which impair the efficiency of PV panels. A dataset with images of some common defects was also built, and it is publicly available to the community.

Together with the captured images, the selection of several previously trained models was essential to evaluate the training and loss performance. The experiments showed that TL provided excellent results with the datasets used in this study. Furthermore, in this work, the network with the best results in all stages was the MobileNetV3Large, realising an accuracy of 99.64% in the test stage with FT.

From the good results of this relatively small network, an in-field test could be performed using a device with less computational power and energy consumption. This approach still needs to be explored in the literature, and it is the main contribution of this study. The results were promising because the embedded system could efficiently identify and classify the defects and healthy PVs present in the dataset.

The results of this study bridge gaps found in the literature. These publicly available codes, trained models, and datasets enable other researchers to replicate and conduct new investigations and real applications based on positive results.

The problem analysis indicates that the methods proposed in this study can be used in embedded systems to help maintenance and engineering teams to make more accurate decisions, thus, optimising the management of maintenance costs. In addition, using this tool with continuous monitoring enables quick decision-making, resulting in gains in electricity generation. Subsequently, from this constant monitoring, it may be possible to build a periodic maintenance schedule, facilitating the projection of maintenance costs of legacy and new solar power plants.

Our approach is still at a prototype level, with images collected on sunny days. Future studies may include other classes of defects to assess whether high performance is maintained along with other shortcomings. Furthermore,

models trained with these methods may be subjected to environments where their performance in analysing unknown images and situations has yet to be assessed. Finally, image segmentation algorithms must be used to generalise the developed system.

**Acknowledgements** The present study was funded by the Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES), Brazil – Funding Code 001, the National Council for Scientific and Technological Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq), the Funding Authority for Studies and Projects (Financiadora de Estudos e Projetos – FINEP), the Araucária Foundation (Fundação Araucária – FA), and the General Superintendence of Science, Technology and Higher Education (Superintendência Geral de Ciência, Tecnologia e Ensino Superior – SETI). The authors also thank the Federal Institute of Paraná (Instituto Federal do Paraná – IFPR), especially the Office of the Vice-Dean for Continuing Education, Research, Graduate Studies and Innovation (Pro-Reitoria de Extensão, Pesquisa, Pós-graduação e Inovação – PROEPPi), for lending the UAV. Author A. E. Lazzaretti would like to thank to CNPq for the research grant 306569/2022-1.

**Funding** Partial financial support was received from the agencies mentioned in Acknowledgments.

## Declarations

**Conflict of interest** The authors have no conflict of interest to declare that are relevant to the content of this article.

## References

- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8, 53.
- Azis, S.F.S., Shahbudin, S., Kassim, M., Mohamad, R., & Rahman, F.Y.A. (2022). Photovoltaic module defects classification analysis using densenet architecture. In *Proc. IEEE symposium on industrial electronics & applications*, pp. 1–5.
- Cao, S., Ma, X., Fan, S., & Wang, T. (2021). IDS-Net: Integrated network for identifying dust state of photovoltaic panels. In *Proc. IEEE international conference on electronic information engineering and computer science*, pp. 89–92.
- Chen, H., Pang, Y., Hu, Q., & Liu, K. (2020). Solar cell surface defect inspection based on multispectral convolutional neural network. *Journal of Intelligent Manufacturing*, 31, 453–468.
- Ding, S., Yang, Q., Li, X., Yan, W., & Ruan, W. (2018). Transfer learning based photovoltaic module defect diagnosis using aerial images. In *Proc. IEEE international conference on power system technology*, pp. 4245–4250.
- Espinosa, A. R., Bressan, M., & Giraldo, L. F. (2020). Failure signature classification in solar photovoltaic plants using RGB images and convolutional neural networks. *Renewable Energy*, 162, 249–256.
- Gallardo-Saavedra, S., Hernández-Callejo, L., & Duque-Perez, O. (2018). Technological review of the instrumentation used in aerial thermographic inspection of photovoltaic plants. *Renewable and Sustainable Energy Reviews*, 93, 566–579.



- Gallardo-Saavedra, S., Hernández-Callejo, L., & Duque-Pérez, O. (2019). Quantitative failure rates and modes analysis in photovoltaic plants. *Energy*, *183*, 825–836.
- Gaviria, J. F., Naráez, G., Guillen, C., Giraldo, L. F., & Bressan, M. (2022). Machine learning in photovoltaic systems: A review. *Renewable Energy*, *196*, 298–318.
- Gutoski, M., Ribeiro, M., Hattori, L. T., Romero, M., Lazzaretti, A. E., & Lopes, H. S. (2021). A comparative study of transfer learning approaches for video anomaly detection. *International Journal of Pattern Recognition and Artificial Intelligence*, *35*, 2152003.
- Haitao, C., Caijia, L., Xin, W., & Qiang, S. (2021). An automatic defects detection system for PV plants with data augmentation. In: *Proc. IEEE international conference on power system technology*, pp. 980–984.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., & Adam, H. (2019). Searching for MobileNetV3. In *IEEE/CVF international conference on computer vision*, pp. 1314–1324.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR* [abs/1704.04861](https://arxiv.org/abs/1704.04861)
- International Energy Agency – IEA: Solar PV (2022). <https://www.iea.org/reports/solar-pv>
- Keras: Keras Applications (2023). <https://keras.io/api/applications/>
- Li, B., Delpha, C., Diallo, D., & Migan-Dubois, A. (2021). Application of artificial neural networks to photovoltaic fault detection and diagnosis: A review. *Renewable and Sustainable Energy Reviews*, *138*, 110512.
- Li, X., Li, W., Yang, Q., Yan, W., & Zomaya, A. Y. (2020). Edge-computing-enabled unmanned module defect detection and diagnosis system for large-scale photovoltaic plants. *IEEE Internet of Things Journal*, *7*, 9651–9663.
- Li, X., Yang, Q., Lou, Z., & Yan, W. (2019). Deep learning based module defect analysis for large-scale photovoltaic farms. *IEEE Transactions on Energy Conversion*, *34*, 520–529.
- Mehta, S., Azad, A.P., Chemmengath, S.A., Raykar, V., & Kalyanaraman, S. (2017). Deepsolareye: Power loss prediction and weakly supervised soiling localization via fully convolutional networks for solar panels. *CoRR* [abs/1710.03811](https://arxiv.org/abs/1710.03811)
- Ozturk, O., Hangun, B., & Eyecioglu, O. (2021). Detecting snow layer on solar panels using deep learning. In *Proc. IEEE 10th international conference on renewable energy research and application*, pp. 434–438.
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*, 1345–1359.
- Queiroz, A. R., Faria, V. A. D., Lima, L. M. M., & Lima, J. W. M. (2019). Hydropower revenues under the threat of climate change in Brazil. *Renewable Energy*, *133*, 873–882.
- Sizkouhi, A. M. M., Esmailifar, S. M., Aghaei, M., & Karimkhani, M. (2022). RoboPV: An integrated software package for autonomous aerial monitoring of large scale PV plants. *Energy Conversion and Management*, *254*, 115217.
- Venkatesh, S.N., & Sugumaran, V. (2021). Fault detection in aerial images of photovoltaic modules based on deep learning. In *Proc. IOP conference series: materials science and engineering*, vol. 1012, p. 012030.
- Victoria, M., Haegel, N., Peters, I. M., Sinton, R., Jäger-Waldau, A., Cañizo, C., Breyer, C., Stocks, M., Blakers, A., Kaizuka, I., Komoto, K., & Smets, A. (2021). Solar photovoltaics is ready to power a sustainable future. *Joule*, *5*, 1041–1056.
- Wang, X., Zhao, B., Cao, S., & Fan, S. (2022). A multi-sensor information fusion monitoring system for photovoltaic power generation. In *Proc. IEEE 3rd international conference on computer vision, image and deep learning*, pp. 955–959.
- Yahya, Z., Imane, S., Hicham, H., Ghassane, A., & Bouchini-Idrissi Safia, E. (2022). Applied imagery pattern recognition for photovoltaic modules' inspection: A review on methods, challenges and future development. *Sustainable Energy Technologies and Assessments*, *52*, 102071.
- Zhang, W., Liu, S., Gandhi, O., Rodriguez-Gallegos, C. D., Quan, H., & Srinivasan, D. (2021). Deep-learning-based probabilistic estimation of solar PV soiling loss. *IEEE Transactions on Sustainable Energy*, *12*, 2436–2444.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.