Studies on Techniques to Improve YOLO in Fault Detection Using RGB Images of Solar Panels

Weng Ti Wong
School of Engineering
Taylor's University
Kuala Lumpur, Malaysia
wongwengti02@sd.taylors.edu.my

Swee King Phang
School of Engineering
Taylor's University
Kuala Lumpur, Malaysia
SweeKing.Phang@taylors.edu.my

2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST) | 979-8-3503-5790-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/GECOST60902.2024.10474793

***Abstract*—Solar panels can have short lifespan and significantly lower efficiency with the occurrence of various faults. Currently, most fault detection methods suffer from cost-ineffectiveness due to either requiring expensive equipment, such as a thermal camera or lack of information on the source of the faults. Most of the research of using You Only Look Once (YOLO) to detect faults in solar panels revolves around thermal images too. This paper aims to propose an alternative method to detect faults (primarily cracks and soiling) in solar panels by applying YOLOv7 on Red, Green, Blue (RGB) images captured by an Unmanned Aerial Vehicle (UAV). To further improve the performance of this scheme, this paper will focus on applying various techniques to improve the accuracy and training time of the model, which include gray scale conversion, data augmentation and change of Gradient Descent (GD) optimizer. Based on preliminary results, it was found that the combination of gray scale conversion and data augmentation leads to an accuracy of 95.91% while Nadam stands out as the best optimizer because it reduces required training time for convergence by 75 epochs.**

Keywords— solar panels, YOLO, cracks, soiling, UAV, RGB camera, thermal camera

I. INTRODUCTION

In the pursuit of a future powered by cleaner energy, solar, hydroelectric power and wind energy are all considered to be viable candidates in replacing the conventional fuel-based energy generation methods [1]. Out of all these prospects, solar energy is deemed to have the most potential due to the advancement in solar Photovoltaic (PV) cells which render solar panels to become an energy harvester that is much affordable and less energy-intensive to be manufactured compared to its counterparts [2]. This is evident by the research result which states that worldwide installed capacity of PV technology increases from 40,334 MW to 709,674 MW in the last decade [3]. Despite being the spearhead that propels the growth of clean energy, solar energy still has its own flaws, particularly with the solar panels. On average, solar panel has an expected lifespan of 25 to 30 years [4]. However, the lifespan can be easily cut short with the occurrence of various faults. Apart from that, the fault of one solar panel can essentially diminish the output of an entire string of solar panels. Therefore, keeping track of the status of a solar panel is crucial in prolonging the effective lifetime of a solar panel and improving the power output efficiency.

A. Faults in Solar Panels

In the metric of product of severity and frequency of occurrences, hotspots and glass breakages are among the most prominent faults that can occur to a solar panel with the former tallying up to approximately 30% and the latter being 20% in the last 10 years [5]. Hotspots occur when a portion of the PV cells are shaded. As a result, the shaded PV cells will have a decreased power output of 19% to 33% and become reverse biased [6]. In fact, out of all the factors, soiling, which refers to the accumulation of dust, algae, bird droppings or any stains that shield the solar panels from light exposure, is accountable for creating 8% to 12% of the hotspots on solar panels [6].

(a) (b)

Fig. 1. Examples of (a) a cracked solar panel and (b) a soiled solar panel

On the other hand, glass breakage is found to have contributed to 10% of the failures in solar panels after 2 years of operation and 33% of the failures after 8 years of operation [7]. Generally, glass breakage can be caused by temperature variations during operations (thermal stress), sudden impacts (mechanical stress) and harsh climatic condition (lightning and hailstorm) [6]. If the glass breakage is left unattended, it will lead to openings where the water vapor can moisturise the insulation of encapsulant layer of the PV cells and lead to ingression of water. Eventually, the expansion and contraction of water vapor can cause mechanical stress within the PV cells and lead to the formation of microcracks. Examples can be seen in Fig. 1.

B. Fault detection using Infrared Thermography

The research world has mostly focused on Infrared Thermography (IRT) to detect faults in solar panels. IRT works with the help of a thermal camera. By capturing images of solar panels, cells that suffer from hotspots will generate abnormal amount of heat and give temperature data that will be represented as bright regions in the output of IRT [6]. An example of a thermal image can be seen in Fig. 2.

The research work by Cubukcu and Akanalci in 2019 showed that they managed to detect 241 faults in solar panels from 19 different projects, with 25% of them being hotspots and 5% being broken modules using thermal cameras [8]. Apart from that, a FLIR thermal camera and a normal RGB camera were installed onto a drone to 3D model the landscape where the solar panels are situated at. It was argued this can greatly assist the fault detection and analysis in solar panels [9] . However, thermal cameras have lower resolution and to match RGB images with thermal images, the drone needs to fly at a lower height thus requiring longer flight time to cover the same area. This method can also lead to influx of huge amount of data and therefore deep learning framework is generally used to assist IRT.

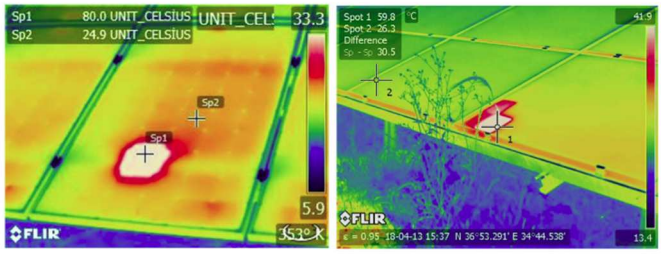


Fig. 2. An example of a thermal image [8].

C. Applications of Deep Learning Algorithms

In terms of computer vision, CNN is the most used type of neural networks. The general architecture of a CNN network can be seen in Fig. 3.

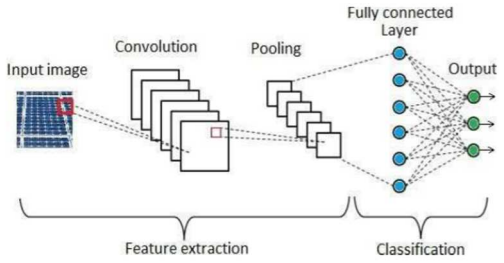


Fig. 3. The general architecture of a CNN [1].

The emergence of YOLOv1 is game changing as it is a CNN that completes drawing of bounding box and classification in one go which leads to high computational speed and is suitable for real-time object detection. Since then, YOLOv1 has evolved for 7 iterations where YOLOv8 was introduced in January 2023. YOLO has been incorporated extensively into the realm of fault detection using drones and IRT because of its high computational capabilities to speed up fault analysis from aerial imageries [10].

For an instance of applying deep learning frameworks in solar panel fault detection, an Autel camera was installed on a drone to capture thermal images of solar panels in a research work done by Terzoglu and his co-researchers in 2023 [11]. Then, YOLOv5 was used to detect and classify the faults. Additionally, Zou and Rajvee use FLIR Duo Pro R thermal camera on a drone to capture images of solar panels, then classify the panels into faulty or non-faulty using YOLOv4 [11]. This classification is less desirable as it does not provide more information on the source of faults.

The main drawback of using thermal camera is the high cost of a thermal camera. As a reference, the cost of all the thermal camera used in the previously discussed paper are in the range of RM 12 881 to RM 36 499. If the drone carries the thermal camera encounter any accidents, the replacement for the thermal camera can be very costly. In addition to that, multiple research papers, such as [9], [11], [13] have combined the use of thermal camera with normal RGB camera as thermal images lack information on the cause of the faults, which is key to creating a solution or strategy in tackling the faults.

There had been ongoing research works in using CNN onto images captured using standard RGB images. For an instance, Pa, Uddin and Kazemi managed to achieve a success rate of 88.6% for multi-class classification, including normal, shadowy, cracks and dusty using a CNN [1]. However, the proposed method was not tested on a drone. Additionally, Ahmed and his co-researchers also used YOLOv5 to detect faults on a dataset consisting of RGB solar panel images after hyperparameter tuning. However, the reasoning for the tuning was not explained. They managed to achieve a mean average precision (mAP) of 86% [14]. Aside from that, Lestary and his co-researchers installed iPad Pro 2020 on a drone to collect RGB images of solar panels. Then, via YOLOv3, the presence of snail's trail (a brown line appeared on solar panels due to microcracks) was detected with an mAP between 78.44% to 99.7% [15]. However, detecting only one type of fault leads to a limited scope.

D. Research Contribution

It is undeniable that application of YOLO on RGB images has high potential in making the maintenance works in solar farms to be much cost-effective. However, this method also suffers from several concerns:

- Low accuracy due to environmental factors, such as the brightness and exposure and camera lens jittering
- Long training time which can make the hyperparameter tuning process in producing an accurate model to be inefficient.

Therefore, this paper will contribute by addressing these 2 main issues by identifying the most suitable techniques in training a YOLO model using the least training effort and return the highest possible accuracy in detecting faults in solar panels, primarily cracks and soiling. The techniques to be reviewed in this paper include gray scale conversion, data augmentation by adjusting brightness and exposure and a change of GD optimizer.

II. RESEARCH METHODOLOGY

A. Overview

YOLOv7 will be implemented in this research due to its stability and good benchmark performance. Theoretically, the techniques showcased in this paper are transferable to any other versions of YOLO. The datasets to be used in the training consists of 1688 images, which are with a split of 85%:10%:5% for training, validation, and testing. This dataset consists of solar panels that are either intact, soiled or cracked. The training of the model will be done using the default settings shipped together with the YOLOv7 from its GitHub repository. The annotation used in this training will be bounding boxes, which is done via LabelImg. The training will be run on a desktop equipped with RTX 3060 with 8GB memory. Since YOLO is coded using Python, the coding environment used in this project will be PyCharm Community Edition.

B. Evaluation of the trained result

The performance of the model can mainly be evaluated using a metric known as mean Average Precision (mAP), which mathematically can be expressed as shown in (1).

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (1)$$

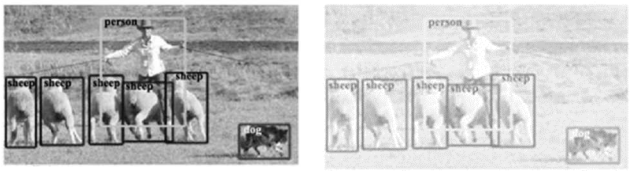
Average Precision (AP) in (1) can be derived by finding the area under the graph of precision against recall. mAP can be evaluated at different Intersection over Union (IOU), which can be defined as the ratio of overlapping area to the total area of predicted bounding box and the ground truth.

Generally, mAP in YOLO will be evaluated at IOU of 0.5. During the training process, the algorithm will keep track of the result that gave the best fitness level and the weight for that epoch will be saved as the best model. The fitness level of the training result is formulated as shown in (2).

$$Fitness = [mAP@0.5 \times 0.1] + [mAP@0.5{:}0.95 \times 0.9] \quad (2)$$

C. Gray Scale Conversion and Data Augmentation

The 2 preprocessing steps that can potentially be used to improve the model's accuracy are gray scale conversion and data augmentation with adjusted brightness and exposure. Gray scale conversion is a technique used in digital photography where every form of colour information is neglected, leaving behind only gray of different shades, with the brightest being white and the darkest being black. On the other hand, data augmentation artificially inflates the data size by oversampling through transformation such as geometric and colour transformation of original dataset to allow more information being extracted [16]. An example of a gray scale image and an augmented image can be seen in Fig. 4.



(a) (b)

Fig. 4. An example of (a) a gray scale image and (b) an augmented image with adjusted brightness and exposure.

Roboflow will be used to help preparing the datasets under these 2 image processing techniques. The dataset will be augmented with ± 25% of exposure and brightness. The target improvement to be achieved using these 2 strategies are 5% compared to the training using default YOLOv7 settings.

D. Selection of Gradient Descent Optimizer

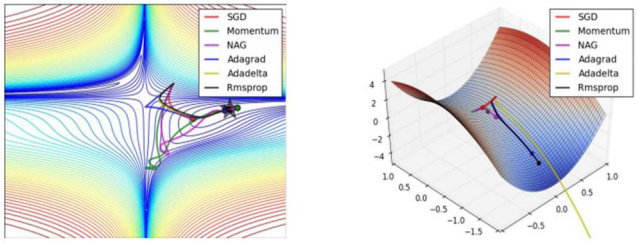
The loss function in YOLO as shown in (3) is the summation of the deviation between predicted and ground truth in the aspect of bounding box (L_{box}), confidence score (L_{conf}), classification score (L_{cls}) and accuracy when there is no detected object (L_{no_obj}). λ_{box} is a constant used to skew the weightage or contribution of bounding box losses to be higher in the resultant loss while λ_{no_obj} is a constant set to skew weightage of no object accuracy to be lower. Meanwhile, s is the total number of grids in the image and k is either 1 or 0 depends on whether there is any detected object.

$$Loss = \lambda_{box} L_{box} + L_{conf} + L_{cls} + \lambda_{no\_obj} \sum_{i=1}^{s} k \times L_{i,no\_obj} \quad (3)$$

The algorithm that is used to decrease the loss to global minima to improve the convergence time and model's accuracy is generally known as GD Optimizer. Most of the popular GD optimizers used today are built as a variation or evolution of Stochastic Gradient Descent (SGD) which can be computed as shown in (4) where the weight of the current iteration, θ_i depends on the product of partial derivative of the loss function in respect to the weight of the previous iteration, $\nabla_\theta J(\theta_{i-1})$ and a learning rate, η.

$$\theta_i = \theta_{i-1} - \eta \cdot \nabla_\theta J(\theta_{i-1}) \quad (4)$$

Analogically, if the gradient descent is a ball rolling downhill to a flat surface where loss is minimum, GD optimizers act as a guide that helps the ball deciding on the path to be taken to reach the end goal. The visualization of the different GD optimizers with this analogy can be seen in Fig. 5.



(a) (b)

Fig. 5. The performance of different types of Gradient Descent Optimizer on (a) a loss contour and on (b) a saddle point [17].

Due to each optimizer using different algorithms, the convergence speed and time will therefore be different, and this has a direct impact on the model's accuracy. The slower the training speed is, the less accurate the model will be unless longer training time is committed until convergence occurs. A preliminary selection had been made to filter out the currently most used and available GD optimizers in PyTorch Libraries (which is the framework where YOLO is built on) due to their obvious flaws. After analysing their characteristics, optimizers to be considered and experimented in this research work other than the default SGD optimizer are Adam, AdamW, Adamax and Nadam.

III. RESULTS AND DISCUSSION

During first round of training, the datasets will be modified and hence a total of 4 different variations will be tested, which are the original dataset (RGB), original dataset with grayscale conversion (GS), original dataset with brightness and exposure augmentation (RGB + AUG) and original dataset with grayscale conversion and brightness and exposure augmentation (GS+AUG). Through augmentation, the training dataset will become larger while the size of the validation and testing dataset remain unchanged and hence the split becomes 92%:6%:2%. The result for these trainings can be seen in Fig. 6. Overall, it can be observed that RGB and GS + AUG dataset perform the best with an approximate improvement of 3% for mAP@0.5 compared to GS and RGB + AUG.

Despite the similar accuracy between RGB and GS + AUG, GS + AUG shows an impressive 3.6% improvement in term of the accuracy for cracks as shown in Fig. 7 but 3.2% drop in the accuracy for the soiling detection as shown in Fig. 8. Hence, it was suspected that the inclusion of dust in the soiling dataset has a significant negative impact on the accuracy of the model. Therefore, the model was re-trained with the exclusion of dust from the datasets.

In fact, it was found that the coal dust generally leads to the least efficiency loss (10.27% to 13.01%) compared to bird droppings and solid soiling (31.25% to 86.53%) as the mass

of contaminants increases from 10g to 50g [18]. Additionally, it was also determined that rainfall is more effective against

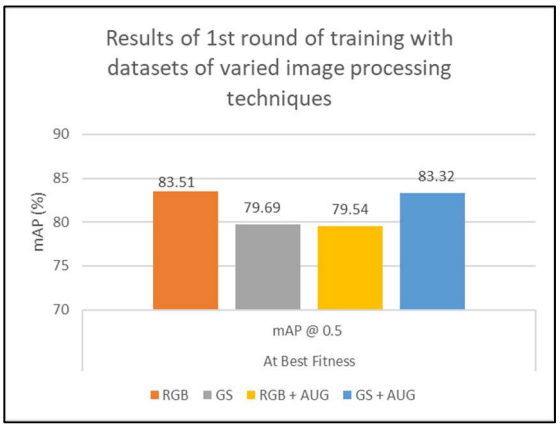


Fig. 6. Best mAP @ 0.5 of 1st round of training.

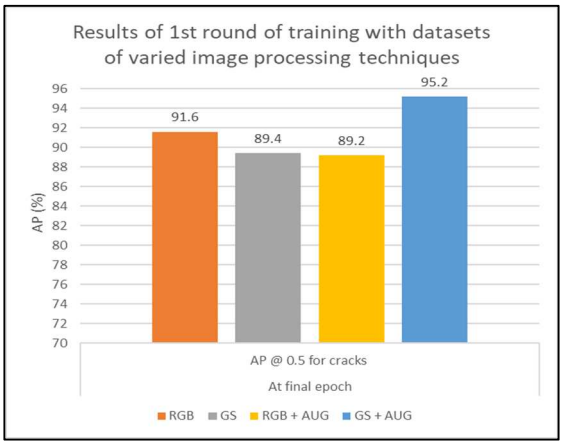


Fig. 7. Best AP @ 0.5 for cracks in 1st round of training.

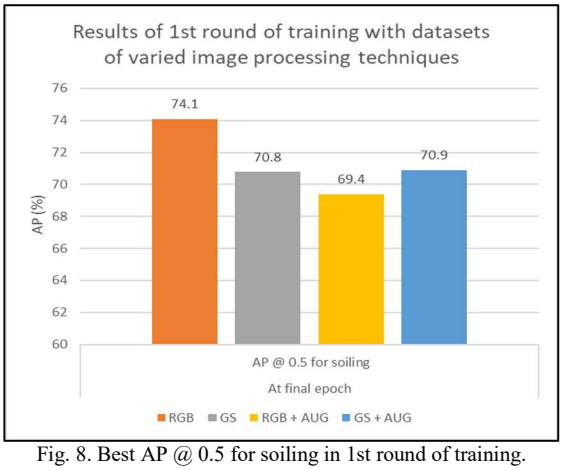


Fig. 8. Best AP @ 0.5 for soiling in 1st round of training.

coal dust compared to bird droppings and solid soiling where the latter tends to have stronger adhesion to solar panels [19]. Considering all these factors and given that the expected outcome of this research is to apply the model on aerial imageries captured by a drone which tends to fly at a height where dust is hardly observable even with human's naked eyes, dust will be excluded from the scope of this research.

Based on the results in Fig. 9, Fig. 10 and Fig. 11, significant improvements can be observed across all datasets. However, the datasets that gave the best performance are RGB + AUG and GS + AUG. Since both have accuracy of approximately 95% @ IOU of 0.5, the tiebreakers will be based on their convergence speed which GS + AUG is superior as seen in Fig. 12 where GS + AUG converged earlier than RGB+AUG.

Finally, the results in Fig. 13 clearly shows that using different optimizers lead to similar mAP @ 0.5 in the end. However, Nadam stands out for reaching convergence at the earliest, which is as early as 25 epochs, which is a significant improvement compared to its other counterparts. Given that one epoch generally takes 2.5 minutes, saving approximately 75 epochs compared to using SGD leads to saving training time of 187.5 minutes.

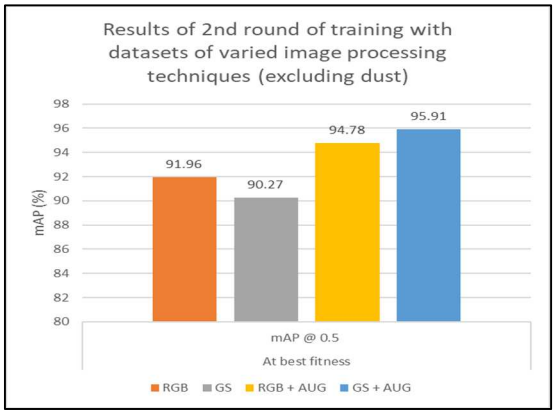


Fig. 9. Best mAP@0.5 of 2nd round of training (without dust).

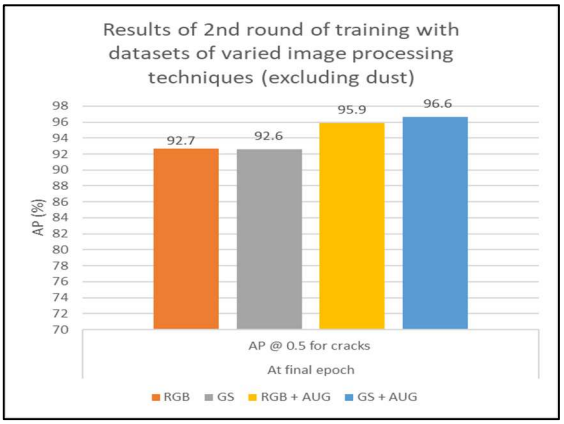


Fig. 10. Best AP @ 0.5 for cracks in 2nd round of training (without dust).

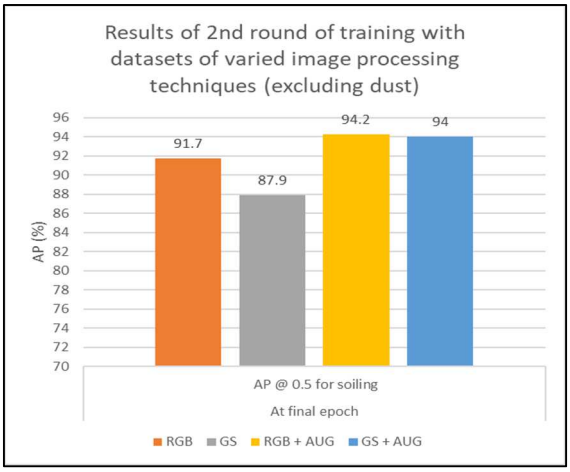


Fig. 11. Best AP @ 0.5 for soiling in 2nd round of training (without dust).

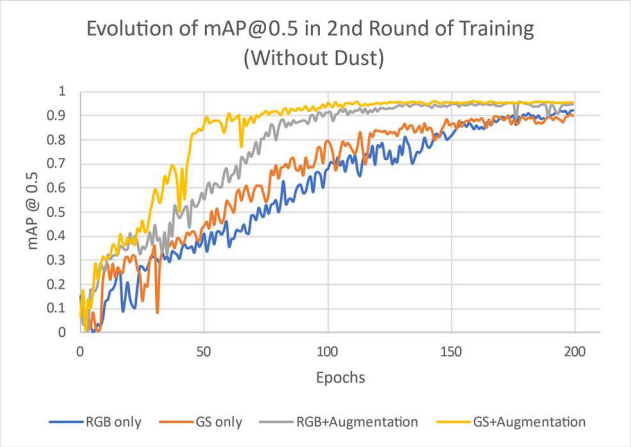


Fig. 12. Convergence of different datasets in 2nd round of training (without dust).

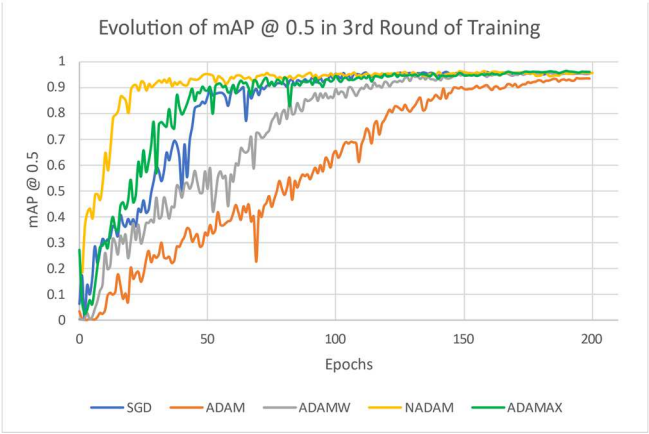


Fig. 13. Convergence of the training using different GD optimizers.

IV. CONCLUSIONS AND FUTURE WORKS

Based on the preliminary results, it can be observed that both gray scale conversion and data augmentation with adjusted brightness and exposure effectively increase the accuracy of the model by 3 – 4 %. On the other hand, Nadam stands out as the best performing GD optimizer as it reduces the training time down to only requiring 25 epochs which is equivalent to saving 187.5 minutes compared to the default SGD. Moving forward, the biggest issue to be tackled will be overfitting which potentially can be addressed via hyperparameter tuning of weight decay and Early Stopper.

REFERENCES

[1] M. Pa, M. N. Uddin, and A. Kazemi, "A Fault Detection Scheme Utilizing Convolutional Neural Network for PV Solar Panels with High Accuracy," in 1st IEEE Industrial Electronics Society Annual On-Line Conference, ONCON 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ONCON56984.2022.10126746.

[2] A. G. Olabi and M. A. Abdelkareem, "Renewable energy and climate change," Renewable and Sustainable Energy Reviews, vol. 158, Apr. 2022, doi: 10.1016/j.rser.2022.112111.

[3] A. O. M. Maka and J. M. Alabid, "Solar energy technology and its roles in sustainable development," Clean Energy, vol. 6, no. 3, pp. 476–483, Jun. 2022, doi: 10.1093/ce/zkac023.

[4] T. Rahman et al., "Investigation of Degradation of Solar Photovoltaics: A Review of Aging Factors, Impacts, and Future Directions toward Sustainable Energy Management," Energies, vol. 16, no. 9. MDPI, May 01, 2023. doi: 10.3390/en16093706.

[5] D. C. Jordan, T. J. Silverman, J. H. Wohlgemuth, S. R. Kurtz, and K. T. VanSant, "Photovoltaic failure and degradation modes," Progress in Photovoltaics: Research and Applications, vol. 25, no. 4, pp. 318–326, Apr. 2017, doi: 10.1002/pip.2866.

[6] V. Sugumaran, "Fault diagnosis of visual faults in photovoltaic modules: A Review," International Journal of Green Energy, vol. 18, no. 1. Bellwether Publishing, Ltd., pp. 37–50, 2021. doi: 10.1080/15435075.2020.1825443.

[7] M. P. Tas and W. G. van Sark, "Experimental repair technique for glass defects of glass-glass photovoltaic modules – A techno-economic analysis," Solar Energy Materials and Solar Cells, vol. 257, Aug. 2023, doi: 10.1016/j.solmat.2023.112397.

[8] M. Cubukcu and A. Akanalci, "Real-time inspection and determination methods of faults on photovoltaic power systems by thermal imaging in Turkey," Renew Energy, vol. 147, pp. 1231–1238, Mar. 2020, doi: 10.1016/j.renene.2019.09.075.

[9] D. H. Lee and J. H. Park, "Developing inspection methodology of solar energy plants by thermal infrared sensor on board unmanned aerial vehicles," Energies (Basel), vol. 12, no. 15, Jul. 2019, doi: 10.3390/en12152928.

[10] S. K. Phang, T. H. A. Chiang, A. Happonen and M. M. L. Chang, "From Satellite to UAV-based Remote Sensing: A Review on Precision Agriculture," IEEE Access, vol. 11, pp. 127057-127076, doi: 10.1109/ACCESS.2023.3330886.

[11] G. Terzoglou, M. Loufakis, P. Symeonidis, D. Ioannidis, and D. Tzovaras, "Employing deep learning framework for improving solar panel defects using drone imagery," in 2023 24th International Conference on Digital Signal Processing (DSP), IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/DSP58604.2023.10167960.

[12] J. T. Zou and V. G. Rajvee, "Drone-based solar panel inspection with 5G and AI Technologies," in Proceedings of the 2022 8th International Conference on Applied System Innovation, ICASI 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 174–178. doi: 10.1109/ICASI55125.2022.9774462.

[13] M. Vlaminck, R. Heidbuchel, W. Philips, and H. Luong, "Region-Based CNN for Anomaly Detection in PV Power Plants Using Aerial Imagery," Sensors, vol. 22, no. 3, Feb. 2022, doi: 10.3390/s22031244.

[14] S. U. Ahmed, M. Affan, M. I. Raza, and M. Harris Hashmi, "Inspecting Mega Solar Plants through Computer Vision and Drone Technologies," in Proceedings - 2022 International Conference on Frontiers of Information Technology, FIT 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 18–23. doi: 10.1109/FIT57066.2022.00014.

[15] F. D. Lestary, Syafaruddin, and I. S. Areni, "Deep Learning Implementation for Snail Trails Detection in Photovoltaic Module," in 2022 FORTEI-International Conference on Electrical Engineering, FORTEI-ICEE 2022 - Proceeding, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 41–46. doi: 10.1109/FORTEI-ICEE57243.2022.9972952.

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," J Big Data, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[17] S. Ruder, "An overview of gradient descent optimization algorithms *," 2017. *arXiv preprint arXiv:1609.04747*.

[18] S. Shaik et al., "Experimental analysis on the impacts of soil deposition and bird droppings on the thermal performance of photovoltaic panels," Case Studies in Thermal Engineering, vol. 48, Aug. 2023, doi: 10.1016/j.csite.2023.103128.

[19] A. K. Sisodia and R. Kumar Mathur, "Impact of bird dropping deposition on solar photovoltaic module performance: a systematic study in Western Rajasthan," Environmental Science and Pollution Research, vol. 26, no. 30, pp. 31119–31132, Oct. 2019, doi: 10.1007/s11356-019-06100-2.