ISSN 1870-4069

Assessing the Impact of Data Augmentation on Photovoltaic Module Faults Detection Using Deep Learning Models

Luis E. Montañez[1], Luis M. Valentín-Coronado[1,4], Daniela Moctezuma[2], Diego A. Mercado-Ravell[3,4]

[1] Centro de Investigaciones en Óptica, Aguascalientes, Mexico

[2] Centro de Investigación en Ciencias de Información Geoespacial A. C., Aguascalientes, Mexico

[3] Centro de Investigación en Matemáticas, Zacatecas, Mexico

[4] Consejo Nacional de Humanidades Ciencias y Tecnologías, Ciudad de México, Mexico

{montanezlef, luismvc}@cio.mx, dmoctezuma@centrogeo.edu.mx, diego.mercado@cimat.mx

Abstract. Accurately identifying photovoltaic (PV) module failures is critical to ensuring their reliability and efficiency. Deep Neural Networks (DNNs) have emerged as a highly effective tool for this purpose, particularly when utilizing infrared images. However, it is important to note that optimal DNN performance commonly requires a substantial amount of high-quality and annotated data. Data augmentation techniques have been developed to address this challenge. These techniques involve applying transformations such as rotation and flipping to augment the size of a dataset. However, it is essential to acknowledge that using data augmentation methods carries the inherent risk of introducing biases or inaccuracies into the augmented data. In this study, we conducted a comparative analysis of four data augmentation methods. Our primary objective was to assess the impact of data augmentation on DNN performance through the k-fold cross-validation technique in the PV module failure classification task. Furthermore, we have also tested the generalization capabilities of five different DNN models when utilizing data augmentation methods. This analysis provides valuable insight into the most effective data augmentation methods for enhancing DNN performance and ensuring the accuracy of PV module failure classification. This analysis highlights that the way one uses data augmentation is critical to achieving realistic and reliable results and, hence, reliable models.

Keywords: Infrared images, data augmentation, CNN, classification, photovoltaic modules.

1 Introduction

Photovoltaic (PV) systems are the primary technology utilized to harness solar energy and, in recent years, have gained tremendous popularity as a renewable energy source. The installed PV capacity exhibits an average annual growth rate of 15%. In the past decade, there has been a significant cost reduction of over 40% in installing PV plants. This reduction has resulted in the emergence of more plants and a rise in their size. However, PV plants, while a great source of renewable energy, require regular maintenance and inspections, which can be inconvenient.

The scientific community has explored various options and technologies to decrease the amount of time needed for inspections and enhance their accuracy. One innovative technique that has emerged is using unmanned aerial vehicles (UAVs) equipped with cameras capable of capturing images in both visible and infrared spectrums, also known as thermal images. Aerial inspections performed through this method have been particularly useful for inspecting PV installations. In this sense, thermographic inspection, which involves using infrared imaging to identify defective modules in PV installations, has gained popularity as a reliable and efficient tool [5].

Thermography allows the identification of temperature distributions in PV modules, enabling the detection of non-uniform distributions. Based on this temperature distribution, it is possible to correlate different problems. Nevertheless, a careful image analysis over a large set of infrared images is needed to perform this identification task. Then, to address this challenge, intelligent and/or autonomous systems have been proposed as possible solutions. These include statistical analysis and, more recently, deep learning-based approaches such as Deep Neural Networks in particular Convolutional Neural Networks (CNN) [7, 9, 10, 12].

Despite the large number of CNN models and their different complexities, reviewing the impact of the information used to train and test these networks is necessary. One of the initial strategies for identifying defective modules involved statistical studies of temperatures in infrared images was demonstrated by Kim et al. [6]. In their work, local and global standard deviations were considered to define defective modules. Additionally, Dotenco et al. [2] proposed a statistical analysis of pixel intensities. Carletti et al. [1] introduced another method, employing a water-filling algorithm for hotspot detection. While these methods showcase innovative approaches, traditional techniques face significant challenges in real-world scenarios.

These methods often lack robustness, struggle to generalize to new, unseen data, and encounter difficulties adapting to variations in image conditions, such as distortions, occlusions, or other unpredictable factors. Deep learning, specifically CNNs, has been used as an alternative solution to the PV module image classification problem. The CNNs have demonstrated significant advantages in handling diverse, large-scale, and complex image classification tasks. In some cases, custom convolutional neural models have been proposed [4, 14, 16]. Also, adaptations and transfer learning to re-train well-known models such as VGG-16 [13], MobileNet [3], AlexNet [7], ResNet [9] and DenseNet [12] have been employed. Datasets, some of which are public, are used for training these models. However, one difficulty with current public databases is their limited size and the imbalance in the number of images per class.

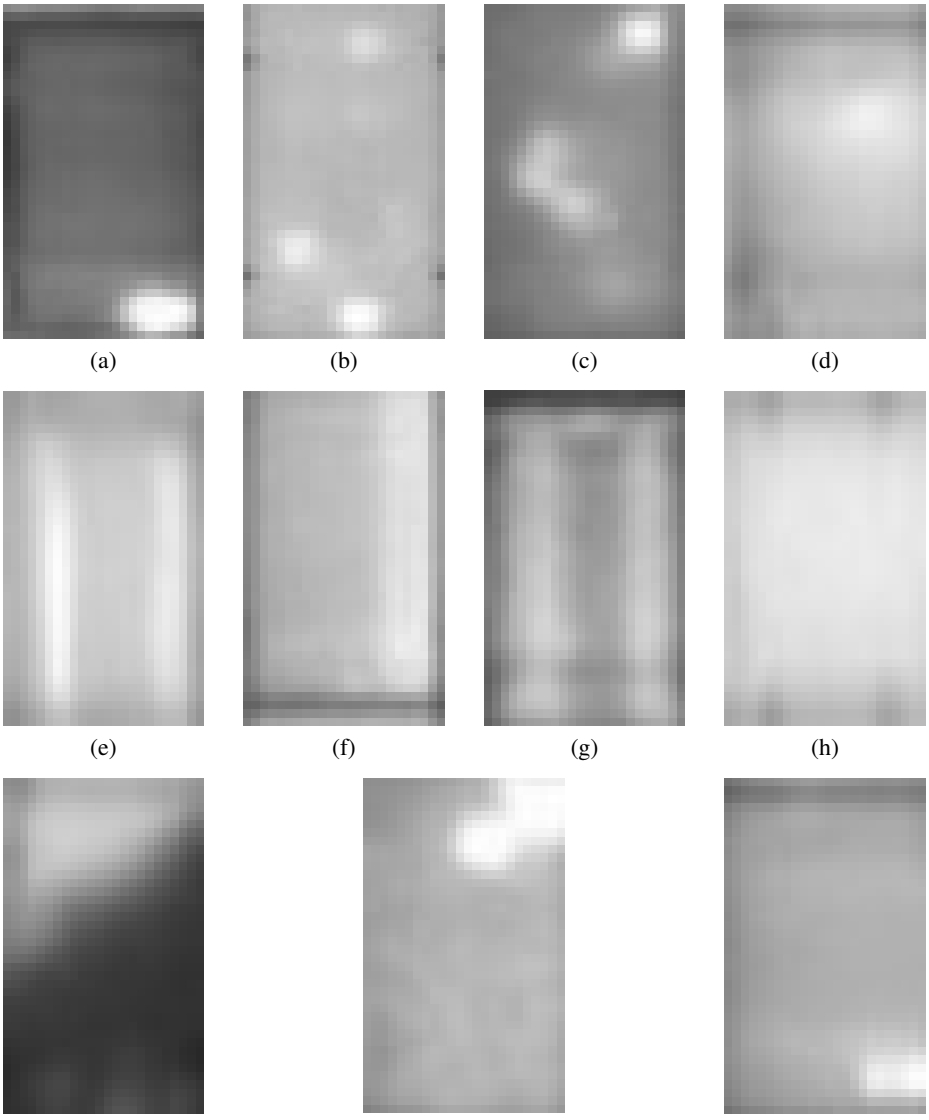

Fig. 1. Example of an image of each class in Millendorf dataset [11]. a) Cell. b) Cell multi. c) Cracked. d) Hotspot. e) Hotspot multi. f) Diode. g) Diode multi. h) Offline. i) Shadowing. j) Soiling. k) Vegetation.

One strategy commonly employed to address these problems is the technique known as data augmentation. Data augmentation consists of applying modifications to existing data, thus creating new instances (synthetic ones) with variations on the original data. In the case of images, geometric transformations such as flipping and rotation operations and brightness modifications have been implemented to generate synthetic images, as reported in the work of Korkmaz [7].

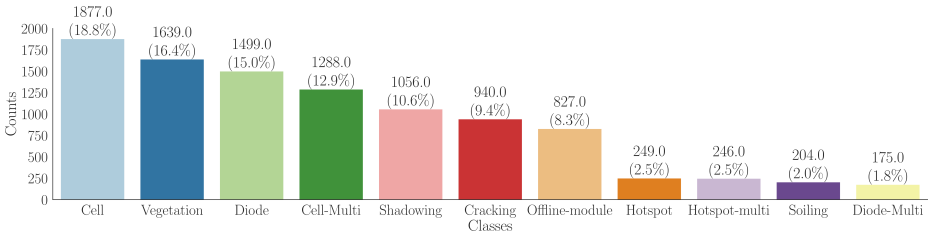

Fig. 2. Distribution of images per class in the dataset [11].

Other works, such as the one presented by Pamungkas et al. [12], use Generative Adversarial Network (GAN) combined with geometric transformations for image generation. However, this combined strategy was outperformed when only geometric data augmentation was used. In this work, we have conducted a comparative analysis of the data augmentation methodologies that some recent works, that address the identification of failures in PV systems, have implemented.

Specifically, we have evaluated the data augmentation methodology of Alves [4], Korkmaz [7], Le [9], and Pamungkas [12], which, in addition, are using the database reported by Millendorf et al. [11]. The analysis performed here primarily aims to assess the impact of data augmentation on these works, since they have reported very different results depending on data augmentation strategies besides the model employed. To ensure a fair comparison, we implemented some of the most recognized and widely accessible deep neural network architectures, modifying only how data augmentation was implemented.

The contribution of this analysis is the evaluation of the effectiveness of different data augmentation techniques to improve the performance of deep learning models in the detection of faults in photovoltaic systems. The remainder of this manuscript is organized as follows. In Section 2 the proposed methodology for comparing training and data augmentation methodologies found in the literature is presented. Section 3 describes the obtained results and their analysis. Finally, in Section 4, the conclusions and future work are presented.

2 Methodology

To assess the impact of data augmentation methodologies, we propose using transfer learning for five well-known deep learning models in the literature: VGG16, ResNet50, MobileNetV2, DenseNet121, and EfficientNetB0. Using widely recognized models, which have proven effective in various classification tasks and can be easily reproduced, the aim is to establish a solid basis for evaluating the proposed methodology. These classification models come with pre-trained weights on the ImageNet dataset [8].

Additionally, the classification models undergo a modification in the last output layer to classify the number of classes of failures present in the thermal image dataset. The employed dataset in this analysis comprises $10,000$ thermal aerial images of PV modules, each measuring 24×40 pixels, with eleven different failure classes. These grayscale images were acquired during various infrared aerial inspections and manually classified by Millendor et al. [11].

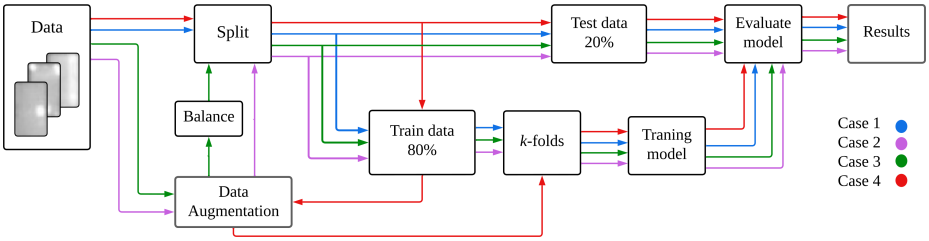

Fig. 3. Proposed methodology to address four cases of data augmentation. Here, k-fold cross-validation is implemented during the training stage of the CNN models. The best-performing model from the k-fold cross-validation is used to evaluate the models.

Depending on the type of failure, each PV module image shows a different thermal distribution, where areas with higher temperatures are highlighted in bright colors, while areas with lower temperatures are highlighted in dark colors. An image for each class is shown in Fig. 1. Due to varying fault frequencies, the dataset exhibits an imbalance in the number of images per class. The classes are Cell, Cell multi, Cracked, Hotspot, Hotspot multi, Diode, Diode multi, Offline, Shadowing, Soiling and Vegetation. Cell failure has the highest incidence at 18.8%, while multiple diode bypass, soiling, hotspot, and multiple hotspots represent the lowest incidence classes at 1.8%, 2%, 2.5%, and 2.5%, respectively.

The distribution of images in each failure class is depicted in Fig. 2. Data augmentation is employed to address imbalance issues. Based on current literature, we have proposed a methodology that incorporates four distinct data treatments: i) No data augmentation, ii) fully augmented data set, iii) data augmentation and post-augmentation class balance, and iv) data augmentation only in the training set. Furthermore, a k-fold cross-validation with $k = 10$ is implemented during the training model stage in all cases.

The k-fold cross-validation involves partitioning the training subset into k distinct folds, where one fold serves as the validation dataset while the remaining folds collectively form the training set of the model. This process is iterated k times, with each fold acting as the validation subset. Using k-fold cross-validation provides a robust metric for assessing the model's generalization capabilities.

The whole methodology is illustrated in Fig. 3, where each color line represents each previously described case of study. In the first case, the dataset is used without augmentation, ignoring the imbalance in the number of images per class, as used in the works by Alves-Fonseca [4], Li [10], Le [9] and Pamungkas [12]. Then, the dataset is split into two subsets: 80% for training and 20% for testing.

In the second case, data augmentation is applied, including flipping, rotation, and brightness operations to increase the number of samples as described in the work of Pamunkgas [12]. Brightness operations involve decreasing and increasing each pixel intensity value by ± 30. Subsequently, flipping and rotation operations are applied to original and brightness-modified images. This augmentation generates 11 additional images per original image. Fig. 4 shows representative examples of the generated synthetic images. The resulting set, which comprises 120,000 images with the original class imbalance, is split into two subsets: 80% for training and 20% for testing.

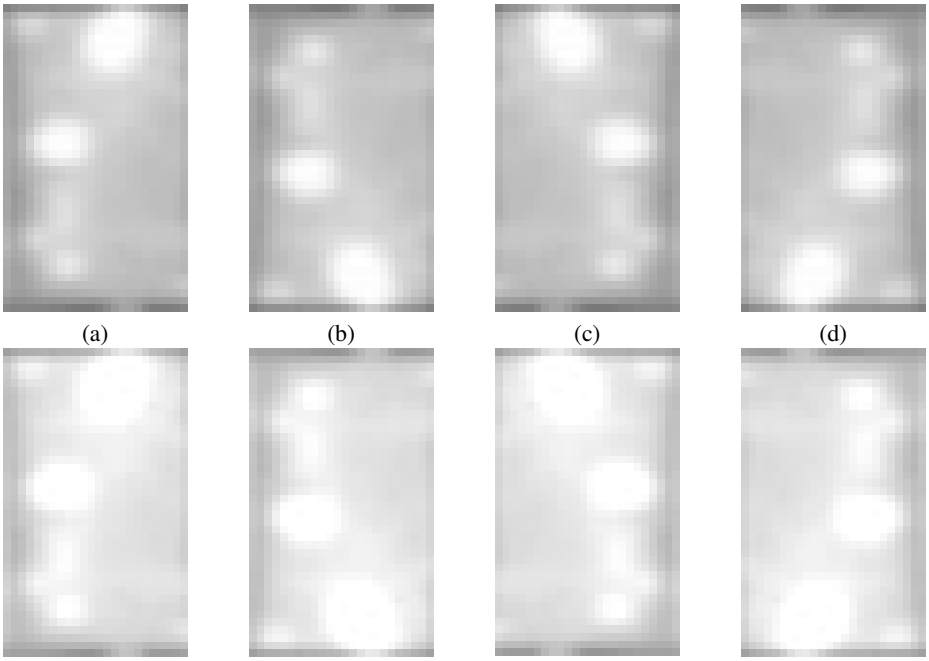

Fig. 4. Example of data augmentation including geometric and brightness operations. a) Original image. b) Vertical flip. c) Horizontal flip. d) 180 degrees rotation. e) Brightness increase. f-h) Vertical, horizontal, and 180 degrees rotation with brightness increase. i) Decrease of brightness. j-l) Vertical, horizontal and 180 degrees rotation with a decrease in brightness.

Similar to the second case, data augmentation is carried out using the same geometric operations and brightness modifications for the third case. However, unlike case two, a data balance is performed, ensuring each class has an equal number of elements. Uniformly random image selection of the augmented dataset is performed to balance the classes. Each class contains the same number of images as the smallest class in the augmented dataset. Then, 1,100 images from each class were uniformly randomly chosen from a final set of 120,000 images.

These images are divided into training and test subsets following an 80-20 proportion. In the fourth scenario, data augmentation is exclusively implemented on the training subset. Then, a training set of 96,000 images is obtained. It is worth mentioning that for all cases, the best-performing model from the k-fold cross-validation is tested on the dedicated test subset. The overall performance of the models has been evaluated using widely known metrics such as Accuracy, Precision, Recall, and F_1-score [15].

3 Results

In this section, the results obtained from the proposed methodology are presented. A desktop computer with Core i9 processor, NVIDIA GPU GeForce GTX 3090 with 24 GB of VRAM, and 64 GB of RAM was used to extract features and train the models.

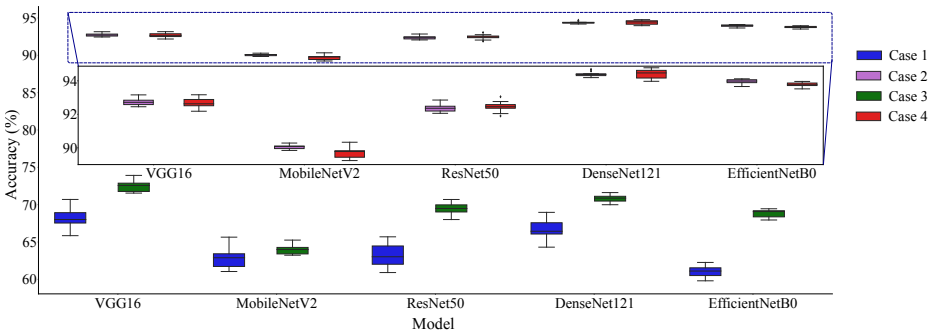

Fig. 5. Comparison of the training performance, after the 10-fold cross-validation implementation, of each of the four cases using the five deep neural networks models.

The implementation was carried out in Python 3.8 and PyTorch framework 1.13.1. In all four cases, the AlexNet model was trained using the Adam optimizer on 25 epochs with a learning rate of 0.0001, a batch size of 32, and categorical cross-entropy as a loss function. For each of the models used in the fine-tuning process, a linear layer of the same size as the last layer of the original network was added, specifically adjusted to match the number of classes required in the classification task.

Fig. 5 provides a graphical representation of the accuracy distribution obtained by each model (VGG16, ResNet50, MobileNetV2, DenseNet121, and EfficientNetB0) in the four different data augmentation cases. The data presented in Fig. 5 indicates an improvement in the performance of all deep neural network models when data augmentation is utilized (cases 2, 3, and 4) compared to the scenario where no data augmentation is applied (Case 1). However, it is worth highlighting that the accuracy reaches only about 75% when data augmentation and post-augmentation class balance are employed (Case 3). This may be attributed to the fact that data sampling is performed after augmentation, which may result in an inadequate representation of each class during the model training phase.

It is crucial to mention that this issue is observable regardless of the model type. Furthermore, it can be observed that the measure of central tendency is higher for Cases 2 and 4 while also presenting a comparatively lower measure of dispersion. The average accuracy values and their corresponding standard deviations obtained through 10-fold cross-validation for each model in four training cases are shown in Table 1. It stands out that DenseNet121 achieved the highest average accuracy, reaching a remarkable 94.4% in Case 4. Also, it is observed that EfficientNetB0 exhibited the lowest standard deviation in Case 4.

In Case 1, the performance is less than 70% in accuracy, so it can be suggested that the imbalance and the small number of images affect the generalization ability. Since the standard deviation is greater than 1 in most cases, it can be concluded that there is significant variability depending on the fold employed. By employing data augmentation in Case 2, an increase of almost 28% in average accuracy is observed, while maintaining a standard deviation of less than 0.24. It is crucial to note that including images to balance the classes (Case 3) leads to a further increase in the average accuracy, as evidenced by comparing Case 1 with Case 3.

Table 1. Mean accuracy (μ_A) and standard deviation (σ_A) in the training of k-folds, presented as percentages.

	μ_A				σ_A			
Model	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 1**	**Case 2**	**Case 3**	**Case 4**
DenseNet121	66.62	**94.39**	70.83	**94.4**	1.36	0.15	0.56	0.28
EfficientNetB0	61.05	93.95	68.82	93.76	0.76	0.15	0.53	0.14
MobileNetV2	62.78	90.06	64.01	89.71	1.38	0.13	0.7	0.32
ResNet50	63.28	92.37	69.44	92.45	1.72	0.23	0.81	0.33
VGG16	**68.3**	92.75	**72.49**	92.67	1.42	0.23	0.78	0.32

Table 2. Classification report showing the percentage performance of the best models on the test subset across four different cases.

Model	**Case 1**					**Case 2**				
	Acc	**P**	**R**	**F1**	**#**	**Acc**	**P**	**R**	**F1**	**#**
VGG16	69.40	**69.03**	63.89	**65.78**	2,000	95.41	94.74	95.02	94.86	13,000
ResNet50	60.25	63.13	51.31	52.69	2,000	97.38	97.08	97.47	97.25	13,000
DenseNet121	**70.35**	67.77	**64.25**	65.40	2,000	97.72	97.73	97.81	97.76	13,000
MobileNetV2	67.05	63.18	59.85	60.84	2,000	94.50	94.60	94.68	94.61	13,000
EficientNetB0	67.15	65.53	63.09	63.76	2,000	**98.52**	**98.78**	**98.54**	**98.52**	13,000
Model	**Case 3**					**Case 4**				
	Acc	**P**	**R**	**F1**	**#**	**Acc**	**P**	**R**	**F1**	**#**
VGG16	**76.58**	**76.09**	**76.58**	**76.58**	4,620	73.90	69.51	67.86	68.31	2,000
ResNet50	74.65	75.22	74.65	74.60	4,620	73.30	72.33	69.28	69.89	2,000
DenseNet121	75.84	75.31	75.84	75.35	4,620	73.25	**73.75**	70.06	**70.82**	2,000
MobileNetV2	66.34	66.25	66.34	66.11	4,620	73.55	71.96	**70.46**	70.79	2,000
EficientNetB0	74.31	74.61	74.31	73.53	4,620	**74.55**	71.75	69.94	70.17	2,000

The differences between Case 2 and Case 4 do not seem to be significant, at least in terms of performance during training.Through cross-validation, the model with the highest accuracy was identified. This model was tested using the test subset, and the results of the best models are detailed in Table 2. In Case 1, the top models were VGG-16 and DenseNet, achieving accuracy close to 70%, with values above 64% in the precision, recall, and F_1 metrics. In Case 2, the most outstanding model was EfficientNetB0, achieving values above 98% in all metrics. For the third case, VGG-16 proved to be the best model with a performance of 76% in all metrics.

DenseNet121 emerged as the best model for the fourth case, presenting two of the highest values in the accuracy and F_1-score metrics, with 73% and 70.82%, respectively. The notably high values are highlighted in bold in Table 2. It can be seen that, for Case 2, all metrics exceed 94%, while for the other cases, they do not reach the average value of 77%. This analysis highlights the remarkable differences in the performance of the models in different training configurations. In cases 1, 2, and 3, no significant differences are observed between the accuracy values reported during training and those obtained in the test phase.

Table 3. Comparison of works employing the dataset reported by Millendorf et al. [11].

Reference	Case	Model	DATe	DATr	Balance	Accuracy
Alves Fonseca 2021 [4]	1	CNN	✗	✗	✗	66.43
Le 2021 [9]	1	Ensamble	✗	✗	✗	85.9
Li 2023 [10]	1	Transformer	✗	✗	✗	**88.5**
Pamungkas 2023 [12]	1	UDenseNet	✗	✗	✗	65.9
This analysis	1	DenseNet121	✗	✗	✗	70.35
		VGG16	✗	✗	✗	69.40
Pamungkas 2023 [12]	2	UDenseNet	✓	✓	✗	96.65
This analysis	2	EfficientNetB0	✓	✓	✗	**98.78**
Korkmaz 2022 [7]	3	CNN Multiscale	✓	✓	✓	**93.51**
This analysis	3	VGG-16	✓	✓	✓	76.58
This analysis	4	EfficientNetB0	✗	✓	✗	**74.55**

However, in Case 4, significant discrepancies of at least 20% are evident. This disparity can be attributed to the fact that the data augmentation in Case 4 is performed exclusively on the training set, after its separation from the test set. As a consequence, the test set contains images that the model has never previously encountered, which may influence the evaluation metrics. This phenomenon suggests that by employing data augmentation on the dataset intended for testing, the illusion could be generated that the model has exceptional generalization ability, when in fact it may be suffering from overfitting to specific patterns present in the training set.

Contrarily, it is found that applying data augmentation exclusively on the training set (Case 4) results in a 4% increase in generalization ability compared to Case 1. This supports the notion that data augmentation on the training set can have a positive impact on the model's ability to generalize to previously unseen data, even when faced with a test set with unpublished images. In addition, a comparison was made between the results obtained in this study and those reported in the literature for models with the best accuracy values in each specific case shown in Table 3. Accuracy is the only metric employed, since the state of the art works only report this metric. The findings are as follows:

– Case 1. The Vision Transformer-based model proposed by Li et al. [10] stands out, outperforming conventional Convolutional Neural Networks (CNN), despite the class imbalance. This result suggests that, under imbalance conditions, transformer-based architectures can offer better generalization capability compared to conventional CNNs.

– Case 2. The EfficientNetB0 model outperforms the model of Pamungkas et al. [12] by almost 3%, even though the model used is less complex. It is important to note that higher accuracy does not always reflect the true generalization ability of a model, as model complexity also plays a crucial role.

Table 4. Comparison of training time of each model.

	Time by model (min)					
Case	**VGG16**	**MobileNet**	**ResNet**	**DenseNet**	**EfficientNet**	μ
I	15	5	9	10	8	9.4
II	235	94	146	158	128	152.2
III	37	14	23	25	20	23.8
IV	189	72	116	149	105	126.2

– Case 3. The model reported by Korkmaz et al. [7] shows exceptional performance, reaching a value of 93.51%, significantly outperforming the best model proposed in this study. However, a discrepancy in the reported accuracy is observed, as the work of Li et al. [10] when replicating the model, obtained an accuracy of 85.1% employing the Case 1 method. These discrepancies highlight the importance of reproducibility and consistency in presenting results in the scientific literature.

– Case 4. By implementing data augmentation exclusively on the training set, a 4% increase in generalization ability is observed compared to Case 1, although it is still not optimal classification. These results emphasize the importance of exploring additional approaches or making adjustments to improve the generalization ability substantially.

Additionally, Table 4 presents the time required for 10-fold cross-validation in the training stage for each model and the four data augmentation cases. As it can be observed, in Case 1, no model exceeded 15 minutes of training. In contrast, Case 2 showed a mean training time of 152.2 minutes, where the VGG16 model achieved a maximum of 235 minutes, whereas the MobileNet achieved the minimum time (94 minutes). Note that Case 2 denotes the scenario with the highest training time.

It is important to note that the relationship between the number of images used and the resulting training time is not linear. When comparing Case 1 and Case 2, in which the number of images used differs by a factor of 12 due to increased data, the training time increases by at least 15 times. In contrast, when Case 1 and Case 3 are compared, where the number of images used differs by a factor of 2.3, the training time increases by at least 2.4 times. Then, it demonstrates that a linear increase in the number of images used does not result in a linear increase in training time. Among all the implemented models, VGG-16 consistently exhibited the longest training times, whereas MobileNetV2 consistently demonstrated the shortest training times.

This discrepancy can be attributed to the substantial contrast in the number of trainable parameters. VGG-16 possesses 138.4 million parameters, whereas MobileNetV2 possesses only 3.5 million parameters, amounting to approximately 40 times fewer parameters. In summary, it has been observed that data augmentation can be a highly effective technique to enhance the performance of deep learning models. For instance, in the work of Pamungkas [12], it was found that augmenting the training and testing sets significantly improved model performance by approximately 30%, compared to no data augmentation.

However, it is important to note that this performance may be biased due to the incorporation of prior knowledge (augmented data) in the testing process, which could lead to overfitting and reduce the model's generalization capability. Nevertheless, it is possible to improve model performance even with data augmentation limited to the training set, as shown by the performance improvement of the EfficientNetB0 model in Case 4 compared to Case 1.

4 Conclusions

In this study, the analysis of the impact on the performance of several data augmentation methodologies reported in the literature was done. Data augmentation was found to be a valuable tool for improving the generalization capability of the model, as well as for achieving a balance in the class distribution. Nonetheless, what was emphasized was that the application of these techniques on both training and test datasets may generate the illusion of an improvement in classification ability.

Crucially, this apparent improvement does not necessarily guarantee the generalization capacity of the model facing new data. In addition, it should be noted that the increase in the number of images used for training prolongs the time required, although this increase does not follow a linear relationship. As a perspective for future work, a more exhaustive exploration of models based on Transformers is contemplated, exploring their analysis and evaluation in more detail. Additionally, exploring more advanced data augmentation techniques, such as synthetic data generation using Generative Adversarial Networks (GANs), is also considered.

Acknowledgments. Thanks to the scholarship 05579 of Becas Nacionales provided by CONAHCYT.

References

1. Carletti, V., Greco, A., Saggese, A., Vento, M.: An intelligent flying system for automatic detection of faults in photovoltaic plants. Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 5, pp. 2027–2040 (2019) doi: 10.1007/s12652-019-01212-6
2. Dotenco, S., Dalsass, M., Winkler, L., Würzner, T., Brabec, C., Maier, A., Gallwitz, F.: Automatic detection and analysis of photovoltaic modules in aerial infrared imagery. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 1–9 (2016) doi: 10.1109/WACV.2016.7477658
3. Dunderdale, C., Brettenny, W., Clohessy, C., van-Dyk, E. E.: Photovoltaic defect classification through thermal infrared imaging using a machine learning approach. Progress in Photovoltaics: Research and Applications, vol. 28, no. 3, pp. 177–188 (2019) doi: 10.1002/pip.3191
4. Fonseca-Alves, R. H., Deus-Júnior, G., Marra, E. G., Lemos, R. P.: Automatic fault classification in photovoltaic modules using convolutional neural networks. Renewable Energy, vol. 179, pp. 502–516 (2021) doi: 10.1016/j.renene.2021.07.070
5. Grimaccia, F., Leva, S., Dolara, A., Aghaei, M.: Survey on pv modules' common faults after an O&M flight extensive campaign over different plants in italy. IEEE Journal of Photovoltaics, vol. 7, no. 3, pp. 810–816 (2017) doi: 10.1109/JPHOTOV.2017.2674977

6. Kim, D., Youn, J., Kim, C.: Automatic photovoltaic panel area extraction from UAV thermal infrared images. Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, vol. 34, no. 6, pp. 559–568 (2016) doi: 10.7848/ksgpc.2016.34.6.559
7. Korkmaz, D., Acikgoz, H.: An efficient fault classification method in solar photovoltaic modules using transfer learning and multi-scale convolutional neural network. Engineering Applications of Artificial Intelligence, vol. 113, pp. 104959 (2022) doi: 10.1016/j.engappai.2022.104959
8. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM, vol. 60, no. 6, pp. 84–90 (2017) doi: 10.1145/3065386
9. Le, M., Luong, V. S., Nguyen, D. K., Dao, V. D., Vu, N. H., Vu, H. H. T.: Remote anomaly detection and classification of solar photovoltaic modules based on deep neural network. Sustainable Energy Technologies and Assessments, vol. 48, pp. 101545 (2021) doi: 10.1016/j.seta.2021.101545
10. Li, S., Chen, H., Zhang, A., Gong, C., Menhas, M. I., Liang, W., Wang, Z., Yang, N.: Photovoltaic panel fault detection and diagnosis based on a targeted transformer-style model. IEEE Transactions on Industry Applications, vol. 60, no. 1, pp. 1814–1826 (2024) doi: 10.1109/TIA.2023.3322688
11. Millendorf, M., Obropta, E., Vadhavkar, N.: Infrared solar module dataset for anomaly detection. In: International Conference on Learning Representations, pp. 1–5 (2020)
12. Pamungkas, R. F., Utama, I. B. K. Y., Jang, Y. M.: A novel approach for efficient solar panel fault classification using coupled udensenet. Sensors, vol. 23, no. 10, pp. 4918 (2023) doi: 10.3390/s23104918
13. Pierdicca, R., Malinverni, E. S., Piccinini, F., Paolanti, M., Felicetti, A., Zingaretti, P.: Deep convolutional neural network for automatic detection of damaged photovoltaic cells. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII–2, pp. 893–900 (2018) doi: 10.5194/isprs-archives-xlii-2-893-2018
14. Po-Ching-Hwang, H., Cheng-Yuan-Ku, C., Chi-Chang-Cha, J.: Detection of malfunctioning photovoltaic modules based on machine learning algorithms. IEEE Access, vol. 9, pp. 37210–37219 (2021) doi: 10.1109/ACCESS.2021.3063461
15. Vujovic, Z. D.: Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 599–606 (2021) doi: 10.14569/ijacsa.2021.0120670
16. Zefri, Y., Sebari, I., Hajji, H., Aniba, G.: Developing a deep learning-based layer-3 solution for thermal infrared large-scale photovoltaic module inspection from orthorectified big UAV imagery data. International Journal of Applied Earth Observation and Geoinformation, vol. 106, pp. 102652 (2022) doi: 10.1016/j.jag.2021.102652