

EMERGING TECH CONFERENCE – Edge Intelligence

Volume 03, 2024, Pages 59 – 65

Proceedings of Emerging Tech Conference:
Edge Intelligence 2024

Computing the Cleanness of the Photovoltaic (PV) Panels

A. Tsagkaropoulos^a, C. Vasilakis^a, G. Venitourakis^a, P. Tz. Amrou^b,
G. Konstantoulakis^b, P. Golemis^b, D. Reisis^{a*}^aComputer Center, Physics Dpt, National & Kapodistrian University of Athens, Greece^bInAccess Networks, 12 Sorou Str, 15125, Athens, Greece* Corresponding author. Email address: dreisis@phys.uoa.gr**Abstract**

Among the major factors that may significantly affect the power production of a photovoltaic (PV) power plant is the performance of the PV panels. Their performance depends on their maintenance and especially on keeping their surface clean to allow the panels to absorb the maximum power of the solar irradiance. To improve the maintenance process especially in remote locations, the current work focuses on an edge computing system using a camera and a Neural Network (NN) to decide the cleanness of the panels. The proposed system determines whether the panel is either clean or not. The proposed system utilizes a proprietary dataset. The execution of the inference on a Raspberry Pi Zero 2W validates the design, the edge computing performance and the accuracy.

Keywords: Neural Network, Photovoltaic (PV) panels, Edge Computing.

1 Introduction

A significant fraction of the energy produced by the renewable energy sources comes from the photovoltaic (PV) power plants. The PV parks' designers utilize the irradiance and count upon clear skies or reduced cloud coverage to maximize the efficiency of the installation. The efficiency depends also on the cleanness of the PV panels, which are cleaned on a three or six months schedule while in between the panels may pile up either dirt or particles such as dust, water drops, snow and sand. Engineers focused on estimating the performance losses in these cases. The authors of [1] have conducted experiments which showed that these effects may reduce the efficiency of the panel up to 85%. Since such phenomena will appear in more than one panels the deterioration can affect the entire park performance and the solution is to notify the park management group so that a cleaning process will restore the levels of power productivity. A system with the ability of estimating whether the panels are soiled, will provide an effective solution especially for the PV parks in remote locations.

Aiming at an effective solution to the problem of alerting the PV park management when the PV panel(s) is(are) no longer clean, this work proposes an edge computing system using machine learning on the PV panel images and deciding the panels cleanness. The motivation for this work came from the last phase of the Archon project [2] forecasting the irradiance and the power of a PV park. During the testing the Archon's research and development team noticed that rain drops and dirt could affect the novel park's power forecasting system. Consequently, they noted that the forecasting system assumes clean PV panels

and that plant's manager has to be aware and take care of this issue. The literature study showed the significance of the problem [1] as well as the complete absence of approaches and/or techniques addressing the problem of categorizing the panel's status. There is only one paper [3] referring to binary classification of solar panels to dusty or clean, employing a CNN (Convolutional Neural Network) named SolNet.

Given the lack of approaches and datasets for solving the problem and our target for a solution effective with respect to the accuracy and efficient with respect to computing resources first, our work develops a relatively small proprietary dataset. Second, it considers a variety of neural networks (NNs) in an effort to identify the extremely advantageous of these models while these perform on the limited size dataset. Considering that a significantly larger dataset will be exploited in the following research stages, this paper selects nineteen (19) neural networks to work on the current dataset, it compares their performance and presents the results of this early research stage. The paper is organized with the following section describing the technique. Section 3 presents the accuracy results achieved by the performance of the nineteen (19) models and Section 4 discusses the following steps of this research.

2 The NN Based Technique

The current section gives the details of the architecture starting with the dataset, then with the details of the training process, the data augmentation, the optimizers and the schedulers.

2.1. The Dataset

In the first stage of the current work, we chose as the first source for the dataset the deep-solar-eye dataset [4]. The considerably small variation of conditions though led to overfitting and it has triggered the need of another source of training data. Compared to the latter the second dataset is very small, and it constructed by gathering images from arbitrary Google sites. These images are explored to evaluate the model in actual clean or soiled PV panel cases. The train/test split was 80-20. For the continuation of this work the research team will install a network of cameras in available PV plants and it will develop an appropriate dataset expected to include a wide variety of images and soiling cases.

2.2. Training

For this research, we consider two training schemes. A first approach follows the standard training, validation and test split with a 72:8:20 % split for each subset correspondingly. A second, more sophisticated scheme utilizes the k-fold cross validation [5] method. The latter first divides the training dataset to k subsets called folds. The training scheme starts by training the model k times per epoch, each time excluding one of the k folds and using it as a validation subset. Each epoch corresponds to $k - 1$ training iterations over the training subset and one iteration over the training subset, mainly used for debugging purposes. Upon the completion of each epoch we validate the model in the validation subset. The application of the latter training scheme to the models improves the prevention of overfitting and presents an overall faster convergence [6]. Both schemes also utilize transfer learning, since all the models used in this study are well-known models and they are pretrained on the ImageNet dataset [7]. All layers except for the output layer of the pretrained models are frozen and therefore not trained. The design of the output layer realizes the functionality of a linear layer of two output features fed to a SoftMax activation function, which are rendered as the two probabilities for soiled and clean panels. The transfer learning technique allows for rapid training of the models, while keeping the high accuracy of the

pretrained model on the ImageNet dataset. The loss function befitting to the problem of the binary classification of images in two classes is the Cross-Entropy Loss function, as the models output two logits, one for each class of soiled and clean panels.

The limited size of the dataset and the small variations among the input data has led us to experiment with data augmentation. Augmenting the input data is common practice for such occasions, while incorporating those methods will increase the gains with respect to training efficiency and accuracy and will reduce the probability of overfitting occurrence. These methods include resizing, cropping, random flipping and rotating, scaling and normalizing the RGB values with the same mean and standard deviation of the ImageNet training dataset.

2.3. Loss Function & Class Probabilities

The quantification of classifier layer's results is accomplished by using the Cross Entropy Loss function. A preprocessing step involves the application of the softmax function to the model's results to enhance the distinctiveness of the probabilities. The Cross Entropy Loss function accepts these probabilities and it computes a value of *loss* with respect to the reference level. This loss value gives the difference from the computation result to the previously known result: either clean or soiled PV panel. The main reason for this approach is that in the future work it may be explored to analyze the class "soiled" into multiple classes. At this early stage the entire architecture is designed for binary classification, mainly due to the limited dataset and consequently, the model outputs two logits.

2.4. Data Augmentation

We incorporate data augmentation techniques to lessen the effect of the dataset's limiting factors mentioned in Section 2.1. The small size of the dataset decreases the training efficiency and results in overfitting of the model on the training data. Thus, the dataset was enriched by methods concerning random image rotations, flipping and cropping. At the same time, we apply the novel TrivialAugment [8] transformation to the images, which also affects the saturation, the contrast and the color profile. We note here that, in many cases, the pretraining of the models uses normalization and scaling of the RGB values. For these models we have to use the normalization and the scaling of the images with the values that have been used during the pretraining. The frame sizes of model's inputs are aggregated in the Results Section in **Table 1**.

2.5. Optimizers & Schedulers

We train models by using one of the following three optimizers. The first optimizer is the SGD (Stochastic Gradient Descent) [9], which updates the model's weights by taking into account the gradient of the loss function with respect to the weights. This is the most trivial optimizer that we have used in our experiments. The hyperparameters corresponding to the SGD were: 0.001/0.01 for the learning rate and, optionally, 0.9 for the momentum. The second optimizer that we applied is the RMSProp, which improves the weight change stability. Its hyperparameters were: 0.01 for the learning rate, 0.99 for the alpha and 1e-08 for the epsilon. The third optimizer is the Adam algorithm [10] that facilitates the optimization of the models' weights. The Adam's hyperparameters were: 0.001 for learning rate, (0.9, 0.999) for betas and 1e-08 for epsilon.

To further enhance the efficiency and consistency of the training procedure we added to the process the

use of one (out of two) scheduler. A scheduler reduces the learning rate whenever the advancement in training stalls. We specify the loss as the metric that evaluates the learning progress. The first scheduler is the StepLR, which decreases the learning rate of the optimizer by a specified amount every constant number of epochs. The ReduceLRonPlateau is the second scheduler: its application will reduce the learning rate by a specified amount if the loss value reduction stalls for a “patience” amount of epochs. To explain the need for a scheduler when the training fails to result in loss reduction: the idea is to let the process follow incrementally small advancements and hence, find local minima in multidimensional spaces.

3 The Results Using Different Models

This section presents the accuracy results of the technique with the use of nineteen different models in the same tests. In the following table the first column shows the nineteen models that we used. These are sorted with respect to the accuracy results that they achieved during the tests, which are listed in the second column. The third column presents the top 1 accuracy that each model achieved with the ImageNet dataset. For sake of results completeness the fourth column displays the frame size that we used with each model. The fifth column gives the number of parameters of each model.

The first in the list, namely the ShuffleNet_V2_X1_5, shows to be the most efficient model with respect to both the accuracy results as well as the number of parameters needed. This high ranking though may be due to the small dataset used. Moreover, the models ranked from 2 to 4 present an efficiency relatively close to the ShuffleNet_V2_X1_5. Furthermore, taking into account that the execution time of the entire panel-cleanness application is practically indifferent to the application (it can be quite slow), the number of parameters plays a significant role to the cost of the edge device. Hence, the low cost implementation of these four models (the upper four in the rank) is a fact that makes them an attractive solution for edge applications.

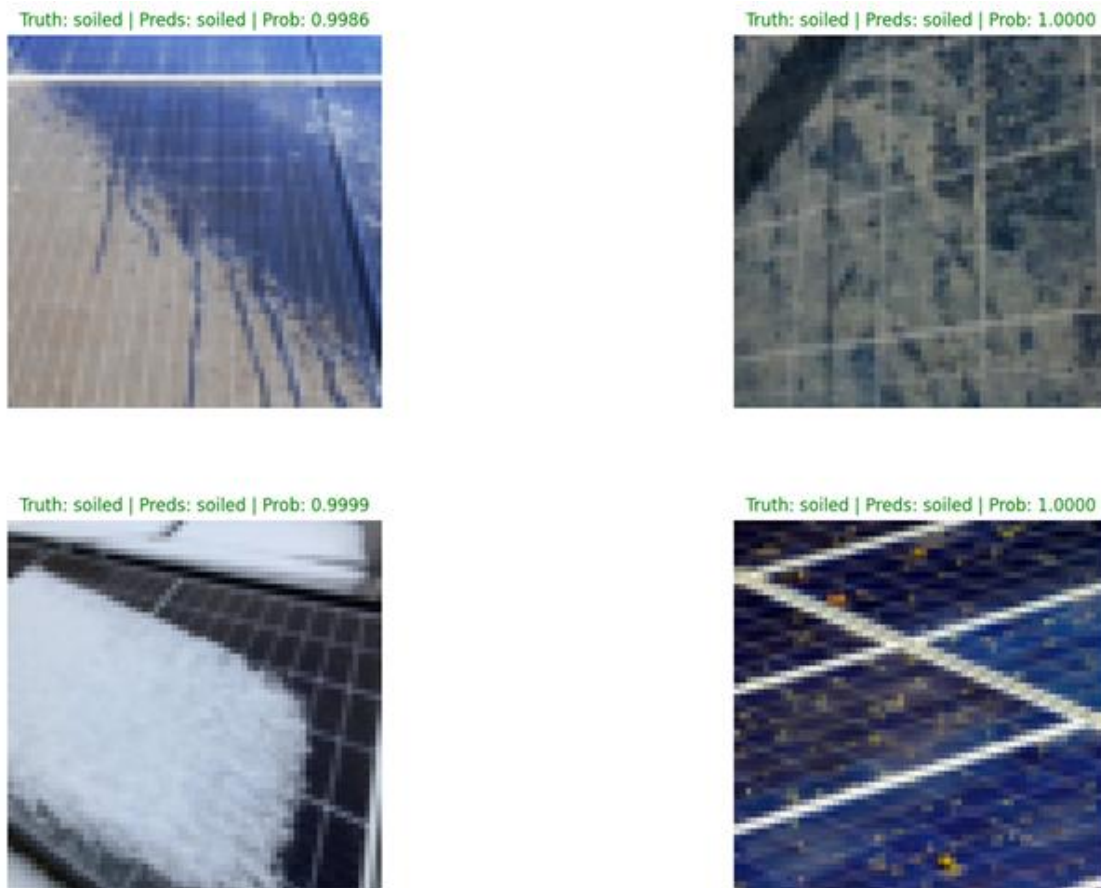
Table 1: Different NN models Ranked with respect to their Accuracy Results

Model Name	Test Accuracy (%)	Acc@1 – ImageNet (%)	Frame Size	Params
ShuffleNet_V2_X1_5 [11]	93.97	72.996	224	3.5M
ResNet34 [12]	91.00	73.314	224	21.8M
ShuffleNet_V2_X2_0 [11]	90.36	76.23	224	7.4M
EfficientNet_B2 [13]	89.30	80.608	288	9.1M
EfficientNet_V2_M [14]	88.89	85.112	480	54.1M
EfficientNet_V2_L [14]	88.50	85.808	480	118.5M
MobileNet_V3_Large [15]	88.45	75.274	224	5.5M
EfficientNet_B7 [13]	88.30	84.122	600	66.3M
MobileNet_V3_Small [15]	87.65	67.668	224	2.5M
EfficientNet_B7 [13]	88.30	84.122	600	66.3M
MobileNet_V3_Small [15]	87.65	67.668	224	2.5M
EfficientNet_B0 [13]	87.34	77.692	224	5.3M

EfficientNet_B1 [13]	86.90	79.838	240	7.8M
ResNet50 [12]	86.02	76.130	232	25.6M
ResNet18 [12]	85.79	69.758	224	11.7M
EfficientNet_B3 [13]	85.77	82.008	300	12.2M
MobileNet_V2 [16]	85.51	71.878	224	3.5M
EfficientNet_B6 [13]	83.73	84.008	528	43.0M
EfficientNet_V2_S [14]	83.78	84.228	384	21.5M
ShuffleNet_V2_X1_0 [11]	81.80	69.362	224	2.3M
ShuffleNet_V2_X0_5 [11]	79.57	60.552	224	1.4M

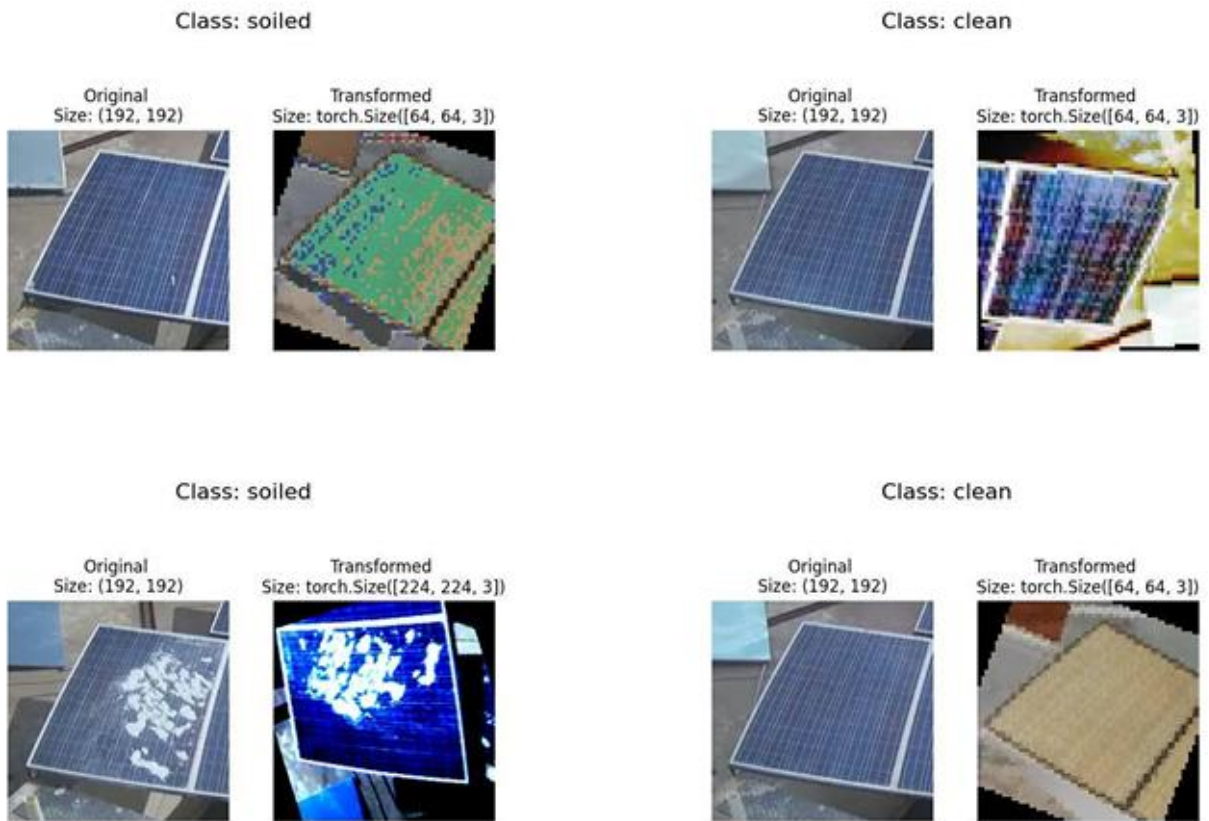
We include two images presenting the results from the inference. Figure 1 depicts these examples.

Figure1: Results from Inference



Furthermore, we give an example of the data augmentation, which is applied only during the training process. The images show the two different classes in two different cases. Figure 2 depicts these examples.

Figure2: Results from Data Augmentation



4 Future Work

This paper has presented an edge computing system that decides whether a (or as many as the camera locates) PV panel is clean or soiled. The research of the system is ongoing and currently we focus mainly on the lack of a dataset and the creation of a new dataset with a very large number of images. For this initial stage we have employed nineteen different models that have been tested to sort them according to their efficiency with respect to the accuracy results and the number of parameters.

Our future work will focus on developing a dataset by installing cameras at PV plants, which will be located at various locations. The variety of locations will cover a multiple of soiling cases. These include dirt, dust, pollen, bird droppings, snow or ice, desert sand, etc. Moreover, we will improve the architecture to let the user choose the level of cleanness of the panels. This can be achieved by enhancing the proposed system so that it will guide the user during the choice of the cleanness threshold.

References

- [1] S. Shaharin, K. S. Atul, M. M. Mior, Mokhtar and A. B.-R. Mohammed, "Influence of Dirt Accumulation on Performance of PV Panels," Energy Procedia, vol. 50, pp. 50-56, 2014.
- [2] "Archon," In Access Nteworks, 2022. [Online]. Available: <http://archonproject.eu/>.

- [3] M. S. H. Onim, Z. M. M. Sakif, A. Ahnaf, A. Kabir, A. K. Azad, A. M. T. Oo, R. Afreen, S. T. Hridy, M. Hossain, T. Jabid and M. S. Ali, "SolNet: A Convolutional Neural Network for Detecting Dust on Solar Panels," *MDPI Energies*, vol. 16, 2023.
- [4] S. Mehta, A. P. Azad, S. A. Chemmengath, V. Raykar and S. Kalyanaraman, "DeepSolarEye: Power Loss Prediction and Weakly Supervised Soiling Localization via Fully Convolutional Networks for Solar Panels," *Arxiv*, 2018.
- [5] M. Stone, "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111-147, 1974.
- [6] D. S. Soper, "Greed Is Good: Rapid Hyperparameter Optimization and Model Selection Using Greedy k-Fold Cross Validation," *MDPI Electronics*, 2021.
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255.
- [8] S. G. Müller and F. Hutter, "TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation," 2021.
- [9] Sutskever, Ilya, J. Martens, G. Dahl and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.
- [10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.
- [11] N. Ma, X. Zhang, H.-T. Zheng and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," 2018.
- [12] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2015.
- [13] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2020.
- [14] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," 2021.
- [15] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le and H. Adam, "Searching for MobileNetV3," 2019.
- [16] S. Mark, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2019.