BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY
Excel@FIT 2024
http://excel.fit.vutbr.cz

Detection and Classification of Photovoltaic Panel Defects from a Drone Thermal Imaging Camera

Filip Haužvic*

Abstract
The paper describes the processing of thermal images of photovoltaic power plants captured by a drone. In contemporary solutions, the images are analyzed manually, where an expert in thermal imaging searches for defects in individual panels. This approach is very time-consuming, and introducing some level of automation could ease the process. Therefore, I trained and utilized a **U-Net** model that detects hot spots in the images. To visualize and present the defects to the user, I designed and created a web-based application that highlights them in a complete orthomosaic of the photovoltaic power plant. Within the application, a user can annotate PV panels in the power plant and manually remove, or add any defect. When the plant is wholly annotated, an export to a spreadsheet can be created, matching defects to the individual annotated panels.

*xhauzv00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Inspections of photovoltaic (PV) power plants are crucial as defects in photovoltaic panels can cause up to 10 % power loss [1]. Some defects may even become a fire hazard, as they manifest as high-temperature areas on the surface of photovoltaic panels. Finding and documenting faults in thermal images can take up to multiple days, and parts of this process could be automated.

The current inspection process that this paper aims to improve utilizes drones to capture thermal images of all the PV panels inside a power plant. These images are then manually inspected using an application, "FLIR Tools"[1], in which found defects are marked. The types of defects are hot spots, increased temperature in a third of a panel, high temperature in a row of panels (two or more), a cracked module (leads to multiple hot spots), and an increased temperature caused by a shade (often on the edge of the panel caused by greenery). Every defect is also manually entered into an Excel spreadsheet, which, along with the annotated images, is the final output of the inspection.

The research of using neural networks for PV defect detection has been rapidly growing in popularity for the last ten years [2]. I found two commercial software solutions[2,3] that help with thermal image PV inspections. Both also contain AI-based fault detection; however, the types of detected defect types or the accuracy these solutions offer are not clear on their websites. The project in this paper was done in collaboration with a professional who conducts PV inspections. None of the current solutions supported their inspection workflow (as described above) well enough for them to utilize such programs.

My solution offers a web-based application that integrates a neural network used for the detection of defects. Defect detection in individual images is just one part of the problem. The application and the resulting report of defects also need to possess information on where exactly these defects are located in the context of the whole PV. PVs are separated into segments, mostly rows, of individual panels, and each row has its identification. Multiple rows grouped together form benches. Therefore, the whole PV

[1] https://support.flir.com/SwDownload/app/RssSWDownload.aspx?ID=1247

[2] Scopito: https://scopito.com/solar-pv-inspection-software/

[3] Raptor Maps: https://raptormaps.com/products/solar-aerial-inspections

needs to be annotated to tell where each defect is located. My application provides supporting tools for such annotation. The panels only have to be annotated once, and the position of the defects is then inferred from those annotations (this is the longest part of manual inspection). Annotation of the panels could not be automated so far, as each PV power plant owner uses a different naming scheme.

The neural network responsible for defect detection focuses on one of the most prominent defect types – a hot spot. The API used by the application allows for easy exchange of the used model once the dataset is extended with better data and a model that detects more classes is trained.

2. Application

I created a web application to allow for visualization and interaction with the output of the trained AI models. The application has two modes: "Panel" and "Fault". The main purpose of the panel mode is to allow the individual PV panels to be annotated. These panel annotations are allowed by the automatic panel detection (described in 2.2). The panels can also be created and deleted if needed. The same goes for faults in Fault mode, where the user may delete and create all the faults the models didn't detect.

The application's main strength lies in working with an image of the entire power plant created using image stitching. In this approach, a sliding window method is used to get samples from the large image, which are then used by the AI detectors. All the results are then aggregated. Once all the faults are annotated, the output table can be generated using the information from the annotations. The application also allows the AI detectors to run on individual, smaller images if the user desires.

2.1 Defect Detection

I tested simple computer vision methods at the start of the project and deemed them unsatisfactory. In order for the detector to be able to work on a wide range of data, I chose an approach using a convolutional neural network. At first, I used **Faster R-CNN** for object detection, and I created a dataset of 300 images where defects were annotated using bounding boxes. I used a PyTorch implementation[4] for transfer learning. The highest resulting mAP[5] of the multiple trained models (with different hyperparameters) was 0.35. Considering these results, I searched for a different approach that could provide better precision.

In the end, I chose the **U-Net** [3] architecture Figure 1 (previously used in [4]). U-Net performs semantic segmentation, which is a task that assigns a class to each pixel in an image. A post-processing is performed on the output of the model, in which each object (non-connected shape) is detected in the output mask. I trained the model from scratch on a dataset of 770 annotated images Figure 2 for 120 epochs. The training loss values and dice score can be seen in Figure 3. The final dice score of the best model is 0.85.

2.2 Panel Detection

Achieving a high accuracy (similar to [5]) is a primary goal as the task is fairly simple, and the need for intervention from a person should be minimal (ideally, none at all).

I used **Mask R-CNN** [6] from the Detectron2[6] library for instance segmentation. I employed a pre-trained model for transfer learning. I created a dataset with 150 images for training. The final mAP of the model is 0.83, and the training graph can be seen in Figure 4.

3. Conclusions

The implemented application aims to speed up inspections of PV power plants, which currently last multiple days. The main steps to achieve this are automatically detecting defects and making an intuitive interface for mapping defects to panels.

The AI defect detector currently detects only the most common defect – a hot spot. With higher-quality training data and an expanded dataset, other defect types can also be detected. The Dice score of the trained model is 0.85, which could be improved with an expanded dataset or using a newer state-of-the-art segmentation model.

The interface for localizing PV panels improves the experience of mapping a panel to its actual position in the power plant. The panel detector achieves high mAP, thanks to which only a few panels from the entire power plant have to be manually created or deleted. Afterwards, those panels can all be annotated.

Acknowledgements

I would like to thank my supervisor Ing. Daniel Bambušek for their help and David Balhar for providing their expertise in the field and also providing images for the datasets.

[4] https://pytorch.org/

[5] mean Average Precision

[6] https://github.com/facebookresearch/detectron2

References

[1] Mahmoud Meribout, Varun Kumar Tiwari, Juan Pablo Peña Herrera, and Asma Najeeb Mahfoudh Awadh Baobaid. Solar panel inspection techniques and prospects. *Measurement*, 209:112466, 2023.

[2] B. Li, C. Delpha, D. Diallo, and A. Migan-Dubois. Application of artificial neural networks to photovoltaic fault detection and diagnosis: A review. *Renewable and Sustainable Energy Reviews*, 138:110512, 2021.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[4] Muhammad Rameez Ur Rahman and Haiyong Chen. Defects inspection in polycrystalline solar cells electroluminescence images using deep learning. *IEEE Access*, 8:40547–40558, 2020.

[5] Jhon Jairo Vega Díaz, Michiel Vlaminck, Dionysios Lefkaditis, Sergio Alejandro Orjuela Vargas, and Hiep Luong. Solar panel detection within complex backgrounds using thermal images acquired by uavs. *Sensors*, 20(21):6219, 2020.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.