Universität Rostock

Traditio et Innovatio

Masterarbeit zum Thema

Visual analysis of solar module imperfections in thermography data

Andrei Purits

M.T.Program:	Visual analytics
Submitted to the:	IEF, University of Rostock
Presented on:	01.06.2023
Matriculation no.:	221202242
Processing period:	20 weeks
First examiner:	Prof. Dr. Stefan Bruckner
Second examiner:	Prof. Dr. Oliver Staadt
Lehrstuhl:	Visual Analytics
Rostock:	Official deadline: 02.11.2023

Master thesis

Faculty of Computer Science and Electrical Engineering

Abstract

Nowadays, the German government is looking forward to using more renewable energy sources. One of the main renewable energy technologies are solar systems which are installed on fields and roofs worldwide. These systems need a lot of engineering solutions. Unfortunately, the environmental parameters in combination with production imperfections can lead to reduced power capacity. The imperfections are primarily not detectable in the visible part of light, but in the infrared spectrum. Therefore, the imperfections can be detected by fluctuating thermal radiation. It is possible to find issues on solar panels using special cameras that can see heat, they called thermographic cameras.

Solar panels are special tools that help us make clean energy from the sun. Germany has an aim to produce as much eco-friendly energy as possible, so it has invested a lot in it.

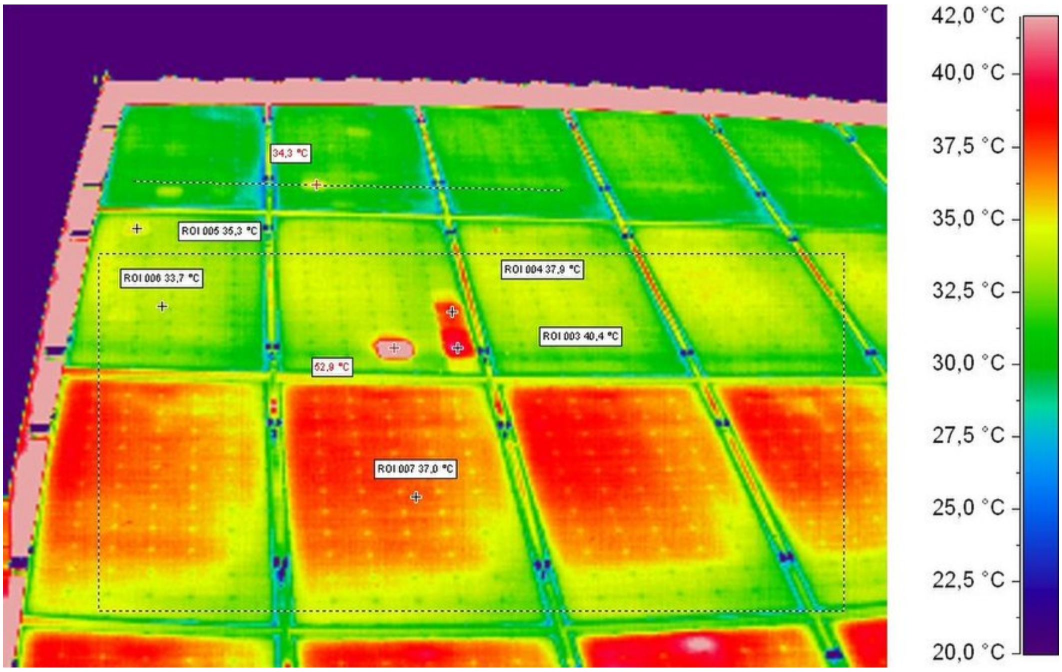

Figure 0.1: Solar panel with hotspot

Image taken from [28]

The primary distinguishing characteristic of outliers is that they exhibit higher temperatures compared to their immediate surroundings. This characteristic simplifies the outlier identification process since it eliminates the requirement for absolute temperatures, which can vary due to changing weather conditions or the panel's cleanliness. Instead, relative temperature differences suffice for this purpose. In Figure 0.1, an example of thermographic data is displayed along with a color legend.

The goals of this thesis were to develop an approach that facilitates searching for the defects in collected data similar to the data presented in Figure 0.1 and create a graphical user interface, as some inputs from user are expected in image processing and outliers detection. As an output, imperfection detection of collected data sets with high accuracy is pending.

This thesis is a part of a bigger project with long-term goal to create a fully automotive system, that will gain data with rover and robotic arm and automatically send report about outliers.

Contents

List of Figures

List of Tables

Acronyms

CNN Convolutional Neural Network.

Cobot Robotic arm.

CPP C plus plus programming language.

GUI Graphical user interface.

HOG Histogram of Oriented Gradients.

HSV Hue saturation value.

IoT Internet of Things.

IQR Interquartile range.

LTS Least Trimmed Squares.

MB Megabyte.

ML Machine learning.

PCP Parallel coordinates plot.

PV Photovoltaic.

RGB The Red, Green and Blue.

SSE Sum of squared errors.

TR Temperature range.

YOLO You Only Look Once.

1 Introduction

1.1 Motivation

The energy transition in Europe is characterized by a rapid growth of renewable energy and infrastructure. In order to reduce the impact of humans on the climate system, a more ambitious development of renewable energies is needed [42].

Solar panels play an important role in Germany's economy by contributing significantly to the country's renewable energy sector. It has shown striking growth in its solar industry. Nowadays, solar panels are a key part of the country's green economy, establishing Germany as a global leader in solar power.

There are different types of solar panels operating in Europe relying on technologies such as photovoltaic (PV) or solar thermal systems. The most common are used in commercial rooftop installations. The German Government wants people to use solar energy motivating with aid programs, such as the Renewable Energy Sources Act (EEG), which makes solar power a very attractive investment [49].

To supervise solar power generation, the Federal Network Agency requires installations to have a monitoring system. This special system helps to monitor how much electricity the solar panels are producing. It helps to make sure the whole process follows the rules and helps managing the power system efficiently [61]. The panels are also regularly checked to make sure they work properly.

In Germany, the widespread adoption of solar panels has been driven by two primary factors. Firstly, it has contributed to improving air quality by enabling the use of cleaner energy sources. Secondly, it has facilitated employment opportunities and economic growth within the country. The extensive utilization of solar energy in Germany aligns with its broader objectives of transitioning towards cleaner energy sources, preventing climate change, and achieving its targets for increased utilization of renewable energy.

Solar panels, like any technology, can experience problems, and one common issue are hotspots. Hotspots occur when certain areas of a solar panel generate uncontrolled heat. This can happen due to manufacturing defects, shading, dirt, or differences in module performance. Hotspots have a negative impact on the panel's performance.

When hotspots occur, the affected area becomes hotter than the rest of the panel, causing an imbalance in electrical currents. This can lead to reduced energy production and sometimes permanent damage to the affected cells. Hotspots not only decrease the panels' energy output but also can affect the entire solar system if not addressed fast.

Hotspots may cause financial losses for solar panel owners, as they decrease amount of consuming energy up to several times. Since solar panels are installed to generate electricity and save money on utility bills or earn revenue, any decrease in energy production means financial losses. To compensate for the reduced output, additional panels or systems may be needed, which can be costly.

Moreover, when some parts of the panels get too hot, their durability may decline. This makes the panels become less efficient over time and replacement may be required sooner than expected, which can make the whole solar panel system less profitable. Regular monitoring and maintenance of solar panels are important steps to minimize the negative effects of hotspots.

This project is driven by a commitment to advancing the adaptation of solar panels through the development of an outlier detection system. By identifying and addressing issues such as hotspots in solar panel systems, this research aims to enhance the robustness and efficiency of solar technology, making it more accessible and reliable for individuals, businesses, and communities seeking to harness the power of the sun for a greener tomorrow. Ultimately, the motivation is to play a part in the successful and accelerated transition to renewable energy sources, supporting a cleaner and more sustainable future not only for Germany but for the world at large.

1.2 Problem

In the past decades, detecting hotspots in thermographic data of PV panels in PV power stations entirely relies on manual experience. With the increasing number and time of PV panel installation, more and more PV panels are featured with hotspot defects of various sizes. Therefore, a more accurate and timely detection system for hotspots of PV panels is urgently needed [52].

The main focus of this thesis is to establish an outlier detection system modified for thermographic data, complete with a graphical user interface (GUI) designed to streamline data handling and simplify the image processing component. The data is stored in the form of a temperature matrix, and efforts will be made to enhance user experience through data visualization while striving for greater precision in results. The system's accuracy will be attentively validated using real solar panel data.

However, various external factors, such as raindrops or sunlight reflections, have the potential to disrupt the outcomes. Hence, the incorporation of a GUI with preprocessing capabilities is considered essential. Additionally, given that the rooftop environment, where the data set is collected, tends to be considerably warmer than the solar panels themselves, a solar panel detection technique must be employed prior to outlier detection. This method must exhibit a high degree of robustness, especially considering the challenges posed by the low resolution of the thermographic camera, which introduces complexity to the image processing aspect.

1.3 Solution

In order to solve the detection of hotspots problem, the GUI component of the program is presented, where user can easily navigate and access all information about the collected data. For example: how hot each part of the solar panel is, the temperature around it, how temperature is spread out. Therefore, the users of the program are able to scroll through every frame and every imperfection in data.

As data can be taken with different external conditions, the program should have a preprocessing part. To address this issue a dashboard is created, that present comprehensive data visualizations with scatter plots. These plots show how attributes in image preprocessing effect the final outlier detection. These attributes are chosen as they influence the result the most, for example size of an outlier or number of iterations of morphological operation.

In Figure 1.1, the main project steps are outlined. Each dataset collected undergoes a preprocessing stage and is subsequently opened in the primary program featuring a GUI. Following this, the selected outlier detection method is applied, and its accuracy is assessed.

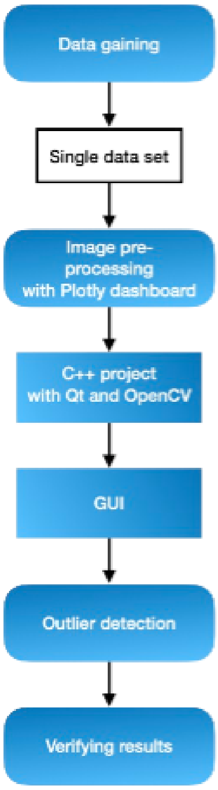

Figure 1.1: Main project steps

The primary objective is to create a powerful and convenient tool, capable of analyzing the collected data to identify any anomalies or irregularities. Through the implementation of advanced algorithms and data processing techniques, the program is able to detect and highlight potential imperfections, allowing the efficient troubleshooting and maintenance.

1.4 Thesis structure

The conducted research has provided valuable insights into the creation and utilization of interfaces between humans and machines. Chapter 2 provides an in-depth exploration of related works and the current landscape in visual analytics research. This research focuses on various approaches to monitoring solar panels, as well as image processing techniques, hotspot detection methods, and the design of graphical user interfaces for data analysis. Chapter 3 introduces the components of the outlier detection system. The primary contribution of this thesis is detailed in Chapter 4, which outlines the methodology encompassing all methods and algorithms, including image preprocessing and processing operations, as well as pixel restrictions for the long-term project objectives. Following this, Chapter 5 provides a comprehensive description of implementation details. Chapter 6 showcases the results of the implementation, demonstrating the capabilities of the approach. The final chapters, 7 and 8, are dedicated to discussions, limitations, conclusions, and future work, respectively.

2 State of the Art

In this chapter, we are focusing to the theoretical base of this thesis. The chapter begins with an exploration of relevant theoretical concepts, including definition of Visual Analytics, process models, and data transformation. Following this theoretical foundation, various methods for monitoring solar panels and acquiring data are discussed. Next, attention turns to prevalent image processing techniques, highlighting their significance in this context. A range of hotspot detection methods is then presented, followed by examples showcasing the integration of GUI in data analysis.

2.1 Definition of Visual Analytics

In "Illuminating the Path" [14], Thomas and Cook define visual analytics as the science of analytical reasoning facilitated by interactive visual interfaces. Though Dainel A. Keim in one of his works gave it more detailed definition: Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets [22].

Visual analytics is more than just visualization. It can rather be seen as an integral approach to decision-making, combining visualization, human factors and data analysis. The challenge is to identify the best automated algorithm for the analysis task at hand, identify its limits which can not be further automated, and then develop a tightly integrated solution with adequately integrates the best automated analysis algorithms with appropriate visualization and interaction techniques [22].

2.1.1 Goals of Visual Analysis

The visual analytics process involves the utilization of automated analysis techniques both prior to and following the interactive visual representation. This necessity arises from the inherent complexity of contemporary, and particularly prospective, datasets. These datasets are characterized by their sheer size, making straightforward visualization impractical [22].

Goals describe the intent with which analysis tasks are pursued. General goals are to explore, describe, explain, or present the data.

- The exploration phase entails the initial observations, the identification of trends, and the detection of outliers. This process also encompasses the recognition of the absence of expected patterns or trends within the data.

- Description involves characterizing an observation by considering the associated data elements. A comprehensive description can serve as a foundation for configuring subsequent analytical steps.

- Explanation entails the identification of all contributing data elements and the determination of the primary causes behind an observation, as highlighted by Christian [55].

- Confirmation is directed at verifying hypotheses. At this stage, concrete evidence is required to either support or disprove a hypothesis. To achieve this, alternative visual representations of the data or the reparameterization of analytical visual representations may be employed.

- Presentation is the process of conveying confirmed analytical results. While explanation and confirmation are primarily concerned with convincing oneself, presentation aims to persuade others regarding the findings within the data.

The workflow progresses from initial data exploration until an observation is made. Following this, the observation is described and an attempt is made to provide an explanation. Subsequently, the hypotheses about the data are validated, confirming their reliability. Finally, the process concludes with the preparation to present the confirmed analysis results. It's noteworthy how this workflow evolves from a state of not knowing that certain observations can be made in the data to acquiring a thorough understanding of them, sufficient for presentation [55].

2.1.2 Data transformation

Initially it is just raw data as input and seek images as output. Models can help us: visualization pipeline and the data state reference model. They conceptually model the data-to-image transformation by defining different data stages and different types of operators, as shown in Figure 2.1.

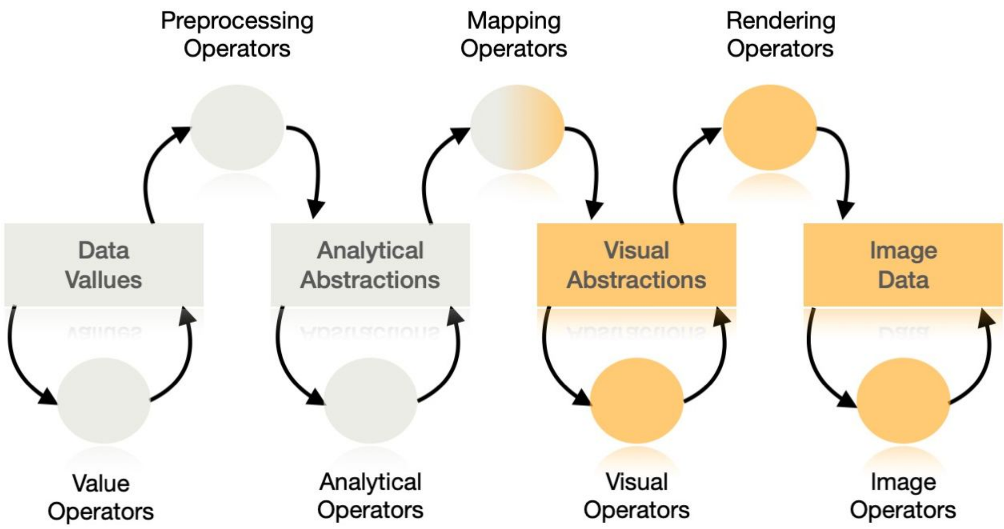

Figure 2.1: Data-oriented and graphics-oriented stages and operators

Image redrawn from [55]

From bits to images, data exists in various states. In order to abstract from particular state details, are commonly considered four basic stages:

- data values
- analytical abstractions
- visual abstractions
- image data

Data values typically represent raw pieces of information. Analytical abstractions, on the other hand, are meticulously organized data, enriched with meaningful derived attributes. This encompasses data tables, hierarchically structured levels of detail, and more advanced abstractions like classifications and clusters.

Visual abstractions models various visual attributes, such as fill color or stroke style. In essence, they serve as a bridge between the analytical abstractions and the final presentation.

Furthermore, image data encompasses the colored pixels that will be displayed on the output device, forming the visual representation of the data.

2.2 Solar panel performance monitoring

Monitoring solar panels is imperative to ensure their efficiency and performance, particularly considering the significance of renewable energy in addressing climate change and global energy demands. Solar panels are vulnerable to environmental factors, wear and tear, and technical issues that can hinder their electricity generation. To address these challenges and optimize solar energy utilization, a range of monitoring methods and technologies have been developed.

These methods include real-time data collection, historical performance analysis, early fault detection, and informed decision-making for maintenance and efficiency enhancement. Effective solar panel monitoring is instrumental in achieving the expected returns on investments, minimizing energy losses, and fortifying the resilience of energy infrastructure.

In this section, we delve into various monitoring techniques, including inverter monitoring, remote sensing, machine learning, and IoT-based monitoring systems. Collectively, these approaches enhance the performance and reliability of solar panels, rendering solar power a more dependable and cost-effective energy source for the future.

2.2.1 Remote sensing and aerial inspection

Remote sensing and drone technology have become invaluable assets for efficiently monitoring and maintaining photovoltaic (PV) solar panels. These tools ensure the sustained performance and return on investment of solar installations. Within this context, further noteworthy articles contribute to the advancement of remote monitoring and drone-based inspection techniques for solar panels.

In the realm of PV system monitoring and inspection, several innovative approaches have emerged, each contributing to the efficiency and precision of maintenance procedures. Gisele [48] introduces a solar-powered autonomous rover, equipped with a suite of sensors, designed for continuous monitoring of PV systems. This rover autonomously gathers essential data about the condition of solar panels, offering a reliable solution for routine inspections. Complementing this effort, Humberto et al. [2] have created a specialized dataset tailored for the recognition of snail trails and hotspot failures in monocrystalline silicon solar panels. This dataset, generated with the help of drone-mounted infrared (IR) cameras, includes thermographic aerial images and environmental data, serving as a valuable resource for the development of advanced image processing techniques to ensure precise panel inspection. Additionally, the studies by Yu and Abdelilah [17, 53] are dedicated to automating the recognition and location of solar panels using drones and thermal cameras, as shown in Figure 2.2. By employing machine vision technology, these systems streamline maintenance processes, significantly enhancing overall efficiency. Collectively, these examples showcase a comprehensive exploration of cutting-edge methods and technologies in the field of PV system monitoring and maintenance, offering promising solutions for the industry.

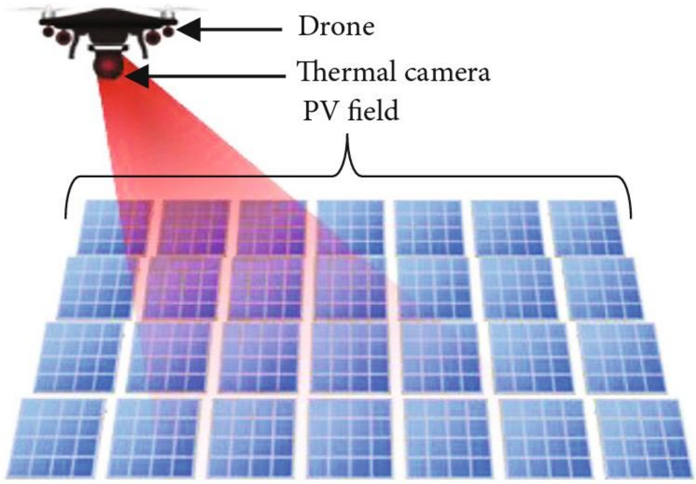

Figure 2.2: Control photovoltaic field using thermal camera on drone

Image taken from [53]

While both remote monitoring utilizing rovers and drones offer valuable capabilities for inspecting solar panels, rovers hold distinct advantages in specific scenarios. Rovers typically operate from a ground-based platform, affording them stability and precision in acquiring images. They are also immune to challenges like wind instability, which can affect the quality of images captured by drones.

2.2.2 Utilizing Machine learning for analysis

Machine learning (ML) has emerged as a powerful tool for monitoring and maintaining solar panels. ML techniques enable the automated analysis of data collected from solar panels and their surroundings, providing insights into panel performance and identifying potential issues. In this context, several articles have explored the application of ML in solar panel monitoring, each contributing valuable insights and methodologies.

Modeling and prediction

In the domain of solar energy prediction, several studies have harnessed machine learning techniques to advance our understanding and capabilities. Enache et al. [12] have introduced a novel approach employing feed-forward neural networks to model PV panels, thereby enhancing our comprehension of their performance characteristics. Building on this, Ulapane et al. [56] have utilized Gaussian processes to determine the maximum power point (MPP) of solar panels, taking into account variations in environmental parameters. This innovative technique contributes to optimizing energy capture and efficiency in PV systems. Further expanding the horizons of solar energy forecasting, Jawaid et al. [20] have focused on predicting daily mean solar power using artificial neural network (ANN) algorithms, offering valuable insights and forecasts for solar energy generation on a daily basis. These studies collectively represent a concerted effort to leverage machine learning to better model, optimize, and predict various aspects of PV systems, ultimately advancing the field's capabilities and understanding.

Defect detection and diagnosis

Machine learning has found versatile applications in enhancing the reliability and efficiency of solar panel systems. Babasaki et al. [3] have demonstrated the utility of machine learning in detecting degraded solar panels afflicted by issues such as breaks, snow, or weed interference, thereby contributing to early defect identification and maintenance. In a complementary vein, Niazi et al. [35] delve into the realm of identifying defective solar panels through binary classification and texture feature extraction from thermography images. This approach provides an effective means of pinpointing panel issues based on visual cues, streamlining the maintenance process. Furthermore, Ferreira et al. [13] focus on the diagnosis of malfunctions within Solar-Powered Wireless Mesh Networks, employing machine learning techniques to detect issues and diagnose individual panels, ultimately bolstering the reliability of such networks. These studies collectively underscore the increasing role of machine learning in addressing various facets of solar panel maintenance, from defect detection to network performance optimization, ensuring the continued efficiency and effectiveness of solar energy systems.

In summary, the application of machine learning techniques in solar panel monitoring may be hindered by the insufficient amount of available data. Some algorithms require a substantial dataset for effective learning and prediction. In cases where data scarcity is a limitation, alternative methods like remote monitoring with rovers may offer more practical solutions for ensuring solar panel efficiency and maintenance.

2.2.3 Internet of Things-based solar panel monitoring

The Internet of Things (IoT) has ushered in a new era on the internet, enabling objects and devices to gain intelligence and exchange information seamlessly. In the context of monitoring solar panels, IoT provides the capability to connect and interact with various sensors and devices over the internet, enhancing the efficiency of solar power plants. By enabling physical objects to be interconnected and self-identifying, IoT holds the potential to revolutionize solar panel monitoring. Some articles regarding utilization of IoT in solar panel monitoring are provided:

The application of the IoT in environmental analysis within the context of solar power plants has yielded a range of innovative solutions. Voicu et al. [59] discuss an IoT-based system designed to analyze the solar power plant environment, employing a data acquisition system to measure solar irradiance and predict energy generation. Sensors connected to a Raspberry Pi facilitate data transmission to a virtual machine for further analysis, enhancing plant efficiency. In a similar vein, Shu et al. [50] introduce a centralized data gathering methodology using the Modbus protocol, streamlining communication between sensors and controllers for efficient bidirectional data transfer. Moving towards embedded IoT solutions, Luo et al. [27] describe an embedded IoT system for measuring temperature and humidity within a communication system, utilizing a DHT11 sensor to upload data to a remote server via a Raspberry Pi, ensuring real-time monitoring. Lastly, Nikhila [36] presents a comprehensive IoT-based system for environmental monitoring, emphasizing the use of various sensors such as LDR, MQ7, DHT11, and an accelerometer. Data collected from these sensors is efficiently uploaded to a webpage through a Raspberry Pi, providing a holistic approach to environmental parameter tracking. Together, these articles highlight the diverse applications of IoT in environmental analysis and monitoring, enhancing the efficiency and capabilities of solar power systems.

While IoT holds great promise for solar panel monitoring, it is important to consider its limitations. In environments with limited data availability or remote locations, the reliance on a stable internet connection and the need for constant data exchange may make IoT less suitable.

2.2.4 Inverter performance analysis

Another method to monitor solar panels is Inverter monitoring. It entails a constant evaluation and management of inverters, which are vital components responsible for transforming the direct current (DC) electricity generated by solar panels into the usable alternating current (AC). This monitoring method is a key driver for real-time performance analysis, fault detection, and data collection, all with the overarching goal of optimizing energy production and ensuring the well-being of the entire solar system. To delve deeper into the realm of inverter monitoring, we turn our attention to a selection of insightful articles that shed light on various approaches and techniques:

Within the domain of inverter monitoring, a series of insightful articles showcase diverse approaches and techniques. Bayrak's article [5] spotlights a LabVIEW-based monitoring system that seamlessly integrates with grid-connected photovoltaic power generation systems, offering a robust framework to assess inverter performance and overall solar system health. Nkoloma et al.'s work [37] introduces an advanced solar power management system known for its dynamic power load resizing capabilities, driven by the Subset Sum Problem algorithm. This system goes a step further by incorporating features like advanced remote metering, automatic load control, and priority-based switching, all of which contribute to improved inverter monitoring and system efficiency. Jubaer's contribution [1] delves into the world of smart inverters powered by solar energy, utilizing stepper motors for precise panel rotation and leveraging the Perturb and Observe (PO) algorithm for solar tracking, thereby optimizing energy generation and inverter performance. Kumar et al. [24] venture into solar photovoltaic smart inverters, introducing a unique two-way communication feature that enhances user control and monitoring. This feature provides valuable insights into load runtimes and user-selected load probabilities, fostering a more personalized and responsive monitoring approach.

These articles collectively paint a comprehensive picture of inverter monitoring within the context of solar energy applications. While each approach brings a unique perspective to the table, they are all interconnected by their shared objective of ensuring the optimal performance of solar power systems. However, the decision to explore a different avenue, focusing on remote monitoring using a rover for solar panels rather than inverters, is guided by specific considerations. A critical factor at play here is data availability. Inverter monitoring often relies on electrical data, which can sometimes be locked behind proprietary systems and protocols. In contrast, the deployment of a rover equipped with a thermographic camera offers a non-intrusive and data-rich approach to assess the physical condition of solar panels. This approach empowers the detection of issues like snail trails and hotspot failures, contributing significantly to the long-term performance and reliability of solar installations. Moreover, the rover-based approach ensures continuous on-site monitoring, allowing for timely intervention and maintenance when necessary.

2.3 Image processing techniques

Having explored various monitoring methods, let us now delve into different image processing techniques employed for the detection and subsequent processing of data collected from solar panels.

2.3.1 Object detection methods

Various object detection methods have been instrumental in the realm of computer vision and image processing. These methods have undergone significant evolution over the years, with traditional techniques like the Roberts operator and Canny operator proving effective in the detection of edges and contours. Additionally, modern approaches, including Convolutional Neural Networks (CNNs) and advanced algorithms like You Only Look Once (YOLO), have brought about a revolution in object detection and image analysis. This discussion will explore and provide examples of these edge detection methods, highlighting their applications and significance in diverse domains such as solar panel monitoring and defect detection.

Edge detection methods

Traditional image edge detection algorithms, which have been in use since the 1960s and 1970s, rely on the utilization of differential operators. Edge detection using Roberts operator by Wang et al. [60]: This study introduces an edge detection technique that employs the Roberts operator. It outlines the process of detecting image edges through the operator's differential approach, contributing to the field of computer vision. Otsu thresholding technique for image segmentation by Tian et al. [62]: Within this article, the Otsu thresholding technique is thoroughly examined as a method for image segmentation. It emphasizes the reliance on linear fitting and the application of the Roberts operator to delineate two distinct boundaries. Optimization-based edge detection with Canny operator: Xianquan et al. [54] place a spotlight on edge detection utilizing the Canny operator. It delves into how the operator operates based on optimization principles and establishes low and high thresholds grounded in the image's histogram gradient, ultimately enhancing signal-to-noise ratios.

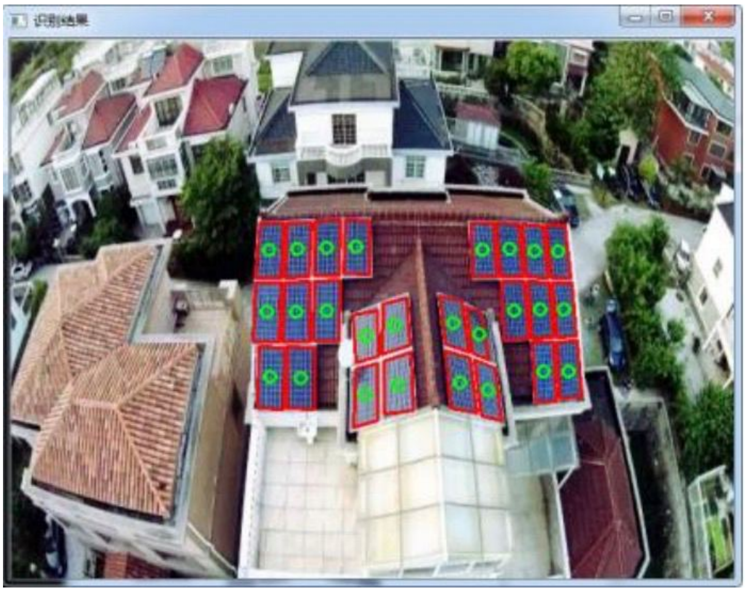

Figure 2.3: Solar panel recognition results

Image taken from [17]

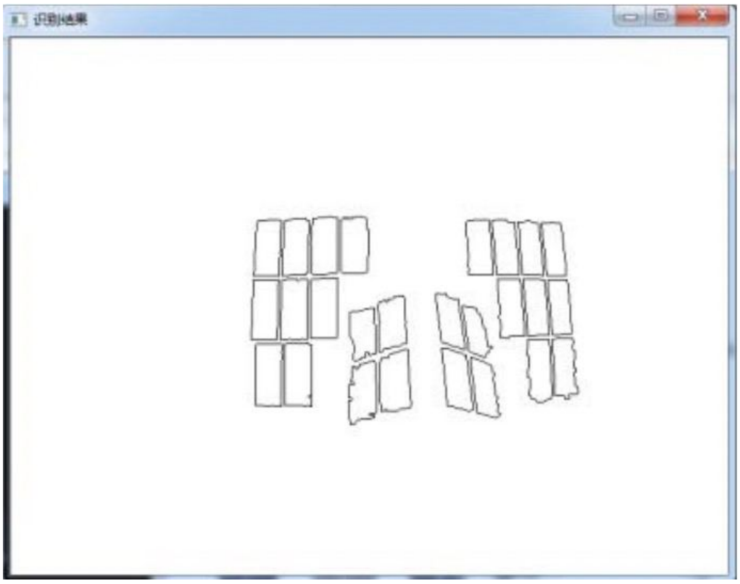

Figure 2.4: Contour recognition results

Image taken from [17]

Another article that has been already discussed in Remote sensing and aerial inspection chapter also applied Canny edge detection. In Article [17], ax example of Canny edge detection is presented. In Figure 2.3, solar panels are highlighted with bounding rectangles based on the building's roof, while Figure 2.4 illustrates only the contours of the solar panels.

Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of artificial neural network specifically designed for processing and analyzing visual data, such as images and videos. They are inspired by the human visual system and consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at feature extraction and pattern recognition in images, making them a powerful tool for tasks like image classification, object detection, and image segmentation. While neural networks, in general, have been in existence since the 1940s, CNNs, particularly for image processing, gained significant prominence and effectiveness in the late 1990s and early 2000s. CNNs have revolutionized computer vision tasks and have been widely adopted in various fields, including solar panel monitoring and defect detection. Notably, Chen et al. [8] have harnessed the power of CNNs to develop a multispectral model capable of classifying diverse defects in solar panels from RGB images, achieving an impressive accuracy of 94.9% in classifying photovoltaic panel images. In a complementary effort, Rahman et al. [45] have leveraged CNNs with a multi-attention U-net architecture to segment and detect cracked solar panels from electroluminescence (EL) images, enhancing the precision of defect identification. Additionally, Pierdicca et al. [40] explored the application of transfer learning using established CNN networks like VGG-16 and Alexnet to differentiate between damaged and defect-free solar panels, utilizing remote sensing and RGB images. Together, these examples exemplify the transformative potential of CNNs in the realm of solar panel monitoring, where they significantly enhance the efficiency and reliability of photovoltaic systems.

Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) has evolved into a fundamental tool in the realms of computer vision and image processing, particularly in the context of object detection. Initially introduced by Navneet Dalal and Bill Triggs in 2005 [11], HOG operates by capturing vital information related to local gradient or edge orientations within an image. Medeiros et al. [32] have harnessed the power of HOG as a feature descriptor to detect and classify solar panels within aerial images obtained from Unmanned Aerial Vehicles (UAVs). This application streamlines the inspection process, automatically identifying solar panels and thereby enhancing the efficiency of monitoring PV systems while also providing a cost-effective solution for ensuring their optimal performance. In a complementary research effort, Ma et al. [29] have seamlessly integrated solar panels with a HOG-based target identification system, significantly enhancing the sustainability of space applications. By outfitting satellites with both solar panels for power generation and HOG-based algorithms for object detection, this innovative approach facilitates more efficient and autonomous operations in orbit, contributing to the long-term success of space missions. Collectively, these instances underscore how HOG's adaptability not only improves solar panel surveillance but also extends its utility to space missions, where effective object detection holds substantial significance.

You Only Look Once

YOLO (You Only Look Once) is a renowned object detection algorithm, initially introduced by Joseph Redmon and Santosh Divvala in 2016 [47]. This algorithm is purpose-built to identify and pinpoint objects within images or video frames. Li et al. [26] present an advanced version of the YOLOv5 algorithm, known as EL-YOLOv5, which incorporates a CBAM hybrid attention module for the precise detection of defects in solar panels. EL-YOLOv5 focuses on identifying five common types of defects, namely hidden cracks, scratches, broken grids, black spots, and short circuits. This is achieved by leveraging publicly available solar panel datasets and real photovoltaic production line datasets. The primary objectives of this algorithm enhancement are to improve defect detection accuracy, robustness, and convergence speed. Evaluation is performed using mAP metrics, where EL-YOLOv5 outperforms the traditional YOLOv5. Qualitative experiments demonstrate effective defect detection, even in challenging industrial conditions.

An additional application of the YOLO machine learning algorithm, specifically YOLOv5, involves real-time wildfire detection using embedded hardware like Raspberry Pi 4 by Johnston et al. [21]. This addresses the need for efficient wildfire detection methods due to the persistent threat of wildfires. Conventional approaches rely on human operators monitoring camera feeds, which can be error-prone and subject to operator fatigue. The article also highlights the benefits of edge processing on embedded devices to reduce bandwidth consumption. The research includes performance comparisons of various YOLO models and assesses the practicality of machine learning-based wildfire detection systems.

Conclusion

In the realm of object detection methods for solar panel monitoring, various techniques have been explored, each offering distinct advantages. Within this context, for the outlier detection system employed in this project, Canny edge detection has been chosen as the method of choice. The decision to utilize Canny edge detection stems from its ability to effectively highlight edges and contours within solar panel images, facilitating the identification of potential defects and irregularities. This technique aligns with the immediate goal of ensuring accurate and efficient outlier detection in solar panel data.

Looking toward the long-term objectives of the project, the incorporation of machine learning techniques holds great promise for enhancing solar panel detection and monitoring. Machine learning algorithms, such as CNNs and YOLO, have demonstrated remarkable capabilities in object detection and image recognition tasks. Leveraging these methods in future iterations of the project can lead to more sophisticated and accurate outlier detection systems.

2.3.2 Image segmentation

Image segmentation with Watershed algorithm

The watershed algorithm is an image segmentation technique that treats an image as a topographical map, with pixel intensities representing elevations. It will be discussed about this approach more in Methodology chapter. Chaudhary [7] centers around the utilization of the watershed algorithm in thermal imagery for the analysis and segmentation of both cement and bird deposits on solar panels. The process initiates with the conversion of the original RGB thermal image into grayscale, followed by the calculation of gradient magnitude to pinpoint regions of interest. To mitigate over-segmentation concerns, a marker-based approach comes into play, wherein foreground and background markers are identified. Subsequently, the watershed algorithm is applied to effectuate segmentation and accentuate these regions of interest. To aid in comprehension, pseudo-colors are judiciously employed for visualization purposes. This method significantly contributes to the study of temperature impacts on solar panels resulting from the presence of cement and bird deposits, thereby streamlining their monitoring and in-depth analysis.

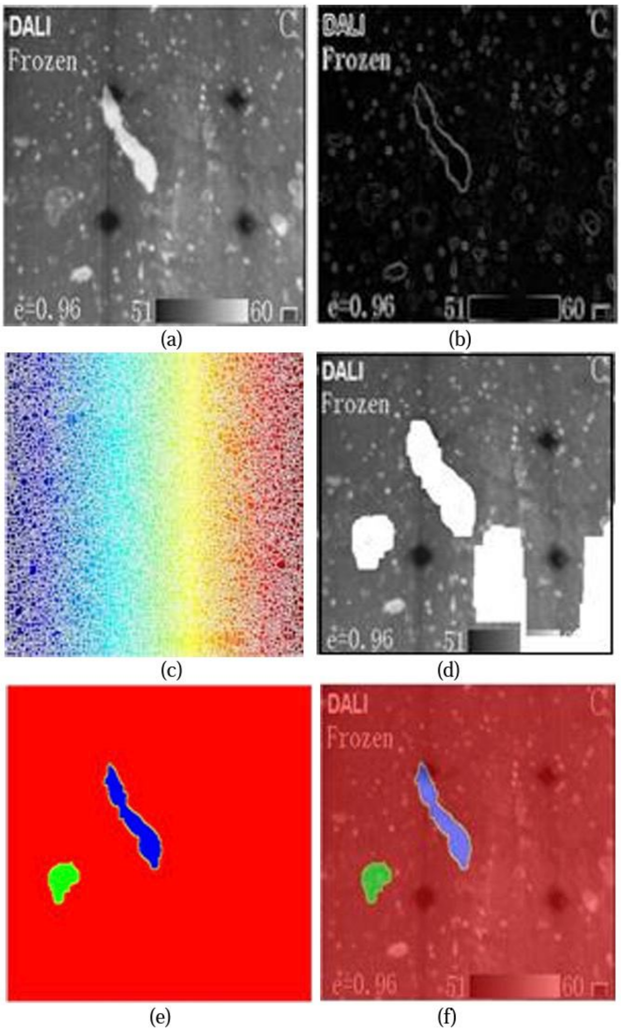

Figure 2.5: Fig. 15(a) Shows gray scale image of original thermal image, Fig. 15(b) gradient magnitude image, Fig. 15(c) over-segmentation of watershed transform, Fig. 15(d) markers and object boundaries super imposed on original thermal image, Fig. 15(e) colored segmented watershed transform label matrix, Fig. 15(f) superimposed segmented image of watershed transform

Image taken from [7]

The figures depicted in Figure 2.5 pertain to bird deposits and follow a similar procedural trajectory as described for cement deposits. Figure 15(a) constitutes the grayscale image derived from the RGB thermal image, while Figure 15(b) unveils the gradient magnitude image, indispensable for detecting the dark areas that encapsulate the desired segmented bird deposits. Figure 15(c) portrays the resultant image suffering from over-segmentation when the watershed algorithm is applied directly to the gradient magnitude image without marker integration. In contrast, Figure 15(d) showcases an image with markers and object boundaries harmoniously superimposed upon the original image, courtesy of the marker-based methodology underpinned by morphological operations. The application of pseudo-colors to the watershed algorithm's label matrix (LRGB) is showcased in Figure 15(e), enabling a clearer delineation of the segmented bird-deposited area. The blue region signifies the segmented bird deposit on a red background, with a green-colored area designating another small deposit, distinguished in proximity to the bird deposit. Lastly, Figure 15(f) emerges as a superimposed image that seamlessly overlays the obtained watershed algorithm label matrix (LRGB) upon the original image, culminating in a more elucidated view of the segmented bird deposit.

K-Means clustering

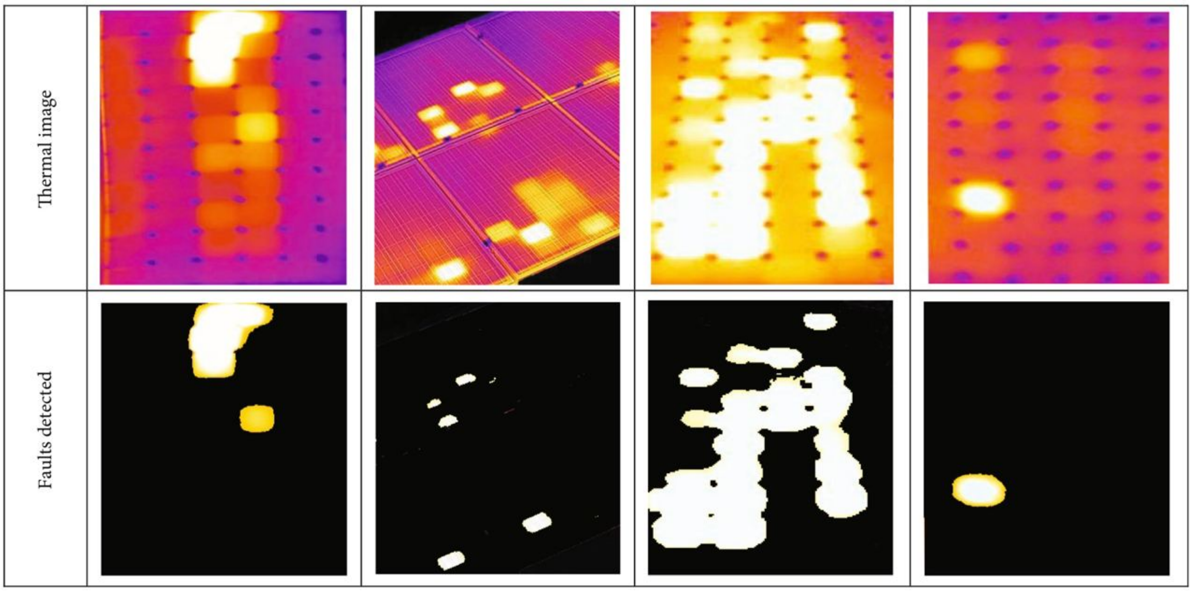

Figure 2.6: The application results of the K-means clustering

Image taken from [53]

K-means clustering, a widely-used unsupervised machine learning technique, is applied in image segmentation. Abdelilah [53] harnesses the K-Means clustering algorithm, initially introduced by MacQueen in 1967 [30], for image segmentation with the goal of detecting faults in a photovoltaic field based on thermal imagery. This research focuses on leveraging the K-means algorithm for the segmentation and precise localization of damaged regions in thermal images of solar panels. Before applying the K-means algorithm, the optimal number of clusters (K) is determined through the Elbow and Average Silhouette methods. To prepare the thermal image for analysis, it undergoes preprocessing by converting it from the RGB color space to the HSV color space, which results in a matrix representation of the image. Subsequently, the K-means algorithm is applied to the processed image, dividing it into K clusters based on pixel intensity. It's worth noting that the choice of the Elbow or Average Silhouette method significantly influences the accuracy of damaged region identification. Validation tests and further assessments are carried out to evaluate the K-means algorithm's efficacy in detecting faults in solar panels. Figure 2.6 illustrates the outcomes, demonstrating the detection of damaged areas on the solar panels.

Conclusion

The watershed algorithm is the optimal choice for outlier detection systems on solar panels due to its unique adaptability to thermal imagery analysis. It efficiently utilizes gradient magnitude data to pinpoint regions of interest, effectively distinguishing faults caused by cement and bird deposits. Marker-based control minimizes over-segmentation, while pseudo-color visualization aids in result interpretation. Its versatility and accuracy in isolating anomalies make it a robust tool for precise fault detection in solar panel inspections, ensuring the reliability of outlier identification systems.

2.4 Statistical outlier detection techniques for solar panel data

Within the domain of solar panel performance monitoring, the utilization of statistical outlier detection techniques emerges as a viable approach to identify irregularities and faults in solar panel data. Recent scientific articles delve into the application of methods such as the Z-Score method and the Interquartile Range method.

Different types of outliers

Before delving into the various statistical outlier detection techniques, it is important to outline the potential types of imperfections that may be encountered. In this project, the verification of results will be conducted manually by the user. Therefore, it is essential to understand the types of outliers that the user might encounter. Tianyi et al. [52] categorized hotspots in PV panels into three primary categories, primarily based on their shapes: round, linear, and square, as shown in Figure 2.7. Round hotspots are typically attributed to issues at power cord junctions and leaf occlusions, while linear hotspots are a consequence of factors such as bird droppings and dust accumulation. Square hotspots, on the other hand, serve as indicators of internal defects or multiple panel failures. These classifications prove invaluable for diagnosing and addressing PV panel issues, ultimately leading to enhanced operation and maintenance.

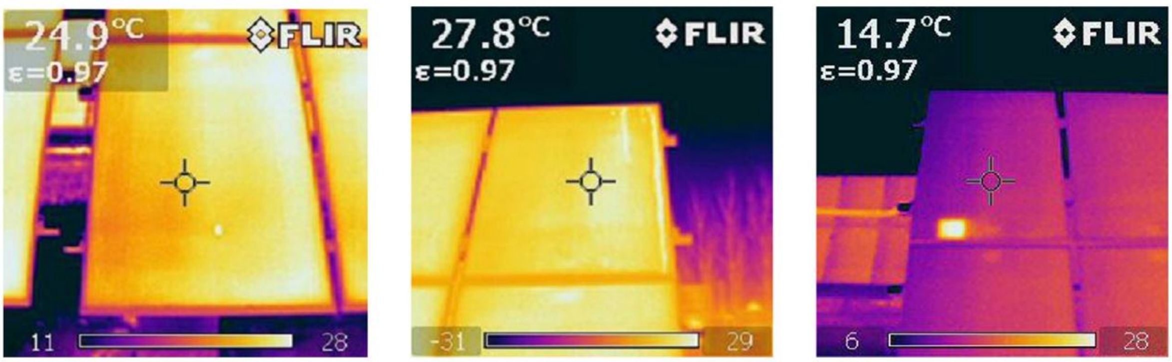

Figure 2.7: Three typical hotspots of PV panels

Image taken from [52]

Additionally, Millendorf et al. [33] have presented a dataset consisting of 20,000 images, each portraying a solar panel with some form of defect. This dataset notably encompasses a broader spectrum of solar panel issues compared to prior works, including concerns such as vegetation, soiling, shadowing, and cracking. The primary purpose of this dataset is to facilitate the training of machine learning models for the detection of outliers and their respective types.

2.4.1 Z-Score method

This statistical measure is utilized to normalize and identify irregularities in solar panel performance data, aiding in the identification of faults and anomalies. Moser [34] in his article applies modified Z-Score method, that is employed to identify outliers on solar panels. The Z-Score is a statistical tool that measures how many standard deviations a particular data point is away from the mean of a dataset. In this case, it is used to assess individual pixel values in a local neighborhood on the solar panel. If a pixel's Z-Score exceeds 2.7, it is considered an outlier, indicating that it deviates significantly from the norm. This can be indicative of potential damage, such as cracks, debris, or other issues on the solar panel's surface. Identifying these outliers using the Z-Score helps ensure the proper functioning and maintenance of solar panels, contributing to their efficiency and longevity. Waleed [51] uses Z-scores for outlier detection in the context of solar panels by applying statistical analysis to the data related to solar panels. By calculating Z-scores for specific attributes associated with solar panels, such as their efficiency, output, cost, or other relevant characteristics, to calculate Z-scores and identify outliers in the dataset associated with solar panels.

2.4.2 The Outlier Detection Rules approach

The Outlier Detection Rules Approach is a method for identifying abnormal operation in solar panel systems. It employs statistical tools like the 3-sigma rule, Hampel identifier, and boxplot to analyze individual string current measurements. These statistical methods establish upper and lower control limits, with the 3-sigma rule considering values beyond three standard deviations as outliers. The Hampel identifier modifies the approach by using the sample median as the reference value. When applied, this approach helps detect and distinguish anomalies, facilitating the early identification of operational issues or faults in solar panel strings based on current data. Zhao et al. [65] employs statistical outlier detection techniques like the 3-sigma rule, Hampel identifier, and boxplot to identify abnormal operation in PV strings based on current measurements.

2.4.3 One-Diode Model Exponentially Weighted Moving Average approach

The ODM-EWMA (One-Diode Model Exponentially Weighted Moving Average) approach is a statistical technique used to monitor the performance of photovoltaic systems. It employs a simulation model based on one-diode model parameters to predict maximum voltage, current, and power generation. By comparing actual measurements with predictions, it generates residuals, which are analyzed using an Exponentially Weighted Moving Average (EWMA) chart in real-time. This approach is effective in detecting failures and distinguishing between open-circuit, short-circuit, and shading faults in PV systems. Fouzi et al. [15] used the one-diode model and the EWMA chart for this purpose.

The Z-score is a preferable method for outlier detection in solar panel data when compared to some of the methods described earlier in the text. It offers a simple and interpretable approach for identifying deviations from the mean. In contrast, previous methods like the "Outlier Detection Rules Approach" may require a larger dataset for reliable results and might not be as effective in dealing with extreme outliers. Additionally, the Z-score is less reliant on the assumption of a normal distribution, making it more adaptable to various data distributions. This flexibility is advantageous in the context of solar panel data, which may not strictly follow a normal distribution.

2.5 Graphical User Interfaces in Data Analysis

In data analysis, Graphical User Interfaces (GUIs) are essential for enhancing user-friendliness and accessibility. Researchers benefit from effortless interaction with complex datasets, as demonstrated by scientific articles. These examples underscore the indispensable role of GUIs in streamlining data analysis.

Lee et al. [25] features a solar panel system integrated into a "Smart Farm" concept designed for agricultural management, with a primary focus on optimizing energy consumption and crop yield. The solar panel, a key component of the system, comprises a photovoltaic module chosen for its effective energy harvesting capabilities. A web-based GUI based on the React framework is provided for real-time monitoring and data analysis. The GUI offers a "Dashboard" tab for real-time information, displaying solar cell output and weather data, alongside a "Database" tab that presents historical data and updated graphs related to solar panel performance, including surface temperature, voltage, current, and charging speed. This comprehensive system empowers farmers to make informed decisions regarding their solar panel systems while promoting sustainable and efficient energy use in agriculture. Hamdani et al. [16] employs a GUI in its solar panel analysis system. The GUI, developed using Visual Basic and Blynk, facilitates real-time monitoring of critical parameters such as current, voltage, light intensity, and panel orientation. Users can easily access and interpret this data, aiding in solar panel performance assessment.

2.6 Conclusion

In conclusion, the exploration of various methods for outlier detection in solar panel data has shed light on the unique advantages and adaptabilities each approach brings to the table. From Canny edge detection for effective contour highlighting to the watershed algorithm's versatility in thermal imagery analysis, and the simplicity of Z-score calculations, each method offers its own set of benefits. However, it's important to note that there is no one-size-fits-all solution for comprehensive outlier detection in solar panel data. These methods, while promising in their respective ways, may have limitations in certain scenarios, such as remote locations with limited data availability.

Looking forward, the integration of machine learning techniques like CNNs and YOLO shows great promise for improving the accuracy of outlier detection systems in the long term. Still, a holistic approach to solar panel monitoring may require a combination of these methods, fine-tuned to specific environmental and data conditions. Therefore, a flexible and adaptable approach that considers the unique challenges of solar panel monitoring remains crucial to ensure the reliability of outlier detection systems in this important field of renewable energy.

3 Components of the system

In this section, important parts of the outlier detection system will be described. It will have two main sections: electrical and technical elements. The electrical part has special electrical devices such as on thermographic camera from Infratec. It can take pictures of the solar panels and show their heat patterns. Complementing this, the technical components involve a detailed data set description. Moreover, the chosen technologies like Qt, OpenCV, and C++ with Python programming languages, help to create a user-friendly screen and advanced methods for image processing.

3.1 Electrical components of the system

The electrical parts of the system include all the devices and elements that handle and control the electrical signals and power in the system. These electrical parts work together to make sure the system can do its tasks and operations effectively.

For the purpose of this project, only a thermographic camera was utilized to record experimental data. This choice was necessitated by the impracticality of obtaining a rover equipped with a robotic arm to access the rooftop where the solar panels in need of testing are situated. Therefore, this chapter incorporates system components that will be subsequently adjusted to align with the long-term project goal of establishing a fully automated system, including the automotive gaining of data.

3.1.1 Radiometric infrared camera module

The primary tool selected for this project is InfraTec's radiometric infrared camera module, the PIR uc 605, that presented in Figure 3.1. This camera module is capable of measuring temperatures ranging from -20 to 400°C, making it well-suited for the project's requirements. With a frame rate of 25 Hz, it ensures that no portions of the panels are overlooked, and it boasts a relatively modest resolution of 640x480 pixels [18].

Figure 3.1: InfraTec camera PIR uc 605

Image taken from [18]

It is connected to the computer with Ethernet cable and has power cable for 12V. A remote-controlled rover with robotic arm is used to move the camera. All data is stored as .irb files and are made with InfraTec software.

3.1.2 Universal robotic arm UR10e

An Universal Robotic Arm UR10e, Figure 3.2, was chosen primarily because of its smooth motion control, and its ability to move in 6 axis and the ability to get into almost every position. The clever motion control methods lower strain on the joints, which means less need for fixing and more working without stopping. It also helps the cobot move quickly and smoothly.

Hence, it does not require to make any movements during the data collection phase with the camera, but the cobot can hold it still, reach far, change its position if needed, save energy, and move precise [57].

Figure 3.2: Universal robotic arm UR10e

Image taken from [57]

3.1.3 Mattro ROVO 2

The ROVO 2 is an off-road vehicle with a top speed of 30 km/h. It is equipped with a durable lithium-ion battery pack that allows for continuous operation for up to 8 hours or 40 km.

Figure 3.3: Mattro ROVO 2

Image taken from [31]

The reason to choose the ROVO 2 for a project lies in its ability to provide efficient and environmentally friendly mobility in off-road environments, where typically solar panels are based. The ROVO 2 operates on electricity and is compact and lightweight, making it easily maneuverable even in challenging locations. Figure 3.3 was taken from the official web-site [31].

3.2 Technical components

The technical components refer to the specific elements and technologies used in the project. Further, choice of specific tools, frameworks, platforms, and how these technologies contribute will be explained in aim to achieve objectives and provide a brief comprehensive overview of both the data set and chosen technologies, focusing on their relevance and the advantages.

3.2.1 Data set description

The data set was filmed on the roof of Fraunhofer IGP in Rostock, where a lot of solar panels are located, which produces energy every sunny day. Due to the difficult accessibility of the roof, the shooting was done not from the rover, but from human hands at approximately the same height and angle as in the calculated parameters for the rover. At least one in five panels has imperfections. The dataset consists of almost 50 panels, recorded on thermographic camera, it is approximately 8 GB of data and 17 .irb files that require exploration.

3.2.2 Chosen Technologies

The application's development was based on the source code provided by Infratec software. To compile the project, CMake [10], in conjunction with the Qt [44] and OpenCV [39] libraries, was utilized due to the necessity of working with image data and creating a graphical user interface (GUI). The primary coding work was carried out in C++ using Visual Studio 2017 [58]. This choice was made because C++ is well-suited for cross-platform development, and Qt, in particular, is a preferred option for building applications that can run on various operating systems and devices. This approach is crucial as it ensures that the application is not confined to a single operation system or device type. It also provides users with the flexibility to select the platform or device that best meets their requirements when using the application.

Qt is an indispensable tool for GUI development. Employing Qt Designer to craft screens featuring labels, sliders, and buttons customized for specific purposes, such as navigating through the dataset, adjusting temperature ranges, or initiating new error detection with updated parameters, is of paramount importance. These elements are then seamlessly integrated into the program code using the Qt library.

Two labels play crucial roles within the interface: the original frame and the frame with highlighted outliers. This design simplifies the process of outlier detection for users, empowering them to identify and report imperfections when necessary. A more comprehensive discussion of the GUI's intricacies will be presented in the Implementation chapter, shedding light on how the user interface enhances the usability and functionality of our outlier detection system.

OpenCV provides many in-build image processing functions, such as Normalization, Thresholding or different image filters, such as Bilateral filter, Gaussian blur, that will be discussed in the Methodology chapter.

4 Methodology

4.1 Overview

The objective of this project is to facilitate user-friendly outlier detection in thermographic data through the implementation of a GUI and an image processing program. This program is designed to identify and highlight frames in a dataset that exhibit imperfections, ultimately reducing the time and effort required for manual error inspection within large datasets. The primary approach focuses on visualizing complex data comprehensively, enabling users to work with information more efficiently.

This chapter presents all the methods and algorithms employed in constructing the outlier detection system. The process begins with data collection through handheld shooting, which often results in data that is not smooth or clear. Consequently, a preprocessing step is necessary to eliminate noise and anomalies unrelated to solar panel defects. This includes the utilization of OpenCV in-build functions and an explanation of why these operations are essential. Feature extraction is another critical aspect, potentially encompassing factors such as temperature variations or visual patterns. Additionally, the chapter covers the selection of an appropriate outlier detection algorithm or model.

The final section of the chapter is dedicated to methods that will be employed in the long-term perspective, intending a fully automated system where user intervention is unnecessary.

4.2 Image preprocessing operations

Before proceeding with image segmentation, each image must undergo preprocessing operations. These operations aim to enhance the frame's quality and enhance the probability of achieving superior results.

4.2.1 Image normalization

The program takes a matrix temperature as an input and applies Min-Max scaling, also known as Normalization, to it. This process linearly scales the pixel values to fit within the range [0, 255]. By doing so, it guarantees that the minimum original pixel value corresponds to 0, and the maximum pixel value corresponds to 255. All other pixel values are adjusted proportionally within this range according to their initial values, ensuring that the image data falls within a standardized range. This normalization removes bias introduced by variations in pixel value scales, which is crucial for fair image comparisons and optimizing the performance of subsequent image processing algorithms and machine learning models.

4.2.2 Convert color operation

In the context of the project, the pixel intensity in a grayscale image signifies the heat level in a specific area of the frame, with 0 representing black and 255 representing white, and the remaining 254 values representing various shades of gray.

Figure 4.1: Image of solar panel

Image taken from [38]

Figure 4.2: Image of solar panel after converting to grayscale

Image taken from [38]

In Figures 4.1 and 4.2, examples of grayscale conversion are presented. Grayscaling provides several advantages. Firstly, it often mitigates the impact of noise in the image by eliminating noise in color channels that aren't relevant to the features of interest. Secondly, grayscale images, having only a single channel, simplify the image data and reduce the computational complexity of subsequent operations. This streamlining enhances contrast between objects or features of interest and the background. Lastly, grayscale images are more memory and storage-efficient than their color counterparts, which is especially beneficial when working with large datasets.

4.2.3 Alternative approaches considered

While the current sequence of image preprocessing steps is well-suited for thermographic data outlier detection, it's essential to explore alternative methods and understand why they were not chosen. Here are some alternative approaches that could have been considered:

Color Retention: An alternative approach could have been to preserve the color information in the images instead of converting them to grayscale. However, this approach was likely discarded because color channels can contain irrelevant data and noise unrelated to defect detection. Retaining color could introduce unnecessary complexity without substantial benefits.

Furthermore, color images require more memory and storage, which could be impractical for large datasets.

Different Normalization Techniques: Instead of Min-Max scaling, other normalization techniques such as Z-score standardization or robust scaling might have been explored. However, the chosen Min-Max scaling method is simple and effective at bringing pixel values within a specified range. Z-score standardization, for instance, might center the data around zero, which may not be suitable for thermographic data. Robust scaling could be considered in the presence of outliers, but if outlier handling is performed later in the pipeline, it may not be necessary. The chosen method offers a balance between simplicity and effectiveness without adding unnecessary complexity.

Additional Preprocessing Steps: The preprocessing pipeline could have included extra steps like contrast enhancement or edge detection. These steps can improve image quality and feature extraction. However, the decision to convert to grayscale already enhances contrast, and subsequent steps in the pipeline may incorporate these operations. Adding more preprocessing steps could increase computational complexity and processing time without a significant improvement in defect detection accuracy.

In summary, the selected sequence of preprocessing steps was chosen for its effectiveness, simplicity, and efficiency. The alternatives considered may have introduced complexity without clear benefits for the specific task of defect detection in thermographic data. The chosen sequence aligns with widely accepted practices in image processing and machine learning, striking a balance between data preparation and computational efficiency.

4.3 Image processing operations

4.3.1 Thresholding operation

After converting a normalized frame into a grayscale image, the next step is to apply a thresholding operation. The fundamental concept behind thresholding is to transform a grayscale image into a binary image, where pixels are classified as either foreground or background based on a predefined threshold value. In the context of this project, Otsu's Thresholding is employed. In Figure 4.3, the grayscale image from the previous section is presented as a binary image.

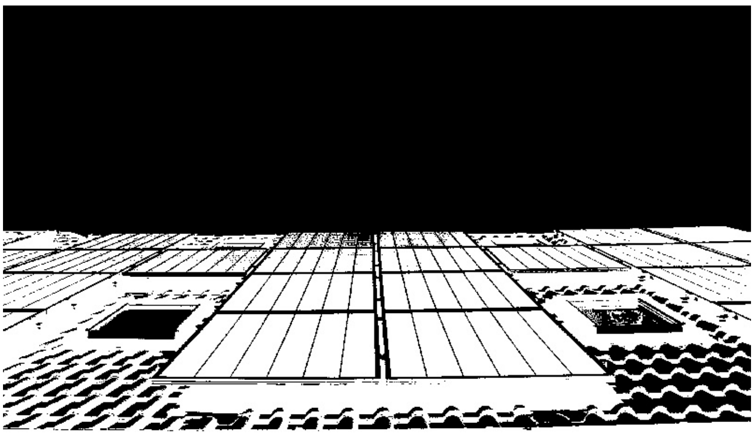

Figure 4.3: Image of solar panel after thresholding

Image taken from [38]

No Manual Threshold Selection: Otsu's Thresholding eliminates the need for manual threshold selection. In contrast, alternative methods like fixed thresholding require a predefined threshold value, which can be unmanagable and often less effective. This is particularly advantageous when dealing with diverse images characterized by varying lighting and contrast conditions. Otsu's method adapts to the specific image content, ensuring that it consistently provides an optimal threshold, regardless of changing input conditions.

Adaptability to Image Content: Otsu's Thresholding excels in adaptability. It automatically computes the optimal threshold based on the image's histogram, enabling it to handle a wide range of images effectively. Whether an image is well-lit or poorly lit, high-contrast or low-contrast, Otsu's method maximizes the separation between the foreground and background, making it a robust choice for a variety of scenarios.

Maximizing Interclass Variance: Otsu's method is preferred because it specifically focuses on maximizing interclass variance. This emphasis makes it particularly effective in distinguishing objects from the background. By maximizing the separation between the foreground and background, Otsu's Thresholding provides more accurate results in object segmentation and analysis.

Considered alternatives were discarded for the following reasons:

- Fixed Thresholding: Manual threshold selection in fixed thresholding may lead to inconsistent results, especially when images exhibit diverse lighting and contrast conditions.

- Adaptive Thresholding: Adaptive thresholding methods compute a threshold value for each pixel based on the local neighborhood. While they can be useful in certain situations, Otsu's Thresholding often outperforms them because it determines a global optimal threshold that maximizes the separation between classes across the entire image.

- Global Histogram-based Thresholding: Some other global histogram-based methods might have been considered. However, Otsu's method stands out due to its specific focus on maximizing interclass variance. This focus allows it to be particularly effective in distinguishing objects from the background.

In summary, the selection of Otsu's Thresholding in the sequence of steps for thresholding was driven by its ability to automate threshold selection, adapt to diverse image conditions, and maximize the separation between the foreground and background. This choice ensures robust and reliable performance across a wide range of images, making it the preferred method for this project.

4.3.2 Morphological operations in image processing

In image processing, morphological operations are employed to manipulate the shape and structure of objects within an image. These operations utilize a structuring element, which is a small template, to process an input image and generate an output image of the same size. Some of the most well-known morphological operations include:

Erosion: During erosion, the structuring element is moved across the image. If all the pixels within the structuring element overlap with the object in the image, the corresponding output pixel is set to 1, otherwise, it is set to 0. Erosion has the effect of shrinking objects and can be used to remove small details.

Dilation: Dilation is the opposite of erosion. It enlarges objects by adding pixels to their boundaries. If at least one pixel within the structuring element overlaps with the object, the output pixel is set to 1, otherwise, it is set to 0.

Opening: Opening is a combination of erosion followed by dilation. It is employed for noise reduction and the removal of small protrusions while preserving the boundaries of objects.

Closing: Closing is the inverse of opening and is used to fill in small holes and gaps in the foreground while preserving larger structures. It is achieved by first applying a dilation operation followed by an erosion operation using the same structuring element [9].

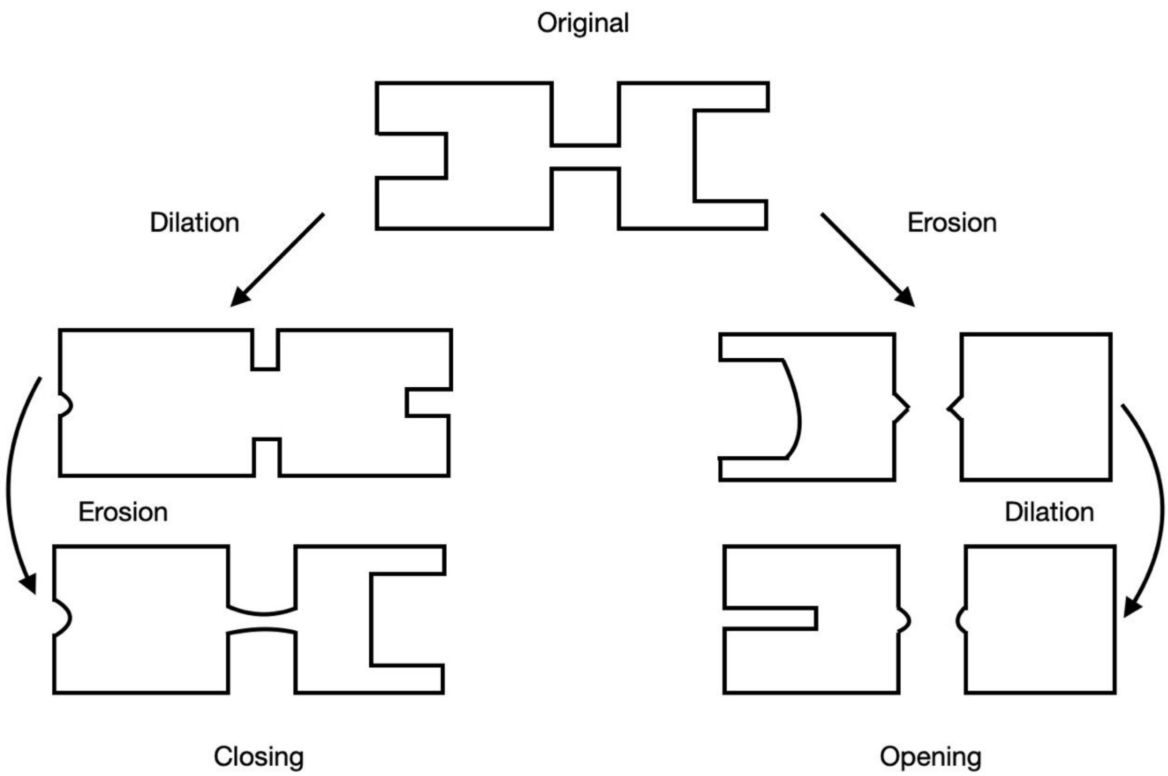

Figure 4.4: Output of Compound operations on an input object

Image redrawn from [43]

Most morphological operations do not rely simply on either dilation or erosion, instead, they involve both operations. Figure 4.4 illustrates compound operations performed on a single object. In the context of the outlier detection program, the opening operation is selected, The choice of "opening" for outlier detection in solar panel data was made to separate solar panels that are in close proximity to each other. Erosion within the opening operation was crucial for noise reduction, which is a common issue in image data. Dilation in opening effectively separated closely spaced solar panels by preserving their boundaries while removing connections between them. Alternative operations, like "closing", would have connected adjacent panels, which was not the desired outcome for outlier detection in this context. Similarly, other alternatives, such as using "dilation" alone, would not have adequately reduced noise or preserved object boundaries, making them unsuitable for this specific task. The choice of "opening" struck a balance between noise reduction and object separation, aligning with the requirements of the outlier detection program in the context of solar panel data. Furthermore, this operation includes an input parameter: the number of iterations. This parameter determines how many times this operation is applied. It is one of the four primary parameters that will need to be configured for each dataset in the future plotly dashboard.

4.3.3 Image segmentation with Watershed algorithm

Now, the Watershed algorithm is ready for application. A grayscale image can be likened to a topographical landscape, where brighter regions resemble elevated hills, and darker regions resemble lower-lying valleys. Imagine pouring different-colored water into separate valleys (low points) in the image. As the water rises, it will mix where there are slopes (changes in brightness). To prevent this mixing, barriers are built between valleys where water is merging, in Figures 4.5 and 4.6, there is an example of applying watershed algorithm on picture with coins. Keeping on filling with water and adding barriers until all the high points are underwater. This is the basic idea of the watershed method [39].

Figure 4.5: Coins picture before segmentation

Image taken from [38]

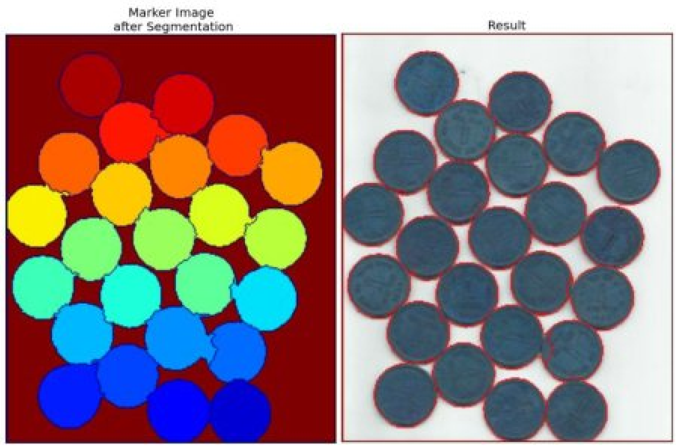

Figure 4.6: Coins picture after segmentation

Image taken from [38]

In more straightforward terms, the watershed method draws an analogy to the concept of pouring water and constructing barriers within an image to segment its different components, hence its name.

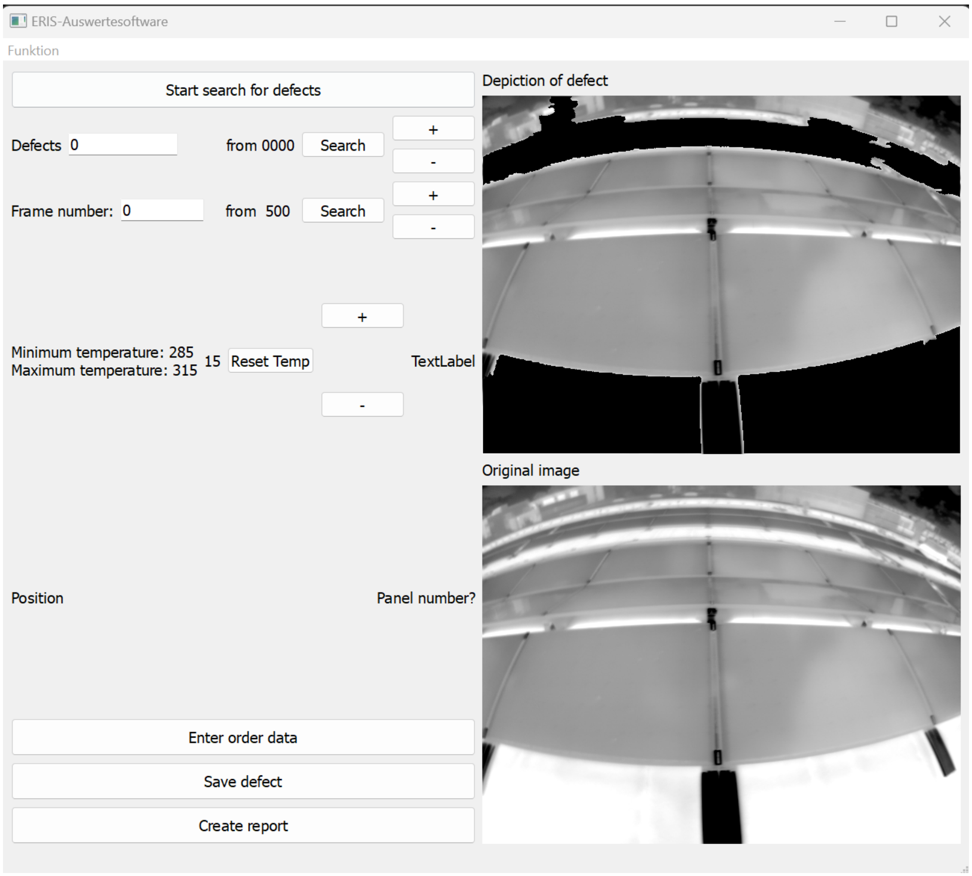

Figure 4.7: Watershed algorithm on GUI example

When exploring alternative methods for image segmentation, various options were examined, such as thresholding, contour-based techniques, and machine learning-driven approaches like Mask R-CNN. However, each of these alternatives exhibited certain limitations. Thresholding, for instance, faced difficulties in accurately segmenting objects with complex shapes and varying intensities, which are often encountered in real-world images. Contour-based methods, on the other hand, struggled when dealing with intricate object shapes and might necessitate additional post-processing to achieve precise segmentation. Machine learning methods, while undeniably potent, demanded a substantial amount of annotated data and computational resources for effective training. Given the relatively low resolution of the frames, employing this algorithm is deemed one of the most effective means to locate the solar panel. The thermographic camera's primary focus is on capturing the panel, positioning it in the foreground. By fine-tuning the algorithm's parameters, it consistently extracts the solar panel while excluding irrelevant elements such as panel borders, flooring, and the sky, effectively generating a new frame. In Figure 4.7, an example of using watershed algorithm with solar panels is presented.

4.3.4 Canny edge detection

To extract boundaries of solar panel, edge detection algorithms are to be used. Edge detection is an essential technique in image processing and computer vision that plays a crucial role in identifying object boundaries and regions of interest. It helps to extract important features by highlighting sharp changes in intensity within an image.

Various edge detection methods have been developed to meet different requirements and scenarios. These methods use different mathematical approaches, such as gradient-based operators, convolutional filters, and zero-crossing techniques, to locate edges and emphasise significant transitions in pixel values. One popular and widely used library for edge detection is OpenCV (Open Source Computer Vision Library) [39].

The process of edge detection involves identifying abrupt changes in pixel intensity, which typically correspond to transitions between different regions or objects within an image. These transitions can occur due to changes in color, texture, or other visual properties. By detecting and highlighting these edges, valuable information about the structure and boundaries of objects in the image is to be extracted [6].

The Canny algorithm applies a series of steps to detect edges effectively. It involves smoothing the image to reduce noise, calculating gradient information to determine the magnitude and direction of intensity changes, applying non-maximum suppression to thin out the edges, and finally, applying hysteresis thresholding to determine the final set of edges [39].

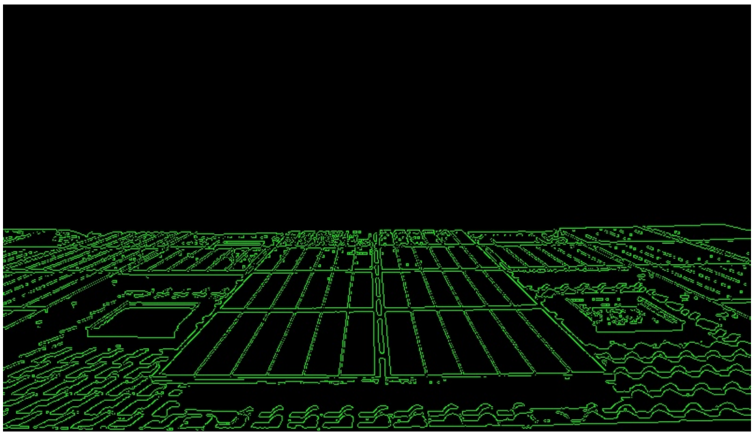

Figure 4.8: Image of solar panel after Canny edge detection

Image taken from [38]

In the specific context of delineating the boundaries of solar panels, it is imperative to utilize edge detection methods that offer a high degree of accuracy and noise resilience. The alternative methods, such as the Sobel and Prewitt operators, the Laplacian of Gaussian (LoG), and basic thresholding techniques, are less suitable due to their heightened susceptibility to noise and propensity for producing thicker edge representations. Given the necessity for precision in edge detection to accurately identify solar panel boundaries, the Canny edge detection algorithm emerges as the more fitting choice for this project.

In Figure 4.8, an image is shown after undergoing normalization, conversion to grayscale, and Canny edge detection. Additionally, it is possible to threshold the resulting contours based on their size. These size-based thresholds will be represented as the second and third parameters in the plotly dashboard, with the second parameter denoting the minimum outlier size and the third parameter representing the maximum outlier size.

4.3.5 Perspective transformation

When capturing the panel using Canny edge detection, there is a requirement to focus solely on the panel itself. In order to align the solar panel's size with that of the entire frame, a perspective transformation becomes essential. Figure 4.9 provides an illustration of how this approach functions, allowing for the redefinition of coordinates to exclusively examine the area of interest.

Perspective transformation refers to the distortion that occurs when a three-dimensional object or scene is projected onto a two-dimensional surface, such as a camera sensor or an image. This distortion arises due to the way our eyes and cameras perceive depth and distance, as explained by Nishan in their work [23].

When a camera captures an image of solar panels, for instance, perspective transformation can cause the panels to appear smaller as they recede into the distance. This phenomenon occurs because objects that are farther away from the camera appear smaller in the resulting image.

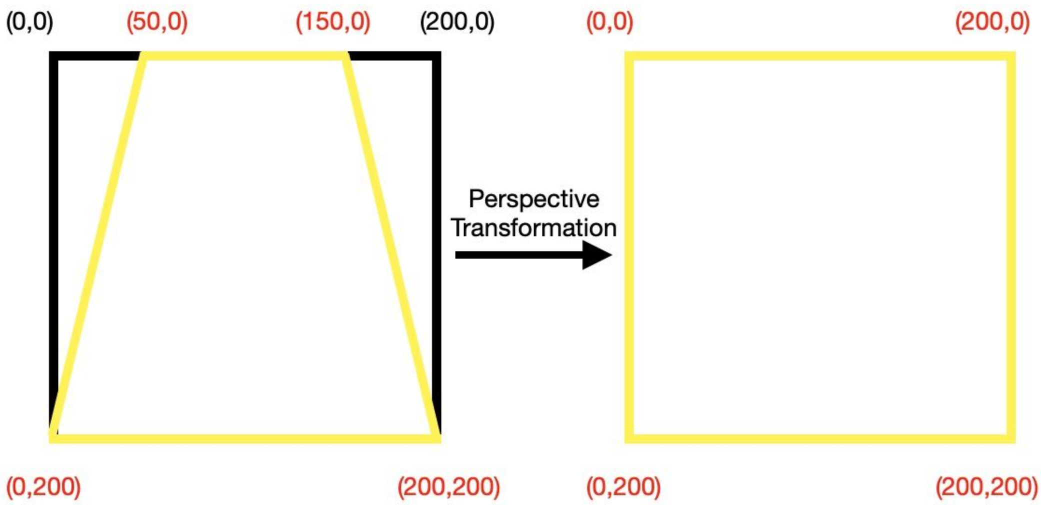

Figure 4.9: Perspective transformation example

Image redrawn from [46]

To illustrate this, imagine you are standing in front of a row of solar panels. The panels closest to you will appear larger, while those farther away will appear smaller. This change in size and shape is a direct result of perspective transformation.

Understanding perspective transformation is crucial in the fields of computer vision and image processing. It allows us to recreate the three-dimensional appearance of objects within two-dimensional images. By comprehending and compensating for perspective distortion, we can accurately analyze and interpret the visual information captured by cameras, as highlighted by Junshu's work [64].

In this project, the preference for perspective transformation over other methods is driven by practical considerations. While camera calibration is undoubtedly beneficial, it necessitates precise knowledge of camera parameters, which can be challenging to obtain when dealing with camera's software. Another alternative, 3D reconstruction, although highly accurate, demands capturing multiple images from various angles, making it resource-intensive and less suitable for real-time or single-image analysis. In contrast, perspective transformation emerges as a more versatile and widely applicable approach capable of effectively rectifying perspective distortion in a single image. This makes it a practical choice for solar panel inspection since it enables the alignment of the solar panel's proportions with the entire frame without the need for extensive prior information or resource-intensive data collection.

4.3.6 Statistical method in finding outliers

Visual inspection is a straightforward and intuitive method to identify outliers in data using scatter plots or other types of graphs. By looking at the scatter plot, it is conceivable to visually examine the distribution of data points and observe if any points stand out or appear different from the majority. Outliers are typically data points that lie far away from the main cluster or trend of the data [4].

Visual inspection is a useful initial step in outlier detection as it allows for a quick overview of the data. However, it may not be the most precise method, especially when dealing with larger data sets where identifying outliers solely by eye becomes challenging [4]. This is where additional measurement methods, such as statistical tests, can prove to be valuable.

Statistical tests offer a more precise approach for identifying outliers, as they provide numerical measures to gauge the extent of deviation of a data point from typical values. Two commonly employed statistical tests for outlier detection include the z-score and the interquartile range (IQR) [19].

The z-score measures how many standard deviations a data point is away from the mean of the data set. A high z-score indicates that the data point is far from the mean and could be an outlier [19]. Generally, data points with z-scores greater than a certain threshold, such as 2 or 3, are considered outliers.

The IQR is the range between the first quartile (25th percentile) and the third quartile (75th percentile) of the data set. Any data point that falls below the first quartile minus 1.5 times the IQR or above the third quartile plus 1.5 times the IQR is considered an outlier [4].

Using this statistical measure, can quantitatively determine whether a data point is an outlier and provide a more precise measure of how extreme the deviation is. This helps in distinguishing genuine outliers from natural variations in the data.

At present, the thermal image comprises individual pixels, each representing the temperature of the corresponding section of a solar panel. To identify outliers within this thermal image, the Z-Score method becomes applicable. This method calculates the extent to which a pixel's temperature deviates from the average temperature, as outlined by Abdulmalik [63].

After the application of perspective transformation, all pixels within the frame become constituents of the solar panel. This transformation enables the utilization of the Z-score method to identify and subsequently highlight outliers.

If the Z-score method is not employed for outlier detection, an alternative approach involves the utilization of the fourth parameter on the dashboard, which pertains to final temperature thresholding.

4.3.7 Overlay annotations for hotspots

When identifying potential outliers in solar panel data, it is essential to mark them with overlay annotations. These annotations enable users to later highlight significant data points or distinctive features. Overlay annotations play a crucial role in enhancing data analysis, as they provide a visual means to emphasize data points or regions that deviate from the norm.

The advantages of using overlay annotations in visual analytics are numerous. They allow for a focused exploration of critical information by directing the user's attention to specific areas of interest. Annotations also provide additional context through supplementary information, such as labels, descriptions, or metadata, thereby deepening the user's understanding of the data. Furthermore, they facilitate comparative analysis, making it easier for users to discern differences between various aspects of the data. Additionally, interactive annotations empower users to access detailed information or perform further analysis by clicking or hovering over annotated elements.

The creation of an overlay annotation system in this project, with the aim of enhancing the value and insights derived from the data, can be achieved by fulfilling to the following steps:

- **Overlay Annotation Generation**: The annotations take the form of markers placed directly on data points representing hotspots. These markers, whether in the shape of circles, crosses, or other symbols, serve as visual signposts, making the hotspots unmistakably visible and readily identifiable.

- **Bounding Boxes**: In addition to markers, bounding boxes are also used to enclose the identified hotspots. These bounding boxes provide a clear, visually distinct representation of the hotspot areas, aiding data analysts in assessing the extent and boundaries of anomalies more precisely.

- **Metadata Inclusion**: To provide context and additional information about each hotspot, metadata is included within each annotation. This metadata typically includes information such as the severity of the hotspot, its precise coordinates, and a timestamp, enabling analysts to quickly comprehend the importance and context of each hotspot and to enhance future outlier reports with additional information.

- **Integration into Data Visualization**: Finally, the overlay annotations are seamlessly integrated into the solar panel data visualization, which may take the form of heatmaps, images, or other relevant formats. Care is taken to ensure that the annotations are presented in a manner that is both clearly visible and non-intrusive, allowing data analysts to assess the hotspots while retaining access to the underlying data.

In conclusion, the use of overlay annotations in solar panel data analysis, in combination with appropriate hotspot detection algorithms, plays a huge role in identifying and categorizing potential hotspots. This approach contributes to informed decision-making, enhances data analysis, and optimizes solar panel performance overall. An example of overlay annotations used in solar panel data is situated in the Results chapter.

4.4 Interactive dashboard with plotly

In the context of image processing parameter optimization, the need for extensive data collection through outlier detection cannot be overstated. With four parameters at play, each requiring multiple evaluations in some range, the process of gathering sufficient data points can be both time-consuming and resource-intensive. To address this challenge, this chapter explores the utilization of plotly [41], a versatile web-based data visualization platform, to streamline the outlier detection process. This approach significantly reduces the time required to compile the necessary data set, making it feasible to run the outlier detection method multiple times in an efficient and organized manner.

The visualization is held with using of scatter plots, that are valuable tools in the context of detecting the best parameters for image processing for several scientific reasons:

- Parameter Exploration: Scatter plots allow us to visualize the relationship between different parameter combinations and the corresponding performance metrics.
- Multidimensional Analysis: Scatter plots can display multiple parameters simultaneously in a two-dimensional space.
- Identification of Optimal Settings: Scatter plots can reveal regions in the parameter space where the image processing algorithm performs optimally.
- Visual Insights: Scatter plots provide a visual representation of data, making it easier to spot patterns, trends, and outliers.

In Figure 4.10, there is a scatter plot that serves as a valuable tool for refining the outcomes of an outlier detection program. This scatter plot is interactive and includes four sliders corresponding to different parameters for fine-tuning data point thresholding. These parameters consist of the maximum and minimum hotspot size, the number of morphological operation iterations, and the final threshold setting. Users can adjust these sliders to influence data point outcomes, enhancing accuracy.

The scatter plot contains more than 7000 data points, which provide users with extensive insights into finding the most suitable parameters for their specific data sets.

The scatter plot's two primary axes represent the number of defects and the average defect size. The size of each data point reflects the area of the outlier within the current frame. Therefore, if a data point appears significantly smaller or larger than the average, it can be considered a potential indicator of inappropriate parameters. When a user selects a data point, the result of the outlier detection program is displayed on the same dashboard, allowing users to optimize parameters based on highlighted hotspots in the figure.

In summary, Figure 4.10 is a practical tool for data analysis and optimization. Its interactive nature, extensive data points, and real-time feedback empower users to better understand their data and fine-tune parameters for improved results. It serves as a user-friendly compass in the field of data analysis, aiding users in achieving accurate and meaningful outcomes.

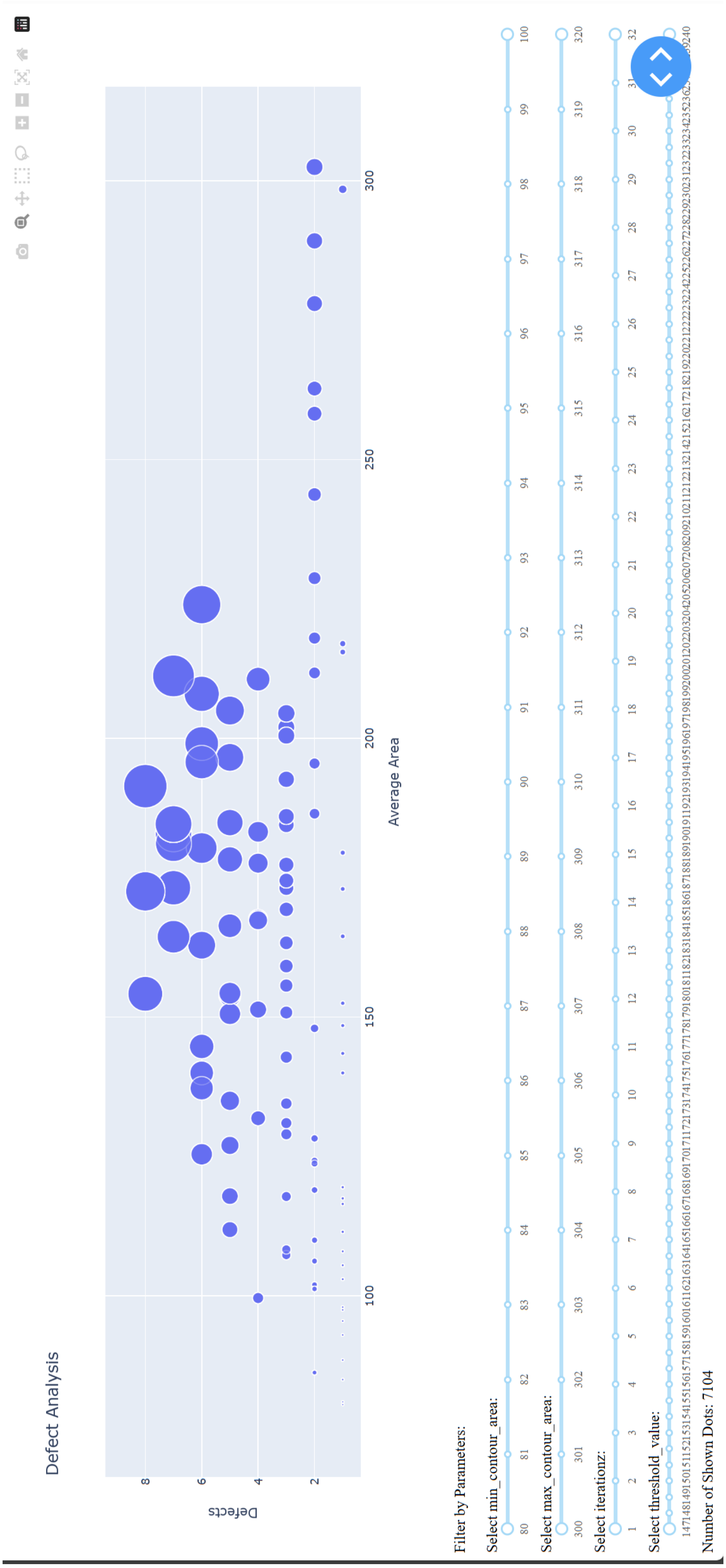

FIGURE 4.10: EXAMPLE OF SCATTER PLOT WITH 4 SLIDERS

4.5 Long-term goal of the project

The final phase of the project involves an automotive system comprising a rover equipped with a robotic arm designed for data collection within solar panel fields. Leveraging the camera's positioning information and the panels' dimensions, it becomes possible to eliminate pixels that display excessive distortion from the computational process. Therefore, in addition to image processing, an algorithm for pixel restriction is to be incorporated.

4.5.1 Determining the optimal camera position

To determine the optimal camera position, we rely on camera and solar panel dimensions as well as angle parameters, which are detailed in Table 1. This information serves as the basis for generating a graph plot in Excel, illustrated in Figure 4.11. In this graph, solar panels designated for inspection are highlighted in brown, featuring a 20° tilt from the ground and a height of 0.3 meters, with a width of 3.5 meters. The green and dark red areas represent the camera's opening angle, which is set at 80°, with an additional 10° reserved to account for rover fluctuations — 8° at the top of the panel and 2° at the bottom. The dark blue regions mark all feasible camera positions attainable by the robotic arm, ensuring that all requirements are met.

Camera				
Opening angle	[°]	80,00	[rad]	1,40
Angle (used)	[°]	70,00	[rad]	1,22
Angle reserve left	[°]	2,00	[rad]	0,03
Angle reserve right	[°]	8,00	[rad]	0,14
Resolution (length)	[px]	640	[px]	640
Resolution	(width)	[px]	480 [px]	480
Solar panel				
Length	[m]	25,00	[m]	25,00
Width	[m]	3,50	[m]	3,50
High	[m]	0,30	[m]	0,30
Slope	[°]	20,00	[rad]	0,35

Table 1: Camera and Solar Panel Dimensions and Angle Parameters

The graphic, Figure 4.11, was generated in Excel using various calculations and tables, which will be presented in full within this thesis. In light of the requirement for the camera to capture the solar panel as closely to perpendicular as possible, the camera's position was determined to be at a distance of 0.309 meters from the panel and a height of 2.172 meters above the ground.

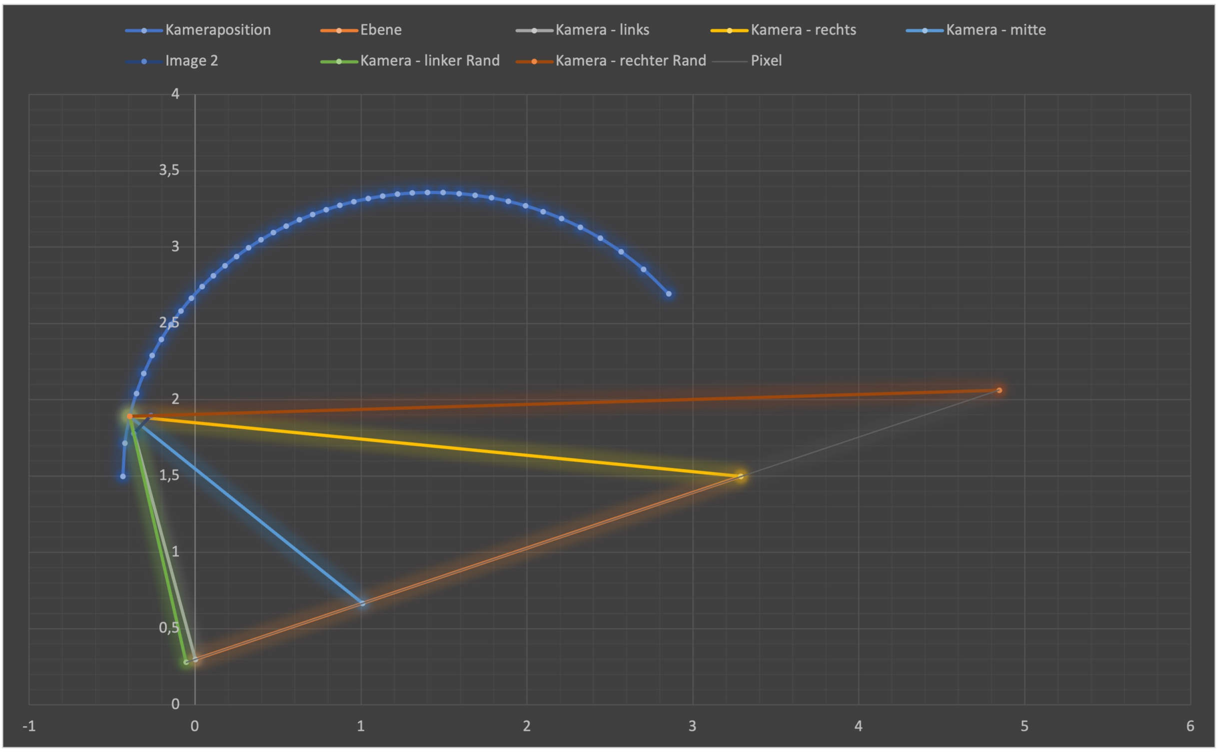

Figure 4.11: Variation of camera positions for solar panel inspection

4.5.2 Pixel restrictions

With the positioning of the solar panel relative to the camera established, the next step involves calculating distortion. To effectively utilize the information captured by the camera while observing the solar panel, specific conditions must be satisfied. Notably, the pixels should maintain at least 80% of their original size, and the angle between the camera and each pixel's perpendicular should fall within the range of 15 to 70 degrees. Otherwise, the data cannot be relied upon.

The calculation of distortion in the camera view of the solar panel was performed using MATLAB. This involved generating a mesh with dimensions of 640x480 to replicate the camera's resolution. The relevant code is located in Appendix 11.

The following figures present the program's outcomes, with the x and y axes representing the camera's pixels and the colors indicating the height above the ground. In the color legend, dark blue signifies a value of 0, indicating that those pixels do not meet the specified criteria. In Figure 4.12, pixels that fail to meet the requirements due to their size are marked. Figure 4.13 displays an ellipsoid, indicating angle restrictions. Lastly, Figure 4.14 combines the information from the first two figures.

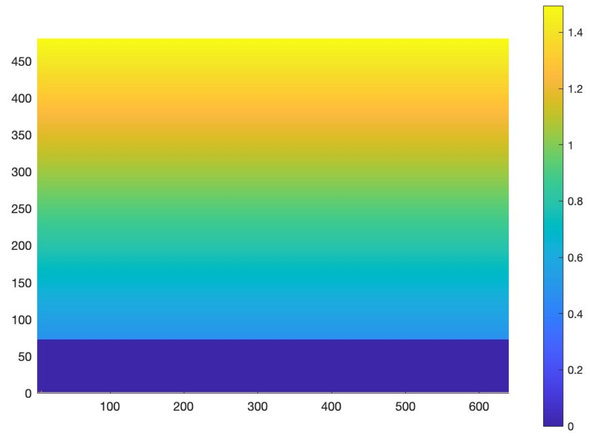

FIGURE 4.12: PIXELS FAILING SIZE REQUIREMENTS

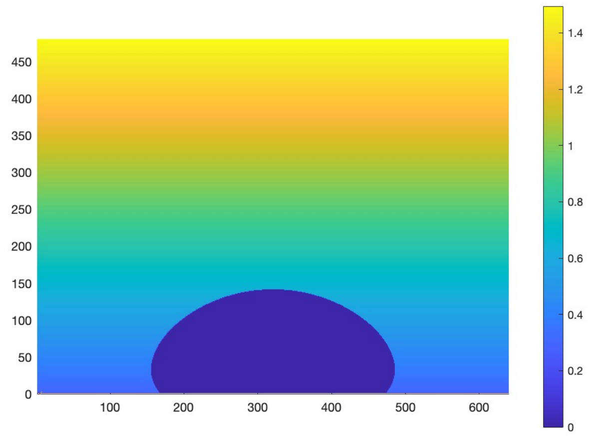

FIGURE 4.13: ANGLE RESTRICTIONS ILLUSTRATED

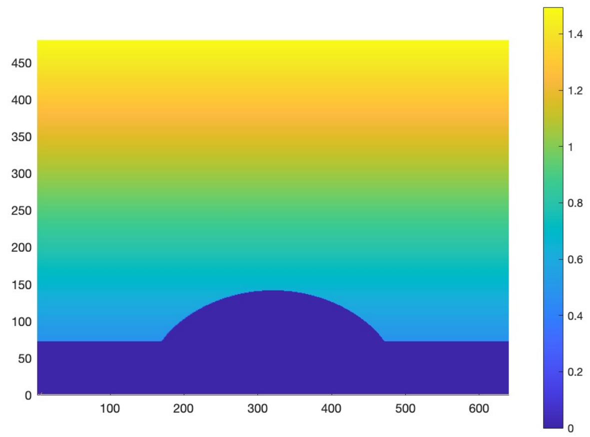

FIGURE 4.14: COMBINED EVALUATION OF SIZE AND ANGLE CRITERIA

5 Implementation

The practical instantiation of the methodologies unravelled in the previous chapter involves the hands-on implementation of the theoretical concepts. This involves the development and explanation of a Graphical User Interface (GUI), which serves as the interface for the CPP (C++ Programming Language) program's execution and result visualization. The GUI is designed to facilitate user interaction and effectively present the outcomes derived from the program's computations.

The creation of the GUI covers a detailed description of its components and functionalities. This includes the design principles, interactive elements, and visual components that collectively provide a coherent and user-friendly interface. The integration of the GUI with the CPP program, as outlined in the Methodology chapter, establishes a seamless interaction between users and the computational core. The practical application of the methodologies described in the Methodology chapter necessitates the integration of a GUI with the CPP program, enabling intuitive result visualization.

5.1 GUI for real-time outlier detection and visualization

Proficiency in statistical graphics necessitates the fusion of intricate concepts with lucid expression, precision, and efficiency. The visual representations should strive to attain a harmonious equilibrium between advanced concepts and a lucid, precise, and efficient presentation.

To enhance visualization, the decision has been made to divide the GUI screen into two distinct sections, as illustrated in Figure 5.1. On the left segment, users will encounter buttons and sliders designed to facilitate real-time data manipulation, accompanied by labels that provide users with relevant information. In contrast, the right portion is exclusively designated for displaying two labels: one that presents the original solar panel image under sign "Original image" and the other under "Depiction of defect" sign that exhibits the same image with highlighted outliers, if any exist.

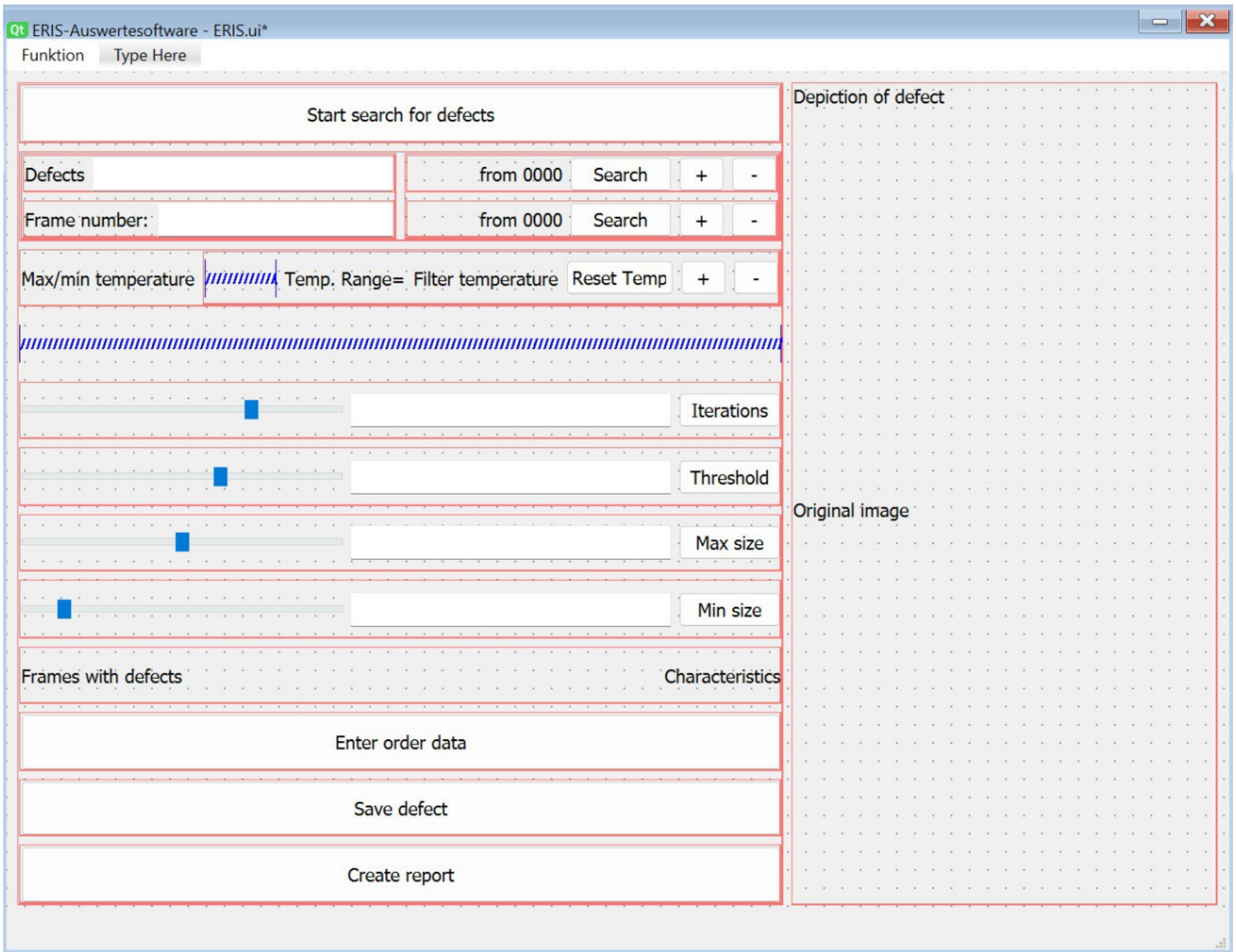

Figure 5.1: Graphical User Interface (GUI) for outlier detection program

After importing the .irb file to the program there are variety of possibilities. The "Frame number:" field displays the current frame's index. The maximum number of frames is capped at 500, equivalent to approximately 600 MB in size. Users have the capability to pinpoint a specific frame number using the "Search" button and adjust it incrementally using the "+" and "-" buttons.

A prominent button, labeled "Start search for defects," resides at the top, serving the purpose of identifying outliers. Upon activation, the program inspects all frames within the file and afterwards records any errors in the "Frame with defects" label. Subsequently, users can navigate through frames containing outliers, as well as traverse frame numbers.

The "Max/Min temperature" section provides the maximum and minimum temperature readings for the current frame. This information empowers users to make informed decisions regarding the Temperature Range (TR) they wish to employ. In cases where there are objects other than solar panels within the frame, proper visualization necessitates consideration of these objects. The presence of objects with significantly different temperatures impacts both, the maximum and minimum temperatures of the frame. For instance, a frame may exhibit a minimum temperature of 267.3 K, a maximum temperature of 323.2 K, and a TR of 30 K (Figure 5.2). Conversely, the same frame with a TR of 10 K (Figure 5.3) minimizes the influence of most obstacles, rendering the outliers more conspicuous.

The "Characteristics" section presents parameters specific to the current frame, including its resolution and environmental temperature. These parameters are determined by the camera settings, which may need to be adjusted as necessary.

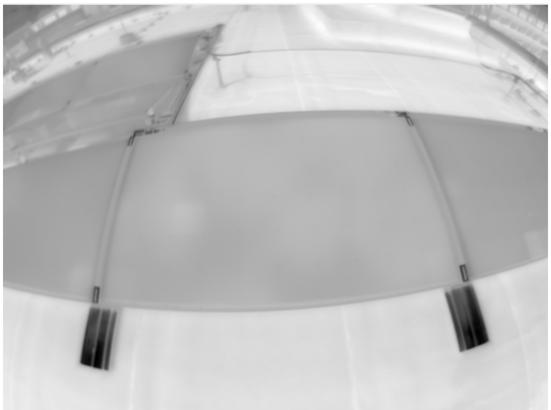

Figure 5.2: Frame of solar panel with 30 K TR

Figure 5.3: Frame of solar panel with 10 K TR

By adjusting the positions of the sliders, the outlier detection program undergoes real-time modifications, accepting fresh inputs and subsequently presenting a new frame with highlighted outliers.

5.2 Data input and reporting outliers

All defects that will be detected should be properly reported. To this purpose additional screens were added to the program.

The button labeled "Enter order data" functions as an initiator for the instantiation of an extra screen, Figure 5.4. Within this screen, user is afforded the capability to input suitable data related to testing procedures, inclusive of his personal identification details, which will subsequently be included into the final diagnostic report.

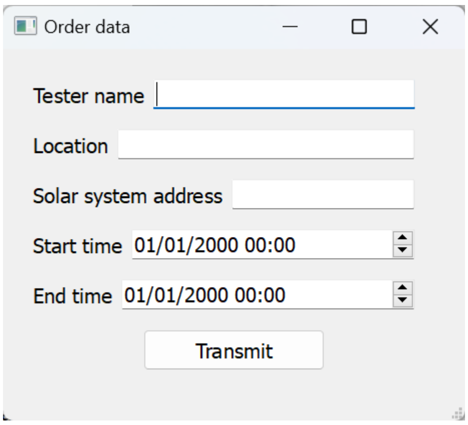

Figure 5.4: Additional screen for entering order data

Below "Enter order data" button, the button titled "Create report" triggers the emergence of a third screen, Figure 5.5. In this interface, the evaluator has the chance to add written comments and determine the seriousness of any abnormalities and variations they identify in the assessed material. The visual images and graphics displayed in this interface are taken from the computer's file system, making it convenient to quickly save instances of any unusual findings by using the "Save defect" button.

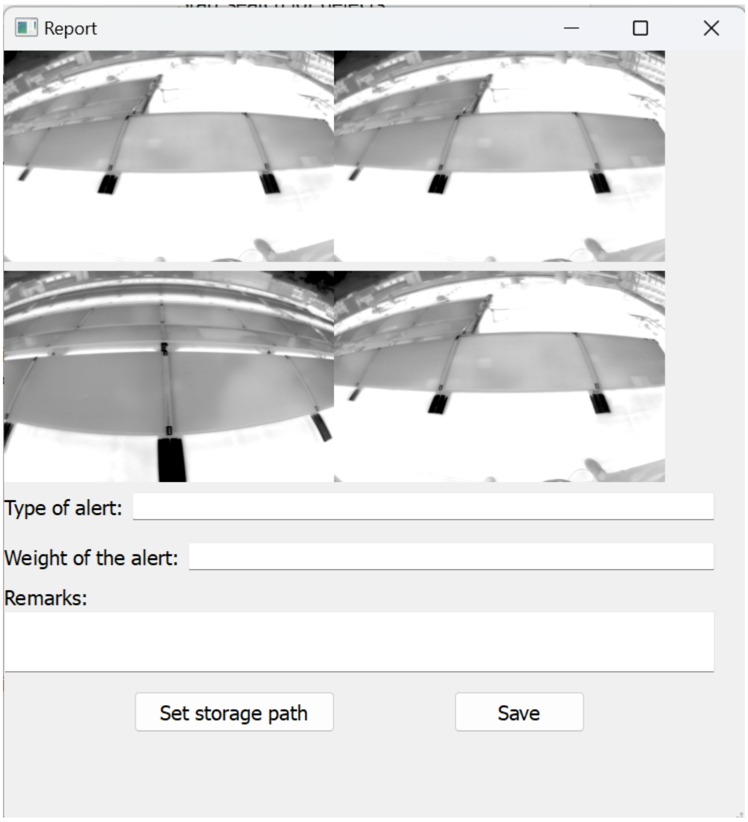

Figure 5.5: Additional screen for reporting the outliers

6 Results

In the following section, this article showcases the capabilities and versatility of the presented approach. It effectively processes low-resolution data, indicating its compatibility with high-resolution data as well. The section will present and discuss the outcomes of image processing operations, along with the dashboard's ability to visualize frames with highlighted outliers. This feature simplifies the selection of optimal parameters for users. Additionally, an example of a functional GUI is provided. When suitable parameters are employed, the system demonstrates highly precise outlier detection results. It is important to note that, given the absence of an open-source dataset containing solar panel images with known outlier parameters, result verification was conducted manually.

6.1 Illustrating outlier detection

To inspect the outcomes of the completed tasks, let us delve into a particular frame, Figure 6.1. Following the application of the watershed algorithm and the subsequent perspective transformation, it becomes possible to reveal outliers with a high degree of accuracy by employing a precise thresholding method. In Figure 6.2, the outliers are accentuated and enclosed within rounded rectangles to enhance the user experience.

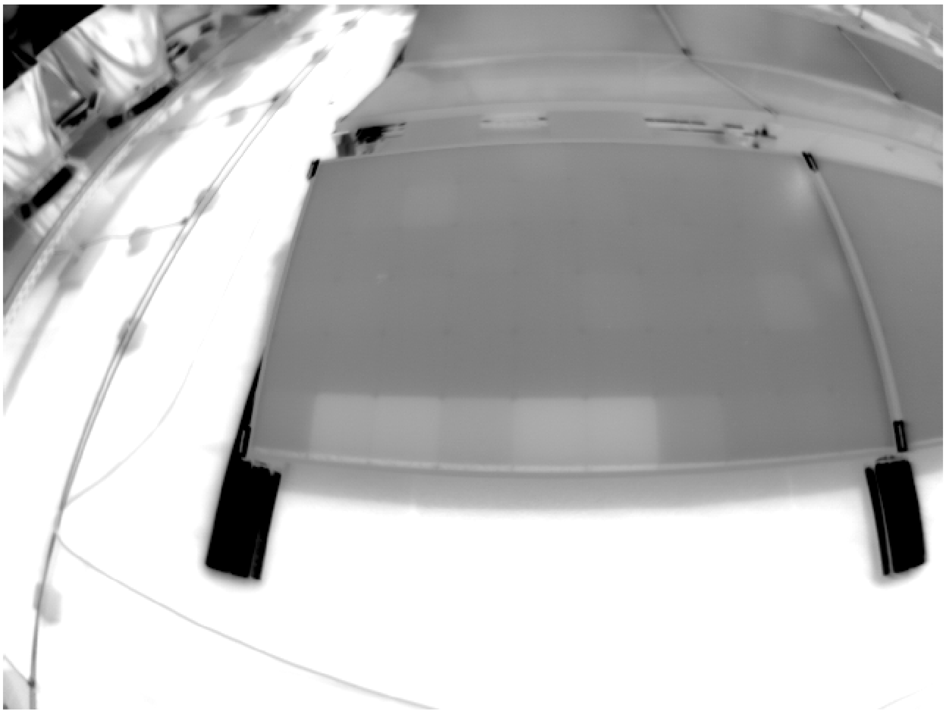

Figure 6.1: An example of analyzed image

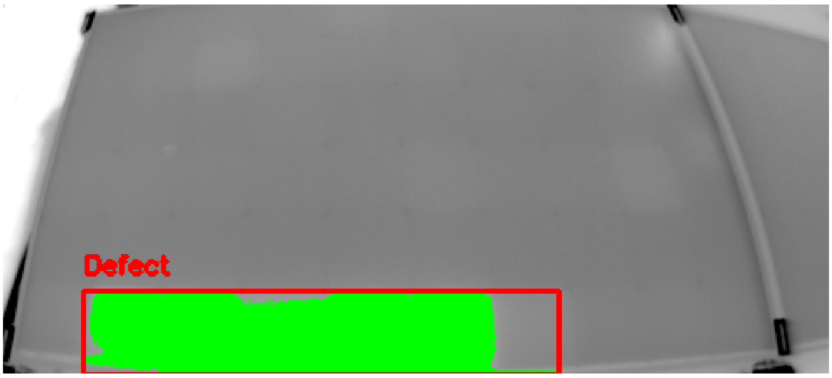

Figure 6.2: Outlier detection results with enhanced visualization

An additional illustrative instance of outlier detection within a distinct frame is presented in Figures 6.3 and 6.4.

Figure 6.3: Another example of analyzed image

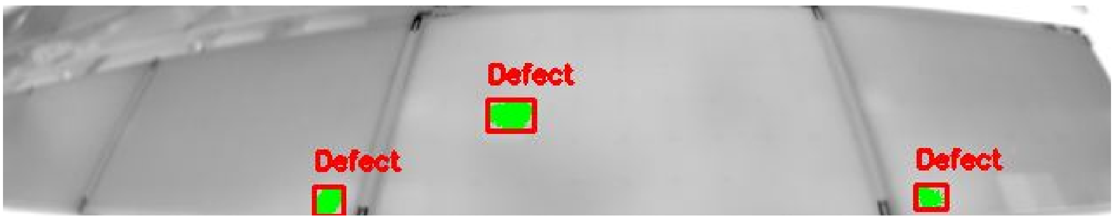

Figure 6.4: Result with highlighted outlier

As observed, an incorrect detection occurs for one outlier, specifically the one situated on the right. To reduce such occurrences, it is important to calibrate the parameters utilized for image processing.

6.2 Parameter optimization and visualization

Optimal parameters tailored to each individual image within the dataset can be determined using the plotly dashboard, as illustrated in Figure 6.6. Within the dashboard, users have the flexibility to navigate the plot by employing features like zooming, adjusting slider positions, applying data point thresholding. The size of each data point corresponds to the total area encompassed by outliers. By selecting specific nodes within the plot, users can access a practical outlier detection example associated with the respective image frame. This functionality enables users to make informed decisions regarding which parameters align better with the characteristics of the dataset. Moreover, the plot provides parameter values for a more comprehensive visualization of outliers, as depicted in Figure 6.5.

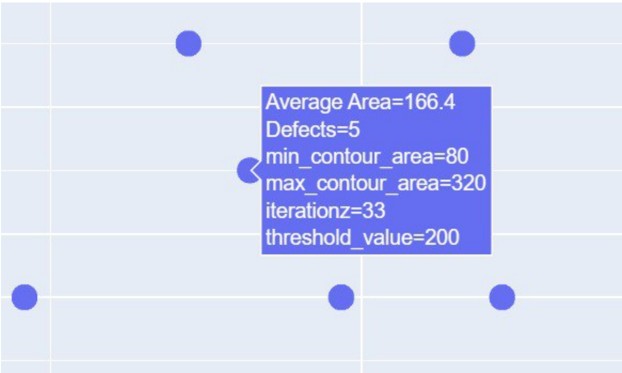

Figure 6.5: Scatter plot's data point example

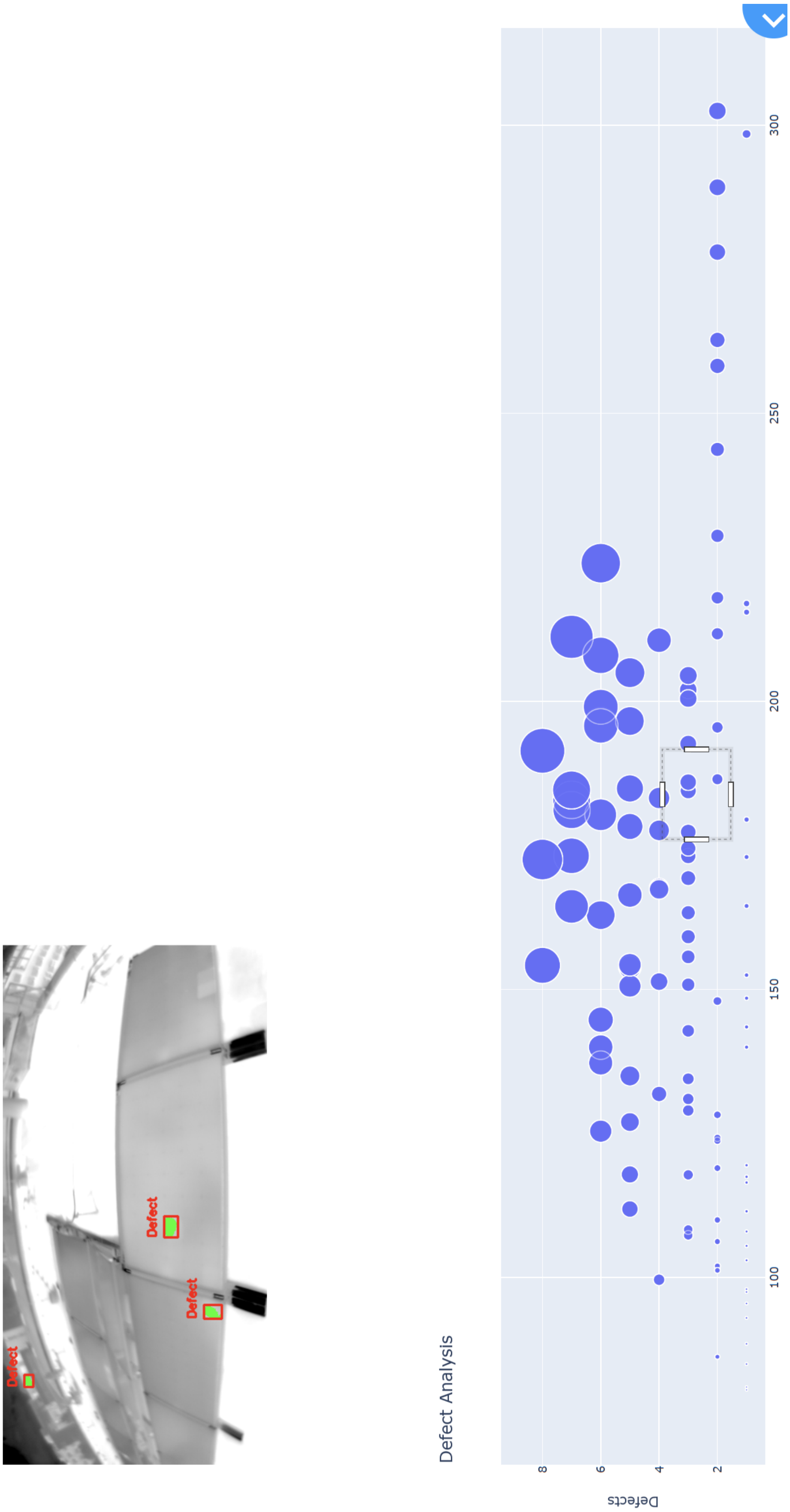

Figure 6.6: Scatter plot with outlier frame

6.3 Functional GUI for outlier reporting and export

The implemented algorithm operates within a graphical user interface (GUI) that allows users to report and export figures containing detected outliers. Figure 6.7 illustrates an instance of imperfection detection, showcasing the utilization of parameters obtained from the scatter plot.

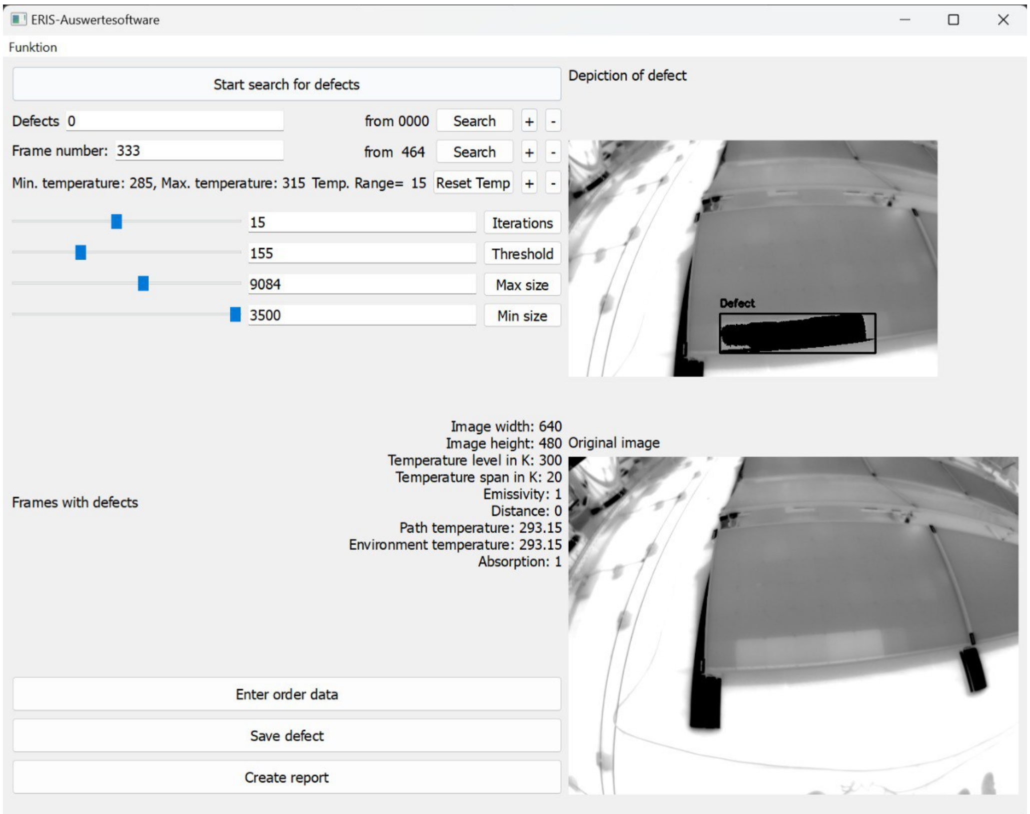

Figure 6.7: Final version of GUI

7 Discussion and Limitations

Despite the achievements and capabilities of the solar panel inspection system, certain limitations and challenges should be acknowledged:

1. Computational Resources: Processing large data sets and running image processing models can demand significant computational resources, potentially impacting system scalability. While the project aimed to create a cross-platform system, allowing users to work with thermographic data from any device, there are still constraints related to computational resources. This is due to the preprocessing computation required for the dashboard parameters.

2. Algorithm Optimization: The effectiveness of defect detection heavily relies on algorithm parameters. Determining the optimal parameters for various environmental conditions and solar panel types can be time-consuming, primarily because of the lengthy preprocessing step in dashboard creation. To enhance the system's time efficiency, machine learning algorithms as presented in the State of the Art chapter can be applied for outlier detection.

3. False Positives: Even with the appropriate preprocessing parameters, the system may still produce false positives or miss certain defects. This necessitates ongoing refinement of the anomaly detection algorithms.

4. Low Resolution Data: Low-resolution data may contribute to these challenges. To improve the outlier detection system's effectiveness, a combination of stereo cameras, capable of building a 3D representation of the panel, and low-resolution thermographic cameras, suitable for marking outliers, may be employed. With better resolution data, the image processing algorithms would function more robustly.

5. Cost: However, it's important to acknowledge that the integration of thermographic cameras, rovers, and computing resources can be costly, limiting the accessibility of such systems to some users. Therefore, this system is primarily designed for companies concerned about the condition of their solar panels and who have the means to invest in it.

In conclusion, the solar panel inspection system's development and integration of various components and technologies have the potential to revolutionise solar panel maintenance and quality assurance. While it shows promise, ongoing research and development are needed to address limitations and enhance its capabilities further.

8 Conclusion and Future Work

The development of the solar panel inspection system represents a significant step forward in the field of renewable energy maintenance and quality assurance. The integration of electrical and technical components and technologies has resulted in a robust and user-friendly system capable of detecting defects and anomalies in solar panels.

The methodology employed, including pixel coordinate calculations, edge detection, segmentation, and parameter optimization, has allowed for precise defect localization and accurate anomaly detection. The use of plotly dashboard for parameter optimization has streamlined the process of finding optimal image processing settings, enhancing the system's performance.

The user interface, created using Qt, offers a user-friendly experience, allowing users to input testing data, generate reports, and save images for documentation. However, certain limitations and challenges, such as algorithm optimisation, and false positives, need to be addressed for further improvement.

Looking ahead, there are further enhancements and refinements to be implemented as part of the ongoing project's development:

- Algorithm Refinement: Continued research and development are needed to further refine the anomaly detection algorithms. This includes optimising parameters and exploring advanced machine learning techniques to reduce false positives and improve overall accuracy.
- Integration of Additional Sensors: To enhance defect detection and characterisation, the system could be extended to incorporate additional sensors, such as ultrasonic sensors or visual cameras, to provide complementary data for a more comprehensive analysis.
- Scalability: Consideration should be given to the scalability of the system to accommodate larger solar panel arrays and diverse environmental conditions. This may involve the use of automated inspection fleets.
- Cost Optimization: Exploring cost-effective hardware solutions without compromising performance will make the system more accessible to a wider range of users.
- Integration with Predictive Maintenance: Integrating the system with predictive maintenance models can help forecast potential defects and optimise maintenance schedules, further increasing the efficiency of solar panel operations.
- User Training and Documentation: Providing comprehensive training and documentation for end-users will be crucial to ensure the system is used effectively and to its full potential.
- Study the impact of environmental parameters: Enhancing the system's adaptability to various environmental conditions, such as extreme temperatures or adverse weather, will make it more robust and reliable.

In summary, the solar panel inspection system represents a promising technology with the potential to improve the efficiency and reliability of solar energy production. Ongoing research and development, coupled with addressing current limitations, will be essential to realise the full potential of this system and contribute to the sustainability of renewable energy sources.

9 Bibliography

References

[1] Jubaer Ahmed and Zainal Salam. "An improved perturb and observe (PO) maximum power point tracking (MPPT) algorithm for higher efficiency". In: *Applied Energy* 150 (2015), pp. 97–108. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2015.04.006.

[2] Estefanía Alfaro-Mejía et al. "Dataset for recognition of snail trails and hot spot failures in monocrystalline Si solar panels". In: *Data in Brief* 26 (2019), p. 104441. ISSN: 2352-3409. DOI: 10.1016/j.dib.2019.104441.

[3] Tadatoshi Babasaki and Yuji Higuchi. "Using PV string data to diagnose failure of solar panels in a solar power plant". In: *2018 IEEE International Telecommunications Energy Conference (INTELEC)*. 2018, pp. 1–4. DOI: 10.1109/INTLEC.2018.8612400.

[4] Zuriana Abu Bakar, Rosmayati Mohemad, and Akbar Ahmad. "A Comparative Study for Outlier Detection Techniques in Data Mining". In: *IEEE Conference on Cybernetics and Intelligent Systems* (2006). DOI: 10.1109/ICCIS.2006.252287.

[5] Gökay Bayrak and Mehmet Cebeci. "Monitoring a grid connected PV power generation system with labview". In: *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*. 2013, pp. 562–567. DOI: 10.1109/ICRERA.2013.6749819.

[6] John Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

[7] Akash Singh Chaudhary and DK Chaturvedi. "Thermal image analysis and segmentation to study temperature effects of cement and bird deposition on surface of solar panels". In: *International Journal of Image, Graphics and Signal Processing* 9.12 (2017), p. 12. DOI: https://doi.org/10.5815/ijigsp.2017.12.02.

[8] Haiyong Chen et al. "Solar cell surface defect inspection based on multispectral convolutional neural network". In: *Journal of Intelligent Manufacturing* 31.2 (Dec. 2018), pp. 453–468. DOI: 10.1007/s10845-018-1458-z. URL: https://doi.org/10.1007%2Fs10845-018-1458-z.

[9] Ravishankar Chityala and Sridevi Pudipeddi. *Image Processing and Acquisition using Python*. Chapman and Hall/CRC, 2014. ISBN: 9781498760577.

[10] *Cmake C++*. Accessed: 2023-July. 2018. URL: https://rvarago.medium.com/introduction-to-cmake-for-cpp-4c464272a239.

[11] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[12] Bogdan-Adrian Enache, Florin-Marian Bîrleanu, and Marin RăduŢ. "Modeling a PV panel using the manufacturer data and a hybrid adaptive method". In: *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. 2016, pp. 1–6. DOI: 10.1109/ECAI.2016.7861108.

[13] Vinicius C. Ferreira, Ricardo C. Carrano, and Silva. "Fault detection and diagnosis for solar-powered Wireless Mesh Networks using machine learning". In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 2017, pp. 456–462. DOI: 10.23919/INM.2017.7987312.

[14] Brian David Fisher. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005, pp. 69–104. ISBN: 0769523234.

[15] Elyes Garoudja et al. "Statistical fault detection in photovoltaic systems". In: *Solar Energy* 150 (2017), pp. 485–499. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2017.04.043.

[16] Hamdani Hamdani et al. "Real Time Monitoring System on Solar Panel Orientation Control Using Visual Basic". In: *Journal of Applied Engineering and Technological Science (JAETS)* 2.2 (2021), pp. 112–124. DOI: 10.37385/jaets.v2i2.249.

[17] Yu-tao Hu and Yi-yong Yao. "Recognition and Location of Solar Panels Based on Machine Vision". In: *2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)* (2017). DOI: 10.1109/ACIRS.2017.7986055.

[18] *Infratec Infrared Camera Module*. Accessed: 2023-July. 2023. URL: https://www.infratec.eu/thermography/infrared-camera/pir-uc-605/.

[19] Claes Isaksson and Margaret H. Dunham. "A Comparative Study of Outlier Detection Algorithms". In: *Machine Learning and Data Mining in Pattern Recognition*. 2009. DOI: 10.1007/978-3-642-03070-3_33.

[20] Faizan Jawaid and Khurum NazirJunejo. "Predicting daily mean solar power using machine learning regression techniques". In: *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*. 2016, pp. 355–360. DOI: 10.1109/INTECH.2016.7845051.

[21] Jordan Johnston, Kaiman Zeng, and Nansong Wu. "An Evaluation and Embedded Hardware Implementation of YOLO for Real-Time Wildfire Detection". In: *2022 IEEE World AI IoT Congress (AIIoT)*. 2022, pp. 138–144. DOI: 10.1109/AIIoT54504.2022.9817206.

[22] Daniel A. Keim, Gennady Andrienko, and Jean-Daniel Fekete. *Visual Analytics: Definition, Process, and Challenges*. Springer, 2008. DOI: 10.1007/978-3-540-70956-5_7.

[23] Nishan Khatri et al. "Perspective Transformation Layerar". In: *International Conference on Computational Science Computational Intelligence (CSCI'22)* (2022). URL: https://arxiv.org/abs/2201.05706.

[24] Aishwarya Kumar et al. "An IOT based smart inverter". In: *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. 2016, pp. 1976–1980. DOI: 10.1109/RTEICT.2016.7808182.

[25] Soomin Lee et al. "Design of Solar Panels Efficiency Monitoring System". In: *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*. 2020, pp. 1–4. DOI: 10.1109/ICCE-Asia49877.2020.9276867.

[26] Jiaqi Li et al. "Deep Learning based Defect Detection Algorithm for Solar Panels". In: Aug. 2023, pp. 438–443. DOI: 10.1109/WRCSARA60131.2023.10261859.

[27] Qian Luo and Min Xie. "Temperature and Humidity Detection System of Communication System Based on Raspberry Pi". In: *2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*. 2018, pp. 214–216. DOI: 10.1109/ICITBS.2018.00062.

[28] Lutz Thomas Weidner. *Bauthermografie Luftdichtheitsprüfung*. Accessed: 2023-October. 2004-2023. URL: https://www.bauthermografie-luftdichtheit.de/.

[29] Yue Ma et al. "FPGA Implementation of HoG-based Space Target Distance Measurement". In: *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*. 2021, pp. 585–589. DOI: 10.1109/IPEC51340.2021.9421215.

[30] J. B. MacQueen. *Some Methods for Classification and Analysis of MultiVariate Observations*. Vol. 1. University of California Press, 1967, pp. 281–297. URL: https://www.bibsonomy.org/bibtex/25dcdb8cd9fba78e0e791af619d61d66d/enitsirhc.

[31] *Mattro ROVO*. Accessed: 2023-August. 2023. URL: https://www.hawe.com/products/robot-platform/rovo-ai/.

[32] Raphael S. Medeiros et al. "Photovoltaic panels identification in aerial images". In: *XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBrT*. 2021.

[33] Matthew Millendorf, Edward Obropta, and Nikhil Vadhavkar. "Infrared Solar Module Dataset for Anomaly Detection". In: *Proceedings of the International Conference on Learning Representations* (2020). URL: https://github.com/RaptorMaps/InfraredSolarModules.

[34] Kevin Moser. *CRDIR: Cosmic Ray Damaged Image Repair for DSLR Cameras*. 2017.

[35] K. Niazi et al. "Binary Classification of Defective Solar PV Modules Using Thermography". In: *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC) (A Joint Conference of 45th IEEE PVSC, 28th PVSEC 34th EU PVSEC)*. 2018, pp. 0753–0757. DOI: 10.1109/PVSC.2018.8548138.

[36] J. Nikhila. "Web based Environmental Monitoring System using Raspberry Pi". In: *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*. 2017, pp. 1074–1080. DOI: 10.1109/CTCEEC.2017.8454964.

[37] Mayamiko Nkoloma, Marco Zennaro, and Antoine Bagula. *SM2: Solar monitoring system in Malawi*. 2011, pp. 1–6.

[38] Open Source Computer Vision. *Watershed algorithm tutorial*. Accessed: 2023-October. 2023. URL: https://docs.opencv.org/3.4/d3/db4/tutorial_py_watershed.html.

[39] *OpenCV*. Accessed: 2023-July. 2023. URL: https://opencv.org/.

[40] Roberto Pierdicca et al. "Deep Convolutional Neural Network for automatic detection of damaged photovoltaic cells". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2 (May 2018), pp. 893–900. DOI: 10.5194/isprs-archives-XLII-2-893-2018.

[41] Plotly Technologies Inc. *Plotly*. Accessed: 2023-October. 2023. URL: https://plotly.com/.

[42] Joanna Ponitka and Stefan Boettner. "Challenges of future energy landscapes in Germany — a nature conservation perspective". In: *Environmental Sciences Europe* 10.1 (2020), p. 1. DOI: 10.1186/s13705-020-00250-9.

[43] Prateek Chhikara. *Understanding Morphological Image Processing and Its Operations*. Accessed: 2023-October. 2022. URL: https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756.

[44] *Qt Software Free Trials*. Accessed: 2023-July. 2023. URL: https://www.qt.io/download.

[45] Muhammad Rameez Ur Rahman and Haiyong Chen. "Defects Inspection in Polycrystalline Solar Cells Electroluminescence Images Using Deep Learning". In: *IEEE Access* 8 (2020), pp. 40547–40558. DOI: 10.1109/ACCESS.2020.2976843.

[46] Raqueeb Shaikh. *OpenCV Perspective Transformation*. Accessed: 2023-October. 2020. URL: https://medium.com/analytics-vidhya/opencv-perspective-transformation-9edffefb2143.

[47] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[48] Gisele A. dos Reis Benatto and Riedel. "Development of outdoor luminescence imaging for drone-based PV array inspection". In: *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*. 2017, pp. 2682–2687. DOI: 10.1109/PVSC.2017.8366602.

[49] Tobias S. Schmidt and Nicolas Schmid. "Policy goals, partisanship and paradigmatic change in energy policy – analysing parliamentary discourse in Germany over 30 years". In: *Climate Policy* 19.4 (2019), pp. 771–786. DOI: 10.1080/14693062.2019.1594667.

[50] Feng Shu, Hanhua Lu, and Yin Ding. "Novel Modbus Adaptation Method for IoT Gateway". In: *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. 2019, pp. 632–637. DOI: 10.1109/ITNEC.2019.8729209.

[51] Waleed T Al-Sit and Rashed Al-Hamadin. "Real estate market data analysis and prediction based on minor advertisements data and locations' geo-codes". In: *International Journal* 9.3 (2020). DOI: 10.30534/ijatcse/2020/235932020.

[52] Tianyi Sun, Huishuang Xing, and Shengxian Cao. "A novel detection method for hot spots of photovoltaic (PV) panels using improved anchors and prediction heads of YOLOv5 network". In: *Energy Reports* 8 (2022), pp. 1219–1229. DOI: 10.1016/j.egyr.2022.08.130.

[53] Abdelilah Et-taleby, Mohammed Boussetta, and Mohamed Benslimane. "Faults Detection for Photovoltaic Field Based on k-means, Elbow, and Average Silhouette Techniques Through the Segmentation of a Thermal Image". In: *International Journal of Photoenergy* 2020 (2020). DOI: 10.1155/2020/6617597.

[54] Zhenjun Tang et al. "Robust image hashing based on color vector angle and Canny operator". In: *AEU - International Journal of Electronics and Communications* 70.6 (2016), pp. 833–841. ISSN: 1434-8411. DOI: 10.1016/j.aeue.2016.03.010.

[55] Christian Tominski and Heidrun Schumann. *Interactive Visual Data Analysis*. 2020. DOI: 10.1201/9781315152707.

[56] Nalika N. B. Ulapane and Sunil G. Abeyratne. "Gaussian process for learning solar panel maximum power point characteristics as functions of environmental conditions". In: *2014 9th IEEE Conference on Industrial Electronics and Applications*. 2014, pp. 1756–1761. DOI: 10.1109/ICIEA.2014.6931452.

[57] *UR10e*. Accessed: 2023-August. 2023. URL: https://www.universal-robots.com/products/ur10-robot/.

[58] *Visual Studio Code*. Accessed: 2023-July. 2023. URL: https://code.visualstudio.com/.

[59] Vladimir Voicu, Dorin Petreus, and Radu Etz. "Data Acquisition System for Solar Panels". In: *2019 42nd International Spring Seminar on Electronics Technology (ISSE)*. 2019, pp. 1–6. DOI: 10.1109/ISSE.2019.8810289.

[60] Aili Wang and Xusheng Liu. "Vehicle License Plate Location Based on Improved Roberts Operator and Mathematical Morphology". In: *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*. 2012, pp. 995–998. DOI: 10.1109/IMCCC.2012.237.

[61] Dr. Harry Wirth. *Recent Facts about Photovoltaic in Germany*. Fraunhofer ISE. 2023. URL: https://www.ise.fraunhofer.de/en/publications/studies/recent-facts-about-pv-in-germany.html.

[62] T. Yang et al. "Otsu thresholding segmentation method based on two boundaries and its fast algorithm". In: *Application Research of Computers* 33.12 (2016), pp. 3872–3875. DOI: 10.1155/2017/1735176.

[63] Abdulmalik Shehu Yaro, Filip Maly, and Pavel Prazak. "Outlier Detection in Time-Series Receive Signal Strength Observation Using Z-Score Method with Scale Estimator for Indoor Localization". In: *Applied Sciences* 13 (2023). DOI: 10.3390/app13063900.

[64] Junshu Zhang et al. "A Perspective Transformation Method Based on Computer Vision". In: *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. 2020. DOI: 10.1109/ICAICA50127.2020.9182641.

[65] Ye Zhao et al. "Outlier detection rules for fault detection in solar photovoltaic arrays". In: Mar. 2013, pp. 2913–2920. ISBN: 978-1-4673-4355-8. DOI: 10.1109/APEC.2013.6520712.

10 Declaration on the independent drafting of a written work

Hiermit erkläre ich, ... (vollständiger Name in Druckbuchstaben), Matrikel-Nr. ...., dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Rostock

(Abgabedatum) (Vollständige Unterschrift)

11 Declaration of Honor

Hiermit versichere ich, ... (vollständiger Name in Druckbuchstaben), Matrikel-Nr. ...., Studiengang (B.A., M.A., LAG etc.) ...., Studienfächer (Erst- / Zweitfach etc.) ...., dass ich mich als Studierender der Universität Rostock den "Regeln zur Sicherung guter wissenschaftlicher Praxis und zur Vermeidung wissenschaftlichen Fehlverhaltens an der Universität Rostock" verpflichtet fühle. Zu diesen Regeln gehört auch die Vermeidung von Plagiaten als einer schwerwiegenden Form geistigen Diebstahls.

Unter einem Plagiat versteht man die ungekennzeichnete oder nicht angemessen gekennzeichnete Übernahme von fremdem geistigem Eigentum unabhängig von dessen Herkunft (d.h. auch aus dem Internet) in eigene Arbeiten. Eine unbefugte Verwertung unter Anmaßung der Autorschaft liegt vor, wenn Fakten, Argumente oder spezifische Formulierungen ohne Quellenangabe übernommen, paraphrasiert oder übersetzt werden.

Mir ist bekannt, dass eine Prüfungsleistung, die nachweislich ein Plagiat darstellt, mit "nicht ausreichend" (5,0) bewertet wird. Ich bin mir dessen bewusst, dass die Aufdeckung eines Plagiatsfalles dem Prüfungsamt gemeldet wird und mit dem Ausschluss von der Erbringung weiterer Prüfungsleistungen geahndet werden kann.

Rostock

(Abgabedatum) (Vollständige Unterschrift)

Appendices

Pixel restriction MATLAB code

```
clear
%coordinates of the camera
Xa = 5;
Ya = -0.309;
Za = 2.172;
%view angle of the camera, where e is horisontal and v is vertical
e = 80;
v = 60;
%distance from camera to the corners of the botton edge of the panel
ha = sqrt((Ya)^2 + (Za-0.3)^2);
%panel has corners and points B C D E
%plotting AB
Xb = Xa + ha*sind(e/2);
Yb = 0;
Zb = 0.3;
plot3([Xa Xb], [Ya Yb], [Za Zb])
hold on
%plotting AC
Xc = Xa - ha*sind(e/2);
Yc = 0;
Zc = 0.3;
plot3([Xa Xc], [Ya Yc], [Za Zc])
Yd = 3.5*cosd(20);
Zd = 0.342020*Yd/0.939692+0.3;
%distance from camera to B point
hb = sqrt((Ya-Yd)^2 + (Za-Zd)^2);
Xd = Xa + hb*sind(e/2);
%plotting AD
plot3([Xa Xd], [Ya Yd], [Za Zd])
Xe = Xa - hb*sind(e/2);
Ye = 3.5*cosd(20);
%that is the equation of the surface of our panel
Ze = 0.342020*Ye/0.939692+0.3;
%plotting AE
plot3([Xa Xe], [Ya Ye], [Za Ze])
%as I have 640x480 pixels we fullfill the arrays X and Y with edges of each
%pixel
for i = 1:481
    for j = 1:(640 + 1)
            X(i,j) = Xc+(Xb-Xc+2*(Xc-Xe)*(i-1)/480)/640*(j-1)-(Xc-Xe)/480*(i-1);
```

```
            Y(i,j) = (Yd-Yb)/(480)*(i-1)+0*j;
    end
end
%calculating perpendicular
Yh = (Ya - 0.10919 + 0.36397 * Za)/1.13247;
Zh = 0.342020 * Yh/0.939692+0.3;
H = sqrt((Yh - Ya)^2 + (Zh - Za)^2);
%all surface of the panel devided by quantity of pixels (average pixel
%size)
S1 = 0.5*(-Xc+Xb-Xe+Xd)*3.5/(480*640);
%calculating center points of every pixel
for i = 1:480
    for j = 1:(640)
            Xcent(i,j) = ((Y(i+1,j)-Y(i,j))*(X(i+1,j+1)-X(i,j))*(X(i,j+1)-X(i+1,
            j))+X(i,j)*(Y(i+1,j+1)-Y(i,j))*(X(i,j+1)-X(i+1,j))-X(i+1,j)*( ...
            Y(i,j+1)-Y(i+1,j))*(X(i+1,j+1)-X(i,j)))/((Y(i+1,j+1)-Y(i,j))*( ...
            X(i,j+1)-X(i+1,j))-(Y(i,j+1)-Y(i+1,j))*(X(i+1,j+1)-X(i,j)));
            Ycent(i,j) = (Xcent(i,j) - X(i,j))*(Y(i + 1, j + 1) - Y(i,j))/(X(i +
            j + 1) - X(i,j)) + Y(i,j);
            %calculating angle from camera to every center point of pixel
            R(i,j) = atan(H/sqrt((Xcent(i,j)-Xa)^2 + (Ycent(i,j)-Yh)^2 + ...
            (0.342020*Ycent(i,j)/0.939692+0.3-Zh)^2));
            %coverting it to degrees
            alpha(i,j) = rad2deg(R(i,j));
            %to find square of the pixel we need to find coefficient k in
            %(y=kx+b) of its diagonals
            k1(i,j) = (Y(i+1,j+1)-Y(i,j))/(X(i+1,j+1)-X(i,j));
            k2(i,j) = (Y(i,j+1)-Y(i+1,j))/(X(i,j+1)-X(i+1,j));
            %then agnle between them
            fi(i,j) = atan((abs((k1(i,j)-k2(i,j))/(1+k1(i,j)*k2(i,j)))));
            %and finally square
            S(i,j) = 0.5 * sqrt((X(i,j)-X(i+1,j+1))^2 + (Y(i,j)-Y(i+1,j+1))^2 +
            (0.342020*Y(i,j)/0.939692 - 0.342020*Y(i+1,j+1)/ ...
            0.939692)^2) * sqrt((X(i+1,j)-X(i,j+1))^2 + (Y(i+1,j)-Y(i,j+1))^2 +
            (0.342020*Y(i+1,j)/0.939692 - 0.342020*Y(i,j+1)/ ...
            0.939692)^2) * sin(fi(i,j));
            %here we are checking if angles is in the range of 15-70 and
            %the size of the pixel should be at least 80% of S1
              if (S(i,j)>0.8*S1)
                Xm(i,j) = X(i,j);
                Ym(i,j) = Y(i,j);
                 if (Ym(i,j) ~= 0)
                Zm(i,j) = 0.342020*Ym(i,j)/0.939692+0.3;
```

```
                end
            end
    end
end



surf(Zm);
shading interp;
axis equal;
```