



# **University of Catania**

Department of Mathematics and Computer Science

Doctorate Research in Computer Science

## **Artificial Intelligence Applied to Solar Energy**

**Marco Grisanti**

Tutor:

Prof. Sebastiano Battiato

Co-Tutor:

Prof. Alessandro Ortis

Academic Year: 2022/2023



## Abstract

This thesis collects all the work done by the PhD candidate at the University of Catania and Enel Green Power, whose collaboration has allowed the possibility of carrying out the doctorate research. Artificial intelligence is a field of computer science that deals with developing algorithms and systems capable of performing tasks that emulates human capabilities trying to achieve human being level of performance. Using complex algorithms and data, artificial intelligence enables machines to perform advanced tasks that, in the past, it would not have been possible to carry out automatically. Such technology is profoundly influencing our lives, entering fields such as medicine, industrial automation, transportation, and even artistic creativity. In this thesis the PhD Candidate will mainly deal with artificial intelligence applied to the world of photovoltaic panels with the aim of improving their maintainability. Initially, the automatic classifications of various types of defects present in photovoltaic panels identifiable with the test based on electroluminescence will be investigated; in this study a new dataset and a preliminary benchmark to make an automatic and accurate classification of defects in solar cells will be proposed. This dataset will include five classes of defects and the pre-trained ResNext50 network will reach 0.07 Hamming Distance. Subsequently, considering the importance of the operating temperature of the photovoltaic modules in the study of the degradation of the photovoltaic panels and that the operative temperature cannot always be recovered due to the absence of sensors, conventional physics models with machine learning models will be compared for the estimation of the temperature of monofacial and bifacial photovoltaic modules installed in two different locations of Italy. In that comparison, machine learning models will turn out to be slightly better than conventional physics models. Furthermore, a benchmark evaluation about compression techniques of stereoscopic images will be performed on large and standardized datasets including 60 stereopairs that differ by resolution and acquisition technique in which an Human Visual System assessment experiment will be implemented which will involve more than a hundred people in order to verify the perceived quality of the decoded images.

## Acknowledgements

# Contents

<b>Abstract</b>	<b>3</b>
<b>Acknowledgements</b>	<b>4</b>
<b>1 Introduction</b>	<b>8</b>
1.1 Dissertation Structure . . . . .	9
1.2 Contributions . . . . .	9
<b>2 Foundations of Artificial Intelligence</b>	<b>11</b>
2.1 Machine Learning . . . . .	12
2.1.1 Principles of Regression . . . . .	12
2.1.2 Principles of Artificial Neural Networks . . . . .	14
2.2 Deep Learning . . . . .	15
2.3 Computer Vision . . . . .	17
2.3.1 Artificial Stereoscopic Vision . . . . .	18
2.4 Technologies for AI Programming . . . . .	19
2.4.1 Python . . . . .	19
<b>3 Artificial Intelligence Applied to Photovoltaics Panels</b>	<b>27</b>
3.1 Introduction . . . . .	27
3.2 Photovoltaic Panels . . . . .	28
3.3 Automatic Detection and Classification of Defects in Photovoltaic Cells . . . . .	29
3.3.1 Introduction . . . . .	29
3.3.2 Luminescence and Electroluminescence . . . . .	30
3.3.3 Electroluminescence Applied to Photovoltaic Panels . . . . .	31
3.3.4 Artificial Intelligence Techniques for Electroluminescence Test Automation	32
3.3.5 State of the Art . . . . .	33
3.3.6 Dataset . . . . .	34
3.3.7 Proposed Methods . . . . .	35
3.3.8 Analysis of Deep Neural Networks . . . . .	37
3.3.9 Results . . . . .	41
3.3.10 Conclusions . . . . .	46
3.4 Machine Learning Against Physics for Photovoltaic Thermal Models . . . . .	46
3.4.1 Introduction . . . . .	47

3.4.2	State of the Art . . . . .	48
3.4.3	Physical Models . . . . .	48
3.4.4	Machine Learning Models . . . . .	50
3.4.5	Dataset . . . . .	51
3.4.6	Proposed Methods . . . . .	53
3.4.7	Results . . . . .	55
3.4.8	Conclusions . . . . .	56
3.5	Papers . . . . .	57
<b>4</b>	<b>A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images</b>	<b>58</b>
4.1	Introduction . . . . .	58
4.2	Motivations . . . . .	62
4.3	Evaluated Pipeline . . . . .	64
4.3.1	Encoding Pipeline: Asymmetric Compression . . . . .	64
4.3.2	NCC-Based Decoding Approach . . . . .	64
4.3.3	Geometry-Based Decoding Approach . . . . .	65
4.3.4	Image Enhancing . . . . .	74
4.3.5	Computational Complexity . . . . .	75
4.3.6	NCC-Based Complexity . . . . .	75
4.3.7	Geometry-Based Complexity . . . . .	76
4.4	Experiments . . . . .	77
4.4.1	Comparative Evaluation . . . . .	84
4.4.2	Subjective Assessment . . . . .	85
4.5	Conclusions . . . . .	86
4.6	Papers . . . . .	87
<b>5</b>	<b>Final Discussion, Remarks and Future Works</b>	<b>88</b>
5.1	Future Directions . . . . .	88
	<b>Appendix</b>	<b>90</b>
<b>A</b>	<b>A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images: Complete Results</b>	<b>90</b>

<b>B Automation of Business Activities</b>	<b>97</b>
B.1 Introduction . . . . .	97
B.2 Techniques and Technologies Used . . . . .	97
B.2.1 Automating the use of REST APIs . . . . .	97
B.2.2 Automating the use of Browser . . . . .	99
B.3 Conclusions . . . . .	100

# 1 Introduction

Enel Green Power (EGP) is a major multinational company in the renewable energy sector, part of the Enel Group, one of the main global operators in the electricity sector. Founded in 2008, EGP is dedicated to the development, production and distribution of energy from renewable sources, significantly contributing to the transition towards a more sustainable and low-carbon energy system. Its global presence extends to numerous countries, spread over several continents, where it develops and manages energy plants that exploit natural resources to generate clean electricity. EGP's international reach allows it to contribute significantly to the reduction of greenhouse gas emissions and the mitigation of climate change. The company not only focuses on the production of renewable energy, but also on constant research and innovation in the sector. Through the adoption of cutting-edge technologies and the optimization of resources, EGP aims to maximize the efficiency and sustainability of its operations. With a strong dedication to environmental sustainability, EGP plays a key role in accelerating the transition to a cleaner and more environmentally responsible energy future. Its global presence, wide range of renewable technologies and commitment to innovation make it a key player in the transformation of the energy sector towards greater sustainability. The company operates in a wide range of renewable energy technologies, with solar energy prominent among them. It is a form of energy resulting from electromagnetic radiation emitted by the sun, which, known as sunlight, travels through space and reaches the Earth in a wide range of wavelengths, including visible, ultraviolet and infrared. Solar energy technologies mainly exploit two main approaches: the photovoltaic approach and the thermal approach. Solar energy offers many advantages, including being a renewable source or reducing greenhouse gas emissions. It stands as a crucial solution to address the energy and environmental challenges of our time. Photovoltaic and thermal technologies continue to evolve, making solar energy increasingly accessible and efficient. It could become a significant driving force in the global energy landscape, providing a more sustainable and ecologically balanced future.

This dissertation collects all the work done by the PhD Candidate for EGP (EGP) and for the University of Catania (UniCT). This Computer Science PhD, directed and supported in every aspect both by the Image Processing Laboratory (IPLab) research group of the UniCT Department of Mathematics and Computer Science and by the Innovation Lab research group of EGP, has focused its attention more on the study of techniques and technologies related to solar energy, in particular the implementation of Artificial Intelligence algorithms for the efficient and effective maintenance of photovoltaic panels. Furthermore, a contribution was carried out relating to a work which concerned an in-depth benchmark on adaptive compression techniques of stereoscopic images.



## 1.1 Dissertation Structure

This dissertation, entitled [...], is structured as follows. Section 2 introduces the artificial intelligence topics that will be used for the various works described in the following chapters, then Machine Learning is discussed (with small insights on regression and artificial neural networks), Deep Learning, Computer Vision (with specific reference to stereoscopic vision) and techniques and technologies used for the implementation of artificial intelligence, therefore speaking of Python and its best-known libraries for this purpose. Section 3 represents the heart of this dissertation, dealing with how it is possible to apply artificial intelligence to the world of photovoltaic panels in the first instance with the automatic detection of defects present on photovoltaic cells [1] (Section 3.3) and in the second instance with the estimation of the temperature of the photovoltaic modules [2] (Section 3.4). The Section 4 is entirely dedicated to the work done on an in-depth benchmark relating to the compression techniques of stereoscopic images [3].

## 1.2 Contributions

The main contributions of this thesis are the following:

- The releasing of a new dataset and a preliminary benchmark to make an automatic and accurate classification of defects in solar cells. The dataset includes five classes of defects and the pre-trained ResNext50 network reaches 0.07 Hamming Distance;
- A comparison of conventional models with machine learning models for the estimation of the temperature of monofacial and bifacial photovoltaic modules installed in two different locations;
- A benchmark evaluation about compression techniques of stereoscopic images on large and standardized datasets including 60 stereopairs that differ by resolution and acquisition technique,
- Creation of advanced scripts for the automation of business processes with the aim of improving the productivity of the employees involved in the analysis of the photovoltaic panels of the EGP headquarters in Catania.

In the following the list of publications of related papers is reported just taking care of group them with respect to the main involved work:

- Artificial Intelligence Applied to Photovoltaics Panels
  - Marco Grisanti, Maria Ausilia Napoli Spatafora, Alessandro Ortis and Sebastiano Battiato, "E-ELPV: Extended ELPV Dataset for Accurate Solar Cells Defect Classification", IntelliSys2023 - September 2023.
  - Marco Grisanti, Gaetano Mannino, Giuseppe Marco Tina, Alessandro Ortis, Mario Cacciato, Sebastiano Battiato, Fabrizio Bizzarri and Andrea Canino, "Thermal Models of Monofacial and Bifacial PV Modules: Machine Learning and Physical Estimation Models Comparison", PVSC50 - June 2023.
- Stereoscopic Vision
  - Alessandro Ortis, Marco Grisanti, Francesco Rundo and Sebastiano Battiato, "A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images", Journal of Imaging - July 2021.

## 2 Foundations of Artificial Intelligence

Artificial intelligence (AI) is a branch of computer science that deals with the creation of algorithms and systems capable of performing tasks that require the use of human intelligence, such as perception, understanding, reasoning and learning. AI is based on the idea of creating machines capable of imitating human intelligence and surpassing human capabilities in certain areas.

The origins of AI date back to 1950, when mathematician and computer scientist Alan Turing proposed a test to evaluate whether a machine was capable of displaying intelligent behavior indistinguishable from that of a human being. One of the earliest examples of AI was the chess game program written in 1951 by mathematician and computer scientist Christopher Strachey. In the 1950s and 1960s, AI research focused primarily on symbolic programming, that is, the creation of algorithms that used logic to process and manipulate information. In 1956, a group of researchers held the Dartmouth Summer Research Project on Artificial Intelligence conference at Dartmouth College in the United States, which is considered the official starting point of AI research. In the 1970s, AI suffered a crisis of confidence, due to the lack of significant progress and the complexity of the problems it faced. However, in the 1980s and 1990s, AI experienced a new impetus, thanks to advances in computing power, the availability of large amounts of data and the creation of new machine learning algorithms. In the 2000s thanks to the availability of cheap hardware and the advent of Cloud Computing, AI became increasingly widespread, powerful and accessible. Today, AI is present in many aspects of daily life, from internet research with modern search engines to autonomous vehicle driving, from medical diagnosis to financial analysis.

In addition to being used in the civilian world, AI is also used in many applications in the military world such as for aerial surveillance and drone guidance. One of the sectors where AI is widely used is the robotics sector. Just to give an example, it might be enough to think of a robot equipped with AI that can thus be used to perform repetitive and dangerous tasks for humans, such as cleaning nuclear sites or collecting data in risky environments. AI can also be used to analyze large amounts of data in order to identify any anomalies, for example to prevent financial fraud or to identify potential cybersecurity threats. AI continues to represent one of the most important frontiers of scientific and technological research, with the potential to radically transform many areas of human life. However, AI also raises several issues, including data security, employment impact and machine liability. In fact, AI could replace some human jobs creating unemployment in some sectors as well as generating unpredictable or even harmful decisions if its algorithms are not properly developed and controlled.

AI is based on various approaches, including Machine Learning, Deep Learning and Computer Vision, the foundations of which, of interest for the thesis topics, will be mentioned in this section.

## 2.1 Machine Learning

Machine learning is a branch of artificial intelligence that deals with creating algorithms capable of learning from the data provided, improving their performance over time and making autonomous decisions in order to solve a specific task. There are several approaches used in machine learning to solve complex problems of classification, regression, grouping and more. Some of the most used approaches in Machine Learning are briefly described below:

- **Supervised Learning:** in this approach, the algorithm is trained on a set of matching input and output data. The goal is to learn a function that can be used to predict output from input. For example, an algorithm based on this approach can be used to classify objects in images or to predict the price of a house based on its characteristics.
- **Unsupervised Learning:** in this approach, the algorithm is trained on a set of input data without knowing the corresponding output. The goal is to identify patterns and hidden relationships in the data. For example, an algorithm based on this approach can be used to identify groups of users with similar interests based on their purchasing behavior.
- **Reinforcement Learning:** in this approach, the algorithm learns by interacting with an environment, receiving rewards or punishments based on the actions it performs. The goal is to maximize the reward obtained by the agent. For example, an algorithm based on this approach can be used to train a robot to perform a specific task, such as walking or picking up objects.

Machine learning has numerous practical applications, such as image classification, product recommendation, medical diagnosis, etc. For example, a machine learning algorithm can be used to recognize people's faces in a photo, or to suggest products similar to those the user has already purchased. In this dissertation two types of machine learning will be used, both in the context of supervised learning: the regression technique and the computational model of artificial neural networks.

### 2.1.1 Principles of Regression

Regression is a technique used in machine learning (and not only) to model and analyze the relationship between one or more independent variables (inputs) and a continuous dependent variable (outputs). The goal of regression is to create a function that can predict or estimate the value of the dependent variable based on the values of the independent variables. In regression, it is assumed that there is a linear or non-linear relationship between the independent variables and the

dependent variable. The regression model attempts to approximate this relationship through the use of a training dataset that contains examples of the independent variables and the corresponding known values of the dependent variable.

Linear regression is one of the most common types of regression and seeks to model the relationship between the independent and dependent variables through a linear function. The general form of a linear regression is given by:

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n$$

where:

- $y$  is the dependent variable;
- $x_1, x_2, \dots, x_n$  are the  $n$  independent variables;
- $a_1, a_2, \dots, a_n$  are the coefficients that measure the effect of each independent variable;
- $a_0$  is called "intercept". It is the value assumed by the dependent variable when the regression line passes through the ordinate axis.

Besides linear regression there are other forms of regression that can be used based on the nature of the data and the relationship between the variables. Among the most common it is possible to single out:

- Polynomial Regression: This form of regression allows you to model nonlinear relationships between variables using polynomial functions of different degrees.
- Logistic Regression: This type of regression is used for binary classification problems, where the dependent variable is categorical. Logistic regression estimates the probability of belonging to one of the two classes based on the values of the independent variables.
- Ridge or Lasso Regression: This form of regression is a variant of linear regression that uses regularization to handle multicollinearity problems that arise when the independent variables are highly correlated with each other in order to avoid overfitting. Ridge regression uses an L2 regularization, while Lasso regression uses an L1 regularization.
- Elastic Net Regression: This type of regression is a combination of Ridge and Lasso regression, using both L1 and L2 regularization. It is useful when there are many variables and some of them are highly correlated.

- **Logarithmic Regression:** This form of regression is used when the relationship between variables can best be modeled using a logarithmic scale.

Regression is widely used in different areas, such as economics, finance and engineering to make data-driven predictions, analyzes and decisions.

### 2.1.2 Principles of Artificial Neural Networks

The artificial neuron, also known as a perceptron, is a key component of machine learning and deep learning systems. The perceptron was introduced in [4], and it serves as a simple model for understanding the basic principles of neural networks. The artificial neuron takes multiple input values, each multiplied by a weight, and then sums up these weighted inputs. The sum is then passed through an activation function, which determines the output of the neuron. Mathematically, the output  $y$  of a perceptron can be represented as:

$$y = f(w_1x_1 + w_2x_2 + \dots + w_nx_n + b)$$

where:

- $w_1, w_2, \dots, w_n$  are the weights associated with the input values  $x_1, x_2, \dots, x_n$ ;
- $b$  is the bias;
- $f$  is the activation function.

An artificial neural network is a computational model inspired by the functioning of the human brain. It is a highly parallel computing structure made up of interconnected artificial neurons, which work together to process data and recognize complex patterns within the data, allowing the neural network to learn from examples and make predictions about new data. The structure of a neural network is made up of different layers of neurons, each of which has a specific function:

- **Input layer:** This layer receives the input data and represents the first stage of processing in the neural network.
- **Hidden layers:** Between the input layer and the output layer there can be one or more hidden layers. They are fundamental for data processing and learning as they constitute the "heart" of the neural network and are responsible for the ability to learn and represent complex models.
- **Output layer:** This layer produces the predictions or final results of the neural network, based on the activations of the last hidden layer.

The learning process in a neural network occurs through an optimization algorithm that updates the weights associated with the connections between neurons to reduce the error between the network's predictions and the correct values known in the training data. Neural networks can be used for various artificial intelligence applications, such as image recognition, text classification, speech recognition, robot control and many other tasks that require complex data processing and the ability to learn from data itself. Note that if the number of hidden layers of a neural network is greater than one, then it is referred to as a deep artificial neural network, which falls under the scope of Deep Learning.

## 2.2 Deep Learning

Deep learning is an advanced machine learning discipline that relies on the structure and training of deep and complex artificial neural networks. This innovative branch of artificial intelligence was inspired by the biology of the human brain and aims to replicate the information processing capabilities of the human brain using a vast network of interconnected artificial neurons. Unlike traditional neural networks, which often contain only one hidden layer, deep neural networks have a hierarchical structure and are composed of many hidden layers. In doing so, an important depth is obtained that allows neural networks to represent and capture complex and abstract characteristics present in the data, making deep learning suitable for highly complex problems, such as image recognition, speech recognition, machine translation and much more.

The heart of deep learning is the learning algorithm called "Back Propagation", which allows neural networks to update the weights of connections between neurons in the training phase, during which input data is presented to the network so as to calculate the error between the model predictions and the known correct values. Then, the Back Propagation algorithm propagates the error backwards through the network, adjusting connection weights to reduce the error and improve future predictions.

One of the hallmarks of deep learning is the ability to learn directly from features extracted from data, without requiring manual engineering of those features. The process of extracting features from data just described is called "Feature Learning" and is one of the reasons that deep learning is so powerful and versatile. However, the success of deep learning is also largely due to the availability of large amounts of data and the growing computational power of graphics processing units (GPUs) that allow training deep neural networks in reasonable time.

Deep learning can be used to detect and recognize text in images or documents. This application is known as Optical Character Recognition (OCR), an area where deep learning has made significant advancements in recent years. Already by the end of the last century, an innovative methodology for

OCR based on convolutional neural networks was introduced [5], which had a significant impact on deep learning. In the article, the authors described how they utilized convolutional neural networks to tackle the problem of recognizing characters and words in documents. Specifically, they demonstrated the effectiveness of automatically extracting relevant features from character images, making the recognition process more robust and accurate. The approach proposed in this article laid the foundation for many subsequent developments in the field of deep learning-based Optical Character Recognition and showcased the potential of convolutional neural networks for pattern recognition in images.

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was an annual competition in large-scale visual recognition that had a significant impact on the development of the deep learning and image recognition field. In 2012, a crucial turning point occurred in deep learning when a model called AlexNet [6] won the ILSVRC competition, demonstrating a significant improvement in performance compared to previous approaches. AlexNet was a well-known deep learning model recognized for its depth, innovative Rectified Linear Unit (ReLU) activation function, Dropout, and Data Augmentation. The use of GPU units accelerated training, contributing to its success. AlexNet marked a milestone in the field of deep learning and computer vision, significantly influencing the subsequent development of neural networks. Figure 1 shows an overview of the AlexNet architecture.

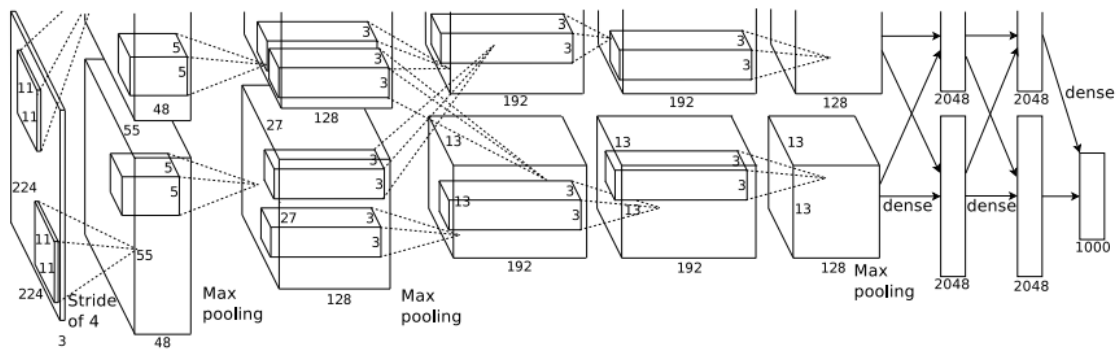


Figure 1: Architecture of AlexNet.

Deep learning represents a significant step in the advancement of AI, enabling machines to learn from complex data and perform AI tasks with a level of accuracy and capabilities comparable to humans. Deep learning is at the heart of modern AI applications and continues to be an active area of research with the potential to transform numerous sectors of industry and science.



## 2.3 Computer Vision

Computer vision is a field of artificial intelligence that aims to enable computers to understand, interpret and analyze real-world visual content, similar to how the human brain does it. This discipline focuses on the design of algorithms and computational methods to extract meaningful and understandable information from images or image sequences. Computer vision has become one of the busiest areas of research and development due to its wide range of practical applications and technological advances in image collection and processing. Applications of computer vision include object detection and tracking, facial recognition, autonomous vehicle navigation, surveillance, diagnostic medicine and much more.

One of the characteristics of computer vision is the ability to learn automatically, which allows to improve performance and the ability to adapt to new situations based on the experience gained from the data. In particular, deep learning has shown enormous promise in computer vision with deep artificial neural networks capable of learning complex feature hierarchies from images, surpassing previous achievements in many computer vision challenges. Computer vision systems consists of several stages as shown in Figure 2:

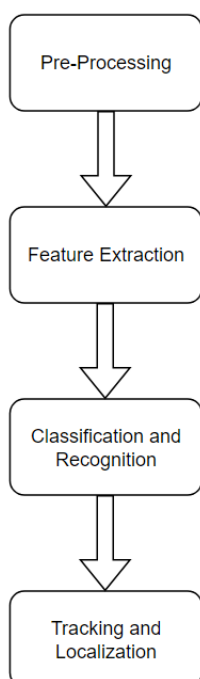


Figure 2: Phases of Computer Vision. (1) Pre Processing is the initial stage of processing in which the images are cleaned, denoised and normalized to improve the data quality. (2) Feature Extraction is the step that involves extracting significant features from images, such as edges, contours, textures and other relevant information. (3) Classification and Recognition is the phase in which the extracted characteristics are used to recognize specific objects, people or patterns within the images. Machine learning techniques, such as deep learning, are often used for this stage to make accurate predictions. (4) Tracking and Localization is the phase that concerns the ability to track and locate objects or people in time and space, allowing recognition and monitoring in video sequences.

Computer vision has made considerable progress in recent years thanks to the advancement of technologies, the large volume of data available and the growing computational power of computer systems. It's an ever-evolving field, set to transform and enrich our lives across industries ranging from security and healthcare to industrial manufacturing and augmented reality. With further developments in computer vision and machine learning, its impact on our society is likely to continue to grow significantly. This dissertation will deal with a very characteristic area of computer vision: artificial stereoscopic vision.

### 2.3.1 Artificial Stereoscopic Vision

Artificial stereoscopic vision is a computer vision technique based on the reproduction of the phenomenon of human binocular vision, which allows us to perceive the three-dimensionality of

the real world. This technique is based on capturing and processing two slightly different images of a scene from slightly different points of view, similar to what our eyes see and combining them to create an illusion of depth. To obtain artificial stereoscopic vision, it is necessary to use special devices, such as stereo cameras, which capture images from the perspective of two distinct points of view, simulating the effect of our eyes placed at a distance from each other. These stereoscopic images are then processed by algorithms to identify corresponding points in the two images, so as to calculate the disparity between them, i.e. the apparent position difference of the points between the two images. Using the calculated disparity, stereo vision systems can create a "disparity map" that represents the depth of the scene at each point. This map can then be used to generate a new stereoscopic image in which objects appear to have a sense of depth.

Artificial stereoscopic vision finds applications in various fields such as robotics, games and entertainment. For example, it is used for autonomous vehicle navigation, precision control in complex environments and for creating immersive experiences in games and 3D cinema. However, for stereoscopic computer vision to work effectively, some challenges must be addressed. For example, the correct registration of the two images and the accurate identification of the corresponding points require sophisticated algorithms and high computational capabilities. Furthermore, the presentation of the stereoscopic images requires the use of special devices to allow the observers to perceive the three-dimensional effect.

With the continuous advancement of computer vision technologies and display devices, stereoscopic computer vision promises to further improve the perception and interaction capabilities of machines with the real world, contributing to a wide range of applications and industries.

## **2.4 Technologies for AI Programming**

There are many programming languages and libraries available for AI programming, each with their own specific characteristics and purposes. In this section some of the AI programming technologies used for the PhD Thesis realization will be discussed.

### **2.4.1 Python**

Python is a high-level programming language that enjoys wide popularity in the developer community due to its simplicity, code readability and flexibility. Created in 1991, Python has earned a solid reputation in the programming world for its ease of learning and the wide range of applications it can be used in. Its main feature is the readability of the code, which derives from a clear and intuitive syntax. Python is based on a simple and consistent programming style, which allows developers to express their ideas concisely and clearly.

Python is a multi-paradigm language, which means that it allows for multiple programming styles such as imperative, functional and object-oriented programming. Such versatility allows developers to use the most suitable paradigm to solve a particular problem, making Python suitable for a wide range of applications. Another distinguishing feature of Python is the wide availability of libraries and frameworks that facilitate the development of complex applications. The Python Standard Library offers a wide range of modules that cover many common needs, while PyPI (Python Package Index) hosts thousands of third-party packages developed by the Python community, which allow developers to use advanced features without having to implement them themselves. zero.

Python is an interpreted language, which means that code is executed line-by-line by an interpreter without needing to be compiled into machine language. This makes it easy to use and portable across platforms. However, compared to compiled languages, Python may be less efficient in terms of performance, a disadvantage that is often offset by higher developer productivity. Python is a versatile programming language, suitable for both beginners and experienced developers. Its syntactic clarity, vast library availability and active, collaborative community make it a powerful tool for solving a wide variety of problems and developing applications of various kinds. While it may have performance limitations compared to compiled languages, its ease of use and high developer productivity make it one of the most popular and well-loved languages among the developer community worldwide.

The following libraries have been mainly used in this dissertation: Pandas, NumPy, Matplotlib, Torch, SciKitLearn and OpenCV.

**Pandas** Pandas library is a powerful and essential tool for data analysis in Python. It offers a wide range of features for manipulating, organizing and analyzing data. Pandas main features include:

- **Powerful Data Structures:** Pandas offers two fundamental data structures, the DataFrame and the Series. The DataFrame is a two-dimensional tabular structure similar to a spreadsheet, which allows you to organize and manipulate heterogeneous data in rows and columns. Array is a one-dimensional structure similar to an array or list, but with additional functionality.
- **Data Loading and Saving:** Pandas provides methods for importing and exporting data to and from different formats such as CSV, Excel, SQL, JSON, HTML and others. This greatly simplifies the process of acquiring and preparing data for analysis.
- **Data Cleanup:** Pandas offers functions to handle missing or duplicate data, reformat columns,

perform transformations and filter data based on certain criteria. These operations are essential to obtain a consistent and analysis-ready dataset.

- **Flexible Indexing:** Pandas offers a wide range of data indexing options, allowing you to access, edit and select specific parts of your data efficiently.
- **Aggregation Operations:** The Pandas library provides powerful methods for performing aggregation operations on large datasets. These operations allow you to calculate sums, averages, standard deviations and other statistics on groups of data based on one or more columns.
- **Data Merge:** Pandas allows you to merge different DataFrames based on common keys or columns, allowing you to combine information from different sources and create new datasets.
- **Powerful Temporal Data Support:** Pandas offers advanced capabilities for manipulating temporal data, including converting, scaling, indexing and calculating specific statistics.
- **Data Visualization:** While Pandas is not a data visualization library, it offers integration with other libraries such as Matplotlib, making it easy to create graphs and visualizations to explore and communicate your analysis results.

Pandas is widely used in data analysis, data science and machine learning tasks in Python. Its flexibility and ease of use make it an indispensable tool for data scientists and analysts working with structured and unstructured data.

**NumPy** The NumPy (Numerical Python) library is one of the fundamental libraries for scientific programming in Python. It provides support for multidimensional arrays and high-speed math functions, making Python a powerful language for manipulating and analyzing numerical data. Key features of NumPy include:

- **Multidimensional Arrays:** NumPy introduces the data type "ndarray," which is a homogeneous multidimensional array. These arrays can be one or more dimensions and allow representing and manipulating data in the form of vectors, matrices, or tensors.
- **Efficient Operations:** NumPy is implemented in C and Fortran, allowing for high-speed mathematical operations on arrays. This feature is essential for analyzing large volumes of data and scientific computing.
- **Broadcasting:** NumPy supports broadcasting, a technique that allows performing operations between arrays of different shapes without the need to explicitly duplicate or expand them.

This significantly simplifies operations on arrays with different dimensions and makes the code more compact and readable.

- **Mathematical Functions:** NumPy offers a wide range of high-speed mathematical functions, including arithmetic, trigonometric, exponential, logarithmic and much more. These functions can be applied to arrays for performing large-scale calculations.
- **Advanced Indexing:** NumPy supports advanced indexing, enabling the selection of elements or subsets of arrays in very flexible ways, including using boolean masks, integer indices and array indices.
- **Linear Algebra:** NumPy provides functions for performing linear algebra operations, such as matrix calculations, decompositions, computation of eigenvalues and eigenvectors, solving linear equations systems and more.
- **Integration with Other Libraries:** NumPy is often used in combination with other Python libraries for data analysis and scientific computing, such as Pandas for data manipulation and Matplotlib for data visualization.
- **Random Number Support:** NumPy offers methods for generating random numbers with various distributions, useful for simulations and statistical analysis.

NumPy has become an essential tool for data scientists, engineers, researchers and anyone working with complex numerical data. Its speed and flexibility make it a popular choice for applications that require intensive numerical computations and structured data manipulation in Python.

**Matplotlib** The Matplotlib library is an essential tool for data visualization and plotting in Python. It offers a wide range of features to generate high quality graphics in different formats and styles. The main features of Matplotlib include:

- **Customizable Graphs:** Matplotlib allows creating highly customizable graphs, providing complete control over aspects such as titles, axis labels, colors, line styles and markers. This flexibility enables users to tailor the graphs to specific visualization needs.
- **Support for Various Chart Types:** Matplotlib supports a wide range of chart types, including line plots, scatter plots, bar charts, histograms, pie charts, scatter plots, surface plots and much more. This versatility allows for effective representation and analysis of different types of data.

- **Interactive Visualization:** Matplotlib offers interactive visualization features that allow users to explore data, zoom in and out, pan and customize the graph directly in the visualization environment.
- **Output Formats Support:** The graphs created with Matplotlib can be saved in various formats, including PNG, JPEG, PDF, SVG and many others. This allows using the graphs in presentations, documents and web pages.
- **Seamless Integration with NumPy and Pandas:** Matplotlib seamlessly integrates with NumPy and Pandas, making it easy to visualize data stored in arrays and DataFrames directly.
- **3D Graphs:** Matplotlib supports the creation of 3D graphs, enabling the representation of data involving three variables, such as surface data or volumetric data.
- **Default Styles:** Matplotlib offers a wide selection of default styles that allow creating professionally-looking graphs with just a few lines of code. Additionally, users can customize styles to meet their specific requirements.

Matplotlib is one of the most popular and widely used visualization tools in Python. Its ease of use and flexibility in creating customized graphs make it an essential tool for exploring, analyzing and communicating data in a clear and engaging manner.

**Torch** The Torch library (TorchScript) is an open-source machine learning framework, primarily developed by Facebook’s AI Research (FAIR) laboratory. It is designed to facilitate the development of deep learning models and the implementation of machine learning algorithms on different platforms. Key features of Torch include:

- **Tensors:** Torch provides an efficient library for tensor manipulation, similar to NumPy. Tensors are multidimensional arrays that form the foundation for deep learning operations and scientific computations.
- **Computational Graph Implementation:** Torch uses a dynamic computational graph to represent the flow of data through the machine learning model. This approach allows for executing complex graphs and dynamic models, enhancing flexibility compared to a static graph used by some other frameworks.
- **Automatic Gradient Calculation Support:** Torch includes functions for automatic gradient calculation, essential for optimizing deep learning models during the learning process.

- **Extensive Collection of Algorithms and Models:** Torch offers a wide range of pre-defined algorithms and deep learning models, including neural networks, recurrent networks, convolutional networks, transformer models and much more.
- **GPU Support:** Torch is optimized to harness the power of Graphics Processing Units to accelerate tensor and machine learning model computations.
- **TorchScript and PyTorch JIT:** TorchScript is a high-level scripting language for PyTorch, enabling serialization of trained models for deployment and execution on various platforms. PyTorch JIT (Just-In-Time Compilation) optimizes PyTorch code to improve runtime performance.

Torch is a powerful tool for developing deep learning models and implementing advanced machine learning algorithms. Its versatility, speed and GPU support make it a popular choice among data scientists and machine learning developers worldwide.

**SciKit Learn** The SciKit-Learn library is one of the most popular and comprehensive tools for machine learning in Python. It offers a wide range of algorithms and tools for data preparation, supervised and unsupervised learning, model selection, performance evaluation and much more. Key features of SciKit-Learn include:

- **Simplicity and Consistency:** SciKit-Learn is designed to be easy to use and follows a coherent design philosophy. Its interface is clear and intuitive, making it simple to use algorithms and experiment with various machine learning models.
- **Wide Range of Algorithms:** SciKit-Learn offers a diverse selection of algorithms for classification, regression, clustering, dimensionality reduction, feature selection and much more. This extensive choice allows tailoring the model to the specific needs of the problem.
- **Data Preparation Tools:** SciKit-Learn provides functions for data management, including preprocessing, transformation, normalization and dimensionality reduction. These tools are essential for preparing data before training machine learning models.
- **Performance Evaluation:** SciKit-Learn offers methods to evaluate the performance of machine learning models through evaluation metrics and cross-validation techniques. These tools help understand how well the model fits the data and select optimal parameters.
- **Model and Pipeline Management:** SciKit-Learn supports the creation of model pipelines to automate the workflow, allowing combining different preprocessing and model training operations into a single object.



- **Scalability and Parallelization:** SciKit-Learn can handle large amounts of data and offers support for parallelization, leveraging the advantages of multi-core and distributed systems.
- **Integration with other Python Libraries:** SciKit-Learn easily integrates with other Python libraries, such as NumPy, Pandas and Matplotlib, enabling users to leverage the power of these libraries in combination with SciKit-Learn.

SciKit-Learn is a popular choice for most machine learning tasks in Python. Its flexibility, ease of use and extensive support make it a fundamental tool for data scientists, engineers and researchers working with machine learning.

**OpenCV** The OpenCV library (Open Source Computer Vision Library) is one of the most popular and widely used libraries for computer vision and image processing in Python. It offers a wide range of features for manipulating and analyzing images and videos. Key features of OpenCV include:

- **Image Processing:** OpenCV provides a wide range of functions for image processing, including loading, saving, pixel manipulation, format conversion, cropping, resizing, rotation and much more.
- **Object Detection:** OpenCV offers algorithms and tools for object detection in images, including line detection, contour detection, geometric shape detection and character detection.
- **Facial Recognition:** OpenCV includes algorithms for facial recognition, such as the Viola-Jones algorithm, which enables face detection and the detection of facial features like eyes and mouth.
- **Object Recognition:** OpenCV supports object recognition using machine learning algorithms, such as object recognition using image descriptors and trained classifiers.
- **Motion Tracking:** OpenCV provides features for motion tracking, which can be used to monitor objects or people in a video.
- **Camera Calibration:** OpenCV offers tools for camera calibration, useful for correcting image distortions and calculating the projection matrix.
- **Mathematical Operations on Images:** OpenCV allows performing various mathematical operations on images, such as convolution, Fourier transform and other filtering operations.
- **Video Processing:** OpenCV supports real-time video processing and manipulation of video frames.

OpenCV is an extremely powerful and versatile library for image and video processing in Python. It is widely used in a broad range of applications, including robotics, computer vision, augmented reality, video surveillance, industrial automation and much more.

## 3 Artificial Intelligence Applied to Photovoltaics Panels

### 3.1 Introduction

This chapter will discuss how to apply artificial intelligence techniques and technologies to the world of photovoltaic panels. In the sustainable energy landscape, photovoltaic panels have achieved an important role as a promising source of renewable energy. The growing adoption of this technology requires innovative solutions to improve the efficiency and reliability of photovoltaic systems. In such a context, artificial intelligence techniques and technologies emerge as powerful tools to address complex challenges in the world of photovoltaic panels, in fact the application of artificial intelligence in this sector offers significant opportunities to improve the design, maintenance, performance prediction and energy management, helping to make solar energy more competitive and sustainable.

Artificial intelligence offers innovative approaches for the predictive maintenance of photovoltaic panels. Timely diagnosis of failures and malfunctions is crucial to ensure the efficiency and life of photovoltaic systems. Therefore, in this discussion the following question was asked: Is it possible to apply machine learning techniques to automate the detection of defects in photovoltaic panels? Being able to automatically identify thermal anomalies, mechanical damage and other hidden problems can make it possible to plan targeted maintenance interventions, reducing downtime and costs associated with corrective maintenance.

Another key area where artificial intelligence can be successfully applied is the performance prediction of photovoltaic panels. In fact, traditional forecasting models are based on analytical and mathematical approaches, which often do not take into account the complexities and non-linear interactions present in photovoltaic systems. In contrast, machine learning models can capture complex relationships between input variables and energy production, providing more accurate and reliable predictions. Such models can be trained using historical production data and environmental variables to estimate future energy production with unprecedented accuracy. Therefore, in this discussion the following question was asked: Is it possible to use machine learning models to estimate the temperature of the photovoltaic panel starting from the environmental data, thus competing with the modern physical models that deal with this? Being able to estimate the temperature of a photovoltaic panel is very important to analyze any anomalies or to study its degradation.

The application of artificial intelligence techniques and technologies in the world of photovoltaic panels offers multiple advantages to improve the efficiency, reliability and sustainability of photovoltaic systems. With the continued development and widespread adoption of artificial intelligence

technologies, solar energy is expected to reach a new level of competitiveness and contribute significantly to the transition to a future low-carbon and sustainable energy society.

### **3.2 Photovoltaic Panels**

Photovoltaic panels are devices that convert solar energy into electricity using the so-called "photovoltaic effect". This effect occurs when sunlight strikes a photovoltaic cell, composed of semiconductor materials, such as silicon, creating an electric potential difference across the cell, which can be used to generate direct current. The efficiency of the photovoltaic panels depends on the quality of the semiconductor materials used, on the structure of the photovoltaic cell and on the presence of any impurities or defects in the cell itself. The semiconductor materials most commonly used in photovoltaic panels are monocrystalline silicon, polycrystalline silicon and amorphous silicon. The first one is the most efficient material, but also the most expensive to produce; the second one is less efficient but cheaper to produce than the first one; the third one is the least efficient but also the least expensive to produce.

Photovoltaic panels can be used for the production of electricity for both domestic and industrial use, powering entire buildings or electricity grids. They can be installed on roofs, on land or on dedicated structures, such as solar car parks. Photovoltaic panels have the great advantage of being able to produce electricity without emitting greenhouse gases or other pollutants. Furthermore, once installed, they require relatively little maintenance and can last for decades. However, photovoltaic panels can be expensive to install and require adequate exposure to sunlight to efficiently produce electricity. Overall, this technology represents a renewable and sustainable energy source that can help reduce the environmental impact of traditional energy sources. With the continuous development of new technologies and materials, the production of photovoltaic panels is becoming more and more efficient and economical, paving the way for an increasingly sustainable and cleaner future.

In addition to the production of electricity, photovoltaic panels can be used for the production of hydrogen starting from water through the so-called "Water Splitting", a process that uses solar energy to divide water into hydrogen and oxygen, which can be used as clean fuel. Photovoltaic panels present some challenges related to their production, which generally involves the use of: rare and expensive materials which can be difficult to find and which can influence the price for the end customer; non-renewable energy sources. In addition to manufacturing challenges, there are also technical challenges for photovoltaic panels, such as the fact that the efficiency of photovoltaic panels is highly dependent on the amount of sunlight hitting the panel surface, which is why mechanical solar tracking systems have been developed that allow the photovoltaic

panel to track the movement of the sun to maximize the amount of "captured" sunlight. Despite these challenges, the use of photovoltaic panels continues to grow worldwide, driven by growing awareness of the importance of environmental sustainability and the need to reduce dependence on traditional energy sources.

The use of photovoltaic panels has also been encouraged by various public policies, such as government subsidies, incentive tariffs and greenhouse gas emission reduction targets. These incentives have helped to make solar panels more affordable and to promote the adoption of this technology around the world. However, despite the increased use of photovoltaic panels, solar power generation still accounts for a small percentage of the global energy mix. There are still many challenges to overcome, one of which, related to the present PhD Thesis, consists in improving the automatic detection of defects present on the cells of the photovoltaic panels.

### **3.3 Automatic Detection and Classification of Defects in Photovoltaic Cells**

In this section I will discuss the topic to which the most time was devoted during the doctorate research, that is the automatic detection and classification of defects present in photovoltaic panel cells [2].

#### **3.3.1 Introduction**

Solar modules are subject to a range of atmospheric events such as rain, wind and snow and for this reason, they are usually built with protection frames. Nevertheless, these measures are insufficient to prevent damages, especially from the mechanical ones (e.g., the fall of tree branches) decreasing the power efficiency of solar modules. Then it is necessary to monitor their healthy conditions and replace or repair defective units. Electroluminescence, a useful inspection modality of solar modules, makes it possible to detect even the finest defects on the surface of solar modules. However, the analysis of these images is usually carried out by human operators, making this inspection practice expensive, time-consuming and it requires very specific knowledge. In state-of-the-art there are several works that distinguish between a healthy cell and defective cell, but a public dataset of possible defects in solar cells has never been published. For this reason, we propose a new dataset and a preliminary benchmark to make an automatic and accurate classification of defects in solar cells. The dataset includes five classes of defects and the pre-trained ResNext50 network reaches 0.07 Hamming Distance.

### 3.3.2 Luminescence and Electroluminescence

Luminescence is a physical phenomenon which involves the emission of light radiations by a body. This phenomenon is based on the interactions between the atoms or molecules of the body and the energy supplied by the excitation. When materials are excited, for example by absorption of light or other forms of energy, the electrons absorb this energy and move towards higher energy states. Subsequently, the electrons relax back to lower energy states, releasing the excess energy in the form of photons of light. Luminescence is of crucial importance in several technological applications. For example, in the medical field, it is used in imaging techniques such as Positron Emission Tomography (PET) and Single Photon Emission Computerized Tomography (SPECT), in which fluorescence is used to obtain functional images of the human body. Furthermore, it is of fundamental importance in modern lighting techniques which exploit the fluorescence in semiconductor materials to convert electrical energy into light. Depending on the cause of the phenomenon, different types of luminescence are distinguished:

- Bioluminescence: caused by biochemical processes;
- Chemiluminescence: caused by chemical reactions;
- Electroluminescence: caused by electrical phenomena;
- Fluorescence and Phosphorescence: caused by photons;
- Triboluminescence: caused by mechanical actions;
- Radioluminescence: caused by nuclear radiation;

Electroluminescence is the basis of the experiments that will be shortly described. It is an optical phenomenon in which materials emit light when subjected to an electric field, or an electric current. This phenomenon occurs mainly in semiconductor materials and is the basis of the operation of devices such as Light Emitting Diodes (LED) and Liquid Crystal Displays (LCD). Electroluminescence has been a revolutionary technology in the lighting and display industry, enabling the production of compact, efficient, high-definition electronic devices. Continuous research and continuous development in the field of electroluminescence have led to constant improvements in terms of energy efficiency, brightness and color gamut of electroluminescent devices, contributing to the diffusion of new and innovative applications in the fields of electronics, communication and lighting.

### 3.3.3 Electroluminescence Applied to Photovoltaic Panels

The electroluminescence test applied to photovoltaic panels is a diagnostic technique used to evaluate the state and integrity of photovoltaic panels. This testing is a non-destructive procedure that identifies any internal defects or damage in the panels, allowing maintenance operators or experts to identify and resolve any problems before they can worsen and affect the overall performance of the photovoltaic system. The electroluminescence test is based on the emission of light by photovoltaic panels when subjected to an electric field. During the test, carried out in low light conditions, an electrical signal is applied to the photovoltaic panel with the aim of exciting the electrons present inside the photovoltaic cells so as to release the excess energy in the form of light. This emitted light is detected by a particular camera, which captures a thermographic or infrared image of the panel. The images resulting from the electroluminescence test show the thermal details of the photovoltaic cells and reveal any anomalies or defects inside the panels, which can cause efficiency losses or even the malfunction of parts of the panel, reducing the overall capacity of the photovoltaic system to generate energy. These anomalies or defects can be:

- **Hot Spots:** Hot spots are localized areas within the photovoltaic cells where there is increased resistance to current flow. This can be caused by material defects, aging or mechanical damage. Hot spots can lead to an increase in local temperature and a reduction in panel efficiency.
- **Cold spots:** Cold spots are areas where electric current flows more easily than in other parts of the photovoltaic cell. This can be caused by material defects or interruptions in the electrical connections. Cold spots can reduce cell efficiency and cause excessive current leakage.
- **Delamination:** Delamination occurs when the layers of the photovoltaic panel separate or peel away, creating gaps between them. This can cause reduced efficiency and make the panel more vulnerable to future damage.
- **Cracks or breaks:** The electroluminescence test can reveal the presence of cracks or breaks inside the photovoltaic cells or the front glass of the panels. Such damage can occur due to mechanical stress, exposure to the elements or incorrect installation.
- **Inclusions:** The electroluminescence test can reveal the presence of inclusions of foreign material inside the photovoltaic cells. Such foreign elements can interfere with current flow and degrade panel performance.
- **Damaged or Defective Cells:** Electroluminescence testing can reveal individual damaged or defective cells within the panel. These cells can be replaced to restore panel efficiency.

Detecting and identifying these defects via electroluminescence testing allows timely corrective measures, such as repairing or replacing damaged panels, to be taken to maintain the efficiency and energy production of the photovoltaic system. This advanced diagnostic technique helps optimize system performance and lifespan, while reducing maintenance costs and increasing the return on solar energy investments. The electroluminescence test is particularly useful in the preventive maintenance of photovoltaic panels. It allows operators to locate hidden problems without having to disassemble panels, thus reducing downtime and costs associated with maintenance operations.

### **3.3.4 Artificial Intelligence Techniques for Electroluminescence Test Automation**

Artificial intelligence can play a vital role in automating the electroluminescence test applied to photovoltaic panels, making the process more efficient, accurate and scalable. The capabilities of AI can be exploited in different phases of the test, helping to improve the diagnosis and maintenance of photovoltaic panels. Here are a few ways AI can be used to automate electroluminescence testing:

- **Image Processing:** AI can be used to analyze and interpret infrared or thermographic images captured during electroluminescence testing. Computer vision algorithms can automatically identify hot spots, cold spots, cracks, delaminations and other anomalies within photovoltaic panels. This allows to obtain more precise results and to detect defects that could escape the human eye.
- **Classification of Defects:** With AI, it is possible to classify defects found in photovoltaic panels according to their severity and possible impacts on performance. This helps maintenance operators to prioritize corrective actions and make informed decisions about repairing or replacing photovoltaic panels.
- **Monitoring:** AI can be used to implement continuous monitoring systems for photovoltaic panels. Such systems can collect real-time data from electroluminescence tests and other measurements, allowing to quickly identify any changes or anomalies in panel behavior. This helps in early detection of failures or performance degradation.
- **Prediction of Useful Life:** With AI, it is possible to analyze the historical data of photovoltaic panels and predict their useful life based on the operating conditions and detected defects.
- **Maintenance Optimization:** AI can be used to optimize the maintenance plans of photovoltaic panels. By analyzing historical data and considering factors such as performance degradation and previous maintenance cycles, AI can suggest more efficient maintenance strategies and reduce associated costs.



- **Decision Support:** AI can help photovoltaic plant operators to make informed decisions about maintenance and performance optimization. Through data analytics and machine learning, AI can provide recommendations based on accurate and reliable information.

The integration of artificial intelligence into electroluminescence test automation leads to many benefits, including faster and more accurate diagnosis, better performance management and reduced maintenance costs. In addition, automation allows to manage large amounts of data efficiently, permitting to monitor and maintain large photovoltaic systems more effectively. AI continues to evolve and improve, providing new opportunities to optimize the use of solar energy and promote the transition to renewable and sustainable energy sources. One of the objectives established for the doctorate research was to use AI for the automatic classification of photovoltaic cells with the aim of being able to distinguish healthy cells from defective ones without the help of a maintenance worker and, in this last case, to identify the various types of defects present.

### 3.3.5 State of the Art

Starting from 44 electroluminescence images of photovoltaic modules, which consisted in 18 monocrystalline modules and 26 polycrystalline modules, the work in [7] proposed a segmentation strategy in order to extract the various cells from the modules. By this process, the authors were able to extract 2624 cells. Subsequently, the authors of [8] dealt with the automatic classification of the various cells by a classifier reaching an average accuracy of 88.42% by considering a CNN based solution. This classification task aimed at indicating how likely a defect is present without specifying any details. The dataset (ELPV Dataset) used for the classification of the cells with the associated labeling has been publicly released. Using the same dataset, but with a little different labelling, the work in [9] implemented an isolated CNN, that is not pre-trained, for the classification of the cells, which achieves an average accuracy of 93.02%. The authors of [10] with a Deep Feature-Based Support Vector Machine (DFB-SVM) technique outperformed the works in [8] and [9] obtaining an average accuracy of 89.63% and 94.52%, respectively. The authors of [11] classified with an average accuracy of 83% two kind of defects: micro-cracks and finger-interruption. However, it used both the public ELPV Dataset and its own dataset, which makes the experiments not replicable.

The aim of this research is to develop a cell classifier able to classify the specific defect detected on the input image, if any, on the basis of a set of predetermined defects. Although previous research works in this field shown high quantitative performances, they only focused on the task of defect detection. This motivated us to extend the ELPV Dataset with a new labeling, concerning 5 specific defect classes and healthy classes (for the samples without defects). The labeling of an already existing large-scale dataset will be useful for the community, as ELPV represents a

standard in the field. In this sense, an important contribution of this work is the release of the new and extended labeling of the ELPV dataset, named E-ELPV (i.e., Extended-ELPV) and a benchmark evaluation for the task of defect classification. To our knowledge, no prior studies addressed the task of defects classification at this level of detail, nor similar public datasets have been released.

### 3.3.6 Dataset

The ELPV Dataset<sup>1</sup> [8] consists of 2624 electroluminescence cells of monocrystalline and polycrystalline photovoltaic modules labeled by an expert, to whom for each cell, in addition to answering the question "Is the cell defective?", also took care of answering the question "Are you sure?". By doing so, if the evaluator indicated with certainty that a cell was defective, a probability of defectiveness equal to 100% was assigned; if the evaluator indicated without certainty that a cell was defective, a probability of defectiveness equal to 67% was assigned; if the evaluator indicated with certainty that a cell was healthy, a probability of defectiveness equal to 0% was assigned; if the evaluator indicated without certainty that a cell was healthy, a probability of defectiveness equal to 33% was assigned. Table 1 resumes the original labelling strategy of the ELPV Dataset with the cardinality of each class. Figure 3 shows a monocrystalline cell and a polycrystalline cell, whereas Figure 4 shows a comparison between a cell with a probability of defectiveness equal to 0% and a cell with a probability of defectiveness equal to 100%.

Condition	Confident?	Probability of Defectiveness	Cardinality
Healthy	Yes	0	1508
Healthy	No	0.33	295
Defective	No	0.67	106
Defective	Yes	1	715
Total			2624

Table 1: The original labelling strategy of the ELPV Dataset with the cardinality of each class.

Cells that have been assigned a defect probability of 0% are marked as *Healthy* on our labeling. The remaining cells have been labeled as follows: *Crack* if the cell presents one or more cracks (not too much otherwise the cell is considered as totally broken); *CellBreakage* if the image presents cracks covering more than 50% of the cell's surface; *DarkArea* if the cell has one dark area; *HotSpot* if the cell has one or more hot spots; *OtherDefect* if the cell has a defect different from the previous ones. Then, the associations in Table 2 were obtained. The number of associations is

<sup>1</sup><https://github.com/zae-bayern/elpv-dataset>

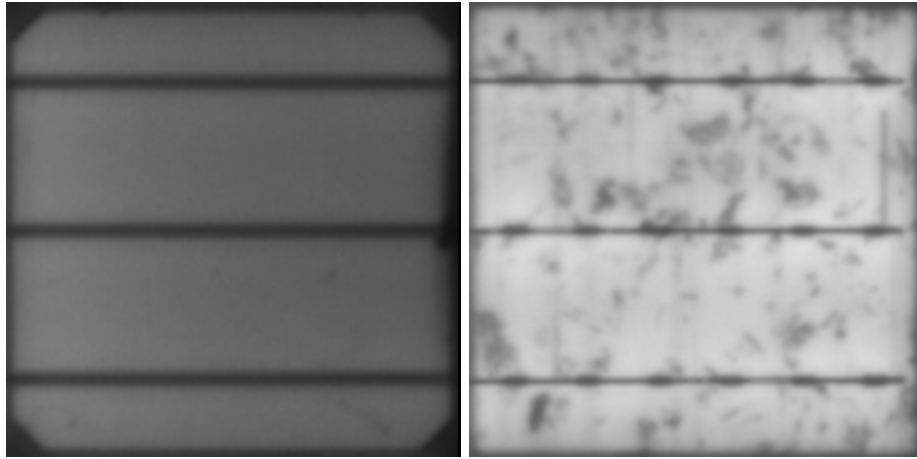


Figure 3: On the left is shown a monocrystalline cell. On the right is shown polycrystalline cell. Both images have a probability of defectiveness of 0%. The monocrystalline cell appears cleaner than the polycrystalline cell, which appears to be dirty.

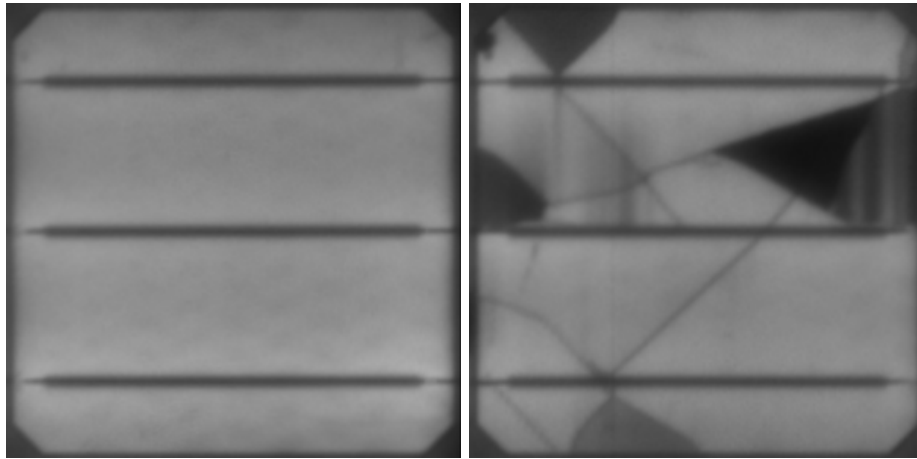


Figure 4: On the left is shown a cell with a probability of defect equal to 0. On the right is shown a cell with a probability of defect equal to 1.

greater than the number of images because to each defective cell could contain one or more defects. In particular, 2551 images are associated to one label and 73 images are associated to two labels. Figure 5 shows the four types of defects studied in this work.

### 3.3.7 Proposed Methods

Our methodology consists in the use of two classifiers: the first deals with classifying the health of a cell, indicating whether it is healthy or defective. In the latter case, the cell is given to the second classifier who is responsible for establishing which defects is present. Furthermore, we have developed an additional classifier that treats the healthy images together with the defective images,

Label	Associations
Healthy	1508
Crack	430
CellBreakage	133
DarkArea	62
HotSpot	171
OtherDefect	393
Total	2697

Table 2: Our custom labelling of the ELPV Dataset.

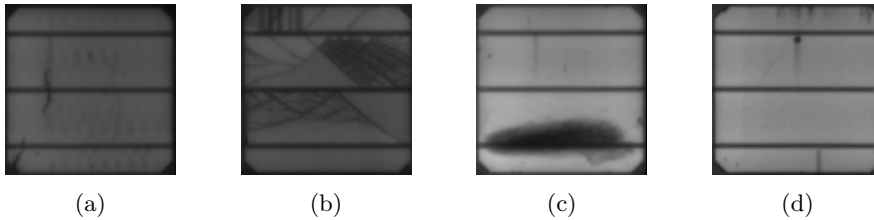


Figure 5: **(a)** cell with Crack defect; **(b)** cell with a CellBreakage defect; **(c)** cell with DarkArea defect; **(d)** cell with a HotSpot defect.

the latter labeled with the respective defects. Our purpose is to answer to the following questions: 1) Is training a Healthy VS Not-Healthy binary classifier better than training a multiclass classifier with the same architecture? 2) Does adding the Healthy class to the other classes in the same training improve the classification rate of the other classes?

All developed classifiers use the same image processing and image augmentation. They are developed with PyTorch library, using Python as Programming Language. Furthermore, they are tested and compared with the following unmodified networks: ResNext-50[12], VGG-11[13], Inception-V3[14] and DenseNet-121[15], on which Fine-Tuning was carried out. These neural networks have different characteristics and depth. Images are normalized using the following formula:

$$\overline{image} = \frac{image - mean}{std} \quad (1)$$

where  $\overline{image}$  is the normalized image;  $image$  is the original image;  $mean$  and  $std$  are respectively the mean and the standard deviation of ImageNet [16] challenge database. This is due to the fine-tuning of a pretrained model on this dataset. Data augmentation is used. Offline image augmentation simply consists of rotating each image  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ . Online image augmentation consists of the following transformations: Gaussian Blur (Kernel  $5 * 5$ ); Color Jitter; Random Horizontal Flip ( $p = 0.5$ ); Random Vertical Flip ( $p = 0.5$ ); Random Rotation ( $-3, +3$ ); Random Translation ( $0.02, 0.02$ ).

**First Classifier (Healthy Classifier)** Given an image of a cell, the first classifier has the purpose of indicating whether the cell is healthy or has defects, without indicating in the latter case which defects are present. The metric used to measure the performance was Overall Accuracy  $OA$  calculated as follows for each *batch*:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

where  $TP$  (True Positives) are defective cells correctly classified as defective;  $TN$  (True Negatives) are healthy cells correctly classified as healthy;  $FP$  (False Positives) are healthy cells incorrectly classified as defective;  $FN$  (False Negative) are defective cells incorrectly classified as healthy. The loss function used for this classifier is the *CrossEntropyLoss* for all networks. This classifier was trained with the following hyperparameters: learning rate is 0.0001 and batch size is 32.

**Second Classifier (Defects Classifier)** This task deals with a multilabelling problem. It means that given an image of a cell marked by the first classifier as unhealthy, the second classifier is intended to indicate which defects are present between Crack, CellBreakage, DarkArea, HotSpot and OtherDefect. For this reason we have chosen a properly metric for the multilabelling problem such as the Hamming Distance  $HD$  calculated as follows for each *batch*:

$$HD = 1 - \frac{|predictions \cap groundtruths|}{|predictions|} \quad (3)$$

The loss function used for this classifier is *BCELoss* for ResNext-50 and *BCEWithLogitsLoss* for the other networks. For this classifier the training setting was learning rate 0.0001 and batch size 32.

**Third Classifier** To assess the need of two different classifiers, we also trained a third classifier that performs the classification including all the classes. Given an image of a cell, the third classifier is intended to indicate if the cell is Healthy or if there are defects between Crack, CellBreakage, DarkArea, HotSpot and OtherDefect. The metric and the loss function are the same of the second classifier. The setting employed for the third classifier was learning rate 0.001 and batch size 32.

### 3.3.8 Analysis of Deep Neural Networks

In this section, the four deep neural networks used for the experiments will be analyzed, namely ResNext50, VGG11, InceptionV3, and DenseNet121.

**ResNext50** The architecture of ResNext50 is an advanced variant of the convolutional neural network ResNet (Residual Network). It is designed to handle very deep and complex neural networks and is particularly effective in the field of image recognition and machine learning. Here are some key features of the ResNext50 architecture:

- **Residual Structure:** Like its counterpart ResNet, ResNext50 extensively uses residual blocks. These blocks allow the direct flow of information through the layers of the network, making it easier to train very deep neural networks without the vanishing gradient problem. In other words, residual blocks help prevent performance degradation when increasing the number of layers in the network.
- **Cardinality:** The most distinctive feature of ResNext is the introduction of the concept of "cardinality". It represents the number of branches or "paths" within each convolutional block. While in previous architectures like ResNet there were blocks with only one convolutional path, in ResNext, there are multiple parallel paths in each block. This diversity of paths helps the network capture a wider range of features, contributing to improved efficiency and performance.
- **Filter Size:** ResNext50 uses 3x3 filters within its convolutional blocks. These filters are applied to extract various features from input images.
- **Depth:** The "50" in ResNext50 indicates the total number of layers in the network. This includes convolutional layers, normalization layers, ReLU activation layers, and fully connected layers. The depth contributes to the network's ability to learn increasingly complex representations of images.

**VGG11** The architecture of VGG11 is a convolutional neural network that belongs to the VGG (Visual Geometry Group) model family. This architecture was developed by the VGG research group at the University of Oxford and is designed for image analysis and recognition. The key features of the VGG11 architecture are:

- **Depth:** The name "VGG11" indicates the depth of the network, which has 11 neural layers, including convolutional layers, normalization layers, and fully connected layers. The depth of the network contributes to its ability to learn complex representations from images.
- **Filter Size:** VGG11 uses very small filters, specifically 3x3, in all its convolutional layers. The use of 3x3 filters in combination with numerous convolutional layers allows the network to learn details at various scales in images.

- **Pooling Layers:** After some convolutional layers, VGG11 employs pooling layers (usually max-pooling) to progressively reduce the size of feature maps while retaining the most relevant features.
- **Fully Connected Layers:** At the end of the network, there are one or more fully connected layers responsible for classifying images into different categories. These layers are often followed by activation functions such as the ReLU function to introduce non-linearity into the model.
- **Symmetric Architecture:** The VGG11 architecture is characterized by significant symmetry, with convolutional blocks having the same number of layers and the same number of filters in each block. This symmetric design simplifies the network's design and implementation.
- **Feature Map Size:** The size of feature maps progressively decreases as you move deeper into the network due to the use of pooling layers. This allows the network to capture increasingly abstract information as it moves towards deeper layers.

**InceptionV3** The architecture of InceptionV3 is an advanced convolutional neural network primarily used for image recognition and other computer vision applications. It is one of the variants from the Inception model family developed by Google, also known as GoogLeNet. Here are some of the key features of the InceptionV3 architecture:

- **Inception Module:** The distinctive feature of InceptionV3 is the use of "Inception" modules. These modules consist of various convolution operations performed in parallel with filters of different sizes (e.g., 1x1, 3x3, 5x5) and pooling layers. This parallel architecture allows the network to capture information at various scales and learn a variety of features from an image. The outputs of these modules are then concatenated to create a more information-rich feature map.
- **Use of 1x1 Filters:** InceptionV3 extensively employs 1x1 filters, known as "bottleneck layers," within the Inception modules. These 1x1 filters serve to reduce the dimensionality of feature maps before applying larger filters, thus helping to reduce the overall number of parameters in the network and improve computational efficiency.
- **Use of Average Pooling:** At the end of the network, instead of using traditional fully connected layers, InceptionV3 uses a global average pooling layer followed by a classification layer. This approach allows the network to adapt to inputs of varying sizes without changing the number of parameters in the classification phase.

- **Regularization and Normalization:** InceptionV3 incorporates regularization techniques such as batch normalization and dropout to enhance the network's ability to generalize from limited training data and reduce the risk of overfitting.
- **Deep Network:** InceptionV3 is a very deep network with numerous layers, which enables it to capture increasingly complex representations of images but also requires training on large datasets.

**DenseNet121** The architecture of DenseNet121 is a convolutional neural network developed for image recognition and other computer vision applications. The name "DenseNet" is a contraction of "Densely Connected Convolutional Network," which suggests one of its key features: dense connectivity between its layers. The key features of the DenseNet121 architecture are:

- **Dense Blocks:** The most distinctive feature of DenseNet is the use of "dense blocks." In a dense block, each layer receives input from all previous layers, and the output of each layer is passed to all subsequent layers. This dense connectivity helps mitigate the vanishing gradient problem, allowing for more efficient information flow and improving feature learning.
- **Bottleneck Layers:** DenseNet uses 1x1 convolution filters, known as "bottleneck layers," to reduce the dimensionality of feature maps before applying 3x3 filters. This helps reduce the overall number of parameters in the network, improving efficiency.
- **Pooling and Transition:** Between dense blocks, pooling and transition layers are inserted. Pooling layers reduce the size of feature maps, while transition layers reduce the number of channels. These layers help control the growth of feature map dimensions and reduce computational load.
- **Global Average Pooling:** At the end of the network, DenseNet121 uses a Global Average Pooling layer followed by a classification layer. This approach allows the network to directly generate a probability distribution for output classes without traditional fully connected layers.
- **Regularization:** DenseNet employs regularization techniques such as batch normalization and dropout to prevent overfitting during training.
- **Deep Network:** DenseNet121 is a deep network with numerous layers. This depth contributes to its ability to learn complex representations of images but also requires extensive training datasets.



### 3.3.9 Results

We have chosen the mentioned neural networks (ResNext-50, VGG-11, Inception-V3 and Densenet-121) to benchmark our dataset. First we computed a baseline for each neural network and then we fine-tuned them. We have chosen this strategy to measure the advantage of training the network instead of using its pre-trained weights on ImageNet. The results of the first classifier, the second classifier and the third classifier are shown in Table 3, Table 4 and Table 5, respectively.

The dataset described in 3.3.6 is clearly unbalanced but we did not balance it because balancing strategies cause overfitting. We can note that with the ResNext50 network, it is possible to pass from a Hamming Distance of 0.41 (Baseline) to 0.08 in only two epochs for the second classifier; moreover, the Baselines of the VGG11, InceptionV3 and DenseNet121 networks have quite low Hamming Distances of 0.08, 0.10 and 0.18 respectively, making the training of the various networks unable to improve performances. Instead, the third classifier reaches convincing Hamming Distances after a significant number of epochs. The best result is obtained with the ResNext50 network passing from a Hamming Distance of 0.50 (Baseline) to 0.07 only after 100 epochs. It is important to underline that the other networks also seem to work well, reaching a Hamming Distance of 0.08. For each class and for each classifier Table 6 shows information about True Positives, True Negatives, False Positives and False Negatives. Instead, Table 7, Table 8 and Table 9 illustrate example of matching and mismatching. A machine with an Nvidia Quadro RTX 6000 was used to carry out the various experiments.

Network	Best Epoch	Overall Accuracy
ResNext50 Baseline	/	0.41
<b>ResNext50</b>	<b>86</b>	<b>0.79</b>
VGG11 Baseline	/	0.55
VGG11	98	0.77
InceptionV3 Baseline	/	0.47
<u>InceptionV3</u>	<u>88</u>	<u>0.78</u>
DenseNet121 Baseline	/	0.38
DenseNet121	91	0.77

Table 3: First Classifier Results. The best and the running up results are highlighted in bold and underline, respectively.

Network	Best Epoch	Hamming Distance
ResNext50 Baseline	/	0.41
<b>ResNext50</b>	<b>2</b>	<b>0.08</b>
VGG11 Baseline	/	0.08
<b>VGG11</b>	<b>97</b>	<b>0.08</b>
InceptionV3 Baseline	/	0.10
<b>InceptionV3</b>	<b>16</b>	<b>0.08</b>
DenseNet121 Baseline	/	0.18
<b>DenseNet121</b>	<b>54</b>	<b>0.08</b>

Table 4: Second Classifier Results. The best and the running up results are highlighted in bold and underline, respectively.

Network	Best Epoch	Hamming Distance
ResNext50 Baseline	/	0.50
<b>ResNext50</b>	<b>100</b>	<b>0.07</b>
VGG11 Baseline	/	0.17
<u>VGG11</u>	<u>92</u>	<u>0.08</u>
InceptionV3 Baseline	/	0.17
<u>InceptionV3</u>	<u>99</u>	<u>0.08</u>
DenseNet121 Baseline	/	0.22
<u>DenseNet121</u>	<u>99</u>	<u>0.08</u>

Table 5: Third Classifier Results. The best and the running up results are highlighted in bold and underline, respectively.

Classifier	Class	TP	TN	FP	FN	Total
First Classifier	Healthy	258	156	45	65	524
Second Classifier	Crack	13	88	40	60	201
	Cell Breakage	2	122	58	19	201
	Dark Area	2	164	26	9	201
	Hot Spot	0	167	3	31	201
	Other Defect	1	110	13	77	201
Third Classifier	Healthy	297	165	36	26	524
	Crack	52	443	8	21	524
	Cell Breakage	21	499	4	0	524
	Dark Area	5	506	7	6	524
	Hot Spot	15	481	12	16	524
	Other Defect	55	389	57	23	524

Table 6: Information about True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for each classifier and for each class.

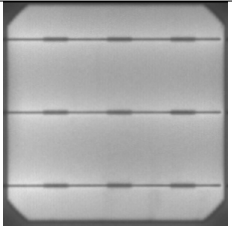
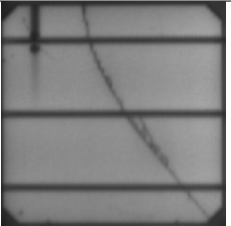
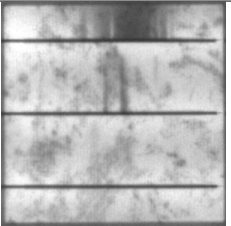
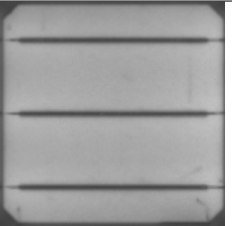
Class	True Positive	True Negative	False Positive	False Negative
Healthy				

Table 7: Some examples about True Positives, True Negatives, False Positives and False Negatives for each class of the First Classifier.


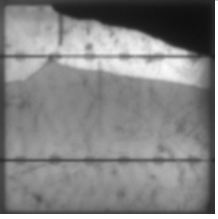
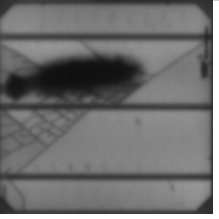
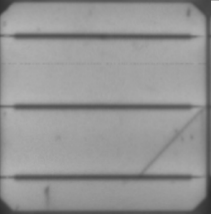
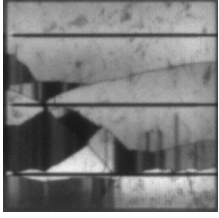
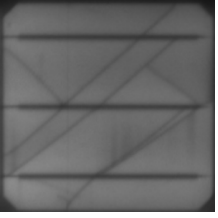
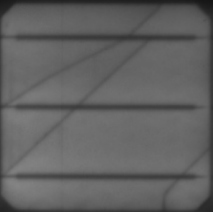
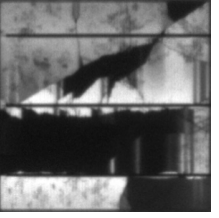
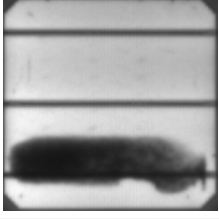
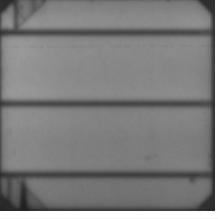
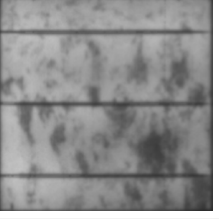
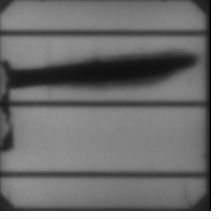
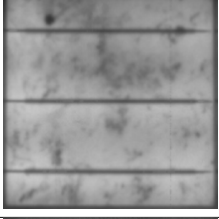
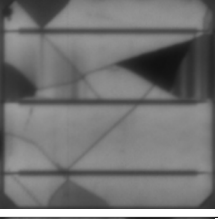
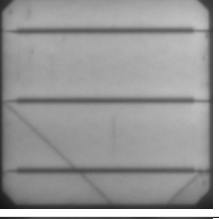
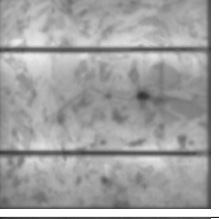
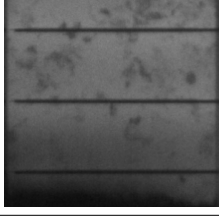
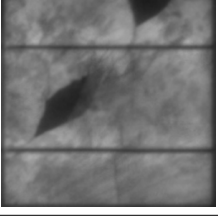
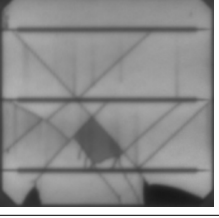
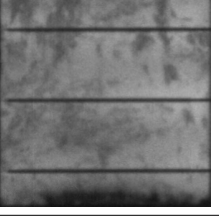
Class	True Positive	True Negative	False Positive	False Negative
Crack				
CellBreakage				
DarkArea				
HotSpot				
OtherDefect				

Table 8: Some examples about True Positives, True Negatives, False Positives and False Negatives for each class of the Second Classifier.

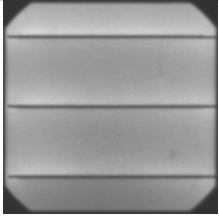
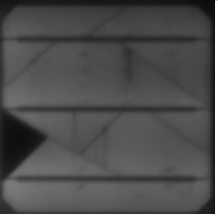
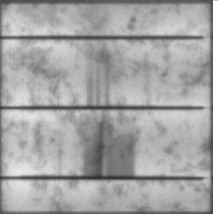
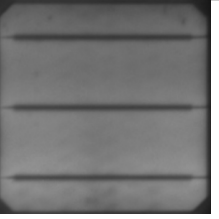
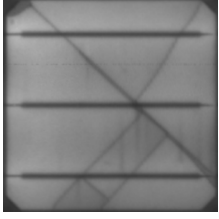
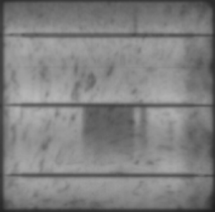
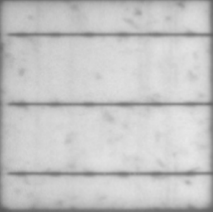
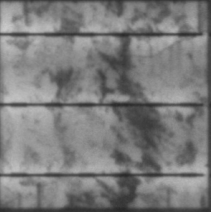

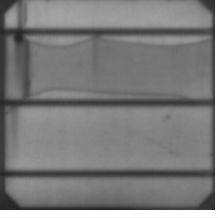
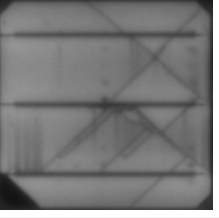
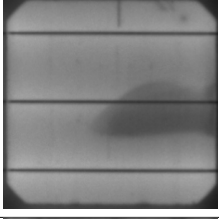
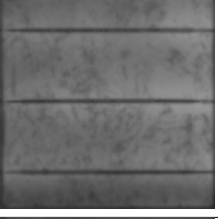
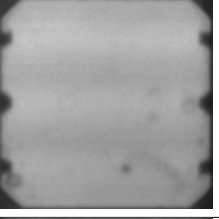
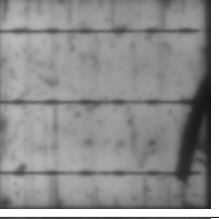
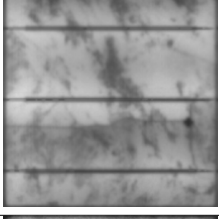
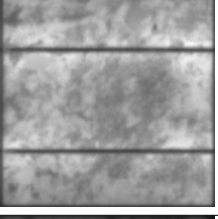
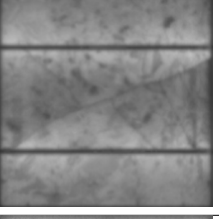
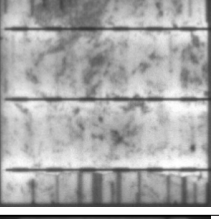
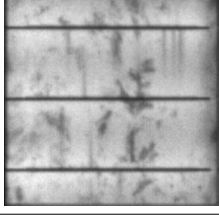
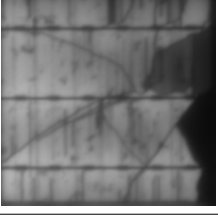
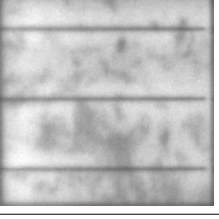
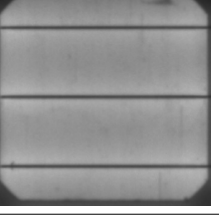
Class	True Positive	True Negative	False Positive	False Negative
Healthy				
Crack				
CellBreakage				None
DarkArea				
HotSpot				
OtherDefect				

Table 9: Some examples about True Positives, True Negatives, False Positives and False Negatives for each class of the Third Classifier.

### 3.3.10 Conclusions

Until now, the scientific community has concentrated its energies on finding techniques that will allow to automatically classify the health or the defect of a photovoltaic cell, without specifying the type of defect or, if it had been searched to classify the types of defects it would have been necessary to resort to private datasets, making the work not very transparent and replicable to researchers interested in this field of study. In fact the ELPV Dataset, that is the most famous public dataset of photovoltaic panel cells on which the most well-known works in the state-of-the-art are based, does not have specific labels regarding defects, indicating only a probability of defect. In our work we have extended this dataset by adding for each cell at least one type of defect among four common and well-known types, giving the scientific community the possibility of dedicating future energy to classify the various types of defects efficiently<sup>2</sup>. Just to give a starting point, we performed benchmarks using pre-trained neural networks. Initially, we tried to use two classifiers, the first to distinguish healthy cells from defective cells and the second to identify various defects in the cells marked by the first classifier as defective.

Subsequently, with the aim of making the discussion more complete and with more food for thought, we decided to implement another classifier that would automatically classify healthy cells and defective cells, indicating for the latter also the various defects present. By doing so, perhaps for the greater quantity of images taken together under examination, we have achieved concrete results, reaching a Hamming Distance of 0.07 with the RexNext50 network. Therefore, we can affirm: 1) Training a Healthy VS Not-Healthy binary classifier is better than training a multiclass classifier with the same architecture. 2) Adding the Healthy class to the other classes in the same training setting further improves the classification rate of the other classes.

Due to the lack of a common dataset and benchmark in the field of the classification of defects in solar cell, we could not compare our results with other works. Although this is a limitation for us, we have made a contribution to the scientific community with a new dataset and a benchmark on it that is a good starting point for future comparisons. As future works, we planned to apply on our task methods for defects detection in other industrial sectors such as Silicon technology.

## 3.4 Machine Learning Against Physics for Photovoltaic Thermal Models

In this section I will discuss the topic to which considerable time was devoted during the doctorate research in collaboration not only with EGP but also with the Department of Electrical, Electronic and Computer Engineering, that is the study of thermal models of photovoltaic panels with the in order to compare them with known regression algorithms applied on the same context [1].

---

<sup>2</sup><https://github.com/MarcoGrisanti/EELPV>

### 3.4.1 Introduction

Thermal models of photovoltaic panels are mathematical and computational tools used to describe the thermal behavior of such devices during their operation. Since temperature has a significant impact on the performance and efficiency of photovoltaic panels, thermal models play a vital role in understanding and optimizing their operation, which is influenced by several factors, including incident solar radiation, ambient temperature, wind speed and cooling rate. Overheating of photovoltaic panels can lead to a reduction in the efficiency of the conversion of solar energy into electricity and can cause premature degradation of materials, thus reducing the lifespan of the system. Thermal models can be of different levels of complexity, depending on the objectives of the analysis. The simplest models can be based on linear equations and simplified assumptions, while the more complex models can use partial differential equations and consider in detail the thermal behavior of each panel component. Thermal models are often used to estimate the operating temperature of photovoltaic panels under various environmental conditions, such as incident solar radiation, ambient temperature and wind speed. This information is critical for evaluating photovoltaic system performance and efficiency and for designing proper cooling devices or thermal management systems. Furthermore, thermal models can be employed to simulate and predict the thermal behavior of photovoltaic panels under different operating conditions, such as high solar irradiance or high temperature environments. This allows to identify potential overheating situations and optimize the design of the photovoltaic panels to maximize their efficiency and life. Thermal models of photovoltaic panels represent an essential tool for understanding and optimizing the thermal behavior of these devices, since thanks to them it is possible to evaluate the efficiency and reliability of the photovoltaic system, identify any overheating situations and design suitable thermal management systems. The use of advanced thermal models makes it possible to maximize the energy efficiency of photovoltaic panels, thus promoting the diffusion of renewable and sustainable energies in the global energy landscape.

As already mentioned, the degradation of photovoltaic modules over time is also influenced by the operating temperature and thermal cycling experienced by the modules. The estimation of the module temperature is therefore of crucial importance if it is not measurable due to the absence of sensors or for preliminary studies. In the present section, conventional models with machine learning regressors are compared for the estimation of the temperature of monofacial and bifacial photovoltaic modules installed in two different locations.

### 3.4.2 State of the Art

The work in [17] proposes a short-term temperature prediction of a Photovoltaic–Thermal module using the Principal Component Analysis (PCA) and a Radial Basis Function (RBF) Network. The simulation results show that the PCA method makes the prediction accuracy of the network model higher and the generalization performance stronger than that of the RBF Network without the main component extraction. The authors of [18] describe the estimation of the photovoltaic module surface temperature using a neural network based on measured ambient temperature and incident solar radiation. Implementing the Error Back Propagation algorithm, the overall accuracy reaches more than 96% of the measured value for clear and sunny days. The work in [19] develops a predictive model using an artificial neural network algorithm to determine the surface temperature of photovoltaic modules for a future period of time with a coefficient of determination between 0.96 and 0.97. The authors of [20] propose a novel temperature prediction method of photovoltaic panels with Support Vector Machine (SVM) for the temperature prediction problem in a complex environment. In order to optimize parameters of SVM, they use a Pigeon-Inspired Optimization method (PIO) with a Delay Factor (DF). Simulation results show that the DFPIO-SVM can obtain better predictive performance than SVM. The work in [21] presents an artificial intelligence-based approach for predicting the temperature of photovoltaic module in indoor and outdoor conditions. It uses a combination of two algorithms (Grammatical Evolution and Differential Evolution) in order to create a customized expression based on the Sandia model, in comparison to which it reduces the error by up to 11% in conditions of variability of sky over short time intervals.

### 3.4.3 Physical Models

This dissertation examines two physical models for calculating the temperature of photovoltaic panels: the Sandia Model and the Faiman Model.

**Sandia Model** The Sandia Model, also known as the Sandia Array Performance Model, is a widely used computational simulation model in the field of solar photovoltaic energy. It was developed by Sandia National Laboratories, a US research organization specializing in several areas, including solar energy. The Sandia Model is an advanced thermoelectric simulation tool that allows to evaluate the behavior and performance of large arrays of photovoltaic panels under different environmental conditions. The model takes into account various parameters, including incident solar radiation, ambient temperature, wind speed and the electrical characteristics of the photovoltaic panels. Such a model is useful for a wide range of applications in the photovoltaic industry, including the design and optimization of photovoltaic systems, the evaluation of the



expected energy yield for different system configurations and the performance analysis of large solar parks. The Sandia Model offers significant advantages over other simulation techniques as it is able to accurately account for thermal variations and the non-uniform distribution of solar radiation within an array of photovoltaic panels. This level of detail is particularly important for accurately predicting the efficiency and actual energy yield of a photovoltaic system, especially in real and complex scenarios. This model has significantly contributed to the development and optimization of solar photovoltaic technologies, enabling better plant design and implementation and providing valuable insights for investors and developers.

With regard to monofacial panels, using the Sandia Model, the temperature of the photovoltaic module  $T_M$  [°C] is calculated as follows:

$$T_M = T_E + G_F \cdot e^{a+bW_S} \quad (4)$$

where  $T_E$  is the ambient temperature [°C],  $G_F$  is the front plane of array irradiance [ $\text{W}/\text{m}^2$ ],  $W_S$  is the wind speed [m/s],  $a$  and  $b$  are two empirical coefficients.

With regard to bifacial panels, using the Sandia Model, the temperature of the photovoltaic module  $T_M$  [°C] is calculated as follows:

$$T_M = T_E + (G_F + cG_R) \cdot e^{a+bW_S} \quad (5)$$

where  $T_E$  is the ambient temperature [°C],  $G_F$  is the front plane of array irradiance [ $\text{W}/\text{m}^2$ ],  $G_R$  is the rear plane of array irradiance [ $\text{W}/\text{m}^2$ ],  $W_S$  is the wind speed [m/s],  $a$ ,  $b$  and  $c$  are three empirical coefficients.

**Faiman Model** The Faiman Model is an important computational simulation model used in the field of photovoltaic solar energy. It takes its name from its creator, Doctor Arza Faiman, a well-known scientist and researcher in the field of photovoltaic technologies. The Faiman Model was developed to predict the performance and efficiency of photovoltaic panels based on various operating and environmental factors. This model is based on mathematical and physical equations which describe the thermal and photovoltaic behavior of the panels, taking into account the solar radiation, the ambient temperature and the electrical characteristics of the device. The Faiman Model has been used extensively in the research and development of photovoltaic technologies, enabling a better understanding of solar panel performance and improvement of their efficiency. It is also a useful design and analysis tool for photovoltaic system engineers, designers and operators, who can use the model to predict and optimize the energy performance of their plants. Unlike

what was done for the Sandia Model, only the application of this model for single-sided panels will be considered.

Using the Faiman Model, the temperature of the photovoltaic module  $T_M$  [°C] is calculated as follows:

$$T_M = T_E + \frac{G_F}{U_0 + U_1 W_S} \quad (6)$$

where  $T_E$  is the ambient temperature [°C],  $G_F$  is the front plane of array irradiance [ $W/m^2$ ],  $W_S$  is the wind speed [m/s],  $U_0$  and  $U_1$  are two empirical coefficients.

#### 3.4.4 Machine Learning Models

The previous thermal models described, the Sandia Model and the Faiman Model, will be compared with two regression models widely used in the scientific literature: the Random Forest Regressor and the Gradient Boosting Regressor.

**Random Forest Regressor** The Random Forest Regressor is a machine learning algorithm used primarily for regression problems. It belongs to the family of Random Forest algorithms, which are based on Ensemble Learning. Random Forest algorithms are very versatile and can be used for classification and regression problems. The Random Forest Regressor works by creating a set of decision trees during the training phase. Each decision tree is constructed using a random subset of the training data and independent variables. Also, during the construction of each tree, a random selection of variables is made to divide the nodes of the tree. This process of random sampling and variable selection makes individual trees different thus reducing the risk of overfitting. Once the decision trees are created, the Random Forest Regressor combines their predictions into a weighted average to get the final prediction. Such a combination improves the accuracy and robustness of the predictions compared to a single decision tree, since each tree can make mistakes but combining their predictions tends to reduce the overall model error.

The Random Forest Regressor is widely used in different applications, such as financial forecasting and environmental forecasting. It is known to be an effective algorithm for handling large amounts of data and complex variables. Also, due to its robust nature and ability to avoid overfitting, it is often preferred for regression problems over other machine learning algorithms.

**Gradient Boosting Regressor** The Gradient Boosting Regressor, also known as Gradient Boosted Trees Regressor or Gradient Boosting Machine Regressor, is a machine learning algorithm used for regression problems. This model belongs to the family of Gradient Boosting algorithms, which are based on Ensemble Learning techniques similar to Random Forest. The operating princi-

ple of the Gradient Boosting Regressor is different from that of the Random Forest. While Random Forest build a set of decision trees in parallel, Gradient Boosting builds a sequence of decision trees iteratively. Initially, a first decision tree is trained using the training data. Subsequently, additional trees are trained to correct the errors of the previous model. Each new tree is trained to try to reduce the residual (i.e. the difference between the current prediction and the real value) of the current model. In this process, each successive tree focuses on the mistakes made by previous trees, creating a more complex and accurate model with each iteration. Finally, the predictions from all trees are combined to get the final prediction of the model.

The Gradient Boosting Regressor is very powerful and can produce highly accurate models. However, it is also more susceptible to overfitting than Random Forest. This algorithm is widely used in various applications such as calculating house prices and predicting system performance based on historical data. Due to its flexible nature and ability to adapt to complex models, the Gradient Boosting Regressor is a great choice for regression problems where high prediction accuracy is required.

### 3.4.5 Dataset

The various experiments that will be described in the next section were carried out on a dataset from the private research center Eurac and on a dataset obtained from EGP. For convenience, the first one will be called Eurac Dataset and the second one EGP Dataset. The Eurac Dataset belongs to a single-sided panel plant, while the EGP Dataset belongs to a double-sided panel plant. The latter will be divided into two datasets: in the first one (known as EGP Monofacial) only frontal irradiation will be present; in the second one (called EGP Bifacial) the frontal radiation and the rear radiation will be present.

**Eurac Dataset** The photovoltaic system relating to the Eurac Dataset was installed at the Bolzano airport in 2010. Various meteorological parameters are recorded such as irradiation, ambient temperature, module temperature, etc. The sensors are systematically cleaned and periodically calibrated to comply with a specific IEC standard. Weather data is logged with a one minute time interval. The dataset contains values for an 8-year period from February 2011 to January 2019. The dataset consists of 279489 rows and 10 columns. The latter are:

- Time
- Ambient Temperature [ $^{\circ}\text{C}$ ]
- Frontal Irradiance [ $\text{W}/\text{m}^2$ ]

- Wind Speed [m/s]
- Wind Direction
- Current (DC) [A]
- Voltage (DC) [V]
- Power (DC) [W]
- Power (AC) [W]
- Module Temperature [°C]

In order to adapt it to the experiments carried out, it was decided to "fix" the dataset with the following operations:

1. Elimination of useless columns. Only the following columns were kept:
  - Time (deleted after cleanup was complete)
  - Ambient Temperature [°C]
  - Frontal Irradiance [ $W/m^2$ ]
  - Wind Speed [m/s]
  - Module Temperature [°C]
2. Elimination of rows in which there is at least one null value;
3. Elimination of rows that do not refer to a specific year. In fact, it was decided to refer to the 2012 data;
4. Changing the sampling interval from 1 minute to 15 minutes for comparison with EGP datasets;
5. Elimination of the lines in which there is an Irradiance value that is too low.  $20 W/m^2$  is the lower limit;
6. Elimination of the lines in which there is an Irradiance value that is too high.  $1200 W/m^2$  is the upper limit.

Once these cleaning operations have been carried out, the dataset consists of 5930 rows and 4 columns.

**EGP Dataset** The EGP Dataset was created ad hoc using the data present in EGP’s office in Catania. The dataset contains values for one year with a sampling interval of 15 minutes. In the construction of the dataset, the limits relating to frontal radiation set in the cleaning of the Eurac Dataset were taken into account. The EGP Dataset Monofacial, consisting of 15866 rows, contains the following columns:

- Ambient Temperature [ $^{\circ}\text{C}$ ]
- Frontal Irradiance [ $\text{W}/\text{m}^2$ ]
- Wind Speed [ $\text{m}/\text{s}$ ]
- Module Temperature [ $^{\circ}\text{C}$ ]

Instead, the EGP Dataset Bifacial, consisting of 15488 rows, contains the following columns:

- Ambient Temperature [ $^{\circ}\text{C}$ ]
- Frontal Irradiance [ $\text{W}/\text{m}^2$ ]
- Rear Irradiance [ $\text{W}/\text{m}^2$ ]
- Wind Speed [ $\text{m}/\text{s}$ ]
- Module Temperature [ $^{\circ}\text{C}$ ]

### 3.4.6 Proposed Methods

Initially, we started by setting the default parameters of the various physical models listed in Table 10. Subsequently, it was decided to carry out Fine-Tuning of the various parameters with the aim of adapting them as much as possible to the photovoltaic systems under examination. The empirical coefficients obtained through Python optimization algorithms are reported in Table 11. For each problem the chosen algorithm was one of BFGS<sup>3</sup>, L-BFGS-B<sup>4</sup>, SLSQP<sup>5</sup>, depending on whether or not the problem has constraints or bounds.

Model	Eurac Dataset Params	EGP Dataset Params
Sandia Monofacial	$a = - 3.5600, b = - 0.0750$	$a = - 3.4700, b = - 0.0594$
Sandia Bifacial	/	$a = - 3.4700, b = - 0.0594, c = 1$
Faiman	$U_0 = 25.00, U_1 = 6.84$	$U_0 = 25.00, U_1 = 6.84$

Table 10: The default parameters for Sandia Model and Faiman Model regarding Eurac Dataset and EGP Dataset.

<sup>3</sup>Broyden–Fletcher–Goldfarb–Shanno (BFGS)

<sup>4</sup>Limited-Memory BFGS

<sup>5</sup>Sequential Least Squares Programming (SLSQP)

Model	Eurac Dataset Params	EGP Dataset Params
Sandia Monofacial	a = -3.3900, b = 0.1125	a = - 3.6500, b = - 0.0747
Sandia Bifacial	/	a = - 3.5790, b = - 0.0699, c = - 0.4080
Sandia Bifacial (Forcing c = 1)	/	a = - 3.7930, b = - 0.0768, c = 1
Faiman	$U_0 = 26.6500, U_1 = 6.0361$	$U_0 = 36.8300, U_1 = 3.8100$

Table 11: The fine-tuned parameters for Sandia Model and Faiman Model regarding Eurac Dataset and EGP Dataset.

After finding the optimal parameters of the physical models, it was possible to compare the real temperatures of the modules with the estimated temperatures of the modules through the calculation of metrics both for the physical models and for the regression algorithms. The metrics used are MSE, RMSE, MAE, MAPE, R2.

The metric Mean Squared Error  $MSE$  was calculated as follows:

$$MSE = \sum_{i=1}^N (\overline{T_{Mi}} - T_{Mi})^2 \quad (7)$$

where  $N$  is the number of samples;  $\overline{T_{Mi}}$  is the estimated temperature of the module;  $T_{Mi}$  is the real temperature of the module. It calculates the mean of the squared deviations between the model-predicted values and the true values of the dependent variable, providing a measure of the model's accuracy in predicting continuous numerical values. A lower value of the MSE indicates a higher accuracy of the model in its predictions.

The metric Root Mean Squared Error  $RMSE$  was calculated as follows:

$$RMSE = \sqrt{MSE} \quad (8)$$

where  $MSE$  is the Mean Square Error metric. It calculates the square root of the Mean Squared Error (MSE), providing a measure of the model's accuracy in predicting continuous numerical values. A lower value of the RMSE indicates a higher accuracy of the model in its predictions.

The metric Mean Absolute Error  $MAE$  was calculated as follows:

$$MAE = \sum_{i=1}^N |\overline{T_{Mi}} - T_{Mi}| \quad (9)$$

where  $N$  is the number of samples;  $\overline{T_{Mi}}$  is the estimated temperature of the module;  $T_{Mi}$  is the real temperature of the module. It averages the absolute deviations between the model-predicted values and the true values of the dependent variable, providing a measure of the mean discrepancy between the predictions and the observed data. A lower value of the MAE indicates a greater

precision of the model in its predictions.

The metric Mean Absolute Percentage Error  $MAPE$  was calculated as follows:

$$MAPE = 100 \frac{1}{N} \sum_{i=1}^N \left| \frac{\overline{T_{Mi}} - T_{Mi}}{\overline{T_{Mi}}} \right| \quad (10)$$

where  $N$  is the number of samples;  $\overline{T_{Mi}}$  is the estimated temperature of the module;  $T_{Mi}$  is the real temperature of the module. It calculates the percent mean of the absolute deviations between the model-predicted values and the true values of the dependent variable, providing a measure of the model's percent accuracy in its predictions. A lower MAPE value indicates a greater accuracy of the model in its predictions.

The Coefficient of Determination Error  $R^2$  was calculated as follows:

$$R^2 = \frac{\sum_{i=1}^N (T_{Mi} - \overline{T_{Mi}})^2}{\sum_{i=1}^N (T_{Mi} - m)^2} \quad (11)$$

where  $N$  is the number of samples;  $\overline{T_{Mi}}$  is the estimated temperature of the module;  $T_{Mi}$  is the real temperature of the module;  $m$  is the mean of all real temperatures of the modules. It measures the proportion of the variance explained by the model relative to the total variance of the data. A value close to 1 indicates a good model that explains the observed data well, while a value close to 0 indicates an unsuitable model for the problem at hand.

### 3.4.7 Results

Table 12 shows the statistical indices for the models applied to the modules of the Eurac photovoltaic plant. Instead, Table 13 and Table 14 show the statistical indices for the models applied to the modules of the EGP photovoltaic plant. The models with the prefixed empirical coefficients, as expected, gave less accurate results than both the same models in which the fine tuning was applied and the machine learning models. Taking into account the irradiance hitting the back of the module has improved the accuracy of the results. However, it is important to note that the improvement deriving from considering the back irradiance in the models, being small, could be compensated by a greater uncertainty when there is a small number of data or for climatic conditions different from those analyzed. Considering the different technology of the modules is important when the models of the present study are to be replicated.

Metric	Sandia	Sandia (FT)	Faiman	Faiman (FT)	RFR	GBR
MSE	13.43	10	10.3	10.11	7.85	7.44
RMSE	3.67	3.16	3.21	3.18	2.80	2.73
MAE	2.78	2.39	2.39	2.39	1.96	1.93
MAPE	0.26	0.29	0.30	0.30	0.18	0.21
$R^2$	0.95	0.97	0.97	0.97	0.97	0.97

Table 12: Statistical indices of physical and machine learning models applied to Eurac Dataset - Monofacial Polycrystalline - Bolzano. FT stands for Fine-Tuning. RFR stands for Random Forest Regressor. GBR stands for Gaussian Boosting Regressor.

Metric	Sandia	Sandia (FT)	Faiman	Faiman (FT)	RFR	GBR
MSE	15.32	6.19	7.40	6.22	5.90	5.87
RMSE	3.91	2.49	2.72	2.49	2.43	2.42
MAE	3.05	1.81	1.95	1.82	1.76	1.76
MAPE	0.12	0.07	0.08	0.08	0.07	0.07
$R^2$	0.89	0.96	0.95	0.96	0.96	0.96

Table 13: Statistical indices of physical and machine learning models applied to EGP Dataset - Bifacial mono-PERC, neglecting  $G_R$  - Catania. FT stands for Fine-Tuning. RFR stands for Random Forest Regressor. GBR stands for Gaussian Boosting Regressor.

Metric	Sandia	Sandia (FT)	Sandia (FT Forcing $c = 1$ )	RFR	GBR
MSE	30.36	5.77	6.38	5.20	5.61
RMSE	5.51	2.40	2.53	2.28	2.37
MAE	4.52	1.73	1.85	1.61	1.70
MAPE	0.17	0.07	0.08	0.06	0.07
$R^2$	0.79	0.96	0.96	0.96	0.96

Table 14: Statistical indices of physical and machine learning models applied to EGP Dataset - Bifacial mono-PERC, considering  $G_R$  - Catania. FT stands for Fine-Tuning. RFR stands for Random Forest Regressor. GBR stands for Gaussian Boosting Regressor.

### 3.4.8 Conclusions

Physical models were compared with two machine learning regressors for estimating the temperature of photovoltaic modules, applied to the case of monofacial and bifacial photovoltaic modules and located in two different parts of Italy. The use of machine learning regressors improved the accuracy of the temperature estimation compared to physical models.



The physical models that use empirical coefficients obtained through fine tuning on the data has allowed to have an accuracy of the results which, although it is close to that of the estimation results from the machine learning regressors, is always exceeded. Further studies will allow a generalization of the use of machine learning regressors for temperature estimation for different technologies and climate zones.

### 3.5 Papers

This section was based on the following scientific publications:

- Marco Grisanti, Maria Ausilia Napoli Spatafora, Alessandro Ortis and Sebastiano Battiato, "E-ELPV: Extended ELPV Dataset for Accurate Solar Cells Defect Classification", IntelliSys2023 - September 2023.
- Marco Grisanti, Gaetano Mannino, Giuseppe Marco Tina, Alessandro Ortis, Mario Cacciato, Sebastiano Battiato, Fabrizio Bizzarri and Andrea Canino, "Thermal Models of Monofacial and Bifacial PV Modules: Machine Learning and Physical Estimation Models Comparison", PVSC50 - June 2023.

## 4 A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images

In this section I will discuss the topic to which considerable time was devoted during the doctorate research, that is the test and evaluation about the performance of the method described in the research paper [22]. We did this by using a wide range of images from commonly used datasets that are meant for testing algorithms on 3D images in real-life situations. We didn't stop at just the technical evaluation – we also had real people participate in a subjective assessment of how well the reconstructed images looked. We had 116 individuals take part in this assessment and each of them reviewed 10 images. This resulted in a total of 1160 tests where we looked at the quality of the images, based on color information (HSV), from the datasets we considered [3].

### 4.1 Introduction

A stereoscopic image, or stereopair, is composed by a pair of images, named left and right views, taken at the same time on the same scene by two cameras from different points of view. The acquisition process aims to emulate the binocular view of the Human Visual System (HVS). In this sense, the distance between the two cameras is set to the distance between the human eyes' pupils. During the presentation of such images, proper filters and devices are used to let the left eye see only the left image and the right eye see only the right image. In this way, the viewer has the perception of a real 3D scene, as the two single views presented to the HVS are merged in the brain visual cortex.

The storage requirement for stereoscopic images is that least twice that of to the storage needed for a single image, hence the motivation for the study of stereoscopic compression techniques applications. The Multi Picture Object (MPO) is a standardized file format used to encode the Multi Picture Format for multi-view images defined by the Consumer & Imaging Products Association (CIPA) [23]. Such a format implements a chain of JPEG-encoded images properly embodied in a unique file (i.e., the MPO file) with a set of additional information useful to recover the single images and correlate them one each other in the context of the multi-view acquisition. In the specific case of stereoscopic images, only two images are encoded in the MPO file. However, the format allows the encoding of an unlimited number of views. Furthermore, the data needed to reconstruct the multi-view image from an MPO file are stored in the meta-data of the image, exploiting the JPEG metadata fields of the first image. As a consequence, the MPO format does not require additional fields or the change of data format.

The JPEG pipeline [24] and its extensions [25] represent a standard for the compression of digital

images, by allowing an optimized encoding of the image palette considering both quality and memory factors. Based on the JPEG encoding process, several works tried to optimize one or more elements of the pipeline. For instance, [26] proposed an alternative strategy to devise proper quantization tables. The experiments showed that the method was able to improve the compression performance of established JPEG compression schemes [24, 27].

MPO provides a standard for multi-view images, bringing several practical advantages. However, most of the state of the art works that address the problem of stereo image compression implement encoding techniques that do not take into account the standardization of the compressed images. Most of the existing approaches store one of the two views, the disparity map and a residual version of the other view [28, 29, 30]. Then, the residual image and the disparity map are used to restore the compressed view, with a certain degree of confidence. This approach needs a method to store the additional data, such as the disparity map and the residual image. In the majority of cases, an entropy-based coding is employed for the disparity map, whereas Discrete Cosine Transform (DCT) is used for the residual image. In particular, the work in [30] proposes the estimation of the stereoscopic disparity map from the local 1D-Fast Fourier Transform (FFT) computed on the left and the right image. Approaches based on the disparity map pose problems in the reconstruction. In particular, although gradual transition is usually observed in disparity maps, object edges may produce abrupt change of the parameter. This will cause two main issues: pixels without any assignment (i.e., Falling Disparity) and matching problems due to double assignments [30]. For this reason, in addition to the left image, the right image and the disparity map, the approach in [30] defines an error image to the encoding pipeline of a stereopair.

Schenkel et al. [31] proposed a joint decoding approach of the two views for the compression of a stereopair. The proposed method then performs an enhancement of the image pairs, previously compressed using the JPEG pipeline. However, some areas of the images cannot be reconstructed. Moreover, the experiments shown that with middle values of the JPEG quality compression, decreases in terms of Peak Signal to Noise Ratio (PSNR) are observed and some ghosting artefacts appear.

The work in [32] proposes a variable size-block coding algorithm for stereoscopic images that jointly optimizes the block sizes and the quality of the disparity map computed from the compressed images. In particular, the system applies a fine-grained pixel blocking on the image areas with more detail, used to encode the disparity map. This approach has been designed with the aim to obtain an high quality of the reconstruction while reducing the bit-rate of the stereopair. Although the experimental results achieved in [32] are promising, it requires an overhead of computational effort due to the block layout definition for the encoding, as well as additional data consisting

of the tree structure describing the blocking layout, the block-length map and other information needed for the stereo pair decoding. All these data structures need to be stored properly with an ad hoc file format, besides the image payload.

In [33], Poolakkachalil et al. presented an approach for symmetric stereoscopic image compression followed by an arithmetic coding named Stereoscopic Image Compression using Curvelet and Arithmetic Coding (SICCAC), which is mostly based on the still image compression method proposed in [34], that applies a curvelet transform for image encoding. This work has been further extended in [35], which proposes an encoding pipeline in which the difference and the average of the two stereo images are quantized and then encoded exploiting an adaptive arithmetic coding (S2ICAC). Other methods exploit entropy-based coding to encode the stereoscopic images, such the method evaluated in [36], also known as Stereoscopic Image Compression using Huffman Coding (SICHC). Most of the above mentioned works assume that the stereopair has been acquired using a parallel camera optical axis stereo system, where the convergent axis setting is ignored. The camera optical axis convergence involves a perspective effect that generates a vertical parallax. With parallel camera optical axis the vertical parallax is removed. Approaches that try to completely reconstruct one view from the other views do not take into account that a perfect reconstruction is not possible, as discussed in [30]. Moreover, in real scenes, there are areas on each image that are not visible in the other image. When the problem of stereoscopic image compression is considered in the context of real and high quality images, approaches that ignore parts of the scenes cannot be considered; issues related to standardization of the storage (i.e., File) and transmission formats also have to be addressed.

The work in [37] introduced an end-to-end deep neural network for stereoscopic image compression. In particular, the homography  $H$  between the left and right image is estimated, then the left image is mapped to the right view by exploiting  $H$  and the residual image is stored. The proposed architecture includes a deep regression model to estimate the homography matrix, two autoencoders and cross-quality enhancement (CQE) network. Although the method in [37] achieves interesting results, it requires very high resources, in terms of data quality and quantity, computational time and specific hardware. Moreover, it forces the input images to a specific resolution as well as introducing a bias related to the training datasets, which are divided among close views (InStereo2K) [38] and far-views (KITTI) [39] stereopairs. In particular, the method in [37] exploits a deep neural network to estimate the  $H$  matrix, which could be simply inferred by using traditional geometry-based methods which only require a few correspondences between the left and right views. During encoding, the left image is geometrically transformed by means of  $H$  and then the two images are further processed through two separate autoencoders. Such a processing breaks the geometric

constrains which are then inferred during the encoding step. For this reason, the CQE network is needed. The authors proposed two deep models, the first requires 50.5 M parameters and the second 69.3 M. The models have been trained on two datasets of high quality stereoscopic images [38, 39] including about 2000 examples each.

The authors of [40] presented a preliminary experiment on the exploitation of stereoscopic image redundancy to reduce the bitrate of stereopairs. In particular, the paper combines two algorithms. One performs better at low and mid-range bitrates, the other at mid and high-range bitrates. One iteratively modifies the disparity map to improve the bitrate-distortion trade-off using a Lagrangian multiplier. The other selects each disparity on a block basis, according to a simplified model of how JPEG deals with the compensation refinement. However, the main contribute of [40] is theoretical, indeed the proposed approach has only been tested on two stereoscopic images.

In [41], the authors presented a strategy for MPO image compression that significantly reduces the space needed to store a stereopair with very low quality loss. One of the main advantage of this method is that the compression phase allows us to obtain an MPO compliant compressed file (see Figure 6). The decoding phase, through a proper restoration phase, reconstructs the original information after the MPO standard decoding of the two images. The paper in [22] presents an improvement of the work in [41] drastically reducing the overall reconstruction phase complexity, while keeping the same reconstruction quality. Compared with respect to previous methods, the methods in [41, 22] are explicitly designed for the MPO format, formalizing a proper coding/decoding pipeline that can be implemented directly on acquisition devices. Therefore they support the standardization and work indifferently on stereoscopic images acquired using both a parallel or a convergent stereo camera system. However, the experiments have been performed on a reduced number (i.e., 23) of selected stereopairs. The two methods achieved the same performances in terms of reconstruction quality; however, the approach presented in [22] drastically improves the method in [41] in terms of computational costs.

In this work, we perform an extended benchmark evaluation of [22] on a large set of images taken from standard and well-known datasets designed for evaluations of algorithms on stereoscopic images in real scenarios. In addition, we present the results of a subjective assessment of the reconstructed image quality, which involved 116 participants who evaluated 10 images each, producing a set of 1160 HSV-based tests on randomly selected images from the considered datasets. Then, we also evaluated the method in [22] on the dataset used in [33, 35].

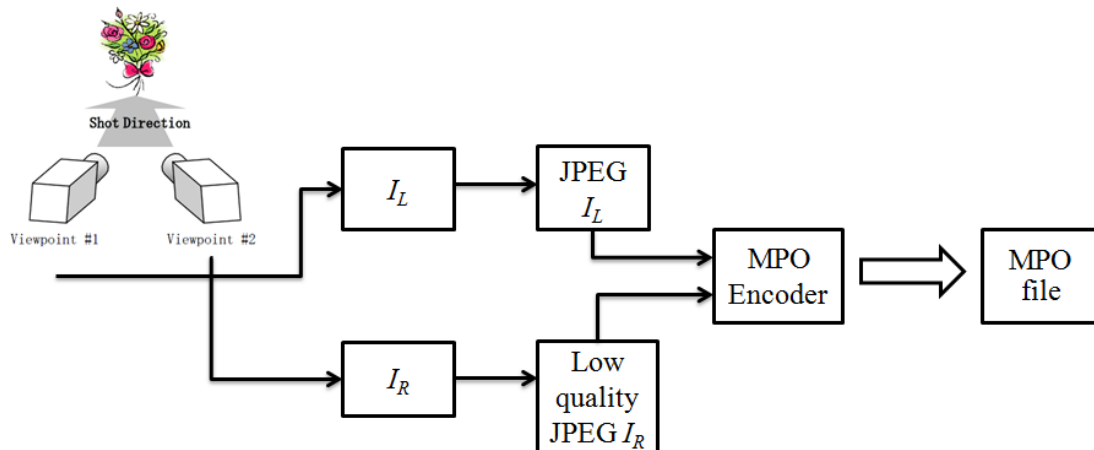


Figure 6: Encoding pipeline. Each view is coded according to the MPO format (i.e., by applying the JPEG compression) using different quality factors.

Section 4.2 clearly states the motivations of the presented benchmarking, by comparing pros and cons of the state-of-the-art methods. The employed coding/decoding pipeline is detailed in Section 4.3, comparing the differences between the approaches proposed in [22, 41]. The two approaches are explained in detail and compared in terms of computational complexity. Section 4.4.1 presents an experimental evaluation of the adaptive stereoscopic image compression approach, considering 16 different experimental settings on 60 stereopairs taken from public standardized datasets with different resolutions and acquisition settings (i.e., parallel or convergent cameras). It also presents a comparative evaluation with other MPO-compliant methods, as well as an HVS-based experiment aimed to evaluate the perceived quality of the reconstructed images. Section 4.5 concludes the work.

## 4.2 Motivations

In the context of multi-view pictures, the MPO format is an established standard by years, currently adopted for storing stereoscopic and 3D pictures taken with a photo camera equipped with multiple lens system by several device producers, including game consoles. In this context, the methods proposed in [33, 35, 41, 22] define encoding/decoding pipelines which maintain the compatibility with MPO format and are independent from the acquisition settings (i.e., parallel/convergent axis, image resolution, etc.). As discussed in the previous Section, other methods often require additional payload to store the stereopair and ad hoc data format [28, 29, 30, 31], or present ghosting artefacts [31].

Standardization of image encoding/decoding pipelines brings several advantages, especially when

such methods are embedded on the acquisition/rendering devices, requiring high performances with limited resources. Methods based on very Deep Neural Networks (DNN) requiring millions of parameters and complex non-linear operations cannot be embedded on such devices [37]. Moreover, DNNs have millions of parameters, each with complex inter-relationships. In this way, Deep Learning models have been criticised to be a black-box, in contrast with deterministic and explainable geometrical based approaches which have full transparency and allow one to directly observe whether the achieved solution will work outside of a training environment [42]. Indeed, traditional Computer Vision techniques are often preferred over DNNs in a range of applications from reducing training time, processing and data requirements, in particular to be applied on geometrical related fields [42] (e.g., structure from motion, Panoramic-Stitching, etc.).

The method assessed by our benchmarking evaluation has been specifically designed for MPO images, as well as the methods considered for the comparative evaluation. All the compared approaches are MPO-compliant and can be easily embedded within low-resource devices as they re-design the encoding-decoding pipeline already implemented in such devices. Moreover, these methods are independent from the stereo acquisition settings (i.e., parallel or converging camera axis). However, the experiments presented so far were limited in quantity, quality and variability of the images. Moreover, no comparison with respect to other methods were presented, as well as a subjective assessment of the perceived image quality. Given the above, we realized need for a standardized, large-scale, benchmark evaluation. The benchmark evaluation presented in this work includes:

- Experiments on 60 stereopairs of the Middlebury-Scenes datasets including versions between 2001 and 2014;
- Images which resolution variates from  $375 \times 450$  to  $2016 \times 2960$ ;
- A total of 16 different evaluation settings, by combining different feature detection and geometry estimation;
- Comparative evaluation with other five methods published more recently, taking into account both bitrate saving and reconstruction quality, on the same publicly available dataset;
- Subjective assessment conducted with high number of tests and high participant population variability confirmed that the reconstructed image is indistinguishable from the high quality one.

### 4.3 Evaluated Pipeline

The following paragraphs present the encoding/decoding strategy. In particular, the two different matching approaches implemented in [22, 41] are described in detail and compared in terms of computational complexity.

#### 4.3.1 Encoding Pipeline: Asymmetric Compression

In the encoding phase, the proposed method encodes one image view with a low JPEG quality factor [24]. During encoding, the redundancy between the two images is exploited to enhance the low quality image by using the high quality one as reference. The enhancing process compares image blocks properly extracted from the two views. In the image areas in which no a reliable level of redundancy is detected (i.e., high difference between low and high quality images), only the information from the low quality patch is considered. As a consequence, there is a certain level of lossy on the resulting enhanced view. However, experiments show that this losing rate is numerically negligible and not perceptible visually. The JPEG encoding pipeline defines a quality level to control the amount of compression. A low-quality image results in a smaller JPEG file, whereas a high-quality image produces a relatively large file. The quality level determines the quantization tables used during the JPEG encoding pipeline, these tables control the amount of loss during the compression and hence the size of the generated file. Therefore, the quality level directly affects the visual quality of the image and the file size.

The amount of JPEG compression is typically measured as a percentage of the quality level. In general, quality levels of 90% or higher are considered high quality images, 80–90% is medium quality and 70–80% is low quality. Images compressed with quality values below 70% are typically a very low quality. With such quality levels, edges are no longer sharp and compression artefacts are visible. For these reasons in our experiments we compressed the low quality image considering quality levels equal to 70% and 65%.

Figure 6 details the proposed encoding pipeline in which the two image views are encoded differently. Each image block of the low quality view  $I_R$  is then reconstructed by exploiting the high quality view  $I_L$ . In particular, we conventionally encoded the left image  $I_L$  with an high JPEG quality rate and the right image  $I_R$  with a low quality rate, but the same pipeline can be applied by inverting the role of the right and left views.

#### 4.3.2 NCC-Based Decoding Approach

The approach proposed in [41] implements a image blocks matching method based on the correlation between image patches. A common way to match a given pattern  $t$  within an image  $I$  is



to consider the Normalized Cross Correlation (NCC) score  $ncc(u, v)$  computed at each possible position  $(u, v)$  of the template  $t$ , which has been shifted by  $u$  and  $v$  steps in the  $x$  and  $y$  direction, respectively. The NCC coefficients are defined as follows:

$$ncc(u, v) = \frac{\sum_{x,y} \{[I(x, y) - \bar{I}_{u,v}][t(x - u, y - v) - \bar{t}]\}}{\{\sum_{x,y} [I(x, y) - \bar{I}_{u,v}]^2 \sum_{x,y} [t(x - u, y - v) - \bar{t}]^2\}^{0.5}} \quad (12)$$

where  $\bar{I}_{u,v}$  is the mean value of the pixels  $I(x, y)$  located within the area of the template  $t$  shifted by  $(u, v)$  and  $\bar{t}$  is the average value of the pixels of  $t$ .

At decoding time, both right and left views are subdivided into not overlapping patches of size  $N \times M \times 3$ . For each block, the decoding procedure exploits the redundancy between blocks extracted from the left and the right image with the aim to enhance the quality of the blocks extracted from the low quality one. The decoding algorithm is applied on each block extracted from each channel.

**Matching Approach** Given the generic  $i^{th}$  extracted from the low quality image  $I_R$ , named  $b_i^R$ , the objective is to find the best sub-image of the high quality image  $I_L$ , which redundancy can be exploited to enhance  $b_i^R$ . To this aim, the algorithm considers two candidate blocks:

- The  $N \times M$  block of  $I_L$  which is located at the same position of  $b_i^R$ ;
- The  $N \times M$  block of  $I_L$  obtained by computing the Normalized Cross Correlation (NCC) [43] between  $b_i^R$  all the  $N \times M$  sub-images of  $I_L$  and considering the sub-image with the highest NCC value.

In practice it is not needed to compute the NCC for every possible position of  $b_i^R$  in  $I_L$ . An optimized approach would compute the NCC in a restricted area of  $I_L$  taking into account the original position of  $b_i^R$ .

The procedures then select the candidate block which minimizes the Sum of Absolute Differences (SAD) with  $b_i^R$ . The selected block is then exploited by the enhancing procedure described in Section 4.3.4.

### 4.3.3 Geometry-Based Decoding Approach

The approach presented in [22] extends the work in [41] by focusing on the improvement of the matching phase efficacy and the optimization of its computational costs. These two objectives have been obtained by leveraging on the geometric constraints of a stereoscopic pair.

**Epipolar Geometry** Epipolar geometry describes the properties and the geometrical relationships between two images that describe the same 3D scene in a stereoscopic image. Such geometry is independent of scene structure, as it is fully described by the parameters of the two cameras and their relative locations [44].

**Definition 1 (Fundamental Matrix)** *The fundamental matrix, denoted as  $F$ , is a  $3 \times 3$  matrix of rank 2 that, for any pair of corresponding points  $x_1$  and  $x_2$  satisfies the following condition:*

$$x_1' F x_2 = 0 \tag{13}$$

where  $x_1$  is a point of one view which corresponds to  $x_2$  in the other image view.

The fundamental matrix encodes the intrinsic geometry of the acquisition setting (i.e., the relative position and orientation of the two cameras) with respect to the scene. Indeed,  $x_1$  and  $x_2$  in Equation (13) correspond to the projection of the same real 3D point on the two camera image planes. Therefore, given two points taken from the left and right views of a stereoscopic image depict the same real-world point, they must satisfy the relationship in Equation (13). This property has been exploited by the method proposed in [22] to drastically reduce the search range of correspondences between the left and the right views. Hence, to reduce the computational cost during the matching phases.

Indeed, according to the point line dualism theorem, given a stereoscopic image, for any point  $x_1$  in the first view, there exists a corresponding epipolar line  $l_2$  on the second view. Moreover, any point  $x_2$  in the second view matching  $x_1$  lies on  $l_2$ . Indeed,  $l_2$  is the projection of the ray from the point  $x_1$  to the second view, which passes through the first camera's centre. We can map  $x_1$  to its corresponding epipolar line  $l_2$  on the second image by exploiting the following Equation:

$$l_2 = F x_1. \tag{14}$$

In other words, the fundamental matrix  $F$  allow us to have a direct relationship between a given point  $x_1$  in the first image and its corresponding epipolar line  $l_2$  in the second view, which contains the corresponding point  $x_2$ . Therefore, the search of the point  $x_2$  can be limited to the points in  $l_2$ .

**Image Blocking** In the decoding approach described in Section 4.3.2, the low quality view is subdivided into a number of not overlapping blocks. The block size is selected adaptively depending on the input image dimensions with the aim to cover all the image area, resulting in a few number of rather big blocks. With such block dimensions, each image is subdivided in about 12 blocks.

Such a low number of blocks allows the execution of the decoding approach in a reasonable time. Then each block extracted from the low quality image is compared with the high quality one using a template matching approach based on the NCC computation to detect the most similar high quality area.

This approach presents several limits. Due to the high dimensions of the blocks, each extracted image patches depict a rather big area of the scene (e.g., with 12 blocks in total, each block includes more than the 8% of the image), including different objects and several details of the pictured scene. Due to the differences between the point of views of the two cameras, each object appears slightly different in the resulting left and right views. Such differences augment with the dimensions of the extracted blocks, as more objects are included in the corresponding patches, especially in cluttered scenes. Moreover, the procedure is repeated for each image channel. As a consequence, the matching procedure results coarse and computationally expensive.

In contrast, the method presented in [22] subdivides the images considering very few blocks. All blocks have the same dimensions and the matching procedure is applied simultaneously to all the colour channels of the block. In particular, a radius  $r$  is set to a constant value ( $r = 20$  in [22]), then the low quality image is partitioned into  $(r + 1) \times (r + 1)$  not overlapping blocks. If needed, to cover all the image area, some overlapping blocks are defined in the right and in the bottom part of the image. An example of block definition is shown in Figure 7. In this example, the image “Cones“ of the Middlebury 2003 dataset [45] is subdivided into 110 blocks, including a number of overlapping blocks (depicted in blue). Reducing the dimensions of the image blocks allows a fine-grained search of the processed data, thus avoiding working with big matrices of coarse pieces of the scene, where several objects’ positions with respect to the camera system can be very different. As an example, considering the same image, the method used in [41] defines 12 blocks with size  $360 \times 360$ , whereas the method in [22] extracts 972 blocks and is computationally more efficient.

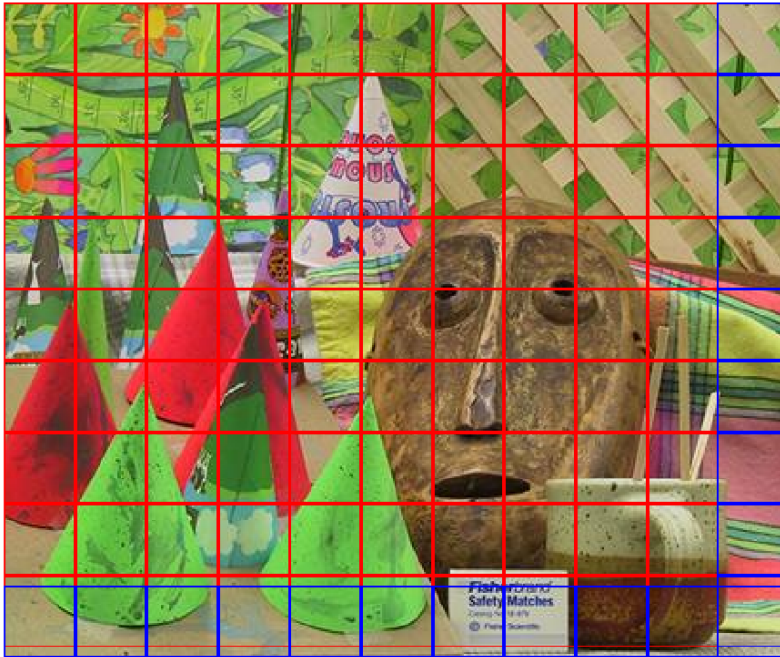


Figure 7: Example of blocking schema employed in [22].

**Matching Approach** The main improvement of the geometry based approach is related to the matching strategy. In the following, we conventionally consider the left view as the high quality image, whereas the right view is considered as the low quality one. Given a block  $b_i$ , extracted from the right view, the system considers the block's centre  $c_i$  and computes the corresponding epipolar line  $l_i$  on the left view by applying the point-line relationship defined by Equation (14). According to the epipolar geometry (see Paragraph 4.3.3), we know that the unknown point that corresponds to  $c_i$  on the left image lies on  $l_i$ . Therefore we can limit the matching of the whole block  $b_i$  to the possible positions of  $b_i$  on the left image obtained by shifting the block centre  $c_i$  along the epipolar line  $l_i$ . Therefore, the main benefit is given by the fact that the matching of the image block is limited to a set of possible positions of  $c_i$  on the epipolar line. The number of such positions is approximately equal to the image width; however, as detailed in the following paragraphs, the employed approach further reduces the search range. The epipolar lines related to the extracted blocks' centres are computed by exploiting the Equation (14); therefore, the estimation of the fundamental matrix  $F$  is needed.

The matrix  $F$  can be estimated starting from a number of correspondences between the left and the right views. Indeed, each pair of matching points between the two images provides a linear constrain on  $F$  (i.e., Equation (13)). As consequence, the fundamental matrix  $F$  can be estimated linearly from at least eight independent pairs applying the eight point algorithm [46]. The eight

point algorithm is simple and effective; however, it is sensible to the precision of the input correspondences. An alternative is represented by the Least Meadian of Squares estimation, which is robust to the presence of wrong correspondences, but requires a number of good pairs of matching points equal or greater than 50% of the input data.

Both the eight point algorithm and the Least Meadian of Squares estimation require a number of input correspondences. These pairs of matching points have been computed considering two different approaches for local features detection, namely the SIFT (Scale-Invariant Feature Transform) [47] and Harris keypoints [48]. Figure 8 shows an example in which three epipolar lines (Figure 8a) have been computed from three sample points taken from the right image (Figure 8b) by applying Equation (14). The points drawn on the left view are the points in the epipolar line which has the same  $x$  of the sample points taken from the right view. Indeed, is possible to note that they do not corresponds to the three sample points of the right view. Figure 9 shows the detail of the points number two and three: these points are placed on the vertex of the cone (point two) and the tip of the masks' nose (point three) in the right view (Figure 9b), their epipolar lines pass correctly by the corresponding points in the left image (Figure 9a). The points with the same  $x$  coordinates which lies on these lines (reported on the left image) appear shifted with respect to the corresponding ones. This observation suggest us that, given a point of the right view, beside from the fact that its corresponding point on the left images lies on the epipolar line (i.e., Equation (14)), the position of this unknown point is close to the point with the same  $x$  of the point selected in the right view.



Figure 8: Considering any point in the right image (b)  $I_R$  (1, 2 and 3), the method in [22] searches the corresponding points in the left image  $I_L$  by computing their epipolar lines (a).



Figure 9: Details of the images shown in Figure 8. The tip of the cone (i.e., point number 2 in the image (b) depicting the right view  $I_R$ ) corresponds to a point that lies on the corresponding epipolar line (i.e., line number 2 in (a)) on the left view  $I_L$ .

**Range Reduction** The estimation of the epipolar line reduces the searching range to a single line. Indeed, given an image block  $b_i$  is possible to estimate the epipolar line of its centre  $c_i$  and perform the block matching considering only the possible patches which centre lies on  $l_i$ . Nevertheless, the number of comparisons can be further reduced by observing that:

1. The  $y$  coordinate of the searched point is given by the  $y$  values of the epipolar line;
2. The  $x$  coordinate of the searched point is close to the  $x$  of the correspondent point on the

right view;

3. Equation (13) must be satisfied by any pair of corresponding points.

With the aim to minimize the searching range, the following approach has been designed in [22]. Given a point  $x_1$ , centre of a block extracted from the right view, the corresponding epipolar line  $l_1$  is computed by exploiting Equation (14). Then, for each point  $x_2 \in l_1$ , the value of  $x_1'Fx_2$  is computed. Equation (13) states that if the corresponding point of  $x_1$  is represented by the left view, the values of  $x_1'Fx_2$  represents a zero crossing line. Furthermore the corresponding point of  $x_1$  lies around the point where the the value of  $x_1'Fx_2$  is zero. In order to further focus the matching search to this region, the following function is defined:

$$\phi = (x_1'Fx_2)^2 \quad (15)$$

The function  $\phi$  defines a parabola. In particular, the vertex of  $\phi$  is close to zero. An example of  $\phi$  is shown in Figure 10. Is possible to observe how the value of  $\phi$  grows rapidly with the distance from its vertex. Moreover, the point with the same  $x$  coordinate of  $c_i$  (represented by a red circle in Figure 10) is placed nearby the vertex.

The reduction of the matching search range is finally obtained by applying a threshold to  $\phi$ . In the experiments, it has been observed that the value 2.3 permitted to drastically remove the number of candidate blocks (see Figure 10). However, instead of using a fixed threshold heuristic, an adaptive method has been applied. It is based on the following adaptive rules:

1. The threshold value is set to 2.3;
2. If the point with the same  $x$  as  $c_i$  is not included by the range obtained after the thresholding, according to the previous considerations, the threshold is augmented until this point is included in the search range;
3. If the search range is empty, the threshold is augmented iteratively by 0.5 until the range is not empty.

These rules allowed the definition of an effective adaptive thresholding strategy.

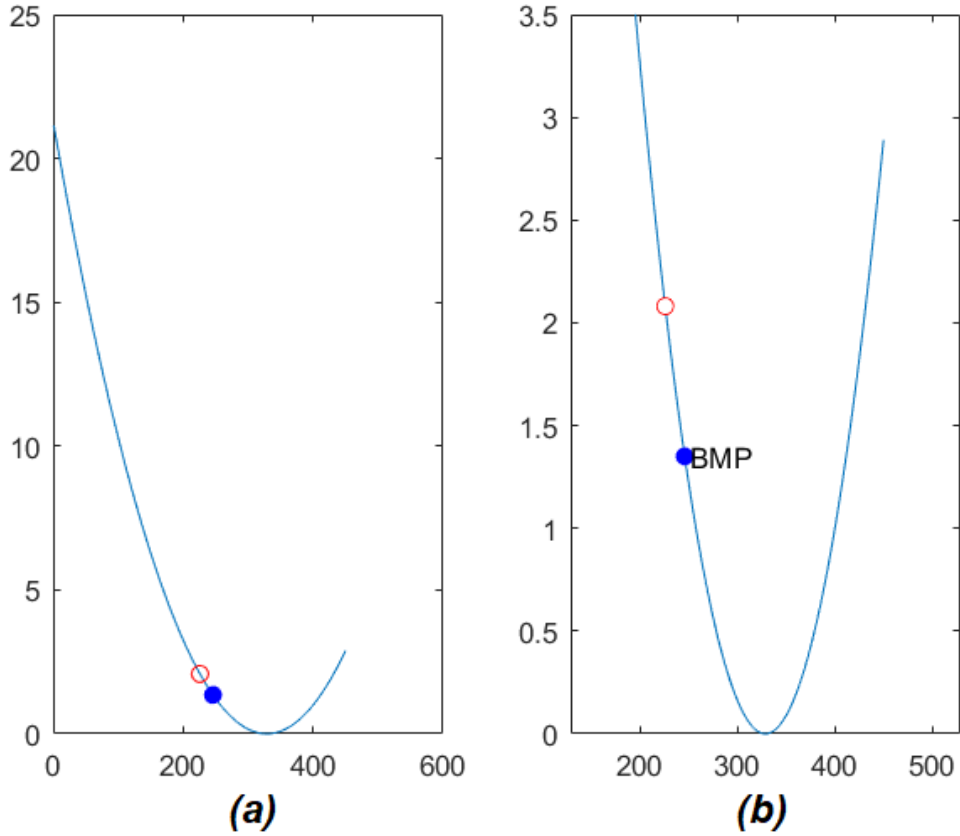


Figure 10: Example of matching by using the approach in [22] (a) and detail near the vertex of the  $\phi$  function (see Equation (15)). The red circle is the point on  $\phi$  with the same  $x$  as the point on the (b). The blue point represents the value of  $\phi$  corresponding to the Best Matching Point (BMP) according to the algorithm.

**Block Matching** Due to the approach described in Paragraph 4.3.3, the search range is drastically reduced. Then, given a block  $b_i$  of the right view, the matching procedure searches the best matching block of  $b_i$  considering the candidate blocks such that the centres lie in  $l_i$  and the  $x$  coordinate is in the reduced range. These constraints allow one to significantly reduce the number of comparisons between blocks. The blocks are compared by computing the Sum of Squared Differences (SSD) between  $b_i$  and the selected candidate blocks. The block which achieves the lowest SSD is then employed during the enhancing phase (see Section 4.3.4).

Experiments revealed that in presence of uniform blocks the above described procedure could include false positive examples and cause the failure of the matching procedure. In these cases, if the threshold is high, the best matching block in terms of SSD could be placed far from the theoretical position computed considering the geometrical properties. In other words, if an image contains a large uniform area (e.g., a wall), the image blocks extracted from such area could match



one each other because there are not enough edges that characterize the selected image patches and hence allow the SSD to catch the differences between mismatching blocks.

Considering that the JPEG compression tends to preserve low frequencies, the method in [22] discards the uniform blocks from the enhancing procedure, by filtering out the blocks with low pixel variance. This strategy allows to further reduce the number of comparisons (i.e., SSD computations) and contextually to avoid the matching issues caused by the presence of uniform candidate blocks. Experiments shown that the results are not affected by this approach in terms of quality, while time performances are significantly improved. Therefore, this simple choice allowed to maintain the quality while the matching process is further sped up.

**Partial Matching** In could happen that some areas of one image view, located nearby the border of the image, correspond to blocks of the other image view that are just partially visualized, due to the difference of the camera point of view and orientation. This happens when some parts of the scene are represented only by one of the two views. To address with the problem of partial matching, the method in [22] allows candidate blocks with lower dimensions. In particular, the procedure include all the blocks whose dimensions are equal or greater than 60% of the processed block dimensions.

To compare  $b_i$  with partial candidate blocks, the missing pixels are filled using the values of  $b_i$ . Figure 11 shows some examples of partial matching and block compositions. In particular, the first row represents an example taken from the stereopair "Flowers 1" in which the block number 111 is partially matched, whereas the second and third rows are related to the stereopair "Cones". The first row of Figure 11 shows that a chandelier is visible in the right view of the stereopair (i.e., column (a) of the first row in Figure 11) and it is depicted in the block  $b_{111}$ , considering the blocking schema described in Paragraph 4.3.3. This object is only partially depicted in the left image view. When the algorithm performs the matching procedure for the block  $b_{111}$ , which represents the part of the chandelier shown by both images (Figure 11a), it takes into account also a partial matching with a  $41 \times 25$  block placed on the left side of the reference image. The blue area in Figure 11b represents the missing part within the image patch, which are then filled with the pixels of  $b_{111}$  in the same positions, obtaining the composed block shown in Figure 11c.

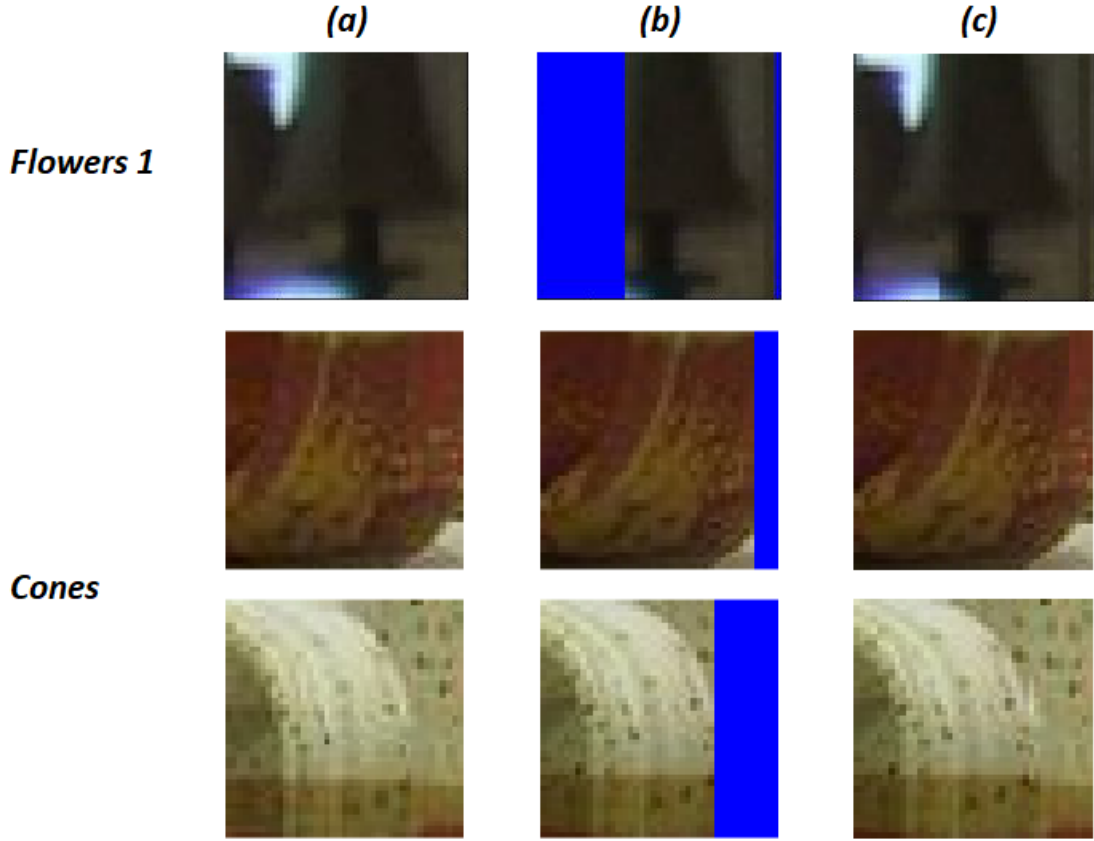


Figure 11: Examples of block composition performed to overcome with partial matching blocks. Each row shows an example of block composition, the first row shows the block number 111 (a) of the image "Flowers 1", the partial matching block (b) and the composed block (c). The second and third rows are examples of partial matching of blocks extracted from the image "Cones", related to the blocks number 17 and 13, respectively.

#### 4.3.4 Image Enhancing

The enhancing step employs the following equation, which is based on a simplified version of Kohonen update rule [49]:

$$\bar{b}_i^R(u, v) = \begin{cases} b_i^R(u, v) + \alpha \cdot d_i(u, v) & \text{if } d_i(u, v) < th \\ b_i^R(u, v) & \text{otherwise} \end{cases} \quad (16)$$

where

$$d_i(u, v) = b_i^R(u, v) - b_i^L(u, v) \quad (17)$$

where  $\bar{b}_i^R(u, v)$  is the enhanced sample,  $b_i^L(u, v)$  is the sample of the block selected by the matching procedure and  $b_i^R(u, v)$  is the sample which has to be enhanced. By applying Equation (16), the

values of some samples of  $b^R$  are moved closer to the corresponding values in  $b^L$ , depending of the similarity between the pair of correspondent samples. The parameters  $\alpha$  and  $th$  (in our case  $alpha = 0,25$  and  $th = 0043$ ) are two coefficients that control the reconstruction procedure [49] based on the distance between the two samples values defined as  $d_i(u, v)$  in Equation (17). The parameters in Equation (16) have been empirically obtained by grid-search evaluation previously performed [41, 22] on the 23 MPO stereoscopic images from the 3DMedia collection<sup>6</sup>. In the extended evaluation presented in this work we used the same parameters, which further assess the generalization capability of the evaluated approach.

#### 4.3.5 Computational Complexity

In this Section the NCC-based matching procedure described in Section 4.3.2 and the geometry-based matching method described in Section 4.3.3 are compared from a computational point of view.

#### 4.3.6 NCC-Based Complexity

The computational complexity of Normalized Cross Correlation (NCC) used to find matchings of a reference template with dimensions  $m \times n$  within a scene image with sizes  $M \times N$  is:

$$O(mnMN) \tag{18}$$

Therefore, the cost of NCC-based matching, for each  $m \times n$  block, is equal to:

$$T_{NCCMatching} = O(mnMN). \tag{19}$$

Let  $D$  be the number of blocks:

$$D = \frac{MN}{mn} \tag{20}$$

Hence, the total cost due to the matching procedure described in Section 4.3.2 is:

$$T_{NCCBased} = D \times T_{NCCMatching} = D \times O(mnMN) = O(M^2N^2). \tag{21}$$

---

<sup>6</sup><http://www.3dmedia.com/gallery>

### 4.3.7 Geometry-Based Complexity

The cost of the computation of the SSD between two blocks of size  $m \times n$  is  $O(mn)$ ; therefore, the cost due to the matching method described in Section 4.3.3 is:

$$T_{GeometryMatching} = k \times O(mn) \quad (22)$$

where  $k$  is the number of blocks that the procedures compares to find the best matching. Thus, if  $D$  is the total number of processed blocks with sizes  $m \times n$ , the total cost of the matching procedure of the geometry based approach described in Section 4.3.3 is:

$$T_{GeometryBased} = D \times T_{GeometryMatching} = D \times k \times O(mnMN). \quad (23)$$

The value of  $k$  is very low due to the range reduction approach described in Section 4.3.3. In particular, the value of  $k$  is negligible with respect to the term  $O(mn)$ . Therefore, the total cost is:

$$T_{GeometryBased} = D \times O(mnMN) = O(MN). \quad (24)$$

Since the enhancing procedure is the same for both the NCC-based and the geometry-based approaches and its computational cost is lower than the matching cost, it has not been considered in the above detailed computational analysis. Indeed, the overall cost of the enhancing procedure is linear, hence it can be simply added to the total cost of the matching step, which has an higher degree. Therefore, is possible to consider just the total costs of the compared matching procedures for their computational evaluation. The above analysis shows that the geometry-based reconstruction improves the efficiency by reducing the order of growth from quadratic (i.e.,  $O(M^2N^2)$ ) to linear (i.e.,  $O(MN)$ ). Figure 12 shows the distribution of the computation time (in seconds) with respect to the image resolution (i.e.,  $M \times N$ ). The blue dots represent the pairs ( $M \times N$ , time), whereas the orange dots represent the same data after subtracting from  $M \times N$  the number of pixels not processed due to the uniform block check. Is possible to observe that the latter set of points (i.e., orange dots) are distributed linearly with respect to the x-axis (i.e.,  $M \times N$ ). The experiments have been performed on a set of images from the Middlebury Stereo Dataset of 2014 (i.e., the one with the highest resolution images).

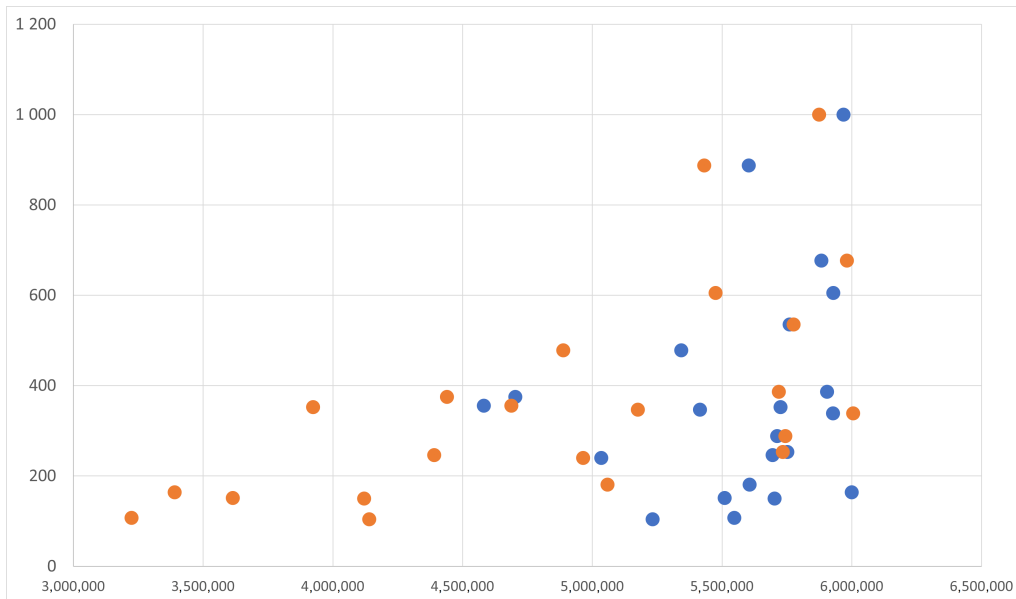


Figure 12:  $M \times N$  (x-axis) versus computation time in seconds (y-axis). The orange dots takes into account when the uniform blocks are ignored from the procedure.

#### 4.4 Experiments

The proposed method has been successfully evaluated in [22] with high resolution images taken from real and challenging cases such as flowers compositions, natural scenes, animals, buildings, etc. The obtained results are reported in Table 15 considering the achieved bit-rate saving and lossy. In particular, the dataset includes 23 stereoscopic MPO images compliant with the [23] standard at different resolutions ( $1440 \times 1080$ ,  $1620 \times 1080$ ,  $1444 \times 1080$ ,  $1924 \times 1080$ , or  $1920 \times 1080$ ). The JPEG quality factor has been set equal to 85 to encode the high quality image, whereas for the encoding of the low quality one it has been set to 65 or 70, by using standard quantization tables. The achieved results show compression gain in terms of total bit-rate, while the quality loss is measured considering the Peak Signal to Noise Ratio (PSNR) measure. In particular, for each MPO image and for each value of JPEG compression factor used to compress the low quality image (i.e., 65 or 70) Table 15 reports the dimensions of the blocks used in decoding (second column) and the lossy (in terms of dB) computed on the reconstructed image obtained after the enhancing procedure. The bitrate saving is computed by comparing the space needed to encode the low quality image with the space needed to encode the same image using a quality factor equal to 85 (third and fifth columns). Note that the size of the blocks has been chosen ad hoc for each stereopair, depending on the resolution of the original images. Experiments show interesting bit-rate values with very low lossy. However the considered dataset only includes 23 stereopairs.

In the proposed experiments, we considered a pool of well-known standard stereoscopic datasets built and publicly shared. Since 2001, the Middlebury Stereo Datasets have been considered to evaluate several stereoscopic algorithms, including stereoscopic image compression approaches. In the experiments here presented, we considered 60 stereopairs from all the Middlebury Stereo Dataset versions published between 2001 and 2014. Indeed, there are several versions that differ by the resolution of images and the employed acquisition technique. In particular, we considered:

- Five datasets of piecewise planar scenes of Middlebury-Scenes 2001 [50];
- Two datasets of Middlebury-Scenes 2003 [45];
- Nine datasets of Middlebury-Scenes 2005 [51];
- Twenty-one datasets of Middlebury-Scenes 2006 [52];
- Twenty-three datasets of high-resolution scenes of Middlebury-Scenes 2014 [53].

MPO Image	N × M	Low Quality 65		Low Quality 70	
		Lossy (dB)	Bit-Rate Saving	Lossy (dB)	Bit-Rate Saving
Flowers1	360 × 360	2.17	40.70%	1.65	34.60%
Flowers2	360 × 481	2	40.60%	1.32	34.50%
Flowers3	360 × 481	2.76	48.70%	2.76	41.70%
Castle	360 × 481	2.62	38.30%	2.18	32.50%
Dorm	360 × 360	2.73	37.10%	2.64	31.10%
Pelion	360 × 481	2.34	37.60%	2.34	31.80%
Hallway	360 × 482	2.33	37.60%	2.33	32.10%
Statue	360 × 483	2.59	41.90%	2.6	35.70%
Library	360 × 270	1.98	38.70%	1.71	32.70%
Hall	360 × 360	1.66	41.10%	1.4	34.90%
Garden					
Bridge	360 × 360	2.11	39.50%	1.82	33.50%
Autumn1	360 × 361	2.73	35.30%	2.53	29.80%
Autumn2	360 × 361	2.6	36.40%	2.4	30.60%
Autumn3	360 × 361	2.38	37.00%	2.15	31.20%
Autumn4	360 × 361	2.65	36.10%	2.44	30.40%
Animals1	360 × 240	2.16	38.80%	2.16	32.80%
Animals2	360 × 240	2.47	37.00%	2.18	31.30%
Cube	360 × 360	2.33	39.30%	2	33.30%
Covered	360 × 360	1.88	39.20%	1.73	33.40%
Garden	360 × 360	2.41	38.50%	2.15	32.50%
Snow	360 × 481	2.62	36.80%	2.45	31.20%
Tree	360 × 360	2.69	37.40%	2.52	31.40%
Zoo	360 × 240	2.67	36.90%	2.33	31.10%

Table 15: Results obtained in the first experiments reported in [41, 22].

The dimensions and variety of stereopair used on the proposed experiments allow us to perform a large evaluation of the pipeline on a standard and well-known benchmark set of stereopair images, as well as compare the performances of the pipeline on input of different resolutions. The approach has been evaluated by varying either the local feature detection and the fundamental matrix estimation algorithms, considering the SIFT or Harris keypoints detection to define the initial set of image point correspondences and the 8-points algorithm or the Least Median of Squares optimization for the fundamental matrix estimation. As expected, the SIFT algorithm provides an high amount of correspondences with the presence of some outliers. The Harris keypoints method detects a lower number of correspondences but with a lower outlier presence rate. As explained in Section 4.3.3, the selection of the best approach to estimate the fundamental matrix depends on the number and the quality of the available point correspondences between the two image views.

The results of the experiments on the Middlebury-Scenes 2014 dataset are shown in Table 16 (low quality factor 65) and Table 17 (low quality factor 70), whereas the results achieved on the Middlebury-Scenes 2006 dataset are reported in Tables 18 and 19. The complete results achieved on all the 60 considered stereopairs are detailed in Appendix A. The achieved results confirmed that the adaptive approach allows to obtain high performances in terms of image quality, while requiring very low computational efforts. Several experiments obtained a reconstruction loss lower than  $10^{-4}$  dB. By observing the experimental results, there is not a setting that strongly outperforms the others. However, considering the average performances in terms of lossy and computational time (reported on the last row of each Table), the methods based on the Least Median of Squares optimization to infer the fundamental matrix performs better in terms of lossy, whereas the methods based on the 8-points algorithm shown slightly better time performances.

ID	Stereo-pair	Size	# of Blocks	Method											
				Low Quality 65											
				Lmeds			Norm8Points								
				Bitrate Saving (%)	Harris Lossy (dB)	Time (s)	SIFT Lossy (dB)	Time (s)	Harris Lossy (dB)	Time (s)	SIFT Lossy (dB)	Time (s)			
1	Adirondack-perfect	1988 × 2880	3,479	43	0.58	933	0.58	252	<10 <sup>-4</sup>	304	1.59	352			
2	Backpack-perfect	2016 × 2940	3,600	39	1.32	846	<10 <sup>-4</sup>	684	1.32	239	1.33	339			
3	Bicycler-perfect	2008 × 2988	3,577	37	1.59	137	1.59	622	1.59	395	1.59	164			
4	Cable-perfect	1984 × 2796	3,381	41	1.59	519	1.59	372	1.59	121	2.18	1,070			
5	Classroom1-perfect	1920 × 3000	3,478	42	1.00	174	1.00	190	1.00	124	1.00	535			
6	Couch-perfect	1992 × 2300	2,793	42	1.59	266	2.00	339	2.59	1,068	2.01	356			
7	Flowers-perfect	1980 × 2880	3,479	38	2.17	306	2.17	378	2.59	1,197	2.59	1,503			
8	Jadeplant-perfect	1988 × 2632	3,185	40	1.74	346	2.32	632	2.91	924	2.92	1,040			
9	Mask-perfect	2008 × 2792	3,381	40	2.74	140	1.41	600	1.83	377	2.75	1,813			
10	Motorcycle-perfect	2000 × 2964	3,577	38	1.74	218	1.74	525	1.74	303	2.07	605			
11	Piano-perfect	1920 × 2820	3,243	40	1.74	170	1.74	304	2.06	178	2.07	347			
12	Pipes-perfect	1924 × 2960	3,431	38	2.00	752	2.91	800	2.00	156	2.01	246			
13	Playroom-perfect	1908 × 2800	3,243	38	2.32	349	2.33	687	2.32	265	2.32	478			
14	Playtable-perfect	1848 × 2724	3,082	38	2.00	133	0.68	369	1.81	325	0.68	240			
15	Recycle-perfect	1924 × 2864	3,290	42	<10 <sup>-4</sup>	154	<10 <sup>-4</sup>	230	<10 <sup>-4</sup>	92	<10 <sup>-4</sup>	151			
16	Shelves-perfect	2000 × 2952	3,577	41	1.41	880	1.00	480	2.00	1,298	1.00	386			
17	Shopvac-perfect	1996 × 2356	2,842	42	<10 <sup>-4</sup>	241	<10 <sup>-4</sup>	365	<10 <sup>-4</sup>	396	<10 <sup>-4</sup>	375			
18	Sticks-perfect	2008 × 2864	3,430	38	2.42	164	1.26	351	2.23	778	1.49	253			
20	Sword1-perfect	2020 × 2912	3,600	39	1.49	978	0.58	572	1.49	208	1.49	677			
21	Sword2-perfect	2000 × 2856	3,430	42	2.59	156	1.00	378	1.00	748	1.00	288			
22	Umbrella-perfect	2016 × 2960	3,650	42	<10 <sup>-4</sup>	183	<10 <sup>-4</sup>	646	<10 <sup>-4</sup>	391	1.00	1,000			
23	Vintage-perfect	1924 × 2912	3,384	36	2.17	201	2.17	189	2.59	257	2.59	887			
Average				40	1.55	375	1.28	453	1.58	461	1.62	596			
St. Dev.				2.02	0.81	296	0.86	179	0.91	367	0.82	446			

Table 16: Results on the Middlebury-Scenes 2014 dataset, considering a low quality of 65.



ID	Stereo-pair	Size	# of Blocks	Bitrate Saving (%)	Method											
					Lmeds						Norm8Points					
					Harris		SIFT		Harris		SIFT		Harris		SIFT	
					Lossy (dB)	Time (s)	Lossy (dB)	Time (s)	Lossy (dB)	Time (s)	Lossy (dB)	Time (s)	Lossy (dB)	Time (s)	Lossy (dB)	Time (s)
1	Adirondack-perfect	1988 × 2880	3479	38	<10 <sup>-4</sup>	208	<10 <sup>-4</sup>	292	<10 <sup>-4</sup>	113	1.17	582				
2	Backpack-perfect	2016 × 2940	3600	33	0.32	679	<10 <sup>-4</sup>	541	1.59	1,184	1.33	474				
3	Bicycle1-perfect	2008 × 2988	3577	32	0.59	111	1.59	726	2.18	359	2.18	484				
4	Cable-perfect	1984 × 2796	3381	36	2.18	882	2.18	674	0.59	183	0.59	201				
5	Classroom1-perfect	1920 × 3000	3478	37	1.00	148	1.00	156	1.00	890	1.00	584				
6	Couch-perfect	1992 × 2300	2793	36	0.59	181	0.59	279	2.01	473	2.01	350				
7	Flowers-perfect	1980 × 2880	3479	33	1.17	851	1.17	395	1.17	444	1.17	385				
8	Jadeplant-perfect	1988 × 2632	3185	34	1.74	441	1.74	848	2.01	489	2.92	1,414				
9	Mask-perfect	2008 × 2792	3381	34	1.84	774	2.16	1,448	2.16	294	1.84	633				
10	Motorcycle-perfect	2000 × 2964	3577	33	1.42	392	1.74	458	1.74	249	2.07	392				
11	Piano-perfect	1920 × 2820	3243	35	1.74	222	2.07	708	1.74	210	2.07	240				
12	Pipes-perfect	1924 × 2960	3431	33	2.01	671	2.33	1,346	2.01	255	2.01	268				
13	Playroom-perfect	1908 × 2800	3243	33	2.07	543	2.07	772	2.07	138	2.33	413				
14	Playtable-perfect	1848 × 2724	3082	32	0.90	966	0.68	168	0.68	207	0.90	340				
15	Recycle-perfect	1924 × 2864	3290	37	<10 <sup>-4</sup>	184	<10 <sup>-4</sup>	191	1.00	95	1.00	446				
16	Shelves-perfect	2000 × 2952	3577	36	<10 <sup>-4</sup>	289	<10 <sup>-4</sup>	307	0.59	525	2.01	1,088				
17	Shopvac-perfect	1996 × 2356	2842	37	<10 <sup>-4</sup>	269	<10 <sup>-4</sup>	306	<10 <sup>-4</sup>	1003	<10 <sup>-4</sup>	315				
18	Sticks-perfect	2008 × 2864	3430	32	0.26	226	0.68	333	1.27	253	1.27	409				
20	Sword1-perfect	2020 × 2912	3600	33	0.91	1,488	0.59	407	1.91	1,737	0.91	319				
21	Sword2-perfect	2000 × 2856	3430	36	2.18	1,696	1.00	546	1.00	1,196	1.00	855				
22	Umbrella-perfect	2016 × 2960	3650	38	1.00	223	1.00	733	1.00	242	1.00	1,134				
23	Vintage-perfect	1924 × 2912	3384	31	1.17	324	1.17	169	2.59	165	1.17	241				
Average				35	1.05	535	1.08	537	1.38	486	1.45	526				
St. Dev.				2.08	0.76	433	0.81	352	0.73	437	0.68	320				

Table 17: Results on the Middlebury-Scenes 2014 dataset, considering a low quality of 70.

ID	Stereo pair	Size	# of Blocks	Bitrate Saving (%)	Method											
					Low Quality 65						Norm8Points					
					Harris		SIFT		Harris		SIFT		Harris		SIFT	
Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)					
1	Aloe	555 × 641	224	37	1.58	6	2.01	11	1.83	4	2.01	11				
2	Baby1	555 × 620	224	40	0.87	4	1.24	6	1.65	26	1.24	6				
3	Baby2	555 × 620	224	39	1.02	12	1.02	9	1.35	18	1.02	9				
4	Baby3	555 × 656	238	40	1.61	36	1.35	13	1.02	6	1.35	13				
5	Bowling1	555 × 626	224	39	1.00	10	1.59	5	1.91	27	1.59	5				
6	Bowling2	555 × 665	238	39	1.81	9	2.27	15	1.81	13	2.27	15				
7	Cloth1	555 × 626	224	36	1.52	11	1.77	6	1.77	11	1.77	6				
8	Cloth2	555 × 650	224	38	1.75	22	1.65	9	1.85	20	1.65	9				
9	Cloth3	555 × 626	224	38	1.85	9	1.85	15	2.18	24	1.85	15				
10	Cloth4	555 × 650	224	37	2.18	27	1.84	8	1.91	10	1.84	8				
11	Flowerpots	555 × 656	238	39	1.59	16	2.01	12	1.59	7	2.01	12				
12	Lampshade1	555 × 650	224	39	3.18	14	3.18	13	1.91	9	3.18	13				
13	Lampshade2	555 × 650	224	39	2.01	8	2.01	6	2.01	14	2.01	6				
14	Midd1	555 × 698	252	36	1.49	5	2.42	31	1.49	8	2.42	31				
15	Midd2	555 × 683	238	36	1.27	5	1.27	10	1.27	4	1.27	10				
16	Monopoly	555 × 665	238	36	1.27	8	1.94	26	1.94	5	1.94	26				
17	Plastic	555 × 635	224	34	0.42	9	0.42	13	0.42	7	0.42	13				
18	Rocks1	555 × 638	224	38	1.65	8	2.40	13	1.65	15	2.40	13				
19	Rocks2	555 × 638	224	38	1.65	9	2.05	8	2.05	5	2.05	8				
20	Wood1	555 × 686	238	42	0.59	15	0.59	6	1.33	29	0.59	6				
21	Wood2	555 × 653	224	40	1.42	31	0.42	8	<10 <sup>-4</sup>	6	0.42	8				
Average					1.53	13	1.68	12	1.57	13	1.68	12				
St. Dev.					1.68	8.73	0.69	6.50	0.54	8.11	0.69	6.50				

Table 18: Results on the Middlebury-Scenes 2006 dataset, considering a low quality of 65.

ID	Stereo pair	Size	# of Blocks	Bitrate Saving (%)	Method									
					Low Quality 70					NormsPoints				
					Harris		SIFT		Harris		SIFT			
					Lossy (dB)	Time (s)	Lossy (dB)	Time (s)	Lossy (dB)	Time (s)	Lossy (dB)	Time (s)		
1	Aloe	555 × 641	224	32	1.26	9	1.45	18	1.52	8	1.52	8		
2	Baby1	555 × 620	224	34	0.66	4	1.32	34	1.32	28	0.87	7		
3	Baby2	555 × 620	224	33	1.61	18	1.61	32	1.61	9	1.02	10		
4	Baby3	555 × 656	238	34	0.91	6	1.17	40	0.91	16	0.91	11		
5	Bowling1	555 × 626	224	33	1.00	10	1.00	26	1.91	16	1.00	5		
6	Bowling2	555 × 665	238	33	2.27	30	1.40	14	1.40	7	2.08	8		
7	Cloth1	555 × 626	224	31	1.83	28	1.47	29	1.47	9	1.71	13		
8	Cloth2	555 × 650	224	32	1.32	13	1.21	15	1.75	10	1.65	7		
9	Cloth3	555 × 626	224	32	1.50	10	1.50	16	1.69	6	1.50	10		
10	Cloth4	555 × 650	224	31	2.05	25	1.49	13	1.49	8	1.78	8		
11	Flowerpots	555 × 656	238	34	1.59	9	1.59	22	2.01	16	2.42	19		
12	Lampshade1	555 × 650	224	33	2.33	11	2.33	14	2.59	26	2.33	8		
13	Lampshade2	555 × 650	224	34	2.01	8	2.01	8	2.01	8	2.33	30		
14	Middl1	555 × 698	252	31	1.27	11	1.27	11	1.27	8	1.49	9		
15	Middl2	555 × 683	238	31	0.68	4	0.68	14	0.68	4	1.49	29		
16	Monopoly	555 × 665	238	30	1.10	5	1.94	29	1.27	15	1.94	20		
17	Plastic	555 × 635	224	29	1.42	22	0.42	6	0.42	18	0.42	8		
18	Rocks1	555 × 638	224	32	2.27	33	1.33	9	2.40	37	1.65	7		
19	Rocks2	555 × 638	224	32	1.50	9	0.91	6	1.65	9	1.65	6		
20	Wood1	555 × 686	238	35	0.56	16	<10 <sup>-4</sup>	12	1.33	29	<10 <sup>-4</sup>	8		
21	Wood2	555 × 653	224	33	1.42	25	0.42	10	<10 <sup>-4</sup>	6	1.42	22		
Average				32	1.47	15	1.26	18	1.46	14	1.48	12		
St. Dev.				1.45	0.53	8.98	0.55	9.89	0.61	9.09	0.62	7.40		

Table 19: Results on the Middlebury-Scenes 2006 dataset, considering a low quality of 70.

#### 4.4.1 Comparative Evaluation

The approaches in [33, 35, 36] have a pipeline similar to those used in the MPO standard [23], with the main difference in the encoding/decoding transform that is applied after quantization, namely curvelets, arithmetic encoding and Huffman encoding, respectively. Such an encoding-decoding pipeline is independent from the stereopair acquisition settings. Those approaches have been evaluated on the popular LIVE 3D image quality database <sup>7</sup> [54] of the University of Texas. In this paragraph, we report the comparative evaluation of the adaptive approach [22] with respect to [33, 35, 36] considering the same evaluation settings. Table 20 shows the detailed results of the adaptive method applied on the LIVE 3D dataset, considering all the experimental settings involved in our benchmark. Note that we reported the PSNR for the right image only, indeed, all the experimental settings apply the same encoding/decoding pipeline to the left image, whereas a specific decoding process is applied on the right view. The PSNR of the left view is the same, regardless the experimental setting and is equal to 38.38744. Table 21 shows the comparison between the adaptive approach and the methods in [33, 35, 36]. Moreover, we also reported the performances obtained by applying the Standard MPO approach [23]. It is possible to observe that the adaptive method significantly outperforms the other approaches in terms of Compression Ratio. Regarding the PSNR measure, the Huffman coding proposed in [36] achieved the best results, with corresponding very low performances in terms of Compression Ratio (CR).

	Lossy (dB)	Bitrate Saving	Time (s)	CR	PSNR Right
LmedSHarris70	1.4893	34.1709	2.0620	14.1925	36.6341
LmedSIFT70	1.4690	34.1709	1.6780	14.1925	36.6544
Norm8PointsHarris70	1.6734	34.1709	3.9440	14.1925	36.4500
Norm8PointSIFT70	1.4604	34.1709	1.5960	14.1925	36.6631
LmedSHarris65	1.6804	40.3686	1.1460	14.7500	36.4430
LmedSIFT65	1.6025	40.3686	4.3960	14.7500	36.5209
Norm8PointsHarris65	1.7414	40.3686	3.4940	14.7500	36.3820
Norm8PointSIFT65	1.6379	40.3686	1.0060	14.7500	36.4856

Table 20: Performance of the [22] method applied on the LIVE 3D Dataset [54].

<sup>7</sup>[http://live.ece.utexas.edu/research/quality/live\\_3dimage\\_phase1.html](http://live.ece.utexas.edu/research/quality/live_3dimage_phase1.html)

Method	CR	PSNR
Adaptive [20]	14.4713	37.4583
Standard MPO	11.6094	38.2526
Lossy SICCAC [11]	8.6400	41.5831
Lossy S2ICAC [13]	6.3600	33.9400
S1CHC [14]	4.4271	49.1446
Lossless SICCAC [11]	4.1488	41.7359
Lossless S2ICAC [13]	3.6100	34.885

Table 21: Compression Ratio (CR) and average PSNR of the evaluated methods on the LIVE 3D Dataset [54].

#### 4.4.2 Subjective Assessment

While the adaptive method is the best in terms of CR, in order to assess the perceived quality of the images obtained by the adaptive method after reconstruction, we set up a subjective quality assessment experiment, described in this Section. In addition to the quality evaluation of the adaptive approach, in terms of bitrate saving, PSNR and computational time reported above, we also set up a subjective assessment of the reconstructed images' quality. The experiments involved 116 people selected by considering variability among gender, age and people with eyesight problems. In particular, each participant indicated if he/she uses corrective lenses or glasses, their gender and their age, by selecting a set of age ranges (18–29, 30–39, 40–49, 50–59 or  $\geq 60$ ). Since we involved only a small number of people aged in the range 50–59 (i.e., 10), we merged the last two age ranges, creating the new category of people aged more than 50. During the acquisition session, each user was required to evaluate the quality of two images, one image is the right view of a stereopair at the original quality, whereas the other one is the same view after the compression and reconstruction pipeline described in this paper. Each user evaluated 10 image pairs randomly selected from the Middlebury Datasets and the position of the low quality image (left or right) is also selected randomly. We counted the number of times the users correctly selected the low quality image (hit) and the times the users selected the other ones (miss). Table 22 shows the percentage of hit and miss considering the different user categories. The general and detailed hit/miss percentage can be also observed in the pie charts shown in Figure 13. From the results is possible to observe that the involved participants were unable to distinguish the reconstructed image than the original one. This result is invariant considering also marginalizing the results by gender, age, or the presence of eyesight problems. We also performed an independence  $Chi^2$  test, which confirmed the independence between the considered categories and the capability of guessing the right image. The test has been repeated considering several values of significance (from 0.01 to 0.05), in which the null hypothesis  $H_0$  (i.e., independence) were accepted with test confidence

always higher than 99%.

	Hit	Miss
Female	0.537931	0.462069
Male	0.593620	0.406380
Without Glasses	0.594990	0.405010
With Glasses	0.562030	0.437970
Age 18–29	0.613269	0.386731
Age 30–39	0.514563	0.485437
Age 40–49	0.581818	0.418182
Age > 50	0.482759	0.517241

Table 22: Hit and miss collected during the subjective assessment experiments.

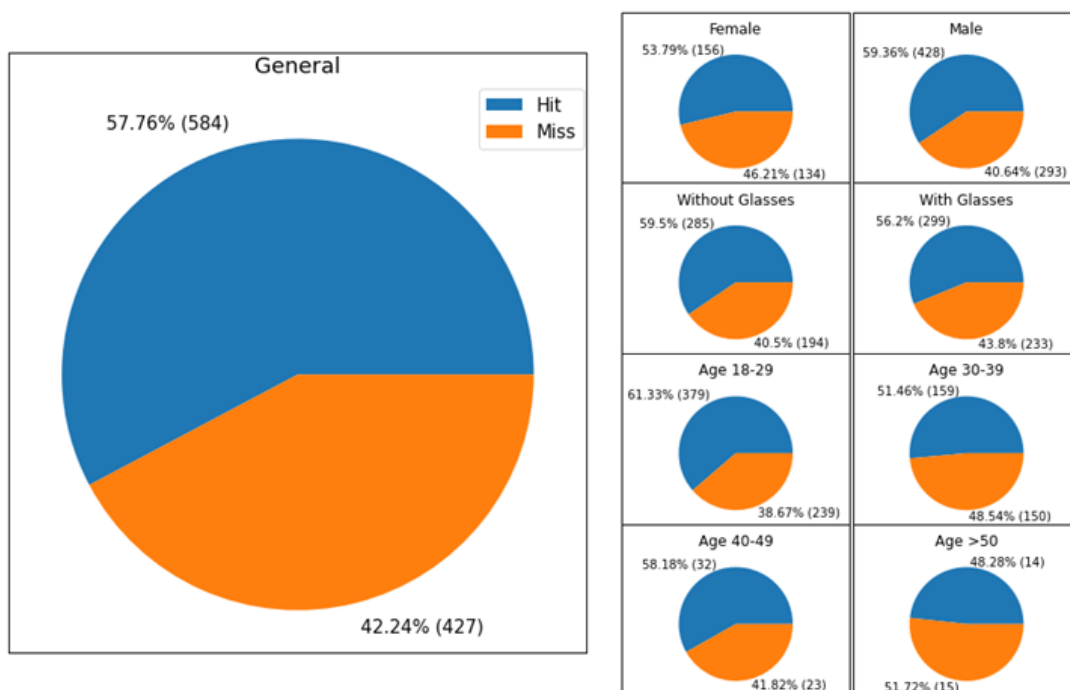


Figure 13: Detailed performances of hit/miss obtained during the subjective assessment.

## 4.5 Conclusions

In this paper, an overview on stereoscopic image compression is first presented, with a focus on the problem of standardization of the compressed images. Then, an adaptive stereoscopic image compression approach first presented in [41] and further extended in [22] has been investigated and detailed. In particular, the advances in the matching strategy of the approach under analysis are presented. Then, the work presents an evaluation of the adaptive stereo compression method under

different settings, taking into account the compressed image quality, two different optimization methods and two keypoints extraction techniques. The 16 resulting evaluation settings have been used to benchmark the compression method on the five Middlebury Datasets published between 2001 and 2014, which represent an established standard for the objective assessment of algorithms in the field of stereoscopy. The results shown that the method is able to obtain high compressed stereoscopic images, while maintaining the visual quality, without requiring additional storing payload and allowing the usage of a file format conform with [23] specifications.

In this benchmarking, we also evaluated the perceived quality of the reconstructed images by involving a large number of participants that were required to select the highest quality image between the two proposed, future extensions of this approach can evaluate the performances considering more advanced and specific Human Visual System based quality metrics [55], to properly assess to what extent we can compress the low quality image while maintaining high quality in the 3D perception of the scene.

Extended experiments could be performed to significantly lower the quality of the images and observing the effects (e.g., blocking artefacts) of using image blocks with resolution dimensions that are powers of two ( $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ , etc.). Moreover, studies on stereoscopic image compression can be also performed on new applications and domains in which stereoscopy is being applied [56].

## 4.6 Papers

This section was based on the following scientific publication:

- Alessandro Ortis, Marco Grisanti, Francesco Rundo and Sebastiano Battiato, "A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images", Journal of Imaging - July 2021.

## 5 Final Discussion, Remarks and Future Works

The main contributions of this thesis are related to the application of artificial intelligence techniques and technologies applied to the context of photovoltaic panels with the aim of improving their maintainability. In particular, Section 3.3 studied how to automatically classify various types of defects present in photovoltaic panels identifiable with the test based on electroluminescence; in this section the creation of a new dataset and an in-depth benchmark on this dataset were described using three classifiers and four pre-trained neural network reaching a Hamming Distance equal to 0.07 on the ResNext50 neural network. Instead, the Section 3.4 relied on a comparison of conventional models with machine learning models for the estimation of the temperature of monofacial and bifacial photovoltaic modules; this assessment focused on monofacial and bifacial photovoltaic modules situated in distinct regions of Italy; Notably, the application of machine learning models yielded enhanced accuracy in temperature estimation compared to the utilization of physical models, in which there were used empirical coefficients obtained in the first case without fine-tuning and in the second case with fine-tuning on available data; these models achieved a level of accuracy in their results that closely approached the temperature estimations derived from the machine learning models, which consistently outperformed the accuracy achieved by the physical models, surpassing their outcomes. Furthermore, in Section 4 was performed a benchmark evaluation about compression techniques of stereoscopic images on large and standardized datasets including 60 stereopairs that differ by resolution and acquisition technique in which the findings demonstrated that the new examined technique [41, 22] successfully reached notable compression of stereoscopic images, which was achieved without the compromise of visual quality and without the need of extra storage capacity. During the evaluation process, a substantial participant pool was engaged to gauge the perceived quality of the reconstructed images; participants were tasked with selecting the superior-quality image from the two options presented.

### 5.1 Future Directions

As regards the application of artificial intelligence related to the world of photovoltaic panels: in relation to the automatic classification of defects, having released a new dataset with an associated benchmark, a good starting point for future comparisons was created. As future works, it could be possible to apply the described task methods for defects detection in other industrial sectors such as Silicon technology; instead, in relation to the comparison between physical models and machine learning models in temperature estimation of photovoltaic modules, further studies will allow a generalization of the use of machine learning models for temperature estimation for different



technologies and climate zones. As far as benchmark evaluation about compression techniques of stereoscopic images is concerned future extensions of the Human Visual System test could be implemented; furthermore extended experiments could be performed to significantly lower the quality of the images and observing the effects; moreover, studies on stereoscopic image compression can be also performed on new applications and domains.

## **A A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images: Complete Results**

In this section there are the complete results of [3].

ID	Stereopair	Size	# of Blocks	Bitrate Saving (%)	Method								
					Low Quality 65				NormsPoints				
					LmedS		SIFT		Harris		SIFT		
Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)				
1	Art	555X695	238	42	2,10	42	1,98	22	1,72	15	1,98	13	
2	Books	555X695	238	15	2,01	15	2,28	23	2,28	11	2,61	37	
3	Computer	555X665	238	13	1,69	13	1,69	8	1,95	7	1,95	12	
4	Dolls	555X695	238	10	1,33	10	1,33	12	1,87	43	1,76	13	
5	Drumsticks	555X695	238	13	2,01	13	2,01	10	2,12	7	2,12	13	
6	Dwarves	555X695	238	13	1,59	13	1,59	11	1,59	16	1,59	10	
7	Laundry	555X671	238	13	2,00	13	1,84	11	2,40	8	2,40	45	
8	Moebius	555X695	238	18	1,62	18	1,62	12	1,62	9	2,17	26	
9	Reindeer	555X671	238	9	1,84	9	1,64	12	1,84	6	1,84	8	
Average					16	1,80	16	1,78	14	1,93	13	2,05	20

Table 23: Results on the Middlebury-Scenes 2005 dataset, considering a low quality of 65.

Stereopair	Size	# of Blocks	Method											
			Low Quality 70						Norm8Points					
			Bitrate Saving (%)		LmedS		SIFT		Harris		SIFT			
Lossy (dB)		Time (sec)		Lossy (dB)		Time (sec)		Lossy (dB)		Time (sec)				
ID														
1	Art	555X695	238	32	1,39	23	1,84	33	1,98	30	1,39	14		
2	Books	555X695	238	31	1,90	16	2,28	37	2,28	28	1,90	12		
3	Computer	555X665	238	30	1,27	9	1,10	9	1,10	6	1,95	9		
4	Dolls	555X695	238	32	1,76	25	1,23	15	1,76	30	1,33	12		
5	Drumsticks	555X695	238	32	1,74	9	1,74	31	2,01	15	2,01	34		
6	Dwarves	555X695	238	32	2,01	29	1,13	9	1,40	16	1,59	24		
7	Laundry	555X671	238	30	1,69	11	1,69	11	2,00	17	2,13	42		
8	Moebius	555X695	238	33	1,47	12	1,14	9	1,47	6	1,62	9		
9	Reindeer	555X671	238	35	2,01	42	1,64	28	1,23	6	1,84	7		
Average			32	1,69	19	1,53	20	1,69	17	1,75	18			

Table 24: Results on the Middlebury-Scenes 2005 dataset, considering a low quality of 70.

Stereopair	Size	# of Blocks	Method									
			Low Quality 65				Norm8Points					
			Bitrate Saving (%)		LmedS		SIFT		SIFT			
Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)			
1	cones	375X450	110	39	1,26	3	1,26	3	1,30	7	1,30	2
2	teddy	375X450	110	40	1,21	3	1,21	2	1,52	8	1,41	6
Average				40	1,23	3	1,23	3	1,41	7	1,36	4

Table 25: Results on the Middlebury-Scenes 2003 dataset, considering a low quality of 65.

Stereopair	Size	# of Blocks	Method									
			Low Quality 70					Norm8Points				
			Bitrate Saving (%)		LmedS		SIFT		Harris		SIFT	
ID			Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)
1	cones	375X450	110	33	1,13	3	1,17	5	1,38	7	1,33	3
2	teddy	375X450	110	34	1,12	4	1,05	3	1,12	3	1,12	2
Average				34	1,13	3	1,11	4	1,25	5	1,22	3

Table 26: Results on the Middlebury-Scenes 2003 dataset, considering a low quality of 70.

Stereopair	Size	# of Blocks	Bitrate Saving (%)	Method										
				Low Quality 65				Norm8Points						
				LmedS		SIFT		Harris		SIFT				
ID				Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	
1	barn1	381X432	110	40	1,34	7	1,17	2	1,17	2	1,17	3	1,34	2
2	barn2	381X430	110	42	0,98	1	0,98	2	0,98	2	0,98	3	0,98	1
3	bull	381X433	110	42	1,00	1	1,00	2	1,00	2	1,00	2	1,14	3
4	sawtooth	380X434	110	39	1,45	3	1,45	2	1,63	2	1,63	2	1,68	3
5	venus	383X434	110	40	1,49	7	1,27	5	1,21	5	1,21	3	1,27	2
Average				40	1,25	4	1,17	3	1,20	2	1,28	2	1,28	2

Table 27: Results on the Middlebury-Scenes 2001 dataset, considering a low quality of 65.

Stereopair	Size	# of Blocks	Bitrate Saving (%)	Method								
				Low Quality 70				NormsPoints				
				LmedS		SIFT		Harris		SIFT		
ID				Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	Lossy (dB)	Time (sec)	
1	barn1	381X432	110	34	1,09	2	1,09	3	1,09	3	1,26	2
2	barn2	381X430	110	35	0,77	1	0,77	2	0,77	2	1,04	4
3	bull	381X433	110	35	0,91	3	1,00	3	0,91	3	1,00	2
4	sawtooth	380X434	110	33	1,34	3	1,34	2	1,86	8	1,58	4
5	venus	383X434	110	34	1,08	7	1,08	4	1,32	4	1,27	2
Average				34	1,04	3	1,06	3	1,19	4	1,23	3

Table 28: Results on the Middlebury-Scenes 2001 dataset, considering a low quality of 70.



## **B Automation of Business Activities**

### **B.1 Introduction**

During my PhD I had to invest a lot of time in automating some activities useful for data collection and organization. In fact, it is not always possible to immediately obtain the various data needed for experiments since they can come from different sources, each of which can transcribe its data in a different format from that used by the other sources. Therefore, in the data collection and organization phase it is essential to automate as many repetitive operations as possible in order to be able to devote more time to the actual data analysis.

### **B.2 Techniques and Technologies Used**

In particular, some of the EGP sources that I had to use made it possible to download data from company websites created ad hoc for this purpose. However, those websites didn't allow for downloading large amounts of data in customizable formats. Therefore, the download and construction of entire datasets could take up to several days where the company's operators had to spend their time performing numerous manual operations, during which, for obvious reasons, errors could be made that compromised the integrity of the dataset.

Therefore, in collaboration with EGP it was decided to invest some time in automating the download and construction of datasets using basically two techniques, sometimes independently, sometimes crosswise: the automation of REST APIs usage and the automation of Browser usage.

#### **B.2.1 Automating the use of REST APIs**

In the digital era Application Programming Interfaces (APIs) play a fundamental role in enabling communication and data exchange between software systems. Representational State Transfer Application Programming Interfaces (REST APIs) have become particularly popular for their simplicity, scalability and flexibility in interacting with web applications and cloud services. However, with the increasing number of available APIs and the growing complexity of applications, automating the usage of REST APIs has become a critical aspect to optimize workflows and improve process efficiency.

Automation of REST APIs involves the use of scripts or automated tools to interact with APIs without manual intervention from the user. This practice offers several advantages, including:

- **Efficiency:** Automation eliminates the need to perform repetitive operations manually, reducing response times and enhancing overall system efficiency.

- **Reliability:** Automated scripts execute operations consistently, reducing the possibility of human errors and ensuring reliable results.
- **Scalability:** Automation allows handling large volumes of requests without excessive efforts from the user.
- **Monitoring:** Automated script can detect any anomalies or service interruptions.
- **Integration:** Automation facilitates integration between different applications and systems, enabling rapid and secure data and information exchange.

Automation of REST APIs usage has become essential for improving operational efficiency and ensuring reliable outcomes in web applications and cloud services. It is crucial to use automation judiciously, ensuring data security and considering its impact on all parties involved in the process. With proper REST API automation, organizations can gain a competitive edge by enhancing user experience and accelerating application development cycles. It was decided to use the "Requests" library for Python in order to do the described automation activity.

**Requests Library** In the context of web application development or automation, interacting with external REST APIs is a crucial element. Python, one of the most popular and versatile programming languages, offers a wide range of libraries to simplify access and management of APIs. Among these, the Python "Requests" library stands out for its ease of use and power, providing an elegant interface for making HTTP requests and manipulating JSON data.

The Requests library is a high-level Python module designed to simplify network operations, particularly HTTP requests. This library is built on top of URLLib3 and offers a user-friendly interface, making the process of interacting with external APIs more straightforward and intuitive.

The Python Requests library offers numerous features that make it a preferred choice for developers:

- **Simple Interface:** Requests provides a clear and easy-to-use interface for making HTTP requests such as GET, POST, PUT and DELETE.
- **Support for Parameters and Headers:** Users can pass parameters, custom headers and data in URL-Encoded or JSON formats to customize requests.
- **Automatic Session Handling:** Requests automatically maintains session state, handling cookies and authentication sessions.
- **SSL Verification:** The library offers support for SSL verification to ensure secure communications.

- **Timeout Management:** Users can set custom timeouts for requests to avoid prolonged blocking.

The Python Requests library is a powerful resource for simplifying network operations and interacting with external APIs. Its simple interface, support for custom parameters and headers and automatic session handling make it an indispensable tool for developers working with web applications and REST APIs services. Using the Requests library allows for improved efficiency in development, optimizing communication operations and manipulating JSON data in Python applications.

### **B.2.2 Automating the use of Browser**

In the realm of web development and internet browsing, browser automation has gained increasing significance. The ability to automate interactions with websites and web applications offers numerous advantages, including process optimization, automated testing and data extraction. There are several tools and libraries that enable browser automation. Browser automation is useful for:

- **Web Application Testing:** Automation tools allow running automated tests across different platforms and browsers, ensuring broader test coverage and reducing manual workload. Automated tests can verify proper functioning of the user interface, error handling and website responsiveness.
- **Data Scraping from Web Pages:** Automation tools can navigate through pages, find specific elements using CSS selectors and extract the required data. This technique is often used for gathering information from websites or performing data analysis from publicly available sources.

Browser automation has revolutionized the way with the which is possible interact with the web. With powerful tools like Selenium, which will be discussed shortly, it is possible to automate complex interactions with web applications and websites. Browser automation finds extensive use in automated testing, ensuring the integrity of web applications efficiently. Additionally, automated data extraction from web pages has become a common practice for gathering information and fueling analysis and decision-making processes. With the continuous development of new technologies and tools, browser automation will continue to evolve, offering new opportunities and solutions to enhance efficiency and user experience on the web.

**Selenium Library** In the realm of browser automation and web application testing, Python's Selenium library is a powerful and widely used tool. It offers a wide range of features for automated

control of web browsers, simulating human interactions and enabling efficient and accurate testing. The key features of Selenium are:

- **Browser Control:** Python's Selenium library allows for the control of popular web browsers such as Chrome, Firefox, Safari and Microsoft Edge. This feature enables developers to conduct cross-browser testing and interact with various browsing environments.
- **Automated Interactions:** Selenium enables the automation of browser interactions, such as clicking, form filling, page scrolling and cookie management. This capability is invaluable for testing the user interface and functionality of web applications.
- **Page Element Selection:** By using CSS or XPath selectors, the Selenium library can locate specific elements within the web page. This allows developers to extract relevant information and data for scientific analysis tasks.

Python's Selenium library is extensively used for web application testing automation in the scientific domain. Researchers can create automated test suites to verify the correctness and stability of web platforms used for scientific research. Furthermore, Selenium is employed for automated data extraction from web pages. Researchers can gather data from various online sources for analysis and study purposes. Moreover, Selenium can be utilized to simulate user interactions with scientific web applications. This enables researchers to conduct realistic usage scenarios to evaluate application performance and responsiveness. Another interesting scenario is the use of Selenium for the automated monitoring of web resources, such as research websites, blogs and scientific news feeds. This keeps researchers updated with the latest discoveries and publications in their research field.

Python's Selenium library is an indispensable tool for browser automation and web application testing in the scientific domain. With the ability to control various browsers and automate complex interactions, Selenium offers a wide range of possibilities for researchers and developers. From automated testing to data collection and web resource monitoring, the Selenium library provides invaluable support to enhance the efficiency and accuracy of online scientific activities. By harnessing its powerful capabilities, scientists can optimize their work, focus on research activities and achieve significant results more quickly and efficiently.

### **B.3 Conclusions**

The various automation jobs carried out for EGP have proved to be very useful both for my research, allowing me to work easily with large quantities of data collected and sorted in a short time and for the company's employees, who have had the opportunity to improve their productivity.

## List of Tables

1	The original labelling strategy of the ELPV Dataset with the cardinality of each class.	34
2	Our custom labelling of the ELPV Dataset.	36
3	First Classifier Results. The best and the running up results are highlighted in bold and underline, respectively.	41
4	Second Classifier Results. The best and the running up results are highlighted in bold and underline, respectively.	42
5	Third Classifier Results. The best and the running up results are highlighted in bold and underline, respectively.	42
6	Information about True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for each classifier and for each class.	42
7	Some examples about True Positives, True Negatives, False Positives and False Negatives for each class of the First Classifier.	43
8	Some examples about True Positives, True Negatives, False Positives and False Negatives for each class of the Second Classifier.	44
9	Some examples about True Positives, True Negatives, False Positives and False Negatives for each class of the Third Classifier.	45
10	The default parameters for Sandia Model and Faiman Model regarding Eurac Dataset and EGP Dataset.	53
11	The fine-tuned parameters for Sandia Model and Faiman Model regarding Eurac Dataset and EGP Dataset.	54
12	Statistical indices of physical and machine learning models applied to Eurac Dataset - Monofacial Polycrystalline - Bolzano. FT stands for Fine-Tuning. RFR stands for Random Forest Regressor. GBR stands for Gaussian Boosting Regressor.	56
13	Statistical indices of physical and machine learning models applied to EGP Dataset - Bifacial mono-PERC, neglecting $G_R$ - Catania. FT stands for Fine-Tuning. RFR stands for Random Forest Regressor. GBR stands for Gaussian Boosting Regressor.	56
14	Statistical indices of physical and machine learning models applied to EGP Dataset - Bifacial mono-PERC, considering $G_R$ - Catania. FT stands for Fine-Tuning. RFR stands for Random Forest Regressor. GBR stands for Gaussian Boosting Regressor.	56
15	Results obtained in the first experiments reported in [41, 22].	78
16	Results on the Middlebury-Scenes 2014 dataset, considering a low quality of 65.	80
17	Results on the Middlebury-Scenes 2014 dataset, considering a low quality of 70.	81
18	Results on the Middlebury-Scenes 2006 dataset, considering a low quality of 65.	82

19	Results on the Middlebury-Scenes 2006 dataset, considering a low quality of 70. . .	83
20	Performance of the [22] method applied on the LIVE 3D Dataset [54]. . . . .	84
21	Compression Ratio (CR) and average PSNR of the evaluated methods on the LIVE 3D Dataset [54]. . . . .	85
22	Hit and miss collected during the subjective assessment experiments. . . . .	86
23	Results on the Middlebury-Scenes 2005 dataset, considering a low quality of 65. . .	91
24	Results on the Middlebury-Scenes 2005 dataset, considering a low quality of 70. . .	92
25	Results on the Middlebury-Scenes 2003 dataset, considering a low quality of 65. . .	93
26	Results on the Middlebury-Scenes 2003 dataset, considering a low quality of 70. . .	94
27	Results on the Middlebury-Scenes 2001 dataset, considering a low quality of 65. . .	95
28	Results on the Middlebury-Scenes 2001 dataset, considering a low quality of 70. . .	96

## List of Figures

1	Architecture of AlexNet. . . . .	16
2	Phases of Computer Vision. (1) Pre Processing is the initial stage of processing in which the images are cleaned, denoised and normalized to improve the data quality. (2) Feature Extraction is the step that involves extracting significant features from images, such as edges, contours, textures and other relevant information. (3) Classification and Recognition is the phase in which the extracted characteristics are used to recognize specific objects, people or patterns within the images. Machine learning techniques, such as deep learning, are often used for this stage to make accurate predictions. (4) Tracking and Localization is the phase that concerns the ability to track and locate objects or people in time and space, allowing recognition and monitoring in video sequences. . . . .	18
3	On the left is shown a monocrystalline cell. On the right is shown polycrystalline cell. Both images have a probability of defectiveness of 0%. The monocrystalline cell appears cleaner than the polycrystalline cell, which appears to be dirty. . . . .	35
4	On the left is shown a cell with a probability of defect equal to 0. On the right is shown a cell with a probability of defect equal to 1. . . . .	35
5	(a) cell with Crack defect; (b) cell with a CellBreakage defect; (c) cell with Dark-Area defect; (d) cell with a HotSpot defect. . . . .	36
6	Encoding pipeline. Each view is coded according to the MPO format (i.e., by applying the JPEG compression) using different quality factors. . . . .	62
7	Example of blocking schema employed in [22]. . . . .	68
8	Considering any point in the right image (b) $I_R$ (1, 2 and 3), the method in [22] searches the corresponding points in the left image $I_L$ by computing their epipolar lines (a). . . . .	70
9	Details of the images shown in Figure 8. The tip of the cone (i.e., point number 2 in the image (b) depicting the right view $I_R$ ) corresponds to a point that lies on the corresponding epipolar line (i.e., line number 2 in (a)) on the left view $I_L$ . . . . .	70
10	Example of matching by using the approach in [22] (a) and detail near the vertex of the $\phi$ function (see Equation (15)). The red circle is the point on $\phi$ with the same $x$ as the point on the (b). The blue point represents the value of $\phi$ corresponding to the Best Matching Point (BMP) according to the algorithm. . . . .	72

11	Examples of block composition performed to overcome with partial matching blocks. Each row shows an example of block composition, the first row shows the block number 111 ( <b>a</b> ) of the image "Flowers 1", the partial matching block ( <b>b</b> ) and the composed block ( <b>c</b> ). The second and third rows are examples of partial matching of blocks extracted from the image "Cones", related to the blocks number 17 and 13, respectively. . . . .	74
12	$M \times N$ (x-axis) versus computation time in seconds (y-axis). The orange dots takes into account when the uniform blocks are ignored from the procedure. . . . .	77
13	Detailed performances of hit/miss obtained during the subjective assessment. . . .	86



## References

- [1] Marco Grisanti, Gaetano Mannino, Giuseppe Marco Tina, Alessandro Ortis, Mario Cacciato, Sebastiano Battiato, Fabrizio Bizzarri, and Andrea Canino, “Thermal Models of Monofacial and Bifacial PV Modules: Machine Learning and Physical Estimation Models Comparison,” in *IEEE Photovoltaic Specialists Conference*, 2023.
- [2] Marco Grisanti, Maria Ausilia Napoli Spatafora, Alessandro Ortis, and Sebastiano Battiato, “E-ELPV: Extended ELPV Dataset for Accurate Solar Cells Defect Classification,” in *Intelligent Systems Conference*, 2023.
- [3] Alessandro Ortis, Marco Grisanti, Francesco Rundo, and Sebastiano Battiato, “A Benchmark Evaluation of Adaptive Image Compression for Multi Picture Object Stereoscopic Images,” *Journal of Imaging*, 2021.
- [4] F. Rosenblatt, “The perceptron - a perceiving and recognizing automaton,” Tech. Rep., Cornell Aeronautical Laboratory, 1957.
- [5] Yann Lecun, Leon Bottou, Y. Bengio, and Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, 1998.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Neural Information Processing Systems (NIPS)*, 2012.
- [7] Sergiu Deitsch, Claudia Buerhop-Lutz, Evgenii Sovetkin, Ansgar Steland, Andreas Maier, Florian Gallwitz, and Christian Riess, “Segmentation of Photovoltaic Module Cells in Uncalibrated Electroluminescence Images,” *Machine Vision and Applications*, 2021.
- [8] Sergiu Deitsch, Vincent Christlein, Stephan Berger, Claudia Buerhop-Lutz, Andreas Maier, Florian Gallwitz, and Christian Riess, “Automatic Classification of Defective Photovoltaic Module Cells in Electroluminescence Images,” *Solar Energy*, 2019.
- [9] M. Waqar Akram, Guiqiang Li, Yi Jin, Xiao Chen, Changan Zhu, Xudong Zhao, Abdul Khaliq, M. Faheem, and Ashfaq Ahmad, “CNN based automatic detection of photovoltaic cell defects in electroluminescence images,” *Energy*, 2019.
- [10] Mustafa Yusuf Demirci, Nurettin Bešli, and Abdülkadir Gümüüşgü, “Efficient deep feature extraction and classification for identifying defective photovoltaic module cells in Electroluminescence images,” *Expert Systems with Applications*, 2021.

- [11] Wuqin Tang, Qiang Yang, Kuixiang Xionga, and Wenjun Yana, “Deep learning based automatic defect identification of photovoltaic module using electroluminescence images,” *Solar Energy*, 2020.
- [12] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [13] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv PrePrint arXiv:1409.1556*, 2014.
- [14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International journal of computer vision*, 2015.
- [17] Li Jiyong, Zhao Zhendong, Li Yisheng, Xiao Jing, and Tang Yunfeng, “Short-term PV/T module temperature prediction based on PCA-RBF neural network,” in *Earth and Environmental Science*, 2018.
- [18] Aiat Hegazy, E. T. El Shenawy, and M. A. Ibrahim, “Determination of the PV module surface temperature based on neural network using solar radiation and surface temperature,” *ARPJN Journal of Engineering and Applied Sciences*, 2019.
- [19] Eun Jung Dong, Lee Chanuk, Han Kim Kee, and Lok Do Sung, “Development of a Predictive Model for a Photovoltaic Module’s Surface Temperature,” *Energies*, 2020.
- [20] Fan Siyuan, Cao Shengxian, and Zhang Yanhui, “Temperature Prediction of Photovoltaic Panels Based on Support Vector Machine with Pigeon-Inspired Optimization,” *Complexity*, 2020.

- [21] L. Serrano-Luján, C. Toledo, J.M. Colmenar, J. Abad, and A. and Urbina, “Accurate thermal prediction model for building-integrated photovoltaics systems using guided artificial intelligence algorithms,” *Applied Energy*, 2022.
- [22] Alessandro Ortis and Sebastiano Battiato, “A new fast matching method for adaptive compression of stereoscopic images,” in *Three-Dimensional Image Processing, Measurement (3DIPM) and Applications*. International Society for Optics and Photonics, 2015.
- [23] “Multi-picture format - white paper dc-x007-2009,” .
- [24] Gregory K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, 1992.
- [25] Graham Hudson, Alain Léger, Birger Niss, István Sebestyén, and Jørgen Vaaben, “JPEG-1 standard 25 years: past, present, and future reasons for a success,” *Journal of Electronic Imaging*, 2018.
- [26] Sebastiano Battiato, Massimo Mancuso, Angelo Bosco, and Mirko Guarnera, “Psychovisual and statistical optimization of quantization tables for DCT compression engines,” in *International Conference on Image Analysis and Processing*. IEEE, 2001.
- [27] Charilaos Christopoulos, Athanassios Skodras, and Touradj Ebrahimi, “The JPEG2000 still image coding system: an overview,” *IEEE transactions on consumer electronics*, 2000.
- [28] Shiping Li, Mei Yu, Gangyi Jiang, Tae-Young Choi, and Yong-Deak Kim, “Approaches to H. 264-based stereoscopic video coding,” in *Third International Conference on Image and Graphics*. IEEE, 2004.
- [29] Woontack Woo and Antonio Ortega, “Stereo image compression with disparity compensation using the MRF model,” in *Visual Communications and Image Processing*. International Society for Optics and Photonics, 1996.
- [30] Udo Ahlvers, Udo Zoelzer, and Stefan Rechmeier, “Fft-based disparity estimation for stereo image coding,” in *International Conference on Image Processing*. IEEE, 2003.
- [31] Markus B. Schenkel, Chong Luo, Pascal Frossard, and Feng Wu, “Joint Decoding of Stereo JPEG Image Pairs,” in *International Conference on Image Processing*. IEEE, 2010.
- [32] Aysha Kadaikar, Gabriel Dauphin, and Anissa Mokraoui, “Joint disparity and variable size-block optimization algorithm for stereoscopic image compression,” *Signal Processing: Image Communication*, 2018.

- [33] T. K. Poolakkachalil and S. Chandran, "Stereoscopic image compression using curvelet," *Journal of Theoretical and Applied Information Technology*, 2019.
- [34] Priya Darshini Kumari, Susant Kumar Panigrahi, and M. Narayana, "Image Compression Algorithm based on Curvelet Transforms and Comparative Analysis with JPEG and JPEG 2000," *International Journal of Electronics Engineering Research*, 2017.
- [35] Thafseela Koya Poolakkachalil and Saravanan Chandran, "Summative Stereoscopic Image Compression using Arithmetic Coding," *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, 2019.
- [36] Thafseela Koya Poolakkachalil and Saravanan Chandran, "Analysis of Stereoscopic Image Compression Using Arithmetic Coding and Huffman Coding," in *International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018.
- [37] Xin Deng, Wenzhe Yang, Ren Yang, Mai Xu, Enpeng Liu, Qianhan Feng, and Radu Timofte, "Deep Homography for Efficient Stereo Image Compression," in *Conference on Computer Vision and Pattern Recognition*, 2021.
- [38] Wei Bao, Wei Wang, Yuhua Xu, Yulan Guo, Siyu Hong, and Xiaohu Zhang, "InStereo2K: a large real dataset for stereo matching in indoor scenes," *Science China Information Sciences*, 2020.
- [39] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision Meets Robotics: The Kitti Dataset," *The International Journal of Robotics Research*, 2013.
- [40] Imen Kadri, Gabriel Dauphin, Anissa Mokraoui, and Zied Lachiri, "Stereoscopic image coding using a global disparity estimation algorithm optimizing the compensation scheme impact," in *2020 Signal Processing: Algorithms, Architectures, Arrangements, and Applications*. IEEE, 2020.
- [41] Alessandro Ortis, Francesco Rundo, Giuseppe Di Giore, and Sebastiano Battiato, "Adaptive Compression of Stereoscopic Images," in *Image Analysis and Processing - ICIAP 2013*. 2013, Springer Berlin Heidelberg.
- [42] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh, "Deep Learning vs. Traditional Computer Vision," in *Science and Information Conference*. Springer, 2019.

- [43] Kai Briechle and Uwe D. Hanebeck, “Template matching using fast normalized cross correlation,” in *Optical Pattern Recognition*. International Society for Optics and Photonics, 2001.
- [44] Richard Hartley and Andrew Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2003.
- [45] Daniel Scharstein and Richard Szeliski, “High-accuracy stereo depth maps using structured light,” in *Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2003.
- [46] H. Christopher Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, 1981.
- [47] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, 2004.
- [48] Christopher G. Harris and Mike Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*. Citeseer, 1988.
- [49] Teuvo Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, 1990.
- [50] Daniel Scharstein and Richard Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International journal of computer vision*, 2002.
- [51] Daniel Scharstein and Chris Pal, “Learning conditional random fields for stereo,” in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2007.
- [52] Hirschmuller H. and Scharstein D., “Evaluation of Cost Functions for Stereo Matching,” in *Conference on Computer Vision and Pattern Recognition*, 2007.
- [53] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling, “High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth,” in *Lecture Notes in Computer Science*. Springer, 2014.
- [54] Anush Krishna Moorthy, Che-Chun Su, Anish Mittal, and Alan Conrad Bovik, “Subjective evaluation of stereoscopic image quality,” *Signal Processing: Image Communication*, 2013.
- [55] Amin Banitalebi-Dehkordi, Mahsa T. Pourazad, and Panos Nasiopoulos, “A human visual system-based 3D video quality metric,” in *International Conference on 3D Imaging (IC3D)*. IEEE, 2012.

- [56] J. Duarte, R. Cassin, J. Huijts, B. Iwan, F. Fortuna, L. Delbecq, H. Chapman, M. Fajardo, M. Kovacev, and W. Boutu, “Computed Stereo Lensless X-Ray Imaging,” *Nature Photonics*, 2019.