TÉCNICO LISBOA

Deep Learning model to detect and differentiate installed solar panels using global satellite imagery from Portugal

Luis Manuel Gonçalves Monteiro

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Sérgio Luís Proença Duarte Guerreiro

Examination Committee

Chairperson: José Carlos Martins Delgado
Supervisor: Sérgio Luís Proença Duarte Guerreiro
Member of the Committee: Rui Miguel Carrasqueiro Henriques

June 2023

Acknowledgments

First and foremost, I want to express my gratitude to Sérgio Guerreiro, my advisor, for taking on the challenge and providing suggestions and direction as the project was being developed.

Second, I want to express my gratitude to EDP for inviting me to participate in this project and for all the assistance and support they gave me as I developed this work. Their support was crucial to the outcome of this experience.

I also want to express my gratitude to my family and friends for their unwavering support.

Abstract

This dissertation explores the use of deep learning models to detect and differentiate installed solar panels using global satellite images of Portugal.

The challenge lies in the incomplete documentation and registration of solar PV installations, which are not centrally registered and vary between countries. Taking advantage of aerial and satellite images, this study addresses the need for automatic identification of photovoltaic solar installations in wide geographic areas.

In this work, two models are used, one for identification and the other for differentiation, and with this strategy, we achieved some good results. For the identification of solar panels, we obtained a precision and recall of 80,11 percent and 83,21percent, respectively. For the differentiation between solar panels, we also obtained a good result, although more modest, with an accuracy of 78,92 percent, but this value opens some doors for future work.

These findings contribute to our understanding of solar panel installation. Using deep learning models and satellite imagery, this research improves the identification and monitoring of solar photovoltaic installations, as well as solar and thermal installations, facilitating sustainable energy systems.

Keywords

deep learning ; renewable energy ; computer vision ; satellite imagery; solar panels

Resumo

Esta dissertação explora o uso de modelos de aprendizado profundo para detectar e diferenciar painéis solares instalados usando imagens globais de satélite de Portugal.

O desafio reside na documentação e registro incompletos das instalações solares fotovoltaicas, que não são registradas centralmente e variam entre os países. Aproveitando imagens aéreas e de satélite, este estudo aborda a necessidade de identificação automática de instalações solares fotovoltaicas em amplas áreas geográficas.

Neste trabalho são utilizados dois modelos um para a identificação e outro para a diferenciação e com esta estrategia conseguimos alguns bons resultados. Para a identificação de painéis solares, obtivemos uma precisão e recall de 80,11 porcento e 83,21 porcento, respectivamente. Para a diferenciação entre painéis solares também obtivemos um bom resultado, embora mais modesto, com uma precisão de 78,92 porcento, mas este valor abre algumas portas para trabalhos futuros.

Essas descobertas contribuem para a compreensão da instalação de painéis solares. Usando modelos de aprendizado profundo e imagens de satélite, esta pesquisa melhora a identificação e o monitoramento de instalações solares fotovoltaicas, bem como instalações solares e térmicas, facilitando sistemas de energia sustentável.

Palavras Chave

aprendizagem profunda; energia renovável ; visão computacional; Imagem de satélite; painéis solares

Contents

List of Figures

1

Introduction

Contents

1.1 Transition to Clean Energy

The energy industry has been responsible for almost three-quarters of worldwide greenhouse gas emissions in latest years, in other words, the climate crisis is mostly an energy challenge. With all this in mind, governments worldwide are working to decrease climate risks, promoting an effective transition to clean new energy industries, accelerating the creation of clean energy technology, and keeping energy security always first. This scenario needs to satisfy several important milestones, including limiting global temperature rise to 1.5 degrees Celsius while preventing temperature overshoot, decreasing global carbon dioxide (CO2) emissions to net zero by 2050, and significant improvements in air quality [8].To achieve those milestones, others aspects cannot be left unanswered. For example, it is imperative to consider the United Nations(UN) Sustainable Development Goals(SDGS) for energy, which aims to *"Ensure access to affordable, reliable, sustainable and modern energy for all."* [9] until the year 2030.

According to the International Renewable Energy Agency (IRENA) [10], solar photovoltaic power would be the renewable energy that will increase more, representing only 2 percent in 2018 to an astounding 25 percent in 2050 of global electricity generation, therefore, a growth of 1250 percent. With the rise in the number of solar panels, it is noteworthy that the energy industry can indeed mitigate 21 percent of the overall energy-sector CO2 emissions (approximately 4.9 Gt CO2).

1.2 Motivation Examples

The fast deployment of solar energy resources is also transforming the electric grid into a cleaner energy network. However, a comprehensive database collecting correct location and size information for PV systems, particularly distributed household solar panels, is still lacking globally. In Portugal, for example, it is known that there is an excess of production in Baixo Alentejo [11], and that energy must be transmitted through the power grid. That's why, there would be increased congestion and voltage fluctuations in the power grid, making monitoring and operation of energy balance management more complex.

Policymakers and solar companies are also interested in this type of remote mapping because, in several regions around the world, this type of stakeholders are using data from specific regions or limited groups of residents and globalizing this data for the entire country, which can be misleading [12]. This work may be of significant interest to these stakeholders since it provides a better knowledge of the geographical coordinates of solar PV systems, allowing solar enterprises to create more efficient and accurate market projections and policymakers to design better policies for supported and encouraged solar energy [13].

1.3 Problem

Not all of this solar photovoltaic generation is precisely documented or registered in any form of the central database, and the depth of understanding about the distributed solar photovoltaic energy varies significantly from one country to another. In the Netherlands, a study carried out by the company Stedin in collaboration with Sobolt estimates that one in four homes with solar panels are not registered [14]. In certain circumstances, network operators maintain records of the precise locations and capacity of all solar PV installations linked to their networks. Even so, it is probably that certain systems are unregistered for or incorrectly reported, or that lacuna in the register exists. Furthermore, the usage of solar PV in off-grid applications is quickly expanding [10] and such systems are likewise unlikely to be registered with any government authority, although it may be important to be aware of them in order to track the overall number of solar PV installations in an area.

As a result, there is significant interest in using satellite and aerial images to enable automatic identification of the exact positions and capabilities of solar PV installations across wide geographic areas.

There are 2 major problems that were addressed in this work, the first was the classification of images with a resolution of 11cm/pixel, these being Portuguese images with typical characteristics of the country, the second problem addressed was the problem of differentiating thermal solar panels from photovoltaic solar panels.

Taking into account the conclusions obtained in the section 2.2 we verified that the best results obtained for images of 10cm/pixel achieve a precision of 64 percent and a recall of 87 percent, this with German images as with the Portuguese images that have its particularities and a small drop in resolution reaches lower values shown in appendix A, and none of the works seen by me pay special attention to different types of solar panels.

1.4 Approach Description

To address the problems mentioned in the above section, the work was divided into two major parts, the first part is the Classification Model and the second part is the Differentiation Model.

This technique of dividing into two large parts is to distance the problems so that the results of one do not interfere with the other.

The objective of the classification model is to categorize images with a resolution of 11cm/pixel that are Portuguese. In order to attack this problem, was to use the technique of intelligent division of the datasets, with the use of neural networks with better results of the state of the art of the ImageNet, which also holds a possibility of being more effective at solving these problems, and also a series of techniques used to combat the problems that were to be obtained throughout the project.

The challenge addressed by the differentiation model is making a distinction between solar thermal

and photovoltaic panels. In order to do this, I built a special data set for the distinction of solar panels and applied the most advanced techniques also utilized by the other model, adjusting to this problem.

2

State of art

Contents

2.1 State of the Art

Solar panel detection using satellite imagery is an active topic of research, with significant progress made in recent years, which can be broadly separated into two categories: standard image processing-based methods and deep learning-based methods.

Deep learning-based algorithms, have showed considerable promise in spotting solar panels in satellite photos. Convolutional neural networks (CNNs) are used in these methods to automatically learn attributes from input photos and detect solar panels. Deep learning-based approaches have the benefit of being able to detect solar panels with complicated geometries and being more resistant to partial occlusion. These approaches, however, necessitate a substantial amount of annotated data for training and can be computationally demanding.

However, applying deep learning-based algorithms for solar panel detection has several significant drawbacks. Some of these challenges include variations in variations in the size and shape of solar panels, in the sense they come in a variety of sizes and shapes therefore it is challenging to train a model that can recognize all different types of solar panels. Another key challenge when it comes to using deep learning-based techniques to detect solar panels is the interference from other objects when obtaining an aerial image. Objects, such as trees, buildings, and highways, may obstruct the detection of solar panels.

There has been a considerable interest in the use of satellite or aerial imagery with the assistance of some machine learning techniques to identify various types of structures, including buildings and roads [15] [16] [17] [18] - [19]. This interest is perceived by the various papers written and the number of published versions and citations of these.

Machine learning has also been used to locate rooftops as well as determine the best locations for a solar panel installation, in the united states of America context [20] [21], the European context [22] and the India context [23]. Although these publications are not directly related to my research, I believe it is important to understand how detection is performed on ”easier to identify” structures.

Although the use of satellite and aerial images for identifying solar PV systems is still in its early stages of development. It is important to note that the first contribution in the field of solar panel identification is [24] and it was only published in 2015, there have been contributed massively of techniques to approach the resolution of the problem.

In the initial phase the technique used was support vector machines (SVMs) [24], and tiny image datasets were used, using images of houses extracted from publicly available satellite images obtained from the U.S. Geological Survey (USGS). The images were collected over the city of Lemoore, California during the year 2014 with a resolution of 0.3m, after manually cutting a rectangular region of the image, the data was extracted around each house, containing the roof of the house and some surroundings. The collection of all images of the houses corresponds to a total surface area of 0.196 square kilometers,

approximately 100 houses.

The research of Stephen Lee [24] was, to the best of my knowledge, the first work dealing with large-scale solar panel detection from imagery. Using a significantly bigger dataset of 12 square kilometers of manually labeled training data with a spatial resolution of 0.3 meters. The size of the dataset is not the only thing new in this project, he also starts using neural network architecture more precisely convolutional neural networks (CNNs), to do the classification of the images.

For the DeepSolar research [25], Jiafan Yu initially uses transfer learning to train a Convolutional Neural Network (CNN) classifier using 366,467 images sampled from over 50 cities/towns across the United States of America with only image-level labels identifying the presence or absence of panels. The model incorporates both image classification and semantic segmentation. The image classification section is to identify whether there are solar panels in the image and the semantic segmentation section is to estimate their size.The classification branch is developed based on Google Inception V3, which is pre-trained on ImageNet. The segmentation branch does not need another forward pass, the results are obtained by implementing a threshold to the Class Activation Maps (CAMs). This model was one of those that achieved incredible values, reaching a precision of 93.2 percent and a recall of 88.9 percent.

The paper [26] reports on one of Switzerland's first initiatives to map the locations and sizes of existing panel solar rooftop installations. This study is noteworthy not only because it is one of the first attempts in Switzerland, but also because it uses a semantic segmentation technique to classify and measure the size of the solar panel. For this purpose, Roberto Castello uses the convolutional neural network in the form of a U-Net, as this is the most popular CNN architecture used for rapid and precise segmentation of images.

One of the latest works on this topic is [12], in this paper, the author converted the original TensorFlow model from the Stanford paper [25] into PyTorch. The author worked to supplement the previous line of research from [25] by accomplishing a precision of 92 percent and a recall of 98 percent using a Google Maps-based model (with a spatial resolution of 5 cm/pixel). For the dataset creation, the author relied in a four-step strategy to compose aerial imagery datasets for efficiently optimizing solar panel system classification models and to reduce the size of the dataset: defining landscape types relevant to the area of interest (ex: urban, rural, farms and pastures, woodlands), determining landscape weighting, retrieving sample images based on weights, and eliminating the image duplicates.

2.2 Conclusion

In summary, we can conclude that the works we have just seen were particularly successful, with extensive terrain mapping, outstanding precision, and recalls. According to one research, the authors has mapped all solar panels systems in the United States with a reported precision of 93.1 percent and re-

call of 88.5 percent. The other one managed to map the most populous state of Germany, North-Rhine Westphalia, with the reported precision of 92 percent and a recall of 98 percent with high resolutions image (5 cm/pixel), and with some more train he can achieve a precision of 64 percent and a recall of 87 percent with a lower resolutions images (10 cm/pixel).

Despite the fact key milestones have already been reached, there are still certain challenges to be addressed, such as the requirement for high-resolution images, the need to sort all training images manually, the difficulty of categorizing images from different locations, and the non-clear distinction among solar PV panels and solar thermal panels.

Various issues are linked to the difficulties of categorizing images from other locations. For example, while solar panels are the same from country to country, there are other structures on rooftops that make categorization difficult. Another example is the usage of Google-based images, because of the quality varies greatly from city to city, even within the same country. For a more concrete example, the company acting in the energy sector that despite the attempting to use different images and using distinctive threshold values, it was feasible to determine that with all of these alterations, the inferred values were not reaching the precision and recall values obtained in the study [12]. These results are attached in Appendix A.

3

Background

Contents

3.1 Solar Panels

Solar thermal panels and photovoltaic panels expertly convert the sun's energy into more useful forms of energy such as heat and electricity. These are two distinct sources of renewable energy. Even though they both collect energy from the sun, the similarity ends there. It is critical to recognize that solar thermal panels and photovoltaic panels are two completely different technologies.

The most common solar panels are thermal solar panels. They have an efficiency rate of roughly 70 percent, compared with the solar photovoltaic panels, at the moment, only have half the efficiency rate (35 percent). To achieve the same level of production, it will be necessary to expand the area they cover. To better understand the magnitude of these efficiency rates, one can use an example: consider a two-person household, where it is essential to use 1 to 2 thermal solar panels to reduce the requirement for hot water, this is equivalent to 2 or 4 square meters of panel. In order to suppress the energy requirement, it would be necessary to have around 12 photovoltaic solar panels, which means it will be necessary 22.8 square meters of panels.

In the following sections, I will go over the most popular forms of solar panels. Figure 3.1 illustrates a few examples.

Figure 3.1: Top Left: Flat-Plate Collectors with active collector system; [1] Top Right: Evacuated-Tube Collector with Thermosiphon; [1] Bottom Left: Monocrystalline solar panels; Bottom center: Polycrystalline solar panels; Bottom Right: Thin-film solar panels [2]

3.1.1 Solar Thermal Panels

Solar thermal panels, also known as solar collectors, are roof-mounted devices that collect the heat of the sun and use it to heat the liquid in the tubes, which ultimately heats the water contained in a cylinder, leaving this hot water ready for use. The liquid that circulates through the panels is a combination of water and antifreeze. This technology's primary function is water heating using as little space as possible. This type of technology is a very popular choice for swimming pool heating. Solar thermal panels are categorized into two major categories: Flat-plate collectors and Evacuated (or vacuum) tube collectors.

The most common type of solar collector in Europe is a flat-plate solar collector, which looks extremely similar to a solar photovoltaic panel, therefore distinguishing between the two is challenging [27]. They are made up of a dark absorbing surface, a transparent cover, a heat insulating backing, and, most crucially, a fluid that allows heat to be conducted from the absorber to a water tank. The absorber can be manufactured of a variety of materials. The technology is typically used in residential buildings where there is a high demand for hot water, which influences expenses. Flat-plate collectors are commonly

used in commercial applications such as car washes, swimming pool complexes, laundromats, military laundry facilities, and restaurants.

In contrast, in China, evacuated-tube collectors are dominant, and they are composed of an absorber strip that is mounted in evacuated and pressure-proof glass tubes [27]. The heat transfer fluid is pumped straight from the absorber into a U-tube system or a tube-in-tube system. The special fluid that circulates in the heat pipe collector evaporates very easily even at low temperatures, causing steam to rise in the individual heat pipes and warm up the fluid in the main pipe, resulting in heat generation. Evacuated tube collectors when compared to flat plate panels are more efficient, especially in cold environments. On the negative side, they lose efficiency throughout hotter seasons due to the possibility of overheating. This happens because vacuum tubes are created to prevent heat loss and hence differ from flat plate panels, which tend to lose more heat.

In addition to these two main types of solar collectors, it is also important to understand a little better the types of systems behind the collectors. These are divided into two groups, passive collector system or thermosiphon and active collector system. An easy way to see the difference is that a passive collector system transfers heat from the collector to the storage tank, which is located just above the collector, simply following the natural convection principle.

This type of system is deployed around the world as domestic systems. In 2015 they had 77 percent of the total number of panels, thus making the most popular, but the popularity has been declining, so in 2019 the percentage was only 57 percent [27]. The active collector heating systems, on the other hand, employ electric pumps, controlled by temperature sensors, to transmit the heat collected from the solar collectors into the building [28].

3.1.2 Solar Photovoltaic Panels

Solar photovoltaic panels are also roof-mounted, but these devices absorb photons and generate flowing electricity. This process varies depending on the type of solar technology used, however, there are a few processes that are shared by all solar PV cells. When light reaches a photovoltaic cell, photons are absorbed by the semiconducting material, which is typically silicon. This photon flux energy knocks electrons free in the silicon. In photovoltaic cells, two layers of silicon are utilized, and each is particularly treated to form an electric field, which means that one side has a positive charge and the other side has a negative charge. This electric field causes loose electrons to move in one direction through the solar cell, resulting in the generation of an electrical current. When an electrical current is created inside the panel, metal plates on the sidewalls of each solar cell gather and transport the electrons to wires. Electrons may now flow as energy through the wire to a solar inverter and subsequently throughout the household.

Inaccurate placement of solar panels can majorly reduce the effectiveness of electricity generation.

If you live in an area surrounded by large buildings or tall trees, the efficiency of your solar panels will be very limited. Ensuring there are no buildings or trees blocking sunlight is much more important for solar photovoltaic panels than for solar thermal panels because they have a much lower efficiency rating. Currently there are some models of photovoltaic panels on the market, which present substantial variances between them, from the raw material to the efficiency of each model. Monocrystalline panels, polycrystalline panels, and thin-film solar panels are the most predominant forms of residential photovoltaic panels.

Monocrystalline solar panels derive their name from the fact that they are composed of a single silicon crystal. They are the oldest and the most efficient type of solar photovoltaic panels on the market and may be distinguished by their sleek, black appearance. This black appearance is achieved due to the use of monocrystalline silicon which is sliced into a wafer during the manufacturing process. These slices allow electrons to move freely, resulting in high-efficiency rates ranging from 15 to 20 percent. As a result, they are also the most space-efficient and expensive type of household solar photovoltaic panels.

Polycrystalline solar panels are created by melting many chunks of silicon together and creating the solar cells in square molds. Because each solar cell has several crystals, there is less area for electrons to flow around, resulting in a reduced efficiency rate compared to monocrystalline cells but still reaching an efficiency rate between 13 to 16 percent. Polycrystalline panels are easier to manufacture than monocrystalline panels, making them less expensive and generating fewer wastes during the manufacturing process. Instead of the black tint of monocrystalline panels, they typically have a blue appearance.

Thin-film solar panels are less expensive to manufacture since they need less material. As a result, they are the most affordable form of solar photovoltaic panels on the market. Thin-film solar panels are created by sprinkling one or more Photovoltaic materials onto a substrate and can be formed into a flexible panel. Thin-film cells come in a variety of materials, including silicon, cadmium, and copper. Thin-film panels have the lowest efficiency on the market, ranging from 6 to 16 percent, and therefore take up the greatest space. They are more heat tolerant, making them appropriate for hotter climates.

3.2 Theoretical Foundation

As we could see in the chapter on ”State of the Art” there have already been several iterations with very similar themes with various techniques and different ways of trying to reach a satisfactory result.

The abrupt development of artificial intelligence and deep learning has led to the development of numerous techniques.

3.2.1 Data augmentation

Data augmentation is a technique particularly useful in computer vision tasks, where the input data is often images. For instance, let's say you have a feature that includes photographs, but your dataset doesn't include enough examples of photographs for the model to develop useful correlations so that it can successfully label the photographs.

Data augmentation consists of artificially increasing the quantity and diversity of training examples by generating new examples from the existing ones. To achieve this, the original data is subjected to a series of transformations, generating new images that are visually similar to the original ones, but with different characteristics that the model can learn from. Some examples can include rotation, flipping, cropping, scaling, and different levels of brightness and contrast. This can enhance the model's generalization capabilities and minimize overfitting by exposing it to a wider range of input data variations.

This type of manipulation is not always effective since some manipulations make learning more challenging. For example, rotating an image of a dog by 180 degrees should still result in an image of a dog, not a cat, but if apply the same rotation on an image of a "6" become an image of a "9" and not a 6" [29].

3.2.2 Hyperparameters

Hyperparameters in deep learning are parameters made previous to the training process as opposed to ones that are learned from the training data. They specify how the learning algorithm should be configured and behave, and they can have a big impact on how effectively the model performs.

A deep learning model's efficacy and generalizability are greatly influenced by its hyperparameters. To produce the best outcomes for a particular task and dataset, they must be properly chosen or adjusted.

Several methods, such as grid search and random search, can be used to find the optimal hyperparameter values. To determine the best configuration, one must experiment with various combinations of hyperparameters and assess how well they perform on a validation set. It is an iterative process that calls for developing and testing various models.

It's critical to remember that fine-tuning hyperparameters can be a time-consuming and computationally expensive procedure. Therefore, it is important to strike a balance between the investigation of a wide range of hyperparameter values and ensuring the effective utilization of computational resources.

3.2.3 Transfer Learning

A machine learning technique called transfer learning involves employing knowledge obtained while resolving one problem to help solve a different issue different but related. In the context of computer

vision, that is our situation, transfer learning refers to using a pre-trained deep neural network, which has previously learned features from a large dataset, as a starting point for training a new neural network on a smaller dataset.

Transfer learning is an effective technique because it allows us to avoid spending a lot of time and effort that would otherwise be required to train a new model from scratch. In addition, transfer learning has been demonstrated to improve the performance of deep neural networks in a variety of computer vision applications, including object detection and image classification, as we can see in the paper [30].

4

Methodology

Contents

4.1 Prototype Framework

For the development of the prototype, two models were created, the Classification Model and the Differentiation Model.

The prototype works as follows: an image is first run through the Classification Model, which will categorize the images with one of two possible outputs ”Negative” or ”Positive”. If no solar panel is identified, the output will be ”Negative” the procedure finishes here, and the categorization of the image is without a solar panel. If a solar panel is identified, the image is classified as ”Positive” and will be delivered to an additional algorithm for the Differentiation Model. The Differentiation Model classifies the solar panels present in the images as, ”Solar Panel Thermal” or ”Solar Panel Photovoltaic”, As shown in the Figure 4.1.

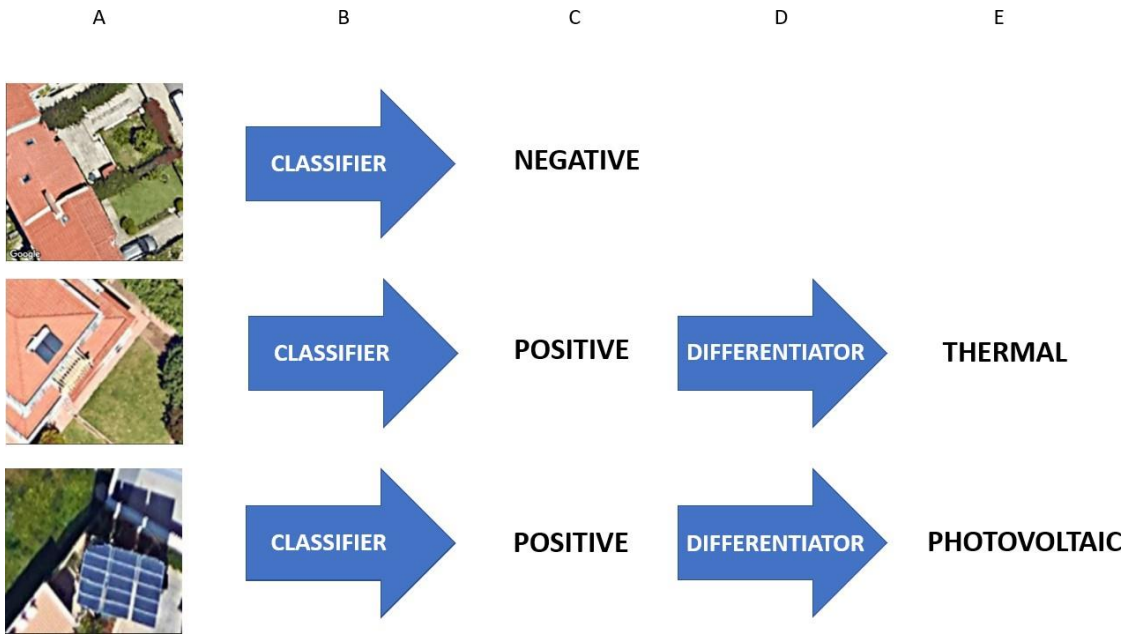

Figure 4.1: Prototype Framework

A - The input satellite imagery obtained by Google Maps.

B - The Classification Model is applied

C - The results returned by the Classification Model

D - The Differentiation Model is applied

E - The results returned by the Differentiation Model

4.2 Data

I'll describe how the data was gathered in this section, along with some challenges I encountered. I'll begin by discussing the difficulties in the subsection 4.2.1 before moving on to the development and methods employed to make the collection in the subsection 4.2.2.

4.2.1 Data Problems

Some of the datasets that EDP made available and that were used for testing and as well as training. They have several issues, such as incorrectly classified photographs, distorted photographs, and an absence of a wide variety of different types of regions and solar panels.

In terms of image distortion, we can see in the Figure 4.2 below, which displays the 4 most significant distortion issues that the images had. In the lower right corner, you can see several lines cutting the images from edge to edge, in the lower left corner we can see that the upper half of the images meet with a resolution and the lower half with a much lower resolution, in the upper right corner we can see black spots on the images and in the upper left corner we can also see the black spots but I wanted to draw attention on the other problem that this images has which is the light spots that resemble white noise.

In terms of lack of variety in regions and topographies, the images were taken from two localities Belas and Birre, and from the Centro Solar Fotovoltaico de Ourika, which gives us a view of different typologies but not in sufficient numbers because it does not meet the division of the paper [12], which I wanted to replicate with the division between urban, rural, fields and forests, thus giving a larger amount of image to the urban, a smaller amount to the rural but also significantly later with a smaller amount the rural, and with a very small amount we have the fields and forests

A large part of the images in Belas were of forests, without buildings or with few buildings, with few positive examples, while Birre had some good images with houses and cleaner land but without much variety, because he did not have images of buildings or more densely populated areas such as Greater Lisbon, and the Centro Solar Fotovoltaico de Ourika, as the name implies, is a Photovoltaic Center that consists of large areas with solar panels that can simulate, roughly, solar panels of large companies such as large areas of solar panels, in a not very accurate way because the solar panels of the Centro Solar Fotovoltaico de Ourik are installed on the ground which is quite homogeneous, whereas in large companies the panels are usually installed on roofs together with skylights and machinery that makes it very difficult to your identification of the solar panels.

Figure 4.2: Images With Anomalies

4.2.2 Data Collection

The datasets provided by EDP had been analyzed, and after the cleaning, and the reclassification, I discovered that there simply were insufficient photographs containing solar panels in other words photographs classified as positive.

After finding the dataset problems described in subsection 4.2.1 in the datasets provided by EDP. I conducted an analysis of the images provided by EDP, and after reclassifying and cleaning each photograph that had a defect, I found that the final dataset simply had an insufficient number of photographs, both in terms of diversity and the number of photographs including solar panels, i.e., photographs classified as positive. These positive images represented only 3.4 percent of the images in the entire dataset, there was also a bias with the images as they were very similar in terms of the zones in which they were taken, as they were all taken in zones with essentially the same topography, which does not accurately represent Portugal as a whole. This bias was explained in subsection 4.2.1.

So to fill in the lack of photographs of solar panels and the lack of diversity I made a plan to collect

photographs from different regions of the country with different topographies, including Restelo, Faro, Birre, Belas, Benfica, and some areas with more enterprises, such as the Sport Lisboa e Benfica Pool Complex (Aquared) and the Ourika Solar Photovoltaic Centre this way makes the model more resilient, trying to divide the dataset into urban, rural, fields and forests as was done in the study [12].

The collection of photographs was made using 2 techniques. The first collection technique consists of collecting photographs using software provided by EDP. This software provided by EDP has a cost for each photograph collected, therefore the images collected by this software only represent a small portion of the entire dataset, the majority of the photographs collected with this software were utilized in the test dataset, and the validation dataset. A better way to describe the software is to say that it asked for two geographic positions and the appropriate zoom value and that with these two positions on the map, images were made consecutively, all of which had the same size, as shown in the images below. Given the two points in the image are represented by the two blue dots, a grid of photographs all of the same size is created with this one represented by the yellow grid, as shown in Figure 4.3.

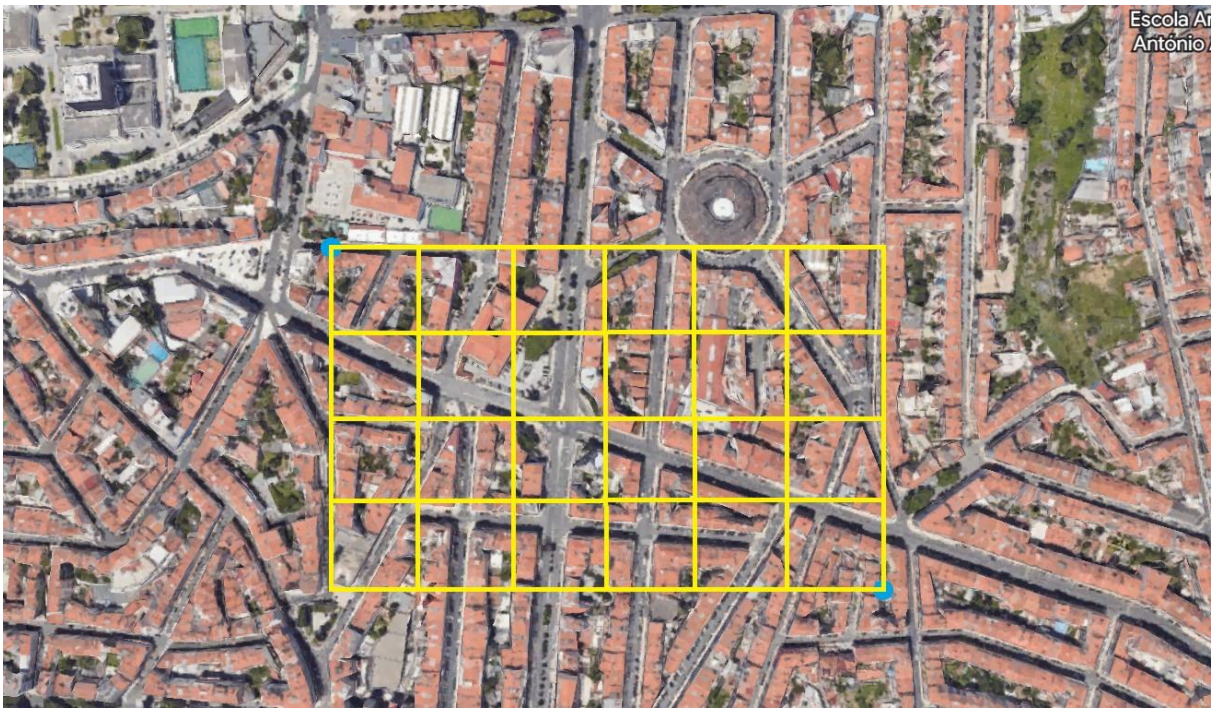

Figure 4.3: Software Provided By EDP

This method enables us to create an uninterrupted map between the two geographic positions. With the photographs collected with this software, it is possible to create an easy and homogeneous map.

The second technique used was a manual collection of photographs, which was inspired by the technique used in the paper [12] which mentions the significance of selecting different landscape types, with and without solar panels, and certain percentages of the landscapes because not all of them the

landscapes are equally significant for a good dataset.

The photographs were collected using Google Maps, these photographs have different sizes and resolutions between them, that's why the section 5.3 is so important. The fact that the images have different sizes and resolutions makes the model more resilient to different photographs.

The difference between the two techniques is the image sizes and the resolutions are what sets them apart the most, with the first technique having all images with a resolution equal to 11 cm/pixel and the second having no defined resolution, despite the two collections techniques, it should be noted that all the labeling was done manually.

4.3 Model Architecture and Techniques

In the next chapters, I will talk about the techniques and architectures that were used in this project, we will discuss how they work and their strength. Let's start with the architecture before turning our attention more toward the techniques used.

4.3.1 Architectures

Deep convolutional networks have been at the heart of the most significant breakthroughs in image recognition performance in recent years. The Inception-V3 architecture is a good example of that when we see the quality of the classifications, for example, has been demonstrated to deliver extremely excellent performance at a minimal computational cost that we can see in the chapter 2.

At present, there are some architectures that achieve better results and converge faster, requiring fewer epochs to accomplish the results. Two good examples are Inception-ResNet-V2 and EfficientNetV2-S, which I will describe in detail next.

4.3.1.A Inception-ResNet-v2

Inception-ResNet-v2 is a state-of-the-art deep convolutional neural network architecture proposed in 2016 by Google researchers. It combines the strengths of two previously suggested architectures, Inception and ResNet, to obtain increased performance, which we will discuss following.

Inception architecture presented by Szegedy et al. [31], published in 2014, tries to extract features at numerous scales by employing several concurrent convolutional layers with varying kernel sizes. This enables the network to capture information at many levels of abstraction effectively, making it appropriate for a wide range of computer vision applications. However, because of their vast number of parameters, Inception networks are prone to overfitting.

He et al. [32] presented the ResNet design in 2015 to address the problem of disappearing gradients in very deep networks by incorporating residual connections between layers. When compared to standard deep neural networks, this allows the network to have very deep structures with substantially fewer parameters.

Inception-ResNet-v2 combines the strengths of these both architectures by providing residual connections between inception modules. This enables the network to have extremely deep structures with far fewer parameters than standard deep neural networks while efficiently collecting information at different scales.

The architecture of Inception-ResNet-v2 includes various performance-enhancing features such as batch normalization, factorized 7x7 convolutions, global average pooling, and auxiliary classifiers. Batch normalization is used to normalize the input of each layer and minimise the influence of covariate shift. Factorized 7x7 convolutions are used to minimize the number of parameters and increase network efficiency. While global average pooling replaces typical fully linked layers by averaging the activations of each feature map to yield a single output. And auxiliary classifiers are used in the network's intermediary layers to increase gradient flow and regularize the network.

In conclusion, Inception-ResNet-v2 has demonstrated cutting-edge performance on a variety of computer vision tasks such as picture classification, object identification, and segmentation. For example, in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2016, Inception-ResNet-v2 outperformed prior state-of-the-art models with a top-5 error rate of 3.08 percent [33].

In [3], they present several new streamlined architectures for both residual and non-residual Inception networks. The author compares the two pure Inception variations, Inception-v3 and Inception-v4, with similarly expensive hybrid Inception-ResNet versions in this paper. To be sure, those models were chosen somewhat haphazardly, with the major limitation being that the parameters and computing complexity of the models be roughly comparable to the cost of the non-residual models. In fact, the author evaluates larger and broader Inception-ResNet versions on the ImageNet classification challenge dataset, and they performed fairly similarly. The author discovered that improvements in single-frame performance do not transfer into comparably big improvements in ensemble performance. Nonetheless, comparing the results of the first trial with Inception-v3 to the final experiment with Inception-ResNet-v2, we can declare a 3.4 percent improvement in top-1 error on the validation set of the ImageNet classification challenge, as we can see in the following figure.

After examining the data above, we observe that the difference between Inception-v4 and Inception-ResNet-v2 is negligible, with both achieving extremely equivalent accuracy values, with Inception-ResNet-v2 exceeding 0.1 percentage point. The ability that attracted my attention most about Inception-ResNet-v2 over Inception-v4 was the ability to converge faster, therefore decreasing training time.

To make my argument more clear, I've included the Figure 4.5. It can be seen that Inception-ResNet-

Network	Top-1 Error
BN-Inception [6]	25.2%
Inception-v3 [15]	21.2%
Inception-ResNet-v1	21.3%
Inception-v4	20.0%
Inception-ResNet-v2	19.9%

Network	Crops	Top-1 Error
ResNet-151 [5]	dense	19.4%
Inception-v3 [15]	144	18.9%
Inception-ResNet-v1	144	18.8%
Inception-v4	144	17.7%
Inception-ResNet-v2	144	17.8%

Figure 4.4: Results of the experiences of the different models [3]

v2 begins its plateau around epoch number 120, whereas Inception-v4 begins its plateau after epoch number 130.

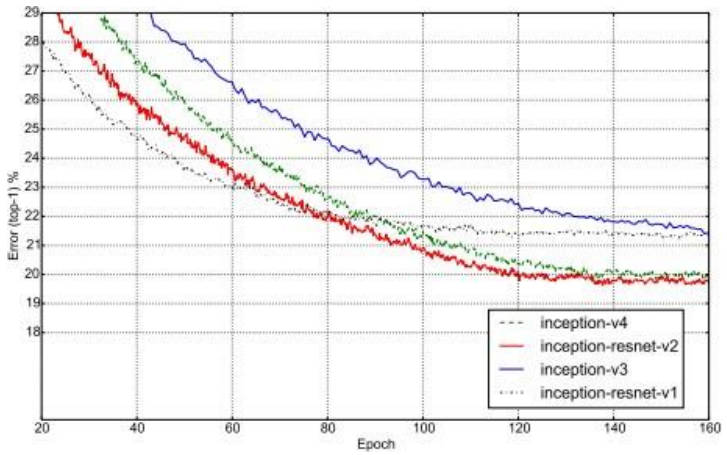

Figure 4.5: Top-1 error evolution of all four models discussed in the paper [3]

4.3.1.B EfficientNetV2-S

EfficientNetV2-S [4] is a family of neural network architectures that improve on the original EfficientNet models by introducing a number of significant breakthroughs that reduce the model size and speed up training. It includes numerous significant advances in order to make the models smaller and easier to train while maintaining the same performance. These techniques include gated linear units, swish activation, dropblock regularization, width-depth product scaling, and stochastic depth.

Swish activation function [34] is a self-gated activation function that has been demonstrated to outperform more conventional activation functions like ReLU. The swish function is defined as 4.1.

$$F(x) = x * \frac{1}{1 + e^{-x}} \tag{4.1}$$

Scaling up the model capacity while retaining high efficiency and accuracy is the key objective of EfficientNet-V2-S [4]. A compound scaling technique is used to build the architecture, which scales

the model's depth, width, and resolution in harmony to improve performance. Additionally, the study demonstrates that EfficientNet-V2-S may perform at the top position on image classification and object recognition tasks while using a smaller model size and shorter training times compared to other architectures.

	Model	Params	ImageNet Acc.	CIFAR-10	CIFAR-100	Flowers	Cars
ConvNets	GPipe (Huang et al., 2019)	556M	84.4	99.0	91.3	98.8	94.7
	EfficientNet-B7 (Tan & Le, 2019a)	66M	84.7	98.9	91.7	98.8	94.7
Vision Transformers	ViT-B/32 (Dosovitskiy et al., 2021)	88M	73.4	97.8	86.3	85.4	-
	ViT-B/16 (Dosovitskiy et al., 2021)	87M	74.9	98.1	87.1	89.5	-
	ViT-L/32 (Dosovitskiy et al., 2021)	306M	71.2	97.9	87.1	86.4	-
	ViT-L/16 (Dosovitskiy et al., 2021)	306M	76.5	97.9	86.4	89.7	-
	DeiT-B (ViT+regularization) (Touvron et al., 2021)	86M	81.8	99.1	90.8	98.4	92.1
	DeiT-B-384 (ViT+regularization) (Touvron et al., 2021)	86M	83.1	99.1	90.8	98.5	93.3
ConvNets (ours)	EfficientNetV2-S	24M	83.2	98.7±0.04	91.5±0.11	97.9±0.13	93.8±0.11
	EfficientNetV2-M	55M	85.1	99.0±0.08	92.2±0.08	98.5±0.08	94.6±0.10
	EfficientNetV2-L	121M	85.7	**99.1**±0.03	**92.3**±0.13	**98.8**±0.05	**95.1**±0.10

Figure 4.6: EfficientNetV2 Performance Results on ImageNet [4]

According to the study, the EfficientNet-V2-S model is more effective and efficient since it can accomplish a high degree of accuracy with fewer parameters and also less computation time.

4.3.2 Regularization

Regularization is a method for keeping machine learning models from overfitting. When a model becomes excessively complex, it begins to memorize the training data rather than understand the underlying patterns, which causes the model to perform badly on data that is not observed. Regularization techniques aim to minimize the complexity of the model and prevent it from memorizing the training data, while still preserving the model's capacity to recognize underlying patterns and correlations. These techniques impose limitations or penalties on the learning process, encouraging the model to discover representations that are more straightforward and reliable. There are several types of regularization techniques. In this thesis, two techniques were used: Early-stopping and Dropout.

4.3.2.A Dropout

Backpropagation learning creates fragile co-adaptations, These complicated relationships will result in sampling noise, which will exist in the training data but will not generalize well to new data. Dropout is a neural network improvement strategy that addresses this issue. The basic concept is to randomly eliminate neurons and connections from the neural network while it is being trained. As a result, the neurons are prevented from over-adapting. Dropout samples from an exponential range of several relatively smaller networks are used during training. By utilizing a single unthinned network with a decreased weight at test time, it is simple to replicate the impact of averaging the predictions of all these smaller networks.

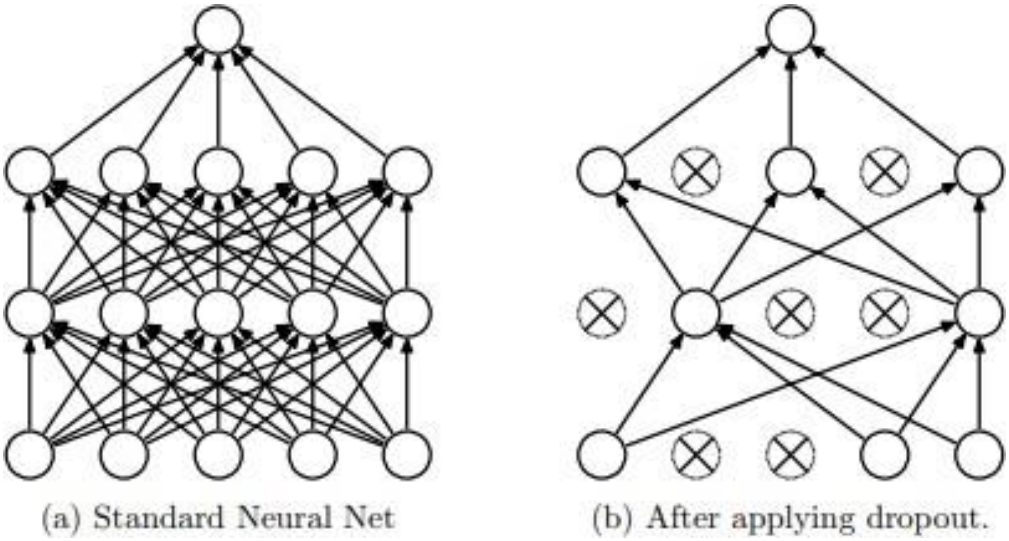

Figure 4.7: Image A without dropout and image B with dropout [5]

As seen in the figure, each hidden layer ignores a percentage p of randomly selected neurons and their corresponding connections during the training process for each training sample and each iteration.

4.3.2.B Early stopping

Early stopping may seem counter-intuitive. That is, it may seem illogical to instruct a model to terminate training while the loss is still lowering. however, overfitting could indeed result from overtraining a model. A model that has been trained for a long period of time may mimic the training data so well that it cannot accurately make predictions on new samples, In other words, when the validation loss stops converging with the training loss, as we can see in the Figure 4.8.

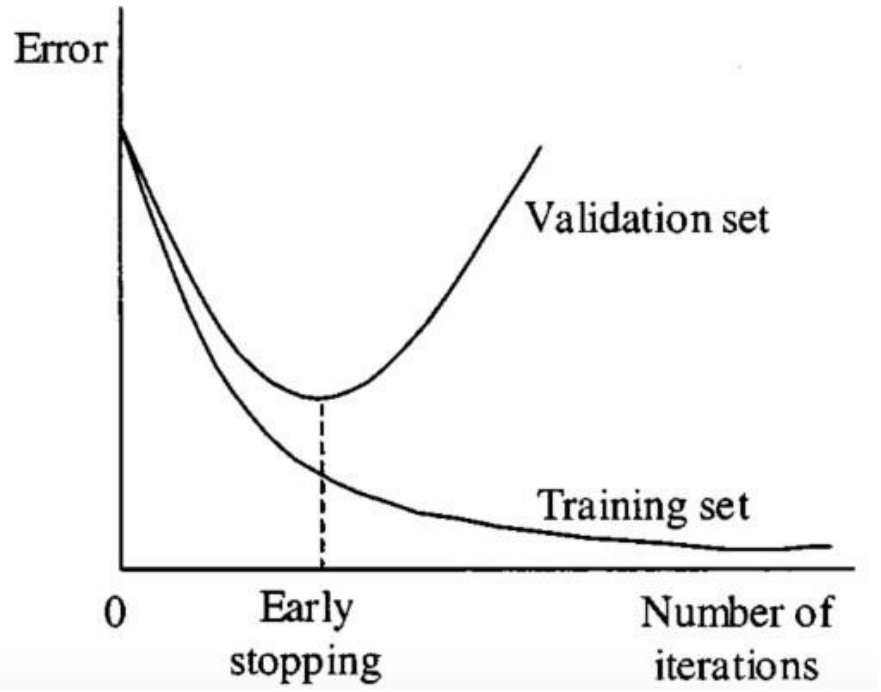

Figure 4.8: Early Stopping [6]

In order to employ this regularization, we deliberately stop training the model when the loss on a validation dataset starts to climb, which is when generalization performance declines. This is not a formal regularization method, but it can successfully reduce overfitting.

4.3.3 Class Weights

”Class Weights” is a technique used to address the common problem when working with the imbalanced classes present in a dataset, an issue we can encounter when solving classification problems. Class imbalances occur when the number of samples in one or more classes is significantly higher than the number of samples in the other classes. As a result, the model may end up behaving as a biased model and poorly performing towards the minority class or classes resulting in the model to frequently predicting in favor of the majority class. One method used to deal with class imbalances is class weights, which are parameters that are used to modify the loss function of the model in order to penalize misclassifications of the minority classes more severely than misclassifications of the majority class.

To solve this issue, class weights are one method. In essence, class weights are employed in training to give the minority class a higher priority or emphasis. This is accomplished by giving the minority class during model training a higher weight.

The ratio of instances in the majority class to instances in the minority class is often the first step in

calculating class weights. Following that, the weights for the majority and minority classes are set to be equal to the ratio's inverse. This makes sure that the weight for each class adds up to 1.

During training, the model then assigns more importance to the minority class by adjusting the loss function based on the class weights. Essentially, errors made by the minority class are penalized more heavily than errors made by the majority class.

$$ClassWeights(i) = \frac{TotalSamples}{NumberClasses * ClassSamples(i)} \tag{4.2}$$

where

- TotalSamples is the total number of samples in the training set
- NumberClasses is the number of classes in the dataset
- ClassSamples (i) is the number of samples in class i

By using class weights, the model is encouraged to learn patterns in the minority class more effectively, which can lead to better performance on imbalanced datasets.

4.3.4 Optimizer

An optimizer is a function or technique that modifies a machine learning model's parameters in order to decrease the model's loss function. In other words, determining the model's optimized parameter values contributes to strengthening the model's prediction performance.

There are several different types of optimizers including Adam and RMSprop. These two are the subject of the following sections.

4.3.4.A RMSprop

RMSprop (short for Root Mean Square Propagation) is an optimization algorithm used in deep learning for updating the weights of a neural network during training. It is similar to AdaGrad, another popular optimization algorithm, but RMSprop overcomes some of the limitations of AdaGrad by using a moving average of the squared gradient.

The RMSprop optimizer uses a different learning rate for each weight in the neural network. The learning rate is adjusted for each weight based on the magnitude of the gradient, which helps to prevent the learning rate from becoming too small or too large.

To calculate the learning rate for each weight, the RMSprop optimizer maintains a moving average of the squared gradient for each weight. The moving average is computed using a decay rate parameter, which determines how much weight to give to previous gradients compared to new gradients.

The RMSprop optimizer can help to accelerate the training process and improve the performance of deep learning models, especially when dealing with sparse data or non-stationary objectives.

4.3.4.B Adam

Adam (short for Adaptive Moment Estimation) is a popular optimization algorithm used in deep learning for updating the weights of a neural network during training. It is an extension of stochastic gradient descent (SGD) that uses adaptive learning rates for each parameter.

Adam optimizer combines two techniques: momentum and RMSprop. Momentum helps the optimizer to accelerate in the relevant direction and dampens oscillations, while RMSprop helps to normalize the learning rate for each weight based on the size of its gradient.

The Adam optimizer maintains a moving average of the gradient and the squared gradient, called the first and second moments respectively. These moments are used to adjust the learning rate for each weight during training.

The learning rate is initially set to a small value, and it is adapted for each weight based on the magnitude of the gradient and the history of the first and second moments. The momentum term is also used to smooth out the updates, which helps to avoid getting stuck in local minima.

One of the benefits of the Adam optimizer is that it is computationally efficient and requires little memory to compute the updates. Additionally, it is less sensitive to the choice of hyperparameters such as the learning rate and momentum, which makes it easier to use in practice.

Adam optimizer has become a popular choice for training deep learning models, especially in natural language processing and computer vision tasks, where large datasets and complex models are common.

4.4 Reduce Learning Rate On Plateau

For training to be effective, determining an ideal learning rate is essential. Even though utilizing a predetermined learning rate may initially be successful, it often becomes difficult to balance rapid convergence with overshooting the best possible solution.

To resolve this problem there is a strategy called Reduce Learning Rate On Plateau or ReduceLROnPlateau that automatically modifies the learning rate while training based on the observed progress. It works especially well in situations where learning progress is slow or the training process hits a plateau, suggesting that the learning rate might be too high right now. ReduceLROnPlateau enables the learning rate to be reduced dynamically, improving the fine-tuning of the model and improved convergence.

The ReduceLROnPlateau technique lowers the learning rate when the observed metric ceases to improve over a predetermined number of epochs. An example of such metrics is validation loss or

validation accuracy. The strategy aims at allowing the model to implement smaller adjustments to the weights by lowering the learning rate, potentially assisting it in overcoming the local minimum and discovering a better solution [35].

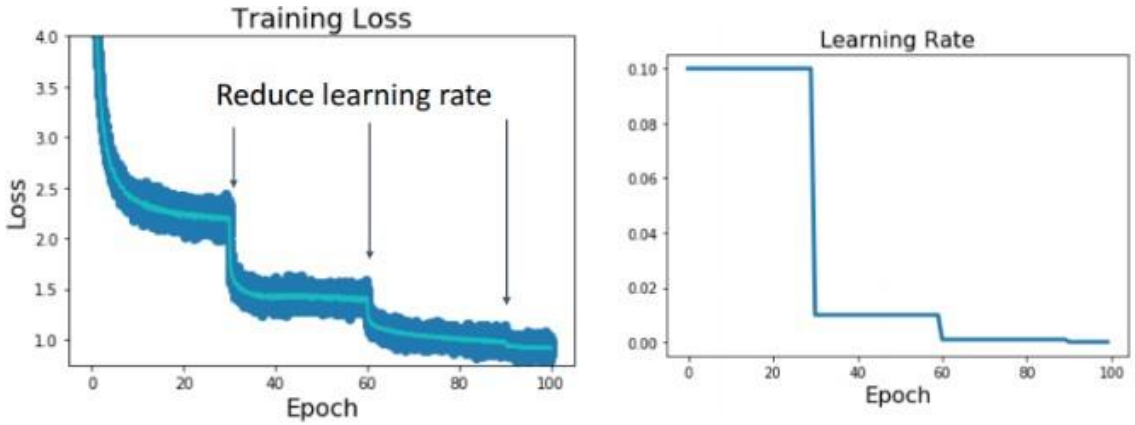

Figure 4.9: Reduce Learning Rate On Plateaus [7]

In the figure above, the technique of Reduce Learning Rate On Plateau is demonstrated in a more visual way, the two images demonstrate the same experience. The figure on the left shows the model's Loss while the image on the right shows the learning rate. As we can be seen, when the model reaches a plateau, the learning rate decreases, enabling the model to achieve lower losses.

4.5 Performance Evaluation Metrics

The classification model will be evaluated in a test set composed of photographs from a few places across Portugal, to make it as close to the Portuguese reality as possible. The metrics to determine the success of the algorithm were not easy to choose.

Normally, accuracy is a good assessment criterion, but it performs better when he has an even class distribution, which is not our case.

Therefore, for the case of the Classification Model that has a dataset imbalance that was chosen that uses are precision (Equation 4.3), recall (Equation 4.4), and F1 score (Equation 4.5).

For the Differentiation Model which has a dataset that is more traditional with a balanced dataset, we also used accuracy (Equation 4.6) as a metric.

Precision is defined as the proportion of accurately predicted positive observations to all expected positive observations. The Recall is the ratio of accurately predicted positive observations to all positive observations in the actual positive class. The F1 Score is calculated as the weighted average of Precision and Recall. As a result, this score considers both false positives and false negatives. The accuracy measures the proportion of correct predictions made by the model over the total number of predictions.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.3}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.4}$$

$$F1score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{4.5}$$

$$Accuracy = \frac{TP + TN}{FP + TN + TP + FN} \tag{4.6}$$

True positives (TP) refer to successfully predicted positive samples. False positives (FP) refer to incorrectly predicted positive samples. False negatives (FN) refer to incorrectly predicted negative samples. True negatives (TN) refer to successfully predicted negative samples.

5

Implementation and Experimental Setup

Contents

5.1 Software and Hardware Setup

During the development of the thesis, several scripts and notebooks were implemented, both for data processing and the implementation of machine learning models. The programming language I have chosen for the purpose of this thesis was Python because is an easy syntax language and I have experience with it. Python also facilitate the development of the project because of the large number of libraries available, including Keras, which was the library that I used throughout the project, the Keras library provided me with all the necessary tools for building and deploying Neural Networks.

Regarding the hardware components used, three different environments were used during the development of this work:

The first environment consists of a CPU (Intel Core i5-8250U 4 cores with 3.40GHz), and a GPU (NVIDIA GeForce 930MX - 2GB RAM). The first environment only served to collect some photographs, the ones done manually and the first environment also acted as a support environment between the other two environments.

The second environment consists of a virtual environment implemented on the Microsoft Azure Machine Learning platform, with a scalable cluster consisting of up to 5 nodes each with a 6-core processor.

The third environment consists of a virtual environment implemented on Google Colab, with a CPU (Intel Xeon CPU @2.20 GHz), with 13 GB RAM, and a GPU (NVIDIA Tesla K80 - 24 GB RAM). The third environment was the fastest environment where it performed the heaviest processing, but the drawback was that it was only available for just a few hours of the day when this environment was unavailable I utilized the second environment.

5.2 Implementation Details

In this section, I will discuss how the experiments were carried out and how the techniques mentioned in the methodology were used, as well as the performance metrics.

I'll start by explaining the implementation of both the Classification Model and the Differentiation Model. The Classification Model was the first model I started to develop so I will use it a lot more as an example, but always bearing in mind that many of the experiments carried out passed from one to another. I'll start talking about the architectures, then the Techniques, and finally the methods used to analyze the results.

5.2.1 Architectures

To create the classification model, I decided to use the architecture of the Inception-ResNet-v2 model because it had a good performance, as we could see in subsubsection 4.3.1.A. This model, having so

many parameters and its training, was obtained by training only a few lower layers of the model, and only then training the whole model with a significantly smaller learning rate. In other words, every training was always split into two phases.

Normally, for the first phase, started with a learning rate of 0.001, then decreased the training course due to the Reduce Learning Rate On Plateau technique. Already in the second phase, I used the new weights from the last phase keeping most of the same parameters, the only adjustments I made were to unlock all the layers, reduced the leaning rate by an order of magnitude to 0.0001 and only saved the weights if this model produced better results than the previous phase.

This technique produced some positive results, but it had several drawbacks, mainly because it became extremely slow; training the two phases would normally require 10 hours. Because of this, there is a need to change to the EfficientNetV2-S model, as this one is smaller but still maintains a strong potential to achieve good results. Because this model is smaller, it is possible to complete the training in a single phase which leads to less computational cost and less time.

5.2.2 Techniques

The BinaryCrossentropy is used to calculate the Loss, this loss is also influenced by the regularization techniques, and in the classification model, the loss function is also influenced by the Class Weights, penalizing the class with the largest size more. Both Adam optimizer and RMSProp were used because both always had very similar values.

5.2.3 Framework for Analysis

The visualization of the results goes through two main phases, the first was the results that the model obtained from the validation dataset during the course of training, and the second was when the model was already trained and the test dataset was utilized.

To visualize the results for the validation dataset, I used three strategies. The first and simplest of all was through the train function provided by Keras, which throughout each training session gives feedback on the results and at the end of each epoch, makes the respective results available for that epoch. The other two strategies derived from these results can then be viewed from the tensorboard where graphs are created from each value, the other takes these results and saves them in tables(for example: excel), this way it is easier to view all data from once and thus be able to identify relationships between them.

It wasn't until the findings were satisfactory that it moved on to this second phase. The second phase, which was with the test dataset, didn't have any progression graphs as the validation phase did; instead, it merely provided the values in a table.

5.3 Data Description

All images received by the model are passed through a filter that scales them to the same size. All images are resized to the chosen size which was ”[255, 255, 3]”, which denotes that the filter transforms the images into images of ”255” by ”255” pixels, and the ”3” denotes that the images have three layers of color and use the RGB standard, indicating that the model is equipped to receive images in color.

The size that was chosen was ”[255, 255, 3]” given that I’m not training the model from scratch I’m using a concept called transfer learning, referred to in subsection 3.2.3. It made perfect sense to utilize the same size considering the neural networks were trained using the ”imagenet” dataset, which contained images of the same size. The basic goal of the transfer learning method is, to begin with a head start, but when I utilized a different dimension, I experienced worse outcomes immediately.

In addition to resizing the training datasets were manipulated using a technique called Augmentation, referred to in subsection 3.2.1. In this particular case, the images were manipulated in terms of Rotation that rotates the image, also use manipulation known as the Shear Angle that slightly deformed the image fixating one axis when stretching the image at a specific, Brightness makes the images darker and lighter.

These decisions were taken in an effort to preserve as much as possible the normal arrangement of solar panels in Portugal. The optimal orientation in Portugal for solar systems is facing south with an inclination of at least 10 degrees and up to a maximum of 45 degrees [36].

5.4 Data division

In the next subsections, I will talk about how the datasets were divided between the Training Set, Validation Set, and Test Set as well as the division within each of these datasets, namely, whether or not the photographs have solar panels for the Classification model, for the Model of Differentiation which type of solar panel is present in the photograph if it is a thermal solar panel or a photovoltaic solar panel.

5.4.1 Classification Model Dataset

The classification model dataset consists of a total of 9105 photographs, which are separated into three sets: training set, validation set, and test set.

These images are not evenly separated between the three sets, as exemplified in Figure 5.1 how the quantity of images in each set differs significantly. The training set is bigger than the others at around 70 percent, with the test set being the second highest with around 20 percent and the Validation Set is the smallest with around 10 percent.

Since the datasets were created from various parts of the country, as indicated in subsection 4.2.2, the test dataset was also divided by regions, and this division helped to validate the circumstances in which the model responded best.

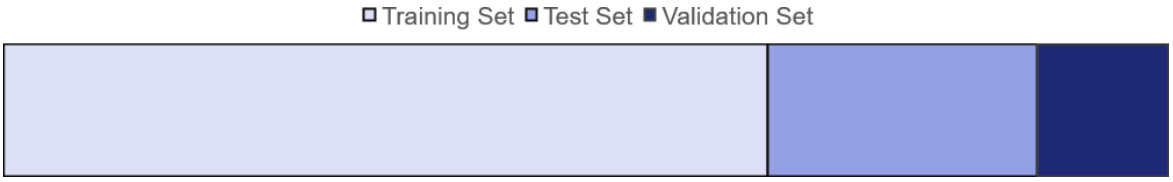

Figure 5.1: Data set partition for the Classification Model

The dataset that was used for the classification model has an extremely large imbalance in the number of negative photographs.

There are many photographs classified as negative and few classified as positive, this is illustrated more clearly in the Figure 5.2. In the training dataset, there is 87 percent of images were classified as negative, which makes this dataset the evenest distribution of the three datasets since the validation dataset has 89.7 percent of negative photographs and the test dataset with 89.4 percent of negative photographs.

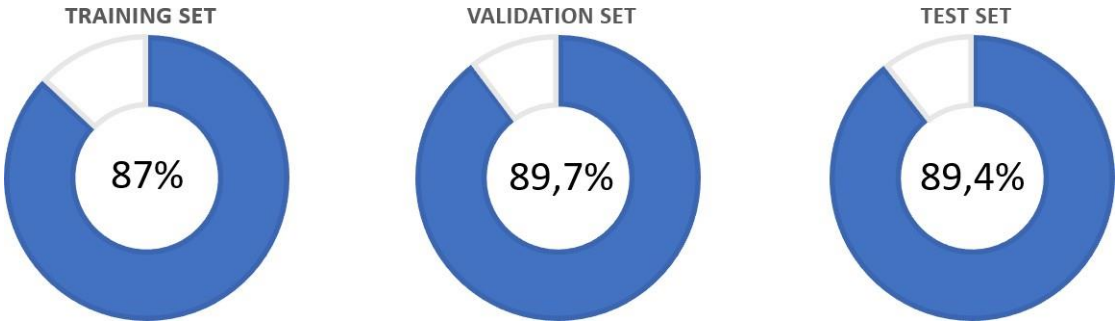

Figure 5.2: Percentage of images without a solar panel per dataset for the Classification Model

5.4.2 Differentiation Model Dataset

For the differentiation model as already mentioned in subsection 4.2.2 the data set for this model was created with the positive images of the classification model. As a result, these datasets are not very extensive, consisting of only a total of 1138 photographs, and it is further divided into three datasets: training set, validation set, and test set.

Similar to the Dataset of the Classification Model, the photographs are not are not distributed equally, between the three datasets, as illustrated in Figure 5.3.

The training set typically contains a higher percentage of images than the other sets; however, for this model, the training set contains even more photographs than the Classification Model, accounting for about 80 percent of the total number of photographs, while the validation set and test set equally divided the remaining photographs, leaving each dataset with roughly 10 percent.

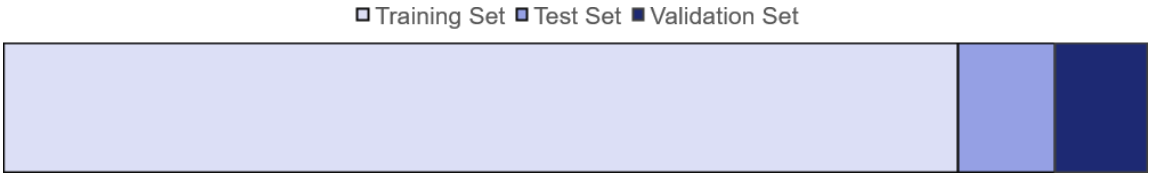

Figure 5.3: Data set partition for the Differentiation Model

In comparison to the Classification model, the Differentiation Model is considerably more balanced in terms of the number of photographs between the two classes, which in the Differentiation Model are photovoltaic solar panels and thermal solar panels. Using Figure 5.4 we can see The training dataset is the most ”unbalanced” of the three, having 55 percent of photographs classified as negative, in this case, it means that the photograph has a thermal solar panel. The validation dataset has 52.2 percent of negative images, and the test dataset has 52.4 percent of negative images.

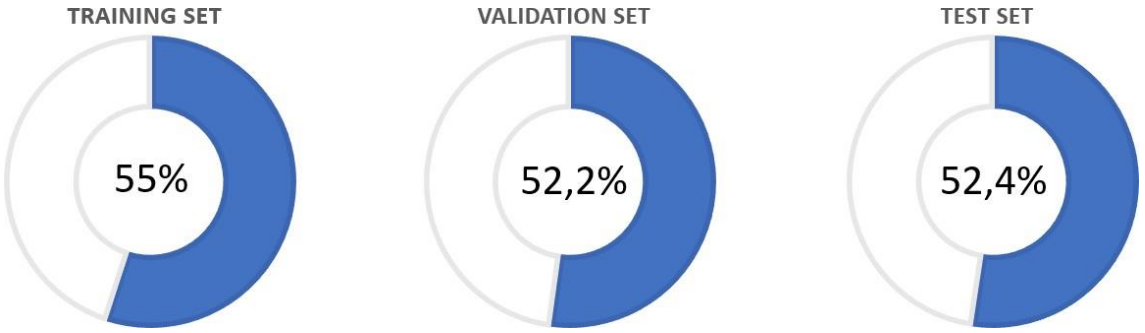

Figure 5.4: Percentage of images with thermal solar panel per dataset for the Differentiation Model

5.5 Baseline and Comparison Methods

A baseline for the outcomes of the experiment should be established before delving into the specifics of the research. Without a solid foundation, it could prove to be challenging to predict the types of values that will be encountered as predictive models are developed.

In order to establish a comparable basis, for the first part of the project, which is the Classification Model, we are using the results obtained by EDP, which are presented in Appendix A, for the second

part, for the Differentiation Model, I was unable to find any comparable projects that may serve as a baseline.

6

Results and Analysis

Contents

6.1 Results

As explained in subsection 5.2.3 only those experiences that showed relevant results with the validation dataset were actually tested with the test dataset.

6.1.1 Classification Model Results

For the classification model we can highlight the best training. The graph on the left is represented by precision and the graph on the right is represented by the Recall graph.

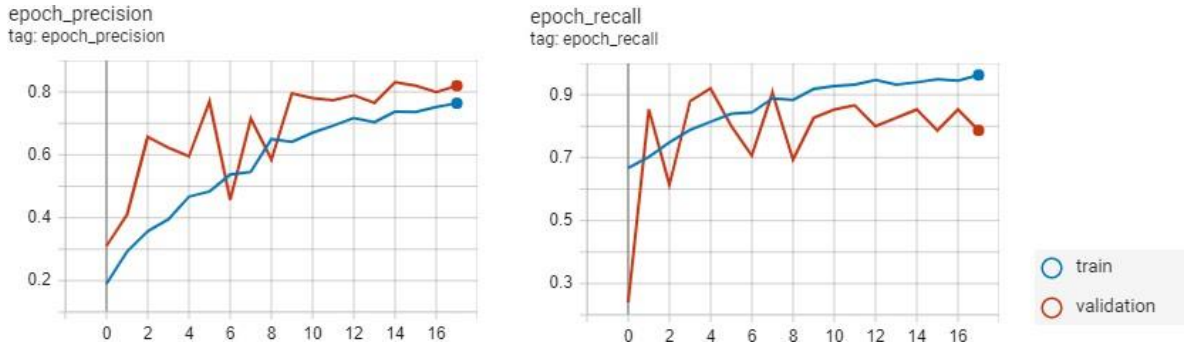

Figure 6.1: Graphs of the best results of the classification model

On the x-axis are the epochs in this case the training took place in 17 epochs, on the y-axis is the numerical value the value of 0.8 represents 80 percent.

The saved weights were from the best result that was obtained in the validation dataset, with a precision value of 0.8242 and a recall value of 0.8533.

In the test dataset we can see Figure 6.2, it was used with the same weights as the validation training.

Region	Precision	Recall	F1-Score
Restelo	0,7943	0,8267	0,8102
Belas	0,7762	0,8406	0,8071
Cascais	0,8074	0,8058	0,8066
B2B	0,8521	0,8633	0,8577
Birre	0,7829	0,8385	0,8097
Faro	0,7938	0,8176	0,8055

Figure 6.2: Table of test dataset results

Similar results were obtained on average with a precision value of 0.8011 and a recall value of 0.8321.

6.1.2 Differentiation Model Results

For the differentiation model I will also present the best training that corresponds to the best result. We can see in Figure 6.3, both the x-axis and the y-axis have the same values as in Figure 6.1

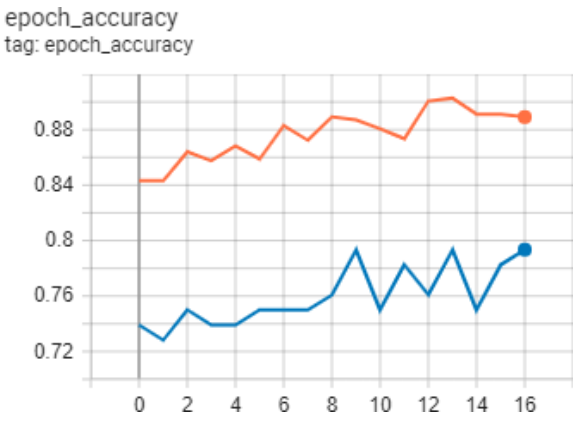

Figure 6.3: Graphs of the best results of the Differentiation Model

The weights saved were from the best result that had an accuracy of 79,35 percent in the validation dataset.

In the test dataset, which was used with the same validation training weights, obtained an accuracy of 78,92 percent.

6.2 Analysis

To carry out the analysis, it also made sense to have it divided into a classification model and a differentiation model, as each of them has its own characteristics.

6.2.1 Classification Model Analysis

Taking into account the results obtained by EDP present in Appendix A, there was a great improvement in terms of the classification of the Birre and Belas datasets. B2b obtained smaller improvements, but even so. In the other datasets that I added for the test dataset to be more complete, they also obtained good results.

These results are excellent taking into account that the model [12] obtained much worse results with the Portuguese images with, with 11cm/pixel, even comparing with the values obtained in the paper with the low resolution images, my model achieved a better result.

6.2.2 Differentiation Model Analysis

As we can see in the graph of the Differentiation Model referring to accuracy, the training line and the validation line are still a little separated, so it would be advantageous to do more fine-tuning, but the big problem with this model is clearly the very small number of samples because although augmentation was used, it is only possible to advance up to a certain point. In order to be able to see the difference that more images make in relation to a better one, since giving 100 images to the training dataset the accuracy of the model increased by 5 percent.

Another thing that has been proven because this speculation already existed, the model is very good at distinguishing solar panels that have a thermosyphon, those without a thermosyphon do not have the best results but many of them can be measured very accurately, which is very good news bearing in mind how difficult it is for a normal person to differentiate.

As expected, the differentiation of solar panels is quite complex, but the model proved to be able to obtain good results. We have to bear in mind that as the test dataset was small, we cannot generalize and say that this model will be perfect in any situation.

7

Conclusions

Contents

7.1 System Limitations and Future Work

Future studies should continue to investigate and improve these methods while taking into account the distinctive conditions and changes in the deployment of solar PV across various locations. As the dataset of both models was small, mainly the differentiation one, with this we can say that for the next works, it would be interesting to increase the size of the datasets.

Both datasets were classified by hand, this always makes the model limited to how good the person is at classifying the images, it would be advantageous in future work to create images synthetically in order to eliminate possible human error.

There is a problem, especially with images taken non-manually because the image can end up right on top of the solar panel, that is, this solar panel is cut, which for the classification model is already challenging, for the Differentiation model it becomes almost impossible, so perhaps it would be advantageous to use images with less zoom, which can also have its problems as it is more difficult to understand details.

7.2 Conclusion

In conclusion, given the lack of a comprehensive central database and the disparate levels of knowledge regarding distributed solar photovoltaic energy across Portugal, this study focused on overcoming the difficulties associated with the identification and recording of solar photovoltaic installations.

Utilizing satellite and aerial images to automatically identify and locate solar PV systems across large geographic areas is becoming more and more popular as a solution to these problems. By enabling the recognition and separation of installed solar panels using global satellite images from Portugal, the deep learning models created in this study support this effort.

The classification model's results showed considerable improvements, especially in the Birre and Belas datasets, while the differentiation model showed promise for differentiating solar panels with accuracy. However, it should be emphasized that the caliber of the photos utilized has an impact on how well the models function. The use of satellite and aerial pictures becomes essential in precisely determining the specific locations and different types of solar PV projects due to the lack of precise documentation.

Even with those setbacks, the results of the Classification model were around 80 percent on both precision and recall on all of the tests, and almost 80 percent of accuracy on the Differentiation Model

The models used in this study represent a development in automatic classification and differentiation, helping to achieve the overall objective of creating a thorough understanding of the deployment of solar PV on a bigger scale. In the end, the results of this study increase the tracking and monitoring of solar PV installations, which supports broader efforts to promote sustainable energy systems.

Bibliography

[1] "What are solar thermal panels?" [Online]. Available: https://www.greenmatch.co.uk/solar-energy/solar-thermal/solar-thermal-panels

[2] M. A. Jeftael Lima, "Evolução das placas fotovoltaicas: Uma análise histórica," Oct 2020. [Online]. Available: https://www.retecjr.com/single-post/2019/05/10/Evolucao-das-placas-fotovoltaicas

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[4] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," *CoRR*, vol. abs/2104.00298, 2021. [Online]. Available: https://arxiv.org/abs/2104.00298

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[6] R. Gençay and M. Qi, "Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 726–734, 2001.

[7] [Online]. Available: https://velog.io/@good159897/Learning-rate-Decay%EC%9D%98-%EC%A2%85%EB%A5%98

[8] IEA, "Net zero by 2050 – analysis." [Online]. Available: https://www.iea.org/reports/ net-zero-by-2050

[9] "United nations official document," Oct 2015. [Online]. Available: https://www.un.org/ga/search/view_doc.asp?symbol=A/RES/70/1&Lang=E

[10] IRENA, "Future of solar photovoltaic." [Online]. Available: https://irena.org/-/media/Files/IRENA/Agency/Publication/2019/Nov/IRENA_Future_of_Solar_PV_2019.pdf

[11] “Solar fotovoltaico.” [Online]. Available: https://www.ordemengenheiros.pt/pt/agenda/webinar-solar-fotovoltaico-producao-distribuida/?utm_term=ARegiaoSulInforma:WebinarSolarFotovoltaico-Producaodistribuidas&utm_campaign=OrdemdosEngenheiros&utm_source=e-goi&utm_medium=email

[12] K. Mayer, Z. Wang, M.-L. Arlt, D. Neumann, and R. Rajagopal, “Deepsolar for germany: A deep learning framework for pv system mapping from aerial imagery,” in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2020, pp. 1–6.

[13] J. de Hoog, S. Maetschke, P. Ilfrich, and R. R. Kolluri, “Using satellite and aerial imagery for identification of solar pv: State of the art and research opportunities,” in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, 2020, pp. 308–313.

[14] “Kwart van de zonnepanelen niet in beeld.” [Online]. Available: https://www.stedin.net/over-stedin/pers-en-media/persberichten/kwart-van-de-zonnepanelen-niet-in-beeld

[15] J. Yuan, “Learning building extraction in aerial scenes with convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 11, pp. 2793–2798, 2017.

[16] N. Audebert, B. Le Saux, and S. Lefèvre, “Semantic segmentation of earth observation data using multimodal and multi-scale deep networks,” in *Asian conference on computer vision*. Springer, 2016, pp. 180–196.

[17] X. Jin and C. H. Davis, “Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information,” *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 14, pp. 1–11, 2005.

[18] S. Ji, S. Wei, and M. Lu, “Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 574–586, 2018.

[19] Z. Zhang, Q. Liu, and Y. Wang, “Road extraction by deep residual u-net,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[20] S. Lee, S. Iyengar, M. Feng, P. Shenoy, and S. Maji, “Deeproof: A data-driven approach for solar potential estimation using rooftop imagery,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2105–2113.

[21] O. ERCOSKUN and T. S. MHLANGA, “Mapping, modeling and measuring photovoltaic potential in urban environments using google project sunroof,” *Gazi University Journal of Science Part B: Art Humanities Design and Planning*, vol. 8, no. 2, pp. 593–606.

[22] “A high-resolution geospatial assessment of the rooftop solar photovoltaic potential in the european union,” *Renewable and Sustainable Energy Reviews*, vol. 114, p. 109309, 2019.

[23] P. Prakash and B. H. Aithal, “A deep learning based approach for rooftop solar potential estimation of a city: A case study of indian metropolis,” in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 8336–8339.

[24] J. M. Malof, R. Hou, L. M. Collins, K. Bradbury, and R. Newell, “Automatic solar photovoltaic panel detection in satellite imagery,” in *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, 2015, pp. 1428–1431.

[25] J. Yu, Z. Wang, A. Majumdar, and R. Rajagopal, “Deepsolar: A machine learning framework to efficiently construct a solar deployment database in the united states,” *Joule*, vol. 2, no. 12, pp. 2605–2617, 2018.

[26] R. Castello, S. Roquette, M. Esguerra, A. Guerra, and J.-L. Scartezzini, “Deep learning in the built environment: Automatic detection of rooftop solar panels using convolutional neural networks,” in *Journal of Physics: Conference Series*, vol. 1343, no. 1. IOP Publishing, 2019, p. 012034.

[27] W. Weiss and M. Spörk-Dür, “Solar heat worldwide 2021,” *Solar Heat Worldwide*, 2021. [Online]. Available: https://www.iea-shc.org/Data/Sites/1/publications/Solar-Heat-Worldwide-2021.pdf

[28] R. Wang, *Advances in solar heating and cooling*. Elsevier, Woodhead Publishing, 2016.

[29] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[34] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[35] A. Al-Kababji, F. Bensaali, and S. P. Dakua, “Scheduling techniques for liver segmentation: Reducelronplateau vs onecyclelr,” in *Intelligent Systems and Pattern Recognition: Second International Conference, ISPR 2022, Hammamet, Tunisia, March 24–26, 2022, Revised Selected Papers*. Springer, 2022, pp. 204–212.

[36] S. P. de Energia Solar, “Em portugal qual deve ser a orientação e inclinação dos paineis solares.” [Online]. Available: https://energiasolare.blogs.sapo.pt/80383.html

A

EDP Experiments

Simulation results using the Pytorch model [12]

Threshold 20, training with images of the original model (Inference with B2B images) Precision: 0.8507 Recall: 0.8531 F1 score: 0.8519

Threshold 20, training with images of the original model (Inferência com imagens de Belas) Precision: 0.7714 Recall: 0.3776 F1 score: 0.5070

Threshold 20, training with images of the original model (Inferência com imagens de Birre) Precision: 0.2993 Recall: 0.6197 F1 score: 0.4036

Threshold 50, training with images of the original model (Inference with B2B images) Precision: 0.8912 Recall: 0.8333 F1 score: 0.8613

Threshold 50, training with images of the original model (Inference with images of Belas) Precision: 0.8033 Recall: 0.3427 F1 score: 0.4804

Threshold 50, training with images of the original model (Inference with images of

Birre) Precision: 0.2993 Recall: 0.6197 F1 score: 0.4036

Threshold 20, re-training with Birre images (Inference with images of Belas) Precision: 0.7714 Recall: 0.3775 F1 score: 0.5069

Threshold 50, re-training with Belas images (Inference with images of Birre) Precision: 0.2993 Recall: 0.6197 F1 score: 0.4036

Threshold 20, re-training with Belas+Birre images (Inference with B2B images) Precision: 0.7124 Recall: 0.8955 F1 score: 0.7935