

## Article

# Cloud Detection and Tracking Based on Object Detection with Convolutional Neural Networks

Jose Antonio Carballo <sup>1,2</sup> , Javier Bonilla <sup>1,2,\*</sup> , Jesús Fernández-Reche <sup>1</sup> , Bijan Nouri <sup>3</sup> ,  
Antonio Avila-Marin <sup>1</sup> , Yann Fabel <sup>3</sup>  and Diego-César Alarcón-Padilla <sup>1</sup> 

<sup>1</sup> CIEMAT, Plataforma Solar de Almería (PSA), 04200 Almería, Spain; jcarballo@psa.es (J.A.C.);

jesus.fernandez@psa.es (J.F.-R.); antonio.avila@ciemat.es (A.A.-M.); diego.alarcon@psa.es (D.-C.A.-P.)

<sup>2</sup> CIESOL, Solar Energy Research Centre, Joint Institute, University of Almería, CIEMAT, 04120 Almería, Spain

<sup>3</sup> Deutsches Zentrum für Luft-und Raumfahrt (DLR), Institute of Solar Research, 04005 Almería, Spain; bijan.nouri@dlr.de (B.N.); yann.fabel@dlr.de (Y.F.)

\* Correspondence: javier.bonilla@psa.es

**Abstract:** Due to the need to know the availability of solar resources for the solar renewable technologies in advance, this paper presents a new methodology based on computer vision and the object detection technique that uses convolutional neural networks (EfficientDet-D2 model) to detect clouds in image series. This methodology also calculates the speed and direction of cloud motion, which allows the prediction of transients in the available solar radiation due to clouds. The convolutional neural network model retraining and validation process finished successfully, which gave accurate cloud detection results in the test. Also, during the test, the estimation of the remaining time for a transient due to a cloud was accurate, mainly due to the precise cloud detection and the accuracy of the remaining time algorithm.

**Keywords:** solar energy; neural network; nowcasting; central receiver system



**Citation:** Carballo, J.A.; Bonilla, J.; Fernández-Reche, J.; Nouri, B.; Avila-Marin, A.; Fabel, Y.; Alarcón-Padilla, D.-C. Cloud Detection and Tracking Based on Object Detection with Convolutional Neural Networks. *Algorithms* **2023**, *16*, 487. <https://doi.org/10.3390/a16100487>

Academic Editors: Guanqiu Qi and Frank Werner

Received: 11 September 2023

Revised: 9 October 2023

Accepted: 16 October 2023

Published: 19 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Solar energy is the most abundant renewable energy source available in the world. There are different technologies that can harness solar radiation directly and transform it into another type of energy, such as concentrated solar power (CSP) and photovoltaic (PV) technology. However, for energy systems based on these technologies, the transients and spatial variation of solar power mainly due to clouds cause technical challenges [1,2]. These challenges have to be addressed in order to achieve meaningful solar penetration under technically and financially viable conditions. For example, in solar central receiver systems (SCR), the central receiver needs to be protected against temperature peaks due to transients caused by clouds in order to extend the plant's entire life cycle [3].

The ability to predict the amount of solar energy that a system can capture and convert is crucial for optimal system performance and longevity. It can also affect the stability of the local power grid, which may become vulnerable to cloud shadow effects [4] due to the growing penetration of solar energy systems. Clouds cause significant intrahour variability in solar power output, which impacts the dispatchability of solar power plants and the management of the electricity grid [5,6].

For all these reasons, the accurate prediction of solar energy resource availability, especially transient cloud behavior, is required [1] and the demand for accurate solar irradiance nowcasting is increasing due to the rapidly growing share of solar energy on our electricity grids [5].

The scientific community is working hard to develop techniques to detect clouds and transients caused by clouds with on-ground cameras and satellite images, as shown by the numerous reviews and works published on this topic [4,7–9]. Researchers propose many different methods for cloud detection but most of the reviews conclude that methods

based on artificial neural networks are needed to overcome the constraints and drawbacks of traditional algorithms [7,8,10]. Cloud detection methods are classified into traditional and smart methods. Among the traditional methods, threshold-based, time differentiation, and statistics methods can be found [8]. The smart methods include convolutional neural networks (CNN), simple linear iterative clustering, and semantic segmentation algorithms. The majority of works rely on traditional approaches; however, recent articles show that this trend might be shifting towards smart approaches [9]. Smart methods have proved to be more efficient on cloud detection tasks than traditional ones.

Cloud detection and classification are usually a first step in solar forecasting/nowcasting [10,11]. As an example of an early step, a paper with a quantitative evaluation of the impact of cloud transmittance and cloud velocity on the accuracy of short-term direct normal irradiance (DNI) forecasts was published using four different computer vision methods to detect clouds [12]. Later, a new computer vision method was proposed using an enhanced clustering algorithm to track the clouds and predict relevant events based on all-sky images, which can deal with the nature of the variable appearance of clouds [13]. This new computer vision method showed that the proposed method can substantially enhance the accuracy of solar irradiance nowcasting methods. Recently, color space operations and various image segmentation methods were investigated to improve the visual contrast of the cloud component and a novel approach to calculate cloud cover under any illumination conditions for short-term irradiance forecasting was presented with a positive linear correlation between cloud fraction and real-time irradiance data [11].

Regarding smart methods, a recent study proposed a deep learning model suitable for cloud detection based on a U-Net network and an attention mechanism for cloud detection for satellite imagery [14]. The results achieved were excellent and show that this network architecture has great potential for applications in satellite image processing. Furthermore, another proposed approach is an integrated cloud detection and removal framework using cascade CNN, which provides accurate cloud and shadow masks and repaired satellite imagery. One CNN was developed for detecting clouds and shadows from a satellite image and a second CNN was used for the cloud removal and missing-information reconstruction. Experiments showed that the proposed framework can simultaneously detect and remove the clouds and shadows from the images and the detection accuracy surpassed several recent cloud detection methods [15]. In [10], the capacity of a CNN to identify the presence of clouds, without ancillary data and at relatively high temporal resolution, was demonstrated. Recently, a cloud detection method using a multifeature embedded learning support vector machine to address cloud coverage occupying the channel transmission bandwidth was presented [16]. Experimental results demonstrate that the proposed method [16] can detect clouds with great accuracy and robustness.

Regarding ground-based observation, a novel deep CNN model named SegCloud was proposed and applied for accurate cloud segmentation. SegCloud showed a powerful cloud discrimination capability and automatic segmentation of the whole-sky images obtained by a ground-based all-sky-view camera [17]. Note that one challenge of artificial neural network approaches is the accessibility of large labeled datasets for training purposes. Therefore, [18] introduced self-supervised-based training methods for semantic cloud classification, using large unlabeled datasets for pretraining. The self-supervised pretraining is followed by a supervised training approach with a small manually labeled dataset. The self-supervised pretraining methods increased the overall accuracy of semantic cloud classification by roughly 9%.

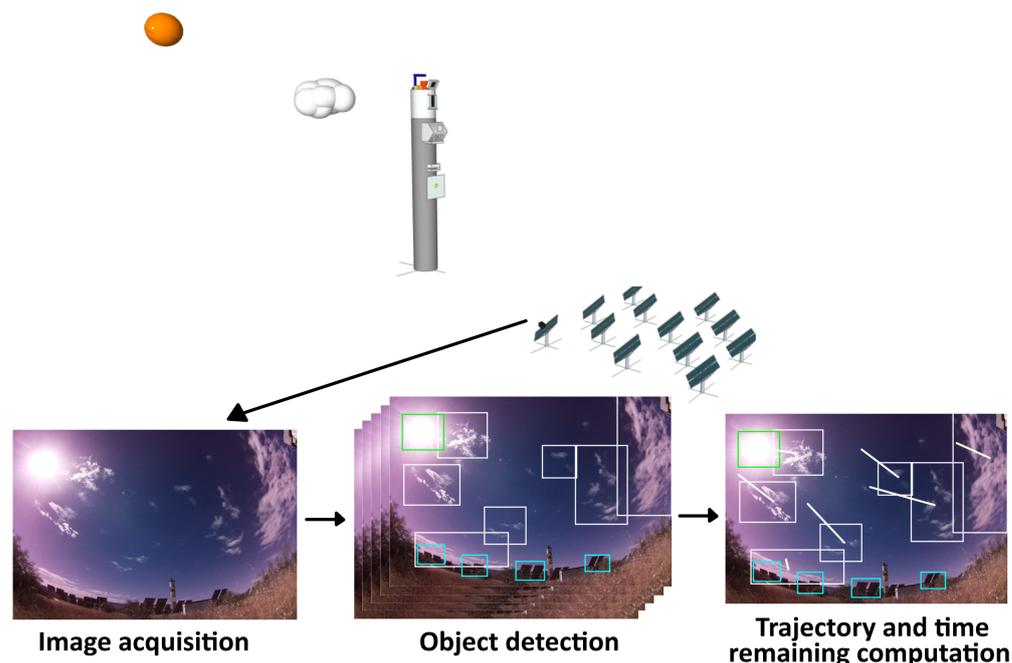
The present work arises from and is a natural further step in the ongoing development of Hel-IoT [19,20] as part of the HelioSun project. Hel-IoT or the smart heliostats concept is based on the application of techniques derived from industry 5.0, mainly artificial intelligence and computer vision techniques. Thanks to these techniques, Hel-IoT incorporates a new, intelligent solar tracking system capable of tracking the Sun with just one camera and one neural network trained for the purpose. The HelioSun project's main goal is to develop

more efficient heliostat fields for SCR among others tools, considering the development of Hel-IoT.

Due to the great relevance of knowing the availability of the solar resource in advance for the correct control of power generation plants based on solar energy, this paper presents the first implementation of a novel approach for cloud detection, trajectory, and transient remaining time prediction making use of computer vision techniques related to object detection with region proposal techniques based on deep learning with CNN. No other method is based on object detection and is able to calculate the trajectory and time remaining. The proposed approach for cloud detection is based on artificial neural networks as recommended in previously mentioned studies. Moreover, it was developed with the main goal of being operable on a wide variety of hardware, including ground-based, low-cost hardware, and having the ability to be used with smart tracker systems. This first implementation of the approach using a low-cost system is analyzed and discussed.

## 2. Methodology

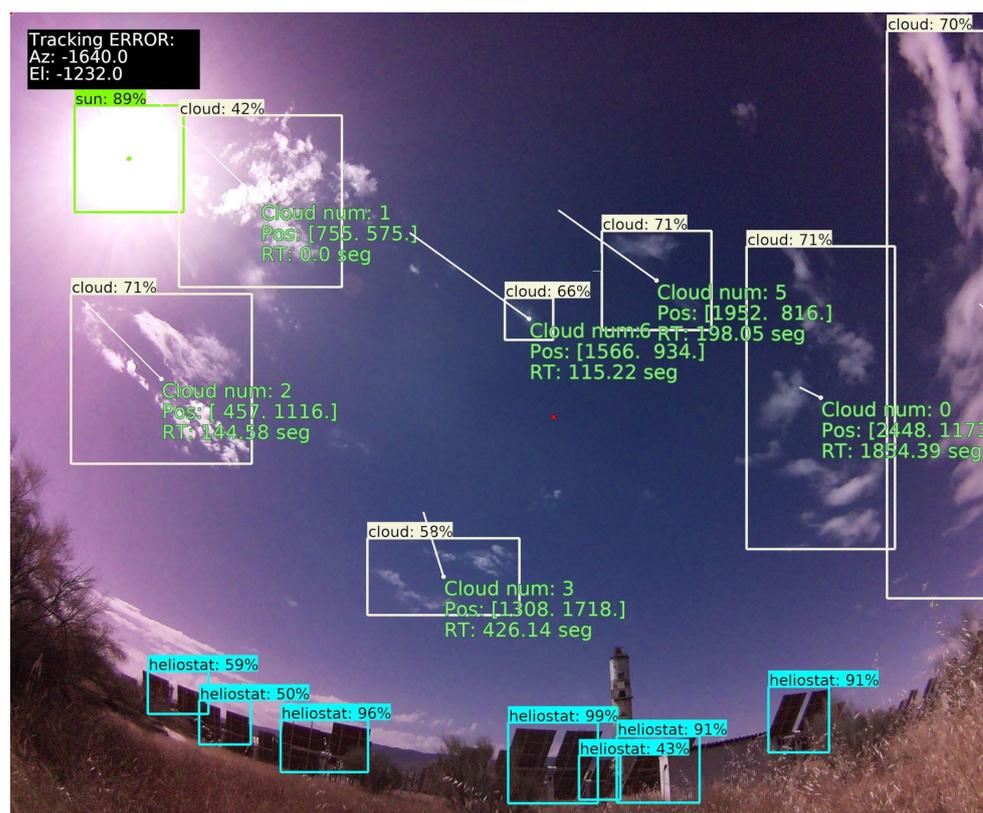
This paper proposes a method based on low-cost cameras and computer vision techniques, specifically a CNN-based object detection technique, to estimate the available solar energy. The method works as follows: First, a low-cost camera takes a picture of the sky or the Sun, as shown in Figure 1, and projects the positions of the objects in the sky onto the camera plane. The image can cover the whole sky or only the sun region, such as the image used for Hel-IoT. The camera was set to full auto mode with a maximum resolution, allowing it to adjust parameters like exposure time according to the lighting conditions. Alternatively, other camera settings such as reducing the exposure time to prevent pixel saturation near the Sun can be used, but they require further investigation.



**Figure 1.** Solar tower system and image taken by a camera located in a heliostat.

The next step consists in the development of a machine learning (ML) model, capable of localizing and identifying multiple objects in a image using a modern CNN [21], in other words, detecting the Sun and cloud positions in the input image of the model. At this point, an algorithm developed for this work is used to analyze the previous results and assign an identifier number to each detected object according to the identifier number of the closest object of the same class in the previous images of the series of images. Then, the algorithm compares the results with previous results to compute the cloud and sun movement vectors

(the white and green lines, respectively, in Figure 2). With the position and the movement vectors (speed and direction), the transient remaining time (RT in Figure 2 for each detected cloud) can be computed as the ratio between the distance in pixels to the Sun and the average of the last five speeds measured for each detected cloud. In Figure 2, in addition to RT and the movement vector, each object detected is also shown according to its class (the colored box), the confidence level of each detection, the id number, and the position of each detected cloud. In this way, a prediction of solar radiation cloud blocking events above the point where the camera is located can be obtained. Several cameras deployed in a region allow the algorithm to infer how the transient will affect each of the areas of the region in which the sensors are installed.



**Figure 2.** Cloud detection results. White, blue, and green boxes to show cloud, heliostat, and sun detection, respectively.

For image acquisition and data processing, the presented methodology uses an open, low-cost hardware platform from the Raspberry Pi Foundation, a Raspberry Pi 4 with 4 Gb of RAM and a Picam, together with a fish eye lens with 8 Mpx of resolution, for a total hardware cost of less than EUR 100. The low-cost camera must be located close to the system that will use the cloud movement information, for example, in the case of Hel-IoT, it can be incorporated together with the smart solar tracking system [19,20]. In others cases or other systems, such as PV systems or existing CSP systems, the proposed system can be located in a representative area of the field. Note that since this new approach can be implemented with low-cost hardware, a network of systems can be distributed over very large solar fields without incurring great cost. In this study, the low-cost camera was installed on a tripod in the northern part of the CESA field. The CESA field is one of the SCR systems at the Plataforma Solar de Almería (PSA) and is composed of 300 heliostats and an 80-meter high tower. The PSA [22], belonging to the Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), is the largest research, development, and testing center dedicated to concentrated solar technologies.

The software was developed in Python 3.10 [23], using TensorFlow [24] and the object detection API [21]. TensorFlow is an open source platform for machine learning that lets researchers push the state of the art in ML by building and deploying ML-powered applications. The TensorFlow object detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train, and deploy object detection models. Object detection models employed for detection in images using ML algorithms involve a CNN that is trained to recognize a series of limited objects. In this work, the EfficientDet-D2 model [25], pretrained on the COCO 2017 dataset [26], was retrained with a PSA dataset in a computer cluster located at the Centro Extremeño de Tecnologías Avanzadas (CETA-Ciemat). CETA manages an advanced data processing center, which is one of the most powerful resources for scientific computing in Spain. The model has an image as input and the detected objects and their positions in the image as output.

Various advances have been made in recent years towards more accurate object detection. But, as we move towards a more accurate object detection network, the network will become more expensive in terms of resource consumption (the number of parameters and floating point operations per second (FLOPS)). The EfficientDet-D2 model was selected for the present work among the many available general models for its good balance between accuracy and resource consumption, see Figure 3. EfficientDet is a new family of object detection models based on a CNN on which the author developed a baseline network that they called as EfficientNet, see Figure 4, which was developed by NAS (Neural Architecture Search) [25]. Before EfficientNet, model architectures were designed by various human experts but that does not mean we completely explored the space of network architectures. For this, a model architecture was developed by NAS which used reinforcement learning under the hood and a EfficientNET was developed with some level of accuracy with the COCO dataset (COCO AP) and FLOPS as the optimization goal. Then, EfficientNet was scaled up under different resource constraints to obtain the family of models EfficientDet (D1, D2, D3... ) [25].

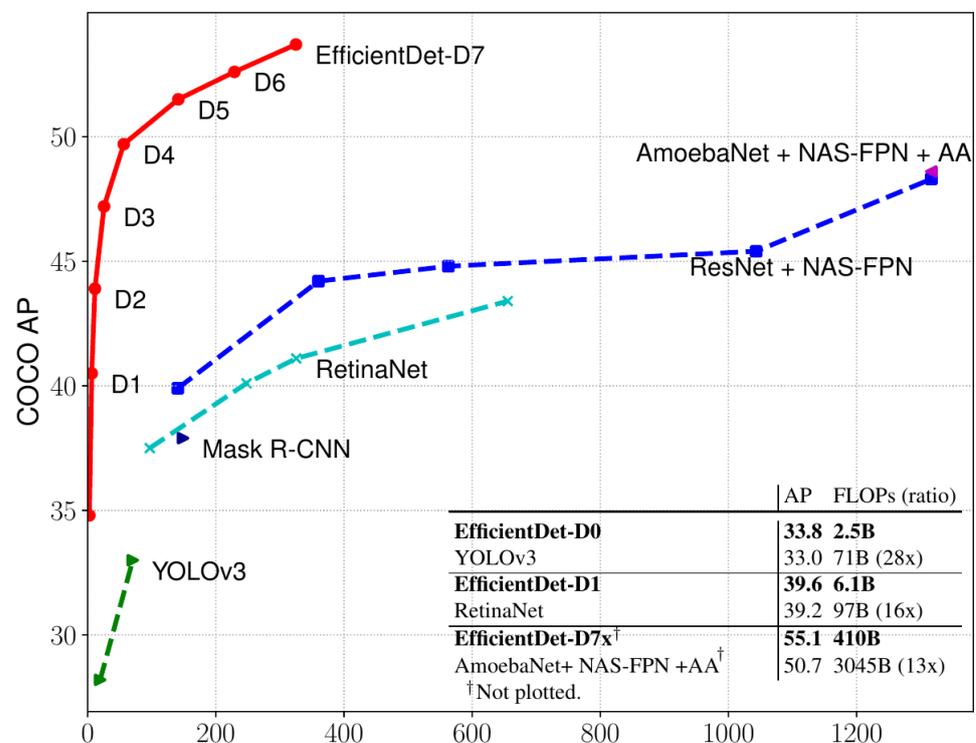
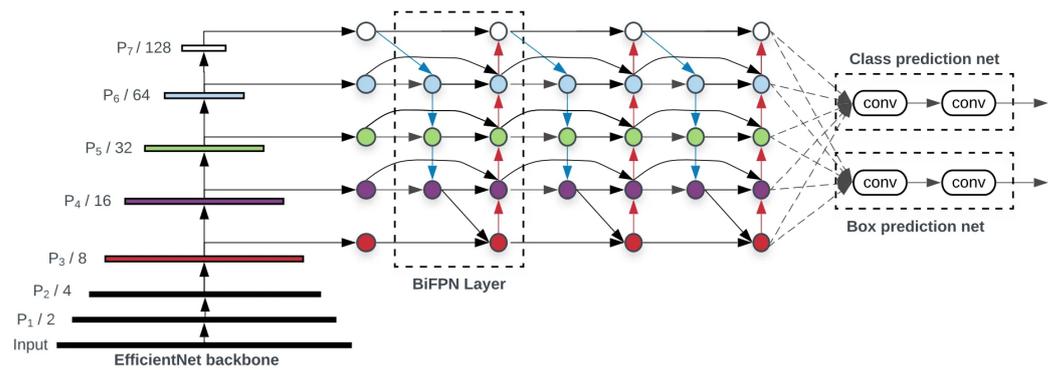


Figure 3. EfficientDet accuracy (COCO AP) vs. resource consumption (FLOPs model) of state-of-the-art object detection models [25].



**Figure 4.** EfficientDet basic architecture [25].

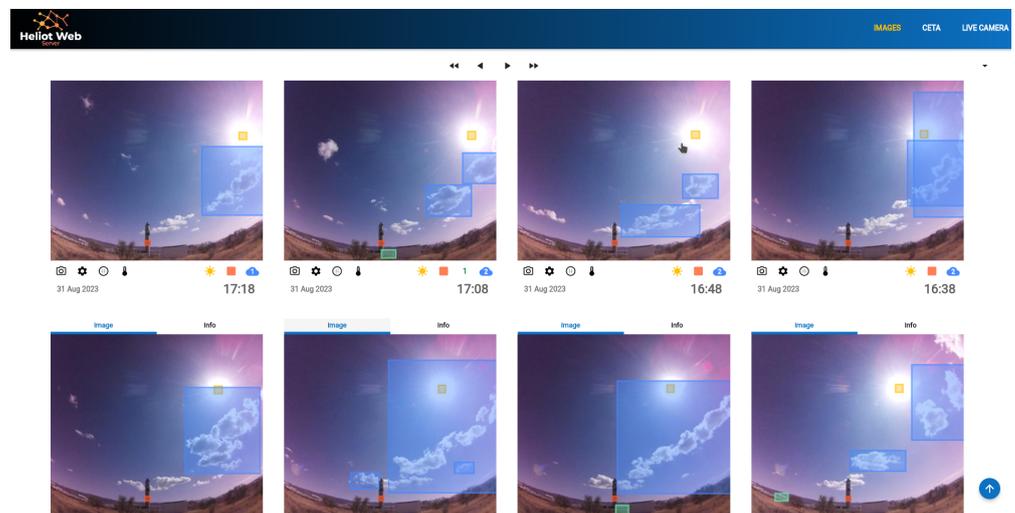
The EfficientNet-B2 backbone is a convolutional neural network that was pretrained on ImageNet and serves as the feature extractor for the object detection task. It has a width coefficient of 1.1, a depth coefficient of 1.2, and a resolution of  $260 \times 260$  pixels. It has a hierarchical structure composed of multiple attention and convolution blocks in cascade, see Figure 4. This modular and scalable structure allows EfficientDet-D2 to strike an optimal balance between computational efficiency and accuracy in object detection. The backbone outputs a set of feature maps at different levels, which are then fed into the BiFPN. The BiFPN is a novel feature fusion module that allows easy and fast multiscale feature integration. It uses weighted residual connections to combine features from different levels in both top-down and bottom-up directions. The weights are learned by a fast normalized fusion operation that avoids the need for extra normalization layers. The BiFPN has a depth of 5 and a width of 112 channels. The bounding and classification box head is a simple yet effective module that predicts the bounding boxes and class labels for each object in the image. It consists of three convolutional layers with  $3 \times 3$  kernels, followed by two separate output layers for box regression and classification. The box head has a width of 112 channels and uses sigmoid activation for both outputs.

Any other model available in the API can be used with just a retraining process to teach the model to detect new objects, unlike most of the previous studies that developed specific models for this task. The advantage of choosing among different models available in the API is that these models are in continuous development by a large community of developers, so models evolve and their performance improves very quickly, unlike with specific models.

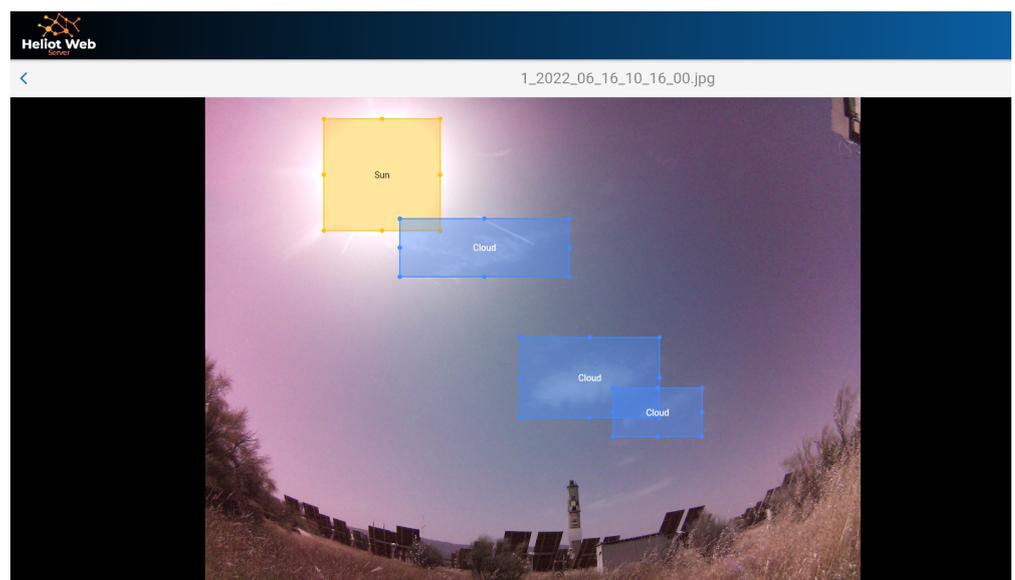
As commented before, the pretrained EfficientDet-D2 model was retrained in this work. A pretrained model is a saved network that has been previously trained on a large dataset. If a model is trained on a sufficiently large and general dataset, such as the COCO dataset, the model will effectively serve as a generic model of the visual world. Then, these learned feature maps can be used without having to start from scratch, i.e., training a large model on a large dataset. This technique to transfer the configuration learned to customize the pretrained model for a given task and reduce the cost of training is known as transfer learning. In this work, transfer learning was used, taking the EfficientDet-D2 model pretrained on the COCO 2017 dataset as the starting point. Then, the model was retrained with the CESA dataset.

The model was retrained using a supervised learning technique which consists of learning a function that maps an input to an output based on an example of input–output pairs [27]. In this case, the model learns to detect objects within an image by training the model with images and associated object location information as inputs, known as a training dataset. As usual in supervised learning, the labels, i.e., the object location information, were added manually in advance. Images together with object location information (hand-labeling was previously commonly used for training purposes) is known as a training dataset. The training dataset used in this work consists of more than 1000 images that were captured from the CESA system. Each image contains several objects of interest, such as the Sun, clouds, and surrounding heliostats, that affect the amount of solar energy that can be harvested. The images were taken

at different times of the day and under different weather conditions to cover a wide range of scenarios and challenges for the object detection task. The labeling work was easily performed thanks to the Hel-IoT web server (see Figure 5), which is a tool for managing the dataset (see Figure 5a) and for the creation and labeling of datasets (see Figure 5b) developed at PSA. One-hundred (100) images were not used during the training, but rather were reserved for use in validating the training process. In order to improve the dataset quality, a random scale-crop data augmentation [25] technique was employed. This technique increases the heterogeneity of the dataset by rescaling and cropping random images from the original dataset to feed the training process. The rest of the configuration training parameters and hyperparameters of the models are available in [28]. The same hyperparameter configuration that the developers of the model optimized was considered, due to it having been optimized by NAS; although a little readjustment of these values due to the fact that the model was used on a different dataset was able to improve the performances of the model for this application. However, the objective of this work was to test the use of the object detection technique, regardless of the model optimization, for cloud detection and time remaining prediction.



(a) Main screen.



(b) Labeling tool.

**Figure 5.** Hel-IoT web server.

The retrained EfficientDet-D2 model in this work can predict four different object classes: the Sun, heliostat, white Lambertian target, and cloud. Although the most relevant objects for this work are clouds and the Sun, the other objects can be used for solar tracking or obtaining valuable information about shadows or blockages. In addition, the model can be retrained to learn to detect other objects of interest, such as another receiver.

### 3. Results

The proposed method can be used with different solar technologies of different sizes. For example, a network of low-cost sensors based on the proposed method can be deployed in the heliostat field to predict the available solar radiation in SCR, improving the control of the receiver and helping to avoid temperature peaks in it, which is one of the key issues in SCR. The same can be done for large photovoltaic plants. Thanks to the low cost of the sensor, a network of sensors can be used to help control the plant and the stability of the grid. These sensors and the information about solar radiation transients can also help with storage management, thermal or electrical, in large plants or small facilities, contributing to an improvement in the competitiveness of solar technologies and their greater deployment. The new approach can be used together with smart solar trackers such as Hel-IoT, using the image for the smart tracking of clouds and predicting the estimated time for a transient. This new capability gives Hel-IoT the ability to track the Sun, detect shadows and blockages, and the ability to track clouds.

The first model retraining results show that the model performed well in object detection and classification and, therefore, produced a good overall result in detecting the new objects. Figure 6 shows the evolution of the classification, localization, and total losses during the 80,000 epochs of the retraining process. This figure shows sufficient training with low error rates. Localization loss is the error generated when placing the detection of the object in the image. In this study, the localization error was quantified using the weighted-smooth-l1 function, based on the Huber loss function [29], which describes the penalty incurred by an estimation procedure. Classification loss is the error made when assigning a label to the detected object; in this case, a weighted-sigmoid-focal function was employed. This function is based on the focal loss approach [30], which improves the learning process with a sparse set of examples and prevents the vast number of negative classifications from overwhelming the detector during training, achieving state-of-the-art accuracy and speed. Total loss is composed of both the classification and localization losses. The optimizer recommended in the API object detection: the momentum optimizer, with a learning rate = 0.0799, was chosen for the training. The training process finished with localization, classification, and total validation error values of  $3.6 \times 10^{-3}$ , 0.04, and 0.06, respectively, so the retraining was sufficient and finished without overfitting. Table 1 summarizes the main training results.

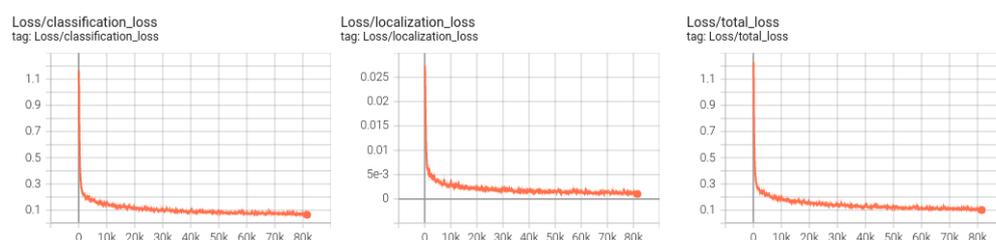


Figure 6. Loss evolution during the first model retraining.

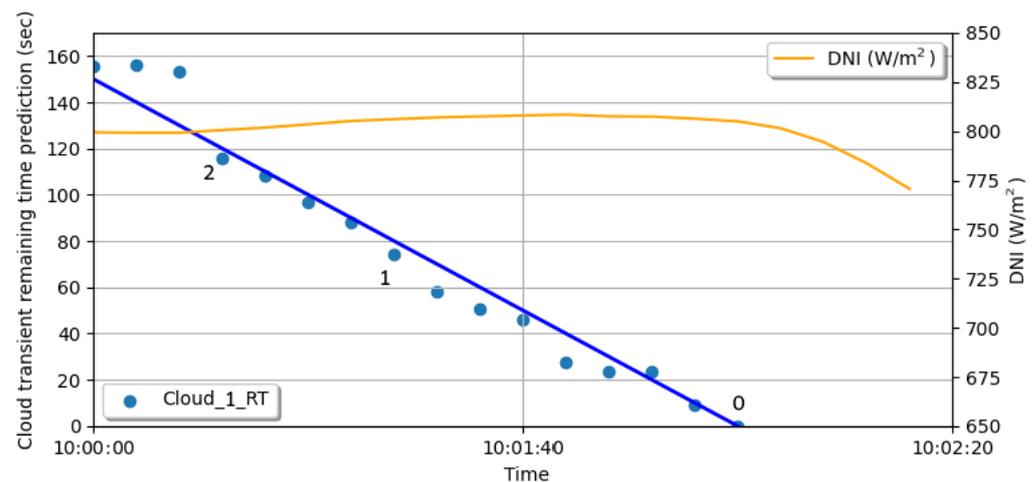
Table 1. Training results and configuration summary.

| N° Epochs | Localization Loss    | Classification Loss | Total LOSS | Learning Rate |
|-----------|----------------------|---------------------|------------|---------------|
| 80,000    | $3.6 \times 10^{-3}$ | 0.04                | 0.06       | 0.0799        |

This paper presents the first version of the retrained model. However, the Hel-IoT web server has been running continuously since April 2022. During this time, the neural network model has been retrained, with supervised learning, several times. The model is undergoing continuous improvement, as it is periodically retrained with new images in cases where the detection fails or does not have enough precision.

To test the cloud detection, trajectory, and remaining time calculation, an image sequence of the PSA sky (Spain) from 24 June 2022 was used. Fifteen images were taken every ten seconds, then the object detection model identified the objects in the images and the trajectory algorithm computed the trajectory of every cloud detected together with the remaining time for the cloud transient.

Figure 7 shows the evolution of the remaining time prediction (blue dots) of the cloud with identification number 1 in Figure 2, since the cloud was detected for the first time in the time series (10:01:30), until it began to generate a transient in solar radiation. The left vertical axis corresponds to the forecast time, the right vertical axis shows the direct normal irradiation (DNI) during the test, and the horizontal axis represents the real time. The orange line shows the DNI values during the test and the blue line represents the cloud transient remaining time if the cloud speed is assumed to be constant. Note that clouds moving with a nonconstant speed also change their shape, which affects their position and speed.



**Figure 7.** Remaining time prediction and direct normal irradiation evolution during the test.

Some instants (numbered in black in Figure 7) were selected and the resulting images are shown in Figure 8, where the trajectory (white line), the cloud number, the cloud position, and the estimated remaining time for the transient are shown.

As shown in Figures 7 and 8, the remaining time and the trajectory predicted were validated by subsequent events. DNI values and image 0 in Figure 8 show a good agreement between the prediction and the reality. Except for small errors, the calculated trajectory and the remaining time prediction were the same as in the real events. The root mean squared error between the remaining time prediction and the remaining time assuming a constant cloud speed was less than 1.7 sec. Note that at the beginning the absolute error was greater than the error at the final point. As mentioned before, this may have been due to the fact that the cloud not only moved, but also changed its shape, which affected the calculation of its position and, therefore, the calculation of its speed, especially at the beginning of the detection when the cloud formed or was close to the horizon line. However, during the rest of the validation, the predicted remaining time and the estimated constant cloud speed were very similar. Finally, the new approach predicted the moment when the cloud began to create the transient without error (0 s in Figure 7).

Although the best model predictions were expected with the image taken using the hardware used to generate the dataset, this model can be used with images from other

hardware, for instance, sky cameras (see Figure 9). The model can also be retrained with additional images to adapt it better to other kinds of hardware.

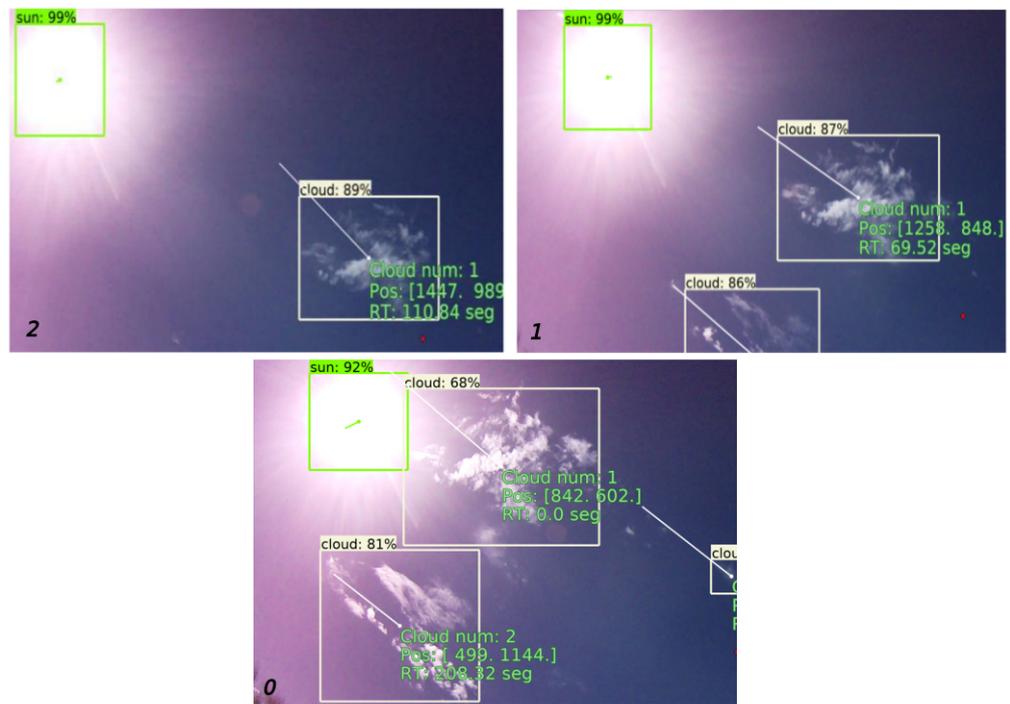


Figure 8. Zoomed images of the selected instants of prediction.



Figure 9. Object detection results from a Mobotix Q25 camera image.

#### 4. Conclusions and Future Work

As mentioned above, knowing the estimated time for a transient in the available solar radiation is key to the proper operation of solar technologies and the grid, since, with this information, it is possible to activate storage systems to supply the transient and keep the grid stable. Therefore, the prediction of transients is an increasingly demanded feature for solar technologies, since this feature makes the solar technologies more competitive and attractive. The method proposed in this paper is not only able to provide the required information (cloud detection and transient remaining time), but it is also capable of being implemented in low-cost systems in a simple and robust way.

In this work, with the exception of retraining, which was carried out in CETA facilities, all the processing was performed with low-cost hardware (object detection model inference, trajectory, and remaining time estimation). This shows the feasibility of implementing this methodology on low-cost hardware. Thanks to this, a large number of sensors can be deployed over large solar fields, increasing the precision of the system without a huge financial investment. In intelligent tracking systems such as Hel-*IoT*, the new cloud detection approach can be implemented in existing hardware together with the tracking system. As a result, the implementation of this new approach does not imply additional costs.

The methodology is based on computer vision and a CNN. It uses a technique known as object detection and an algorithm to compute the trajectory together with the remaining time for a transient to occur. The model presented produced accurate object detection results, both in detection and location estimations during the test, as expected from the accurate results obtained in the retraining and validation process. Other available models can be used if a reduction in the consumption of computational resources or an increase in accuracy is desired. The generation of the dataset for the retraining and the subsequent datasets for successive training was carried out using the Hel-*IoT* web server. This tool, which was developed for this purpose, proved to be of great help. Despite the low number of images in the dataset compared to others, the results of the retraining and test were sufficient thanks to the transfer learning and data augmentation techniques used. However, the dataset needs to be improved and expanded despite it proving sufficient for the first test of the methodology. The prediction results regarding the trajectory and remaining time were in good agreement with the real values in the test. Further periodic tests are planned. Therefore, it can be assured that this first test was a success and the approach should be further studied and tested. Regarding the model and cloud detection, the test, training, and validation results show highly accurate values. On the other hand, regarding the estimated remaining time, the test result demonstrated that the algorithm for calculating the remaining time works properly given the accuracy of the performance of the model among other factors. In conclusion, the first test of the methodology was positive and showed its great potential and room for improvement; however, the methodology must be tested and validated in more depth.

New techniques to improve the approach are already being tested, such as the use of semantic segmentation techniques to obtain greater precision in the estimated time for the transient. Also, a new dataset is being created with different classes of clouds based on the impact they have on solar radiation. With this, an estimate of the reduction in solar radiation caused by a cloud can be obtained, together with the estimated time for the transient event. Future work on this methodology includes the improvement of the model and the dataset, new real tests, cross validation with other methods, the implementation of new previously mentioned techniques, tests to find the best camera configuration, model hyperparameter optimization, improving the calculation of the speed of the clouds and the time remaining, and studying how to reduce the net cost even more.

**Author Contributions:** Conceptualization, J.A.C., J.B. and J.F.-R.; methodology, J.A.C. and J.B.; software, J.A.C. and J.B.; validation, J.A.C., J.B., B.N. and Y.F.; formal analysis, J.A.C. and J.B.; investigation, J.A.C. and J.B.; resources, J.F.-R., A.A.-M. and D.-C.A.-P.; data curation, J.A.C., J.B., B.N. and Y.F.; writing—original draft preparation, all authors; writing—review and editing, all authors; visualization, J.A.C., J.B., B.N. and Y.F.; supervision, J.F.-R., A.A.-M. and D.-C.A.-P.; project administration, J.F.-R., A.A.-M. and D.-C.A.-P.; funding acquisition, J.F.-R., A.A.-M. and D.-C.A.-P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is part of the HELIOSUN project (More efficient Heliostat Fields for Solar Tower Plants) with reference PID2021-126805OB-I00, funded by the Spanish MCIN/AEI/10.13039/5011000011033/FEDER UE, and Plan Andaluz de Investigación, Desarrollo e Innovación (PAIDI 2020) of Consejería de Transformación Económica, Industria, Conocimiento y Universidades de la Junta de Andalucía funds.

**Acknowledgments:** This work was partially supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

|        |  |
|--------|--|
| API    | Application Programming Interface                                      |
| CSP    | Concentrated Solar Power   |
| PV     | Photovoltaic   |
| SCR    | Solar Central Receiver   |
| CNN    | Convolutional Neural Networks  |
| DNI    | Direct Normal Irradiance   |
| ML     | Machine Learning   |
| RT     | Remaining Time   |
| CIEMAT | Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas |
| FLOPS  | Floating Point Operations Per Second                                   |
| NAS    | Neural Architecture Search   |

### References

1. Abutayeh, M.; Padilla, R.; Lake, M.; Lim, Y.; Garcia, J.; Sedighi, M.; Too, Y.; Jeong, K. Effect of short cloud shading on the performance of parabolic trough solar power plants: Motorized vs manual valves. *Renew. Energy* **2019**, *142*, 330–344. [CrossRef]
2. García, J.; Too, Y.; Padilla, R.; Beath, A.; Kim, J.; Sanjuan, M. Dynamic performance of an aiming control methodology for solar central receivers due to cloud disturbances. *Renew. Energy* **2018**, *121*, 355–367. [CrossRef]
3. García, J.; Barraza, R.; Too, Y.; Vásquez-Padilla, R.; Acosta, D.; Estay, D.; Valdivia, P. Transient simulation of a control strategy for solar receivers based on mass flow valves adjustments and heliostats aiming. *Renew. Energy* **2022**, *185*, 1221–1244. [CrossRef]
4. Terrén-Serrano, G.; Martínez-Ramón, M. Comparative analysis of methods for cloud segmentation in ground-based infrared images. *Renew. Energy* **2021**, *175*, 1025–1040. [CrossRef]
5. Nouri, B.; Wilbert, S.; Segura, L.; Kuhn, P.; Hanrieder, N.; Kazantzidis, A.; Schmidt, T.; Zarzalejo, L.; Blanc, P.; Pitz-Paal, R. Determination of cloud transmittance for all sky imager based solar nowcasting. *Sol. Energy* **2019**, *181*, 251–263. [CrossRef]
6. Denholm, P.; Margolis, R. *Energy Storage Requirements for Achieving 50% Solar Photovoltaic Energy Penetration in California*; National Renewable Energy Lab. (NREL): Golden, CO, USA, 2016. Available online: <https://www.nrel.gov/docs/fy16osti/66595.pdf> (accessed on 17 October 2023).
7. Mahajan, S.; Fataniya, B. Cloud detection methodologies: Variants and development—A review. *Complex Intell. Syst.* **2020**, *6*, 251–261. [CrossRef]
8. Li, L.; Li, X.; Jiang, L.; Su, X.; Chen, F. A review on deep learning techniques for cloud detection methodologies and challenges. *Signal Image Video Process.* **2021**, *15*, 1527–1535. [CrossRef]
9. Martins, B.; Cerentini, A.; Neto, S.; Chaves, T.; Branco, N.; Wangenheim, A.; Rütther, R.; Arrais, J. Systematic Review of Nowcasting Approaches for Solar Energy Production based upon Ground-Based Cloud Imaging. *Sol. Energy Adv.* **2022**, *2*, 100019. [CrossRef]
10. Matsunobu, L.; Pedro, H.; Coimbra, C. Cloud detection using convolutional neural networks on remote sensing images. *Sol. Energy* **2021**, *230*, 1020–1032. [CrossRef]
11. Rashid, M.; Zheng, J.; Sng, E.; Rajendhiran, K.; Ye, Z.; Lim, L. An enhanced cloud segmentation algorithm for accurate irradiance forecasting. *Sol. Energy* **2021**, *221*, 218–231. [CrossRef]

12. Li, M.; Chu, Y.; Pedro, H.; Coimbra, C. Quantitative evaluation of the impact of cloud transmittance and cloud velocity on the accuracy of short-term DNI forecasts. *Renew. Energy* **2016**, *86*, 1362–1371. [[CrossRef](#)]
13. Cheng, H. Cloud tracking using clusters of feature points for accurate solar irradiance nowcasting. *Renew. Energy* **2017**, *104*, 281–289. [[CrossRef](#)]
14. Guo, Y.; Cao, X.; Liu, B.; Gao, M. Cloud detection for satellite imagery using attention-based U-Net convolutional neural network. *Symmetry* **2020**, *12*, 1056. [[CrossRef](#)]
15. Ji, S.; Dai, P.; Lu, M.; Zhang, Y. Simultaneous cloud detection and removal from bitemporal remote sensing images using cascade convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 732–748. [[CrossRef](#)]
16. Zhang, W.; Jin, S.; Zhou, L.; Xie, X.; Wang, F.; Jiang, L.; Zheng, Y.; Qu, P.; Li, G.; Pan, X. Multi-feature embedded learning SVM for cloud detection in remote sensing images. *Comput. Electr. Eng.* **2022**, *102*, 108177. [[CrossRef](#)]
17. Xie, W.; Liu, D.; Yang, M.; Chen, S.; Wang, B.; Wang, Z.; Xia, Y.; Liu, Y.; Wang, Y.; Zhang, C. SegCloud: A novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation. *Atmos. Meas. Tech.* **2020**, *13*, 1953–1961. [[CrossRef](#)]
18. Fabel, Y.; Nouri, B.; Wilbert, S.; Blum, N.; Triebel, R.; Hasenbalg, M.; Kuhn, P.; Zarzalejo, L.; Pitz-Paal, R. Applying self-supervised learning for semantic cloud segmentation of all-sky images. *Atmos. Meas. Tech. Discuss.* **2021**, *15*, 797–809. [[CrossRef](#)]
19. Carballo, J.; Bonilla, J.; Berenguel, M.; Fernández-Reche, J.; García, G. New approach for solar tracking systems based on computer vision, low cost hardware and deep learning. *Renew. Energy* **2019**, *133*, 158–1166. [[CrossRef](#)]
20. Carballo, J.; Bonilla, J.; Berenguel, M.; Fernández-Reche, J.; García, G. Machine learning for solar trackers. *AIP Conf. Proc.* **2019**, *2126*, 030012. [[CrossRef](#)]
21. Huang, J.; Rathod, V. Supercharge your computer vision models with the TensorFlow Object Detection API. 2017. Available online: <https://blog.research.google/2017/06/supercharge-your-computer-vision-models.html?m=1> (accessed on 17 October 2023).
22. Ministerio de Ciencia e Innovación, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT)—Plataforma Solar de Almería. Available online: <http://www.psa.es> (accessed on 17 October 2023).
23. VanRossum, G. Python Reference Manual. Department Of Computer Science [CS]. 1995. Available online: <https://docs.python.org/> (accessed on 17 October 2023).
24. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software Available from tensorflow.org. 2015. Available online: <https://www.tensorflow.org> (accessed on 17 October 2023).
25. Tan, M.; Pang, R.; Le Q. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference On Computer Vision And Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790. [[CrossRef](#)]
26. Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. Microsoft coco: Common objects in context. *Eur. Conf. Comput. Vis.* **2014**, *8693*, 740–755. [[CrossRef](#)]
27. Russell, S. *Artificial Intelligence a Modern Approach*; Pearson Education, Inc.: London, UK, 2010.
28. Tensorflow Model Configuration File. 2022. Available online: [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md) (accessed on 17 October 2023).
29. Huber, P. Robust Estimation of a Location Parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [[CrossRef](#)]
30. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference On Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.