

## Article

# Automatic Detection of Photovoltaic Farms Using Satellite Imagery and Convolutional Neural Networks

Konstantinos Ioannou <sup>1,\*</sup>  and Dimitrios Myronidis <sup>2</sup> 

<sup>1</sup> Hellenic Agricultural Organization “DEMETER”, Forest Research Institute, Vasilika, 57006 Thessaloniki, Greece

<sup>2</sup> Department of Forestry and Natural Environment, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; myronid@for.auth.gr

\* Correspondence: ioanko@fri.gr; Tel.: +30-231-046-1171 (ext. 225)

**Abstract:** The number of solar photovoltaic (PV) arrays in Greece has increased rapidly during the recent years. As a result, there is an increasing need for high quality updated information regarding the status of PV farms. This information includes the number of PV farms, power capacity and the energy generated. However, access to this data is obsolete, mainly due to the fact that there is a difficulty tracking PV investment status (from licensing to investment completion and energy production). This article presents a novel approach, which uses free access high resolution satellite imagery and a deep learning algorithm (a convolutional neural network—CNN) for the automatic detection of PV farms. Furthermore, in an effort to create an algorithm capable of generalizing better, all the current locations with installed PV farms (data provided from the Greek Energy Regulator Authority) in the Greek Territory (131,957 km<sup>2</sup>) were used. According to our knowledge this is the first time such an algorithm is used in order to determine the existence of PV farms and the results showed satisfying accuracy.



**Citation:** Ioannou, K.; Myronidis, D. Automatic Detection of Photovoltaic Farms Using Satellite Imagery and Convolutional Neural Networks. *Sustainability* **2021**, *13*, 5323. <https://doi.org/10.3390/su13095323>

Academic Editor: Maria Malvoni

Received: 15 April 2021

Accepted: 7 May 2021

Published: 10 May 2021

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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/>).

**Keywords:** PV farms; deep learning; satellite imagery; CNN; automatic detection

## 1. Introduction

During the last three decades mankind is witnessing an evolution in the energy sector as we notice a shift in energy production methods, from the usage of fossil fuels (petroleum, natural gas, coal, etc.) to more environmentally friendly methods. This is caused mainly due to the fact that a significant portion of the world’s carbon dioxide production is a result of fossil fuels used for energy production [1–3].

However, as electricity consumption plays an important role for modern societies (and its usage cannot be reduced) other forms of energy production must be used in order to satisfy current and future energy demands [3–7].

Renewable energy methods can be considered as a viable solution for energy production and the reduction of CO<sub>2</sub> emissions. These methods include the usage of sustainable sources based on wind, water, biomass, solar and geothermal energy for energy production which are in general called renewable energy sources (RES) [8].

The exploitation of solar energy is considered as one of the most common types of RES. Solar panels are used for transforming energy from indecent sunlight, to electricity using solar cells based on the photovoltaic effect, thus they are also called photovoltaic (PV) panels [9]. Nowadays, massive arrays of PV panels (in the form of solar or PV farms) are used for energy production throughout the world. These farms energy production capability ranges from 1 to 2000 MW, in the case of mega projects covering thousands of hectares [10].

In Europe, PV farms account for 13% of the total RES production. Furthermore, solar power is the fastest-growing source: in 2008, it accounted for 1%. This means that the

growth in electricity from solar power has been dramatic, rising from 7.4 TWh in 2008 to 125.7 TWh in 2019 [11].

In Greece, data provided by the Regulatory Authority for Energy (RAE) indicate that currently there are 9791 PV potential installations (farms) in a variety of stages (licensed investments, licensed installations, licensed production or under evaluation), currently producing 715.6 MW of electric energy.

The variety of the existing stages of PV farms is making difficult to track the infiltration of PV to the Greek market as in many cases the time period from the initial evaluation of the energy production license to production can be years. Financial difficulties, public reaction against the investment as well as technical difficulties can pause the entire installation process.

In this work we investigate a new method of collecting installed PV information which is potentially cheaper and faster than existing methods. The proposed approach uses an algorithm which can automatically detect the existing PV farms based on high resolution free to use satellite imagery, current RAE data for training and deep learning techniques. The entire methodology can be divided in two separate steps.

The first step involves the association of the data provided by RAE with satellite images. For the implementation of this step, we used an algorithm for automatically annotating the images and matching RAE data with satellite images in order to create two datasets. A high-resolution dataset and a low-resolution dataset.

The second step involve the usage of the output produced in the first step in order to train a deep learning (DL) algorithm to automatically detect the PV farm's locations. The algorithm apart from the determination of the locations can also help scientists to extract other information. As it is basically a data unaware algorithm, it can also provide information such as the effect of land use in the selection of PV farm locations, the effect of micrometeorology to the installation locations etc.

The proposed approach offers a series of benefits when compared with other data analysis methods. First it allows the scalability of the produced results as well as the automatic improvement of the data collection. Usage of higher resolution images will provide the user with better results. Thus, the user is free to use data which originate from a variety of sources even from Google Earth, with the best results however, to be expected with data from paid services such as LandSat [12,13].

Additionally, the implementation of the approach using a computer algorithm allows the automation of the process. The entire procedure is easy to use and can be executed multiple times in order to monitor the installation rate. The produced information can also help scientists to predict the level of energy produced as well as help the Government to initiate programs related with RES adoption and provide a valuable tool to enhance the decision-making process regarding the determination of potential installation sites [2,14]. Finally, the presented methodology can be easily adapted in order to monitor other types of RES and reproduced in other regions.

## 2. Literature Review

Computer applications, sensor networks as well as the Internet of Things are responsible for the creation of enormous amounts of data [15]. For this reason, new and innovative techniques must be applied in order to perform sufficient analysis of the accumulated data. Deep Learning is a part of machine learning (ML) methods based on the usage of artificial neural networks with representation learning (supervised, semi-supervised or unsupervised learning) [16].

Essentially DL is a methodology where many classifiers work together, and it is based on linear regression followed by activation functions. DL foundation relay on the same traditional statistical linear regression approach. The only difference is that there are many neural nodes in deep learning instead of only one node (in the case of linear regression). These nodes are known as neural network, and one classifier (a node) is known as perceptron. The network is organized in layers and each layer can have many

hundreds or even thousands of nodes. Layers which are situated between the input and output layers constitute the hidden layer and accordingly the nodes which constitute this layer are known as hidden nodes. In contrary with traditional machine learning classifiers where the user must write complex hypothesis, in deep neural network applications the hypothesis is generated by the network itself, making it a powerful tool for learning nonlinear relationships effectively [16].

ML can be divided into two development phases, shallow learning (SL) and deep learning. The most widely spread SL methods include logistic regression, support vector machine (SVM) and Gaussian mixture models [17–26]. SL main disadvantage is that it cannot handle complex real-world problems such as voice and image recognition [16]. On the contrary DL specializes in solving problems such as image classification, voice recognition etc. For example, image classification of 1000 kinds of images provided a classification error rate of 3.5% which is higher than the accuracy of ordinary people [27].

Various DL algorithms were used for disease determination. Quiroz and Alferez [28] used DL image recognition of legacy blueberries in the rooting stage, planted in smart farms in Chile. For this reason, they used a convolutional neural network (CNN) to detect the presence of trays with living blueberry plants, the presence of trays without living plants and the absence of trays. The model produced results with 86% accuracy, 86% precision, 88% recall and 86%  $F_1$  score.

Other researchers used DL for apple pathology image recognition and diagnosis [29]. For this reason, they trained a CNN that obtained a recall rate of 98.4% using error back propagation analysis of sampled elements. In the study of Liu et al. [30], DL was used for the identification of citrus cancer based on the AlexNet model, with an optimized network structure which could reduce the network parameters while maintaining the same level of accuracy. The results from the application showed that the recognition accuracy reached 98%. In the study of Amara et al. [31], DL was used for detecting two well-known banana diseases. For this reason, they used a deep CNN based on the LeNet architecture, with the results accuracy at 85.9%, precision accuracy 86.7%, recall 85.9% and  $F_1$  score 86.3%.

DL was also used for other types of image recognition. Huang et al. [32] used DL for determining crack and leakage defects on metro shield tunnels which produced very good results with an identification error of 0.8%. Yang et al. [33] used a DL algorithm (in this case a modified AlexNet model) was used in order to determine wind turbine blade damage on images taken from an unmanned aerial vehicle. The model provided better results (97.1% average accuracy) when compared to the unmodified AlexNet model and support vector machine models. In [34], a DL approach was proposed for the classification of road surface conditions. For this, they used a CNN network and created a new activation function based on the rectified linear unit function. Their results showed a classification accuracy of 94.89% on the road state database. DL were also used to perform breast cancer classification. A new method called BDR-CNN-CGN was used to perform classification of breast cancer types, the results showed improved detection rates (accuracy 96.10%) compared to other neural network models [35]. A CNN was also used in order to perform COVID-19 diagnosis. The proposed CNN employed several new techniques such as rank-based average pooling and multiple-way data augmentation. Among the eight proposed models, the model named FGCNet performed better with performance percentage higher than 97% [36]. Finally, Malog et al. [37], used high resolution satellite imagery and deep forest algorithm in order to detect roof top installed photovoltaic arrays. Their data included imagery from an area of 135 Km<sup>2</sup> and the results showed 99.9% pixel-based detection accuracy and 90% object-based detection accuracy. Table 1 presents an overview of the aforementioned literature.

**Table 1.** State of the Art.

| Reference   | DL Usage  |
|---|---|
| Ioannou et al., 2021  | Application of Deep Learning in weather Data gathered from IoT devices  |
| Dong et al., 2021   | Deep learning methods   |
| Freedman, D. 2009   | A survey on deep learning and its applications  |
| Mood, C. 2010; Kleinbaum and Klein, 2002; Hosmer et al., 2013                 | Applied Logistic Regression and applications  |
| Soentpiet, R. 1999; Hearst et al., 1998; Steiwart and Christmann, 2008        | Support Vector Learning, Support vector machines and applications   |
| Schraudolph, N, 2002  | A Gauss–Newton approximation of the Hessian from nonlinear least-squares problems to arbitrary loss functions |
| Li, Z., 2009  | A deep investigation into new ways of applying computer technologies to biomedicine                           |
| Verbeek et al., 2003  | Development of a greedy algorithm for application on unknown ratio mixtures of components                     |
| Gavali, P., Banu, J.S., 2019  | DL for image classification using the CUDA Platform   |
| Quiroz, Alferoz, 2020; Tan et al., 2015; Liu et al., 2020; Amara et al., 2017 | Application of DL for the recognition of fauna diseases   |
| Huang et al., 2018; Yang et al., 2021; Cheng et al., 2019                     | Recognition of structural defects using DL  |
| Zhang et al., 2021; Wang et al., 2021   | Application of CC for medical purposes (cancer type determination, COVID-19 diagnoses, etc.)                  |
| Malog et al., 2016  | Detection of residential PV arrays using DL and aerial imagery.   |

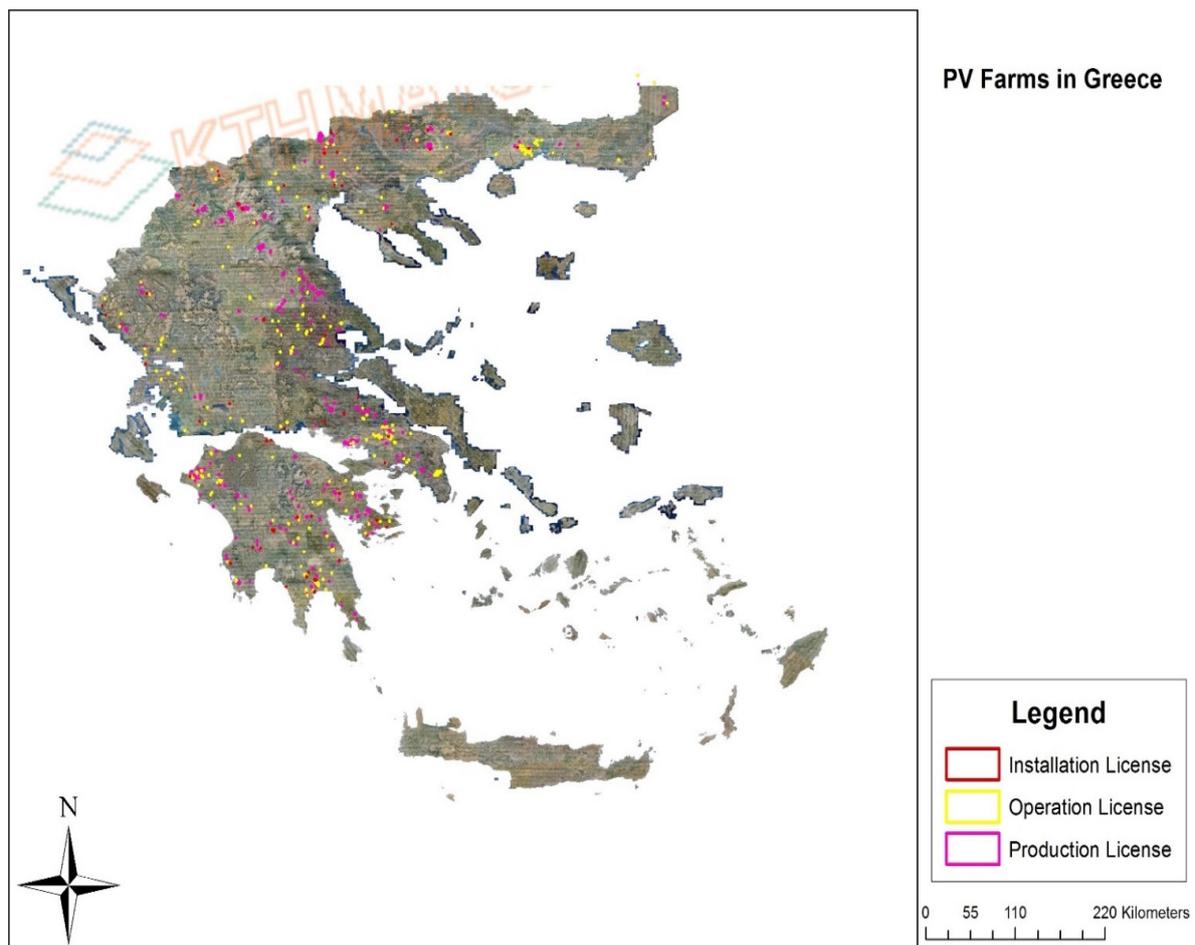
### 3. Materials and Methods

For the creation of the image data sets we used data provided by RAE as well as, data which are available from Apple Maps. Apple Maps is a free map service based on satellite data which are provided from DigitalGlobe. RAE data included a series of polygons (in Shape file form) which included all PV farm investments in Greece (Figure 1). The data were categorized depending on the status of the investment in:

- Investments with installation licenses;
- Investments with production licenses;
- Investments with operation licenses.

Each shape file was at first converted to GeoJSON format. GeoJSON is a geospatial data interchange format compatible with the GNU/General Public License (GPL) guidelines, based on JavaScript Object Notation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents. GeoJSON uses a geographic coordinate reference system, World Geodetic System 1984, and units of decimal degrees [38].

A special PYTHON algorithm was written in order to match the polygons with base map data. The algorithm used a GNU/GPL library called jimumap in order to read each polygon in GeoJSON form and create an image file. Thus, concluding the first step of the methodology. Jimutmap allows the user to select different zoom levels when annotating the data and create images of different resolutions.



**Figure 1.** Map of Greece with PV farms in various phases (Basemap from Hellenic Cadastre).

In Figure 2, we can easily observe that the library user, can easily select the zoom level value, using the zoom variable, and thus determine the resolution of the images created (higher zoom level creates images with lower resolution). This is due to the fact that satellite imagery provided by free services has limited resolution. Additionally, the library allows the usage of multiple core threads in order to perform quicker the required annotations.

```
download_obj = api(min_lat_deg = 10,
max_lat_deg = 10.01,
min_lon_deg = 10.1,
max_lon_deg = 10.11,
zoom = 19,
verbose = False,
threads_ = 5,
container_dir = "myOutputFolder")
```

**Figure 2.** Code snippet from jimutmap library.

The second step included training a convolutional neural network to automatically detect the PV farm's locations. The CNN was developed using Google Collaboratory or Google Colab (GC) for short. GC is a product from Google Research allowing users to write and execute arbitrary PYTHON code using their browser, and is especially well suited to machine learning, data analysis. Additionally, it provides access to advanced cloud resources including the ability for the user to use graphics processor units (GPU's) and

tensor processing units (TPU's). Unlike normal central processor units—CPU's (which are installed on all personal computers using the x64 architecture), GPU's are specialized electronic circuits designed to accelerate the creation and manipulation of images. Their highly parallel structure makes them more efficient than general-purpose (CPUs) for algorithms that process large blocks of data in parallel [39]. TPU's are artificial intelligence accelerator application-specific integrated circuits (ASICs) developed by Google specifically for neural network machine learning using TensorFlow a free and open-source software library for machine learning [40].

### 3.1. Convolutional Neural Networks

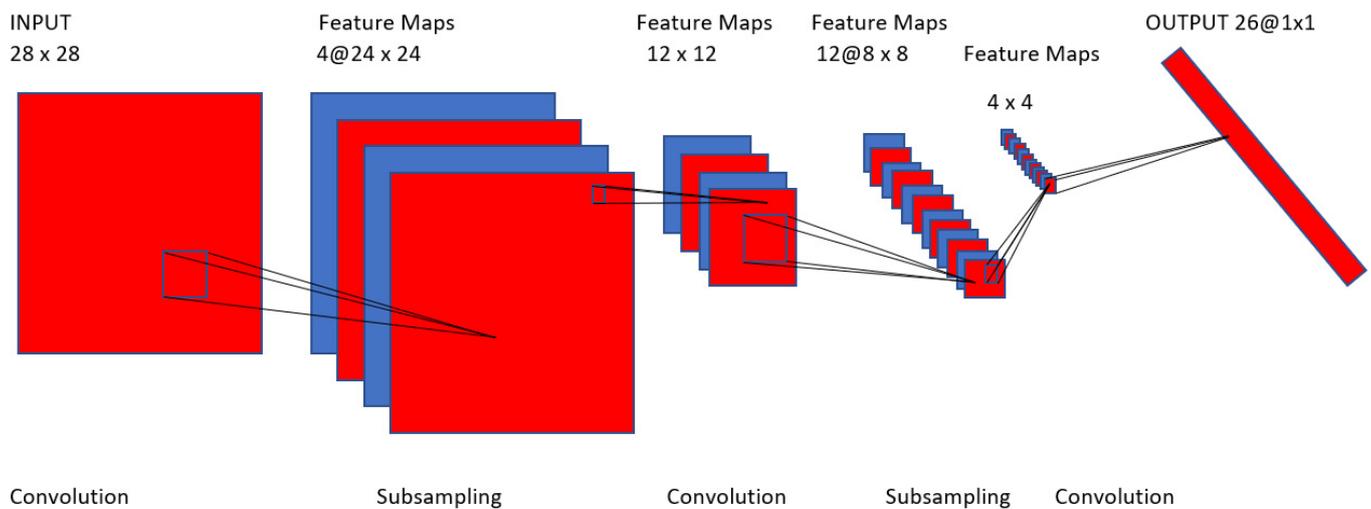
Convolutional neural networks (CNN) are inspired by the cat's cortex and were first proposed in the 1980s [41]. A CNN has similar structure with other multilayer neural networks, and it is comprised of layers. Each layer is composed of a number of two-dimensional planes and each plane has independent neurons. Sparse connections are used between layers, meaning that the neuron in each feature map only connects to the neurons in a small area in the upper map, in contrast with the traditional neural networks. The CNN structure depends mainly in the shared weight, the local experience field and the sub-collector to ensure the invariance of input data [42].

The following figure (Figure 3) presents the layout of a CNN. In this case the network is comprised from an input layer, four hidden layers and an output layer. This network was created for performing image processing. In more detail image recognition of characters written by hand. In this case the input layer is made up using  $28 \times 28$  sensory nodes. This layer receives the images which have been approximately centered and normalized in terms of size. Afterwards the computational layouts alternate between convolution and subsampling as follows:

- The first hidden layer is responsible for the convolution. This layer consists of four feature maps, with each feature map consisting of  $24 \times 24$  neurons. Each neuron is assigned a receptive field of  $5 \times 5$  size.
- The second hidden layer is responsible for subsampling and local averaging. Like the previous layer, it also consists of four feature maps, but each feature map is now made up of  $12 \times 12$  neurons. Each neuron has a receptive field of size  $2 \times 2$ , a trainable coefficient, a trainable bias, and a sigmoid activation function. The trainable coefficient and bias control the operating point of the neuron.
- The third hidden layer is responsible for the second convolution. It consists of 12 feature maps, with each feature map consisting of  $8 \times 8$  neurons. Each neuron in this hidden layer may have synaptic connections from several feature maps in the previous hidden layer. Otherwise it operates in a manner similar to the first convolutional layer.
- The fourth hidden layer is responsible for performing a second subsampling and local averaging. It consists of 12 feature maps, but with each feature map in this case consisting of  $4 \times 4$  neurons. Otherwise, it operates in a manner similar to the first sampling layer.
- Finally, the output layer is responsible for the final stage of convolution. This layer consists of 26 neurons, with each neuron assigned to one of 26 possible characters. As before each neuron is assigned a receptive field of size  $4 \times 4$  [42].

The result of the previously described processes is the application of a bipyramidal effect. This means that with each convolutional or subsampling layer, the number of features maps is increased while the spatial resolution is reduced, compared to the corresponding previous layer.

CNN's first usage was for the identification of handwritten checks in banks, but they were incapable of recognizing large images. For this reason, ref. [43] developed LeNet-5 which was a classical model of convolutional neural network with low error rates (only 0.9% on the MNIST data-set).



**Figure 3.** Convolutional Network for image processing.

The main bottleneck on the application of CNN is the long training time due to many hidden nodes on the networks. However, weight sharing which is a characteristic of the CNN allows parallel processing of weights if the proper infrastructure exists. Today as modern graphics processor units (GPU's) support parallel computing the application of CNN's is easier. In [44], a GPU algorithm was used in order to solve the ImageNet problem.

The CNN implemented for automatically detecting PV farms was based on Keras 2.3.0, a deep learning application programming interface written in PYTHON 3.7, running on top of the machine learning platform TensorFlow 2.4.1 supported by Google Colab. Keras was developed with a focus on enabling fast experimentation.

### 3.2. Building the Model

Keras supports various image classification models (Xception, ResNet, MobileNet, VGG, etc.). In this study we used the InceptionV3 model mainly because it performs significantly better than the other Keras Supported models [45]. The images that will be used were randomly divided in two categories, Training Images used for training and validating the model and evaluation images used for determining the network performance against new, unseen, images.

Before presenting the images to the network we perform a series of augmentations which will ensure that our model would never use twice the exact same picture thus, the model will try to overfit on the training data. For this reason, we used the image data preprocessing function of Keras. This function has a series of arguments for manipulating the training image datasets. The following arguments were used for the manipulation:

- Rotation range, rotates the images randomly;
- Height shift range, shifts the image along the X axis;
- Width shift range, shifts the image along the Y axis;
- Horizontal flip, flips the image across the X axis;
- Vertical flip, flips the image across the Y axis;
- Validation split, determines the fraction of images reserved from the training dataset for model validation;
- Zoom range, determines the zoom factor;
- Brightness range, modifies the image brightness level;
- Rescale, determines if the image is rescaled to specific dimensions;
- Shear range, determines the image distortion across an axis in order to create or rectify perception angle;
- Fill mode, determines the image location inside the canvas.

Continuing, we must determine the training epochs as well as the image batch size. Epochs refers to the number of times the network is trained through the entire dataset, whereas batch size determines the number of samples processed each time (before the model is updated).

In InceptionV3 we have the capability to use predefined training weights using the imagenet or initialize them randomly. Imagenet is an image database which is organized according to the WordNet hierarchy in which each node of the hierarchy is depicted by thousands of images [46]. The usage of this database is proven to significantly increase a CNN's performance [47]. Figure 4 displays the entire workflow of the model applied.

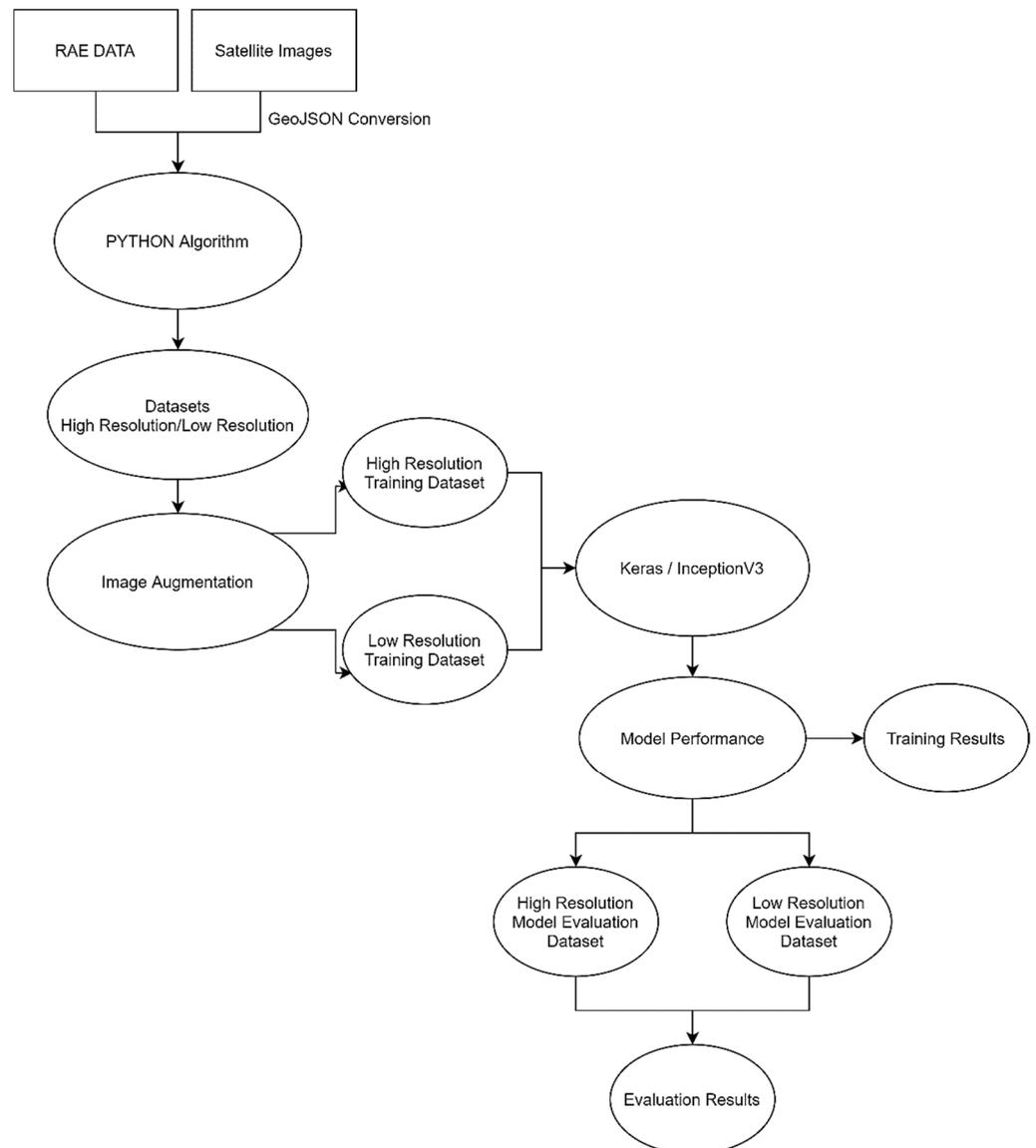


Figure 4. Methodology flowchart.

#### 4. Results

The PYTHON algorithm used for extracting the images of PV farms created 570 images files. Of them, 220 where high-resolution images (approximately 1 MB each) and 350 where low-resolution (approximately 16 KB each). These images where divided randomly in Training and Evaluation datasets as show on Table 2.

**Table 2.** Image per dataset.

| Resolution | Training Dataset | Evaluation Dataset |
|------------|------------------|--------------------|
| High Res   | 75               | 145                |
| Low Res    | 250              | 100                |

Following that, the datasets were augmented using the image data processing function. The parameters used in this function are presented in Table 3.

**Table 3.** Image data processing parameters.

| Parameter         | Value    |
|-------------------|----------|
| Rescale           | 1./255   |
| Zoom_range        | 0.3      |
| Rotation_scale    | 360      |
| Width_shift_range | 0.5      |
| Shear_range       | 0.5      |
| Horizontal_flip   | True     |
| Vertical_Flip     | True     |
| Brightness_range  | 0.6, 1.4 |
| Fill_mode         | Nearest  |
| Vallidation_split | 0.2      |
| Shear Range       | 0.2      |
| Fill mode         | Nearest  |

Next, the images were imported to Keras and the InceptionV3 algorithm was applied, for 15, 20 and 25 epochs with a batch size of 15 using the ImageNet pre-trained weights. Batch size number was selected mainly because the number of the images used for training and validation is rather small. Generally, we use larger batch sizes when we have large datasets. The selection on the number of training epochs is based on the produced results (there is no guideline regarding the train period of a neural network). This means that if we notice overfitting in the results (meaning that the network cannot generalize properly), then we reduce training epochs.

Table 4 includes the results taken from the three training sessions applied. The results show the percentage of correct prediction using the training dataset and the validation dataset. From the table it is evident that the applied model does not provide better results when trained for more than 20 epochs, as it can also be seen in the graphical representation of the results in Figure 5. From Figure 5 it is also obvious, that the model performs erratically during the last validation session with large fluctuations during the validation of the model. This means that the model must have overfitted during training for 25 epochs.

**Table 4.** Training and validation results.

| Accuracy            | 15 Epochs | 20 Epochs | 25 Epochs |
|---------------------|-----------|-----------|-----------|
| Train Accuracy      | 94.23%    | 95.38%    | 87.31%    |
| Validation Accuracy | 90.77%    | 90.77%    | 86.15%    |

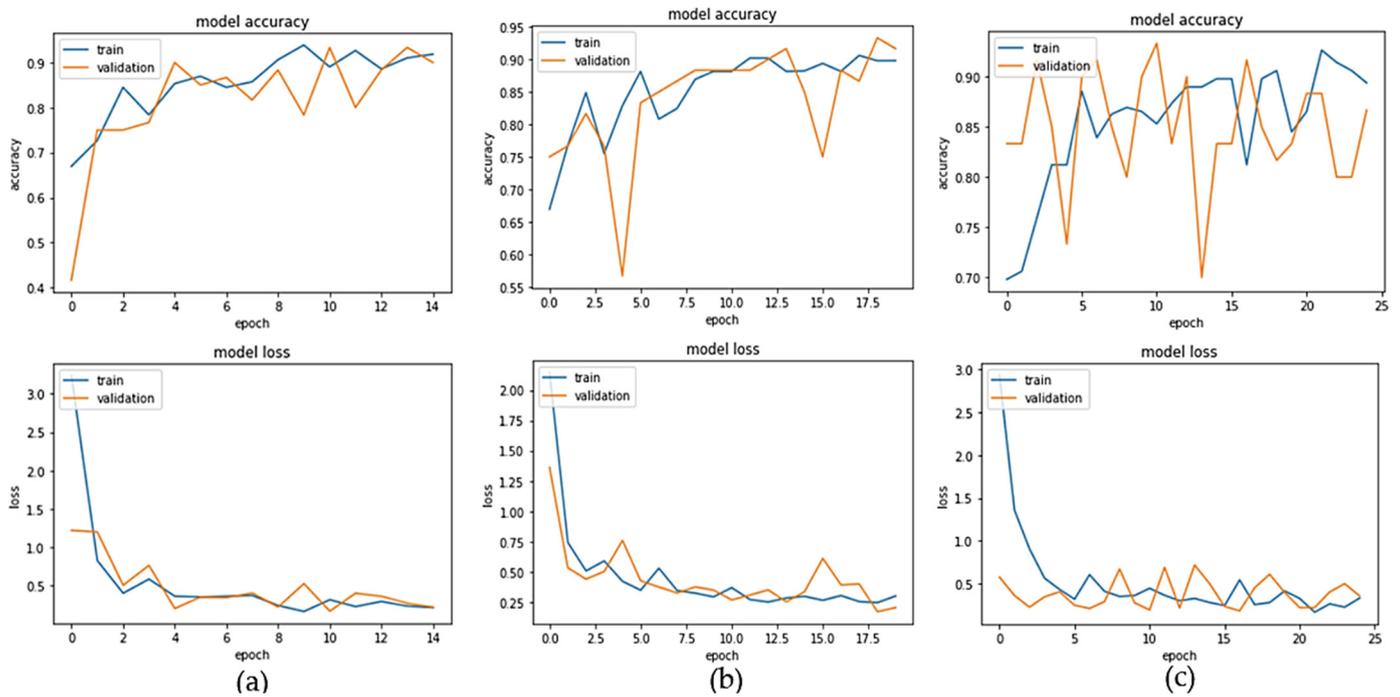
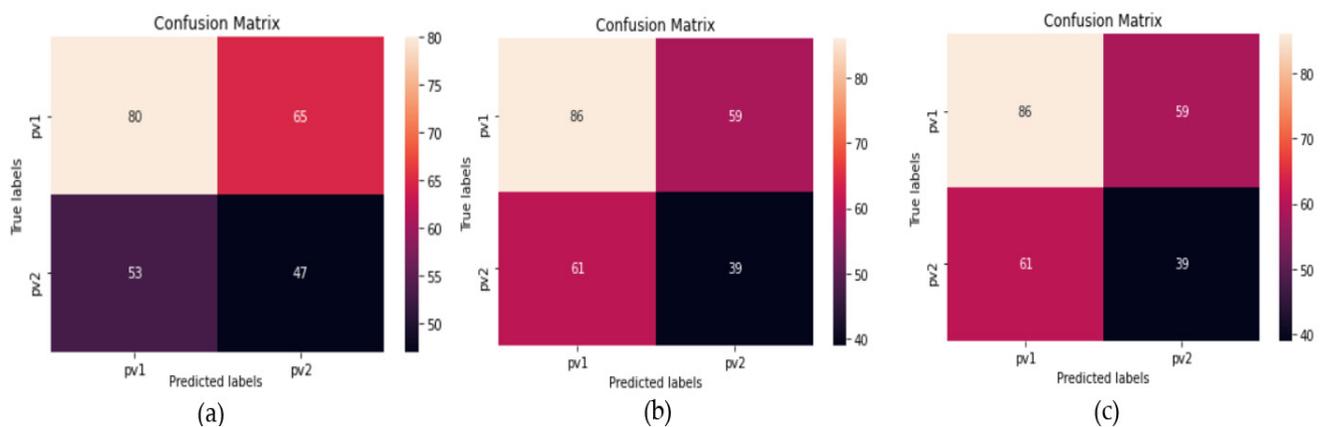


Figure 5. Model train and accuracy results for 15 epochs (a), 20 epochs (b) and 25 epochs (c).

Additionally, from the same figure it is also obvious that the model performs better when trained for 15 epochs (although the training performance in this session is slightly smaller compared to the performance during the next training session). As it can be seen in model accuracy section of the diagrams, the model validation line follows more closely the training line. Generally, models with a smaller curve fluctuation during accuracy elevation have better training convergence. Furthermore, model training is better when the two curves (train and validation) are closer. After training completion, the model is also tested against new data which were not used during train and validation sessions. The produced evaluation results are shown on Table 5 and Figure 6 which also prove that the model trained for 15 epochs provides the best overall predictions.

Table 5. Evaluation results.

| Results      | Precision | 15 Epochs |                      |         | 20 Epochs |        |                      |         | 25 Epochs |        |                      |         |
|--------------|-----------|-----------|----------------------|---------|-----------|--------|----------------------|---------|-----------|--------|----------------------|---------|
|              |           | Recall    | F <sub>1</sub> Score | Support | Precision | Recall | F <sub>1</sub> Score | Support | Precision | Recall | F <sub>1</sub> Score | Support |
| Pv1          | 0.60      | 0.55      | 0.58                 | 145     | 0.57      | 0.53   | 0.55                 | 145     | 0.59      | 0.59   | 0.59                 | 145     |
| Pv2          | 0.42      | 0.47      | 0.44                 | 100     | 0.38      | 0.42   | 0.40                 | 100     | 0.40      | 0.39   | 0.39                 | 100     |
| Accuracy     |           |           | 0.52                 | 245     |           |        | 0.49                 | 245     |           |        | 0.51                 | 245     |
| Macro Avg    | 0.51      | 0.51      | 0.51                 | 245     | 0.48      | 0.48   | 0.48                 | 245     | 0.49      | 0.49   | 0.49                 | 245     |
| Weighted Avg | 0.53      | 0.52      | 0.52                 | 245     | 0.49      | 0.49   | 0.49                 | 245     | 0.51      | 0.51   | 0.51                 | 245     |



**Figure 6.** Confusion matrices for 15 epochs (a), 20 epochs (b) and 25 epochs (c).

On Table 5, Pv 1 refers to the high-resolution images' dataset, whereas Pv 2 refers to the low-resolution images' dataset. Precision is the ability of the classifier not to label as positive a sample that is negative. Or in other terms, precision is the number of correct results divided by the number of all returned results.

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \quad (1)$$

Recall is the ability of the classifier to find all the positive samples. Or in other terms, recall is the fraction of relevant documents that are successfully retrieved.

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \quad (2)$$

$F_1$  score is a measure of the test's accuracy. It is the harmonic mean of the precision and recall:

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \quad (3)$$

The worst value for this measure is 0 whereas the best is obtained when it equals to 1.

Accuracy is the weighted arithmetic mean of Precision and Inverse Precision (weighted by Bias) as well as the weighted arithmetic mean of Recall and Inverse Recall (weighted by Prevalence). Inverse Precision and Inverse Recall are simply the Precision and Recall of the inverse problem where positive and negative labels are exchanged. Higher accuracy values demonstrate better model performance.

Macro Average, computes the  $F_1$  for each label and returns the average without considering the proportion for each label (in our case high- and low-resolution PV images) in the dataset. Weighted Average computes  $F_1$  for each label (in our case high- and low-resolution PV images) and returns the average considering the proportion of each label to the dataset. Finally, support is the number of occurrences of the given class (or label) in the dataset.

The results on Table 4 indicate that the trained model produce's better results for 15 epochs in both datasets (high and low resolution).

## 5. Discussion

For most researchers, terms such as deep learning and machine learning seem interchangeable concerning the world of artificial intelligence. However, this approach is mistaken. Deep learning is a specialized subset of machine learning which, in turn, is a specialized subset of artificial intelligence. Deep learning describes algorithms that analyze data with a structure which is similar to how a human would draw to a conclusion. The

only drawback in the application of DL is the requirement of incredibly vast amounts of data and the need for substantial computing power for its usage.

However, the application of deep learning algorithms nowadays is a necessity. The evolvement of Internet of Things has created multiple devices capable of collecting a variety of unstructured data, ranging from simple arithmetic values to images from satellites. Therefore, the need arises to evaluate this data and extract useful patterns. DL algorithms have no requirement for human intervention as the nested layers in the neural networks put data through hierarchies of different concepts, and eventually learn through their own errors. Therefore, the usage of DL algorithms can greatly help toward the process of collected data, mainly because these algorithms ignore the data types which are processing. Thus, they can (if trained properly) be used for solving many problems, including image detection and classification.

This study presents a novel approach towards the problem of automatic recognition of PV farms. The recognition is based on the usage of satellite imagery and image classification techniques which until recently were used for other purposes (face recognition, flora and fauna species recognition, etc.). According to our research it is the first time that neural networks (in particular a CNN) was used for the automatic detection of PV farms. From the literature review we conducted, the only similar research used a CNN for the determination of small rooftop installed PV arrays, however we did not find any other similar research, which indicates that our approach is pioneering.

Furthermore, another novelty of our approach is that the used dataset's as well as the software (libraries, functions algorithms) used for the implementation of this research are freely available to the researchers, thus making our methodology easily replicable.

The results showed that (even though the original dataset was rather small) we can expect correct identification accuracy reaching 60% when using high resolution imagery and lower results in case we use lower resolution. From the confusion matrixes we can determine that for 15 epochs 127 correct identifications were performed, 125 correct identifications were performed for 20 epochs and 125 were also recognized correctly for 25 epochs.

However, the identification results can be further improved if we use larger datasets. Additionally, the results showed that, increase in the number of training epochs does not provide significant improvements. Table 5 presents the results showing that 15 training epochs can be considered adequate for the dataset used.

Finally, the application of the algorithm also proven that high resolution images perform significantly better even in smaller datasets compared to low resolution imagery. This result was not expected because we believed that increasing the number of low-res input data could compensate for the lower resolutions, mainly due to the fact that input data are characterized by a specific geometry.

The approach presented in this work can also be applied in the recognition of other types of RES, if trained properly. It can also be used in other cases where automatic image recognition is necessary. The results could be improved by using images provided from paid services (and therefore high resolution) and by using larger datasets. Further improvements can be achieved if the user performs some kind of image pre-processing on the dataset (edge detection, color corrections, etc.), or deeper networks (more hidden layers).

## 6. Conclusions

Image recognition can provide a valuable tool for monitoring the adaption rate of renewable energy sources. Modern deep learning methods are unaware of the processing data and therefore can be easily used in order to recognize the various forms of RES (wind turbines, PV panels, hydroelectric stations, etc.). However, there is a need for large datasets in order to train properly the algorithms. The existence of various satellite imagery services allows the user to collect these data in a variety of resolutions and create datasets which contain images of RES forms in a variety of installation environments, various angles,

different weather and time. Therefore, it is possible to create a tool which will be capable of identifying them with increased accuracy.

This paper examined a first approach towards this goal. The dataset is based on the usage of PV farms in Greece and the results proved to be adequate given the size of the training dataset. As the years pass and more installations complete the algorithm can be trained again in order to increase its efficiency. Furthermore, advancements in computer technology and DL algorithms can also help towards this goal.

Finally, the combination of these algorithms with other types of software capable of calculating the annual solar energy output can help local and regional authorities to plan their energy policy. The methodology can also be used from the national authorities in an attempt to continuously monitor current RES status, determine the investment/adoption rate of RES in the various regions and regional units, and act as an overall tool for the application of national policy.

**Author Contributions:** Conceptualization, K.I.; methodology, K.I.; software, K.I.; validation, K.I. and D.M.; formal analysis, K.I.; resources, K.I. and D.M.; data curation, K.I.; writing—original draft preparation, K.I. and D.M. Both authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

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

## References

- Rafaj, P.; Barreto, L.; Kypreos, S. Combining policy instruments for sustainable energy systems: An assessment with the GMM model. *Environ. Model. Assess.* **2006**, *11*, 277–295. [CrossRef]
- Konstantinos, I.; Georgios, T.; Garyfalos, A. A Decision Support System methodology for selecting wind farm installation locations using AHP and TOPSIS: Case study in Eastern Macedonia and Thrace region, Greece. *Energy Policy* **2019**, *132*, 232–246. [CrossRef]
- Tsantopoulos, G.; Arabatzis, G.; Tampakis, S. Public attitudes towards photovoltaic developments: Case study from Greece. *Energy Policy* **2014**, *71*, 94–106. [CrossRef]
- Rahman, S.M.; Miah, M.D. The impact of sources of energy production on globalization: Evidence from panel data analysis. *Renew. Sustain. Energy Rev.* **2017**, *74*, 110–115. [CrossRef]
- Arto, I.; Capellán-Pérez, I.; Lago, R.; Bueno, G.; Bermejo, R. The energy requirements of a developed world. *Energy Sustain. Dev.* **2016**, *33*, 1–13. [CrossRef]
- Andreoni, V. Energy Metabolism of 28 World Countries: A Multi-scale Integrated Analysis. *Ecol. Econ.* **2017**, *142*, 56–69. [CrossRef]
- International Energy Outlook 2020*; International Energy Agency: Paris, France, 2020.
- Doukas, H.; Karakosta, C.; Psarras, J. RES technology transfer within the new climate regime: A “helicopter” view under the CDM. *Renew. Sustain. Energy Rev.* **2009**, *13*, 1138–1143. [CrossRef]
- Ohl, R.S. Light-Sensitive Electric Device. US Patent Office: 2,402,662, 25 June 1946.
- Wolfe, P. An overview of the world’s largest solar power plants. *PV Magazine International*. 18 June 2019. Available online: <https://www.pv-magazine.com/2019/06/18/an-overview-of-the-worlds-largest-solar-power-plants/> (accessed on 30 March 2021).
- Eurostat Renewable Energy Statistics. Available online: [https://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable\\_energy\\_statistics](https://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable_energy_statistics) (accessed on 30 March 2021).
- Jayanthi, S.; Vennila, C. Performance improvement in satellite image classification using adaptive supervised multi-resolution approach. *Comput. Commun.* **2020**, *150*, 200–208. [CrossRef]
- Liu, M.; Fu, B.; Xie, S.; He, H.; Lan, F.; Li, Y.; Lou, P.; Fan, D. Comparison of multi-source satellite images for classifying marsh vegetation using DeepLabV3 Plus deep learning algorithm. *Ecol. Indic.* **2021**, *125*, 107562. [CrossRef]
- Ioannou, K.; Tsantopoulos, G.; Arabatzis, G.; Andreopoulou, Z.; Zafeiriou, E. A Spatial Decision Support System Framework for the Evaluation of Biomass Energy Production Locations: Case Study in the Regional Unit of Drama, Greece. *Sustainability* **2018**, *10*, 531. [CrossRef]
- Ioannou, K.; Karampatzakis, D.; Amanatidis, P.; Aggelopoulos, V.; Karmiris, I. Low-Cost Automatic Weather Stations in the Internet of Things. *Information* **2021**, *12*, 146. [CrossRef]
- Dong, S.; Wang, P.; Abbas, K. A survey on deep learning and its applications. *Comput. Sci. Rev.* **2021**, *40*, 100379. [CrossRef]

17. Freedman, D. *Statistical Models: Theory and Practice*; Cambridge University Press: Cambridge, UK, 2009.
18. Mood, C. Logistic Regression: Why We Cannot Do What We Think We Can Do, and What We Can Do About It. *Eur. Sociol. Rev.* **2010**, *26*, 67–82. [[CrossRef](#)]
19. Kleinbaum, D.; Klein, M. *Logistic Regression: Why We Cannot Do What We Think We Can Do, and What We Can Do about It*; Springer: Berlin/Heidelberg, Germany, 2002.
20. Hosmer, W.; Lemeshow, S.; Sturdivart, X. *Applied Logistic Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
21. Soentpiet, R. *Advances in Kernel Methods: Support Vector Learning*; MIT Press: Cambridge, MA, USA, 1999.
22. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Appl.* **1998**, *13*, 18–28. [[CrossRef](#)]
23. Steinwart, I.; Christmann, A. *Support Vector Machines*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
24. Schraudolph, N.N. Fast Curvature Matrix-Vector Products for Second-Order Gradient Descent. *Neural Comput.* **2002**, *14*, 1723–1738. [[CrossRef](#)]
25. Li, Z. *Encyclopedia of Biometrics I-Z*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
26. Verbeek, J.J.; Vlassis, N.; Kröse, B. Efficient Greedy Learning of Gaussian Mixture Models. *Neural Comput.* **2003**, *15*, 469–485. [[CrossRef](#)] [[PubMed](#)]
27. Gavali, P.; Banu, J.S. Chapter 6—Deep Convolutional Neural Network for Image Classification on CUDA Platform. In *Deep Learning and Parallel Computing Environment for Bioengineering Systems*; Arun, K.S., Ed.; Academic Press: Cambridge, MA, USA, 2019; pp. 99–122. ISBN 978-0-12-816718-2.
28. Quiroz, I.A.; Alférez, G.H. Image recognition of Legacy blueberries in a Chilean smart farm through deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105044. [[CrossRef](#)]
29. Tan, W.; Zhao, C.; Wu, H.; Gao, R. A deep learning network for recognizing fruit pathologic images based on flexible momentum. *Nongye Jixie Xuebao/Trans. Chin. Soc. Agric. Mach.* **2015**, *46*, 20–25. [[CrossRef](#)]
30. Liu, Z.; Xiang, X.; Qin, J.; Ma, Y.; Zhang, Q.; Xiong, N.N. Image Recognition of Citrus Diseases Based on Deep Learning. *Comput. Mater. Contin.* **2020**, *66*, 457–466. [[CrossRef](#)]
31. Amara, J.; Bouaziz, B.; Algergawy, A. A Deep Learning-based Approach for Banana Leaf Diseases Classification. In *Datenbanksysteme für Business, Technologie und Web (BTW 2017)—Workshopband*; Mitschang, B., Nicklas, D., Leymann, F., Schöning, H., Herschel, M., Teubner, J., Härder, T., Kopp, O., Wieland, M., Eds.; Gesellschaft für Informatik: Bonn, Germany, 2017; pp. 79–88.
32. Huang, H.W.; Li, Q.T.; Zhang, D.M. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. *Tunn. Undergr. Space Technol.* **2018**, *77*, 166–176. [[CrossRef](#)]
33. Yang, X.; Zhang, Y.; Lv, W.; Wang, D. Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier. *Renew. Energy* **2021**, *163*, 386–397. [[CrossRef](#)]
34. Cheng, L.; Zhang, X.; Shen, J. Road surface condition classification using deep learning. *J. Vis. Commun. Image Represent.* **2019**, *64*, 102638. [[CrossRef](#)]
35. Zhang, Y.-D.; Satapathy, S.C.; Guttery, D.S.; Górriz, J.M.; Wang, S.-H. Improved Breast Cancer Classification Through Combining Graph Convolutional Network and Convolutional Neural Network. *Inf. Process. Manag.* **2021**, *58*. [[CrossRef](#)]
36. Wang, S.-H.; Govindaraj, V.V.; Górriz, J.M.; Zhang, X.; Zhang, Y.-D. Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. *Inf. Fusion* **2021**, *67*, 208–229. [[CrossRef](#)]
37. Malof, J.M.; Bradbury, K.; Collins, L.M.; Newell, R.G. Automatic detection of solar photovoltaic arrays in high resolution aerial imagery. *Appl. Energy* **2016**, *183*, 229–240. [[CrossRef](#)]
38. Butler, H.; Daly, M.; Doyle, A.; Gillies, S.; CiteHagen, S.; Schaub, T. Internet Engineering Task Force RFC: 7946. 2016. Available online: <https://datacenter.ietf.org/doc/html/rfc7946> (accessed on 30 March 2021).
39. Harding, S.; Banzhaf, W. Fast genetic programming on GPUs. In Proceedings of the 10th European Conference on Genetic Programming; Ebner, M., O’Neill, M., Ekárt, A., Vanneschi, L., Esparcia-Alcázar, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 90–101.
40. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; USENIX Association: Berkeley, CA, USA, 2016; pp. 265–283.
41. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *J. Physiol.* **1962**, *160*, 106–154. [[CrossRef](#)]
42. Haykin, S. *Neural Networks A Comprehensive Foundation*, 2nd ed.; Subsequent Edition (1 July 1998); Prentice Hall: Hoboken, NJ, USA, 1999; ISBN 978-0132733502.
43. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten Digit Recognition with a Back-Propagation Network. In Proceedings of the Advances in Neural Information Processing Systems; Touretzky, D., Ed.; Morgan-Kaufmann: Burlington, MA, USA, 1990; Volume 2.
44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
45. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *CoRR* **2015**. [[CrossRef](#)]

- 
46. Yang, K.; Qinami, K.; Fei-Fei, L.; Deng, J.; Russakovsky, O. Towards fairer datasets. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*; ACM: New York, NY, USA, 2020; pp. 547–558.
  47. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]