



# PyPVRoof: a Python package for extracting the characteristics of rooftop PV installations using remote sensing data

Yann Trémenbert, Gabriel Kasmi, Laurent Dubus, Yves-Marie Saint-Drenan, Philippe Blanc

## ► To cite this version:

Yann Trémenbert, Gabriel Kasmi, Laurent Dubus, Yves-Marie Saint-Drenan, Philippe Blanc. PyPVRoof: a Python package for extracting the characteristics of rooftop PV installations using remote sensing data. 2023. hal-04209569

**HAL Id: hal-04209569**

**<https://hal.science/hal-04209569v1>**

Preprint submitted on 18 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# PyPVRoof: a Python package for extracting the characteristics of rooftop PV installations using remote sensing data

Yann Tremenbert<sup>\*1</sup>, Gabriel Kasmi<sup>1,2</sup>, Laurent Dubus<sup>2</sup>,  
Yves-Marie Saint-Drenan<sup>1</sup>, and Philippe Blanc<sup>1</sup>

<sup>1</sup>MINES Paris, Université PSL, Centre Observation Impacts  
Energie (O.I.E.), 06904 Sophia Antipolis, France

<sup>2</sup>RTE - Réseau de transport d'électricité, 92073 Paris La Défense,  
France

## Abstract

Photovoltaic (PV) energy grows at an unprecedented pace, which makes it difficult to maintain up-to-date and accurate PV registries, which are critical for many applications such as PV power generation estimation. This lack of qualitative data is especially true in the case of rooftop PV installations. As a result, extensive efforts are put into the constitution of PV inventories. However, although valuable, these registries cannot be directly used for monitoring the deployment of PV or estimating the PV power generation, as these tasks usually require PV systems *characteristics*. To seamlessly extract these characteristics from the global inventories, we introduce **PyPVRoof**. **PyPVRoof** is a Python package to extract essential PV installation characteristics. These characteristics are tilt angle, azimuth, surface, localization, and installed capacity. **PyPVRoof** is designed to cover all use cases regarding data availability and user needs and is based on a benchmark of the best existing methods. Data for replicating our accuracy benchmarks are available on our Zenodo repository [1], and the package code is accessible at this URL: <https://github.com/gabrielkasmi/pypvrproof>.

## 1 Introduction

Photovoltaic (PV) installed capacity grows quickly [2, 3] as it is key to decarbonizing energy systems [4]. Keeping track of the deployment and characteristics of the PV installed capacity can be difficult, and public authorities and

---

<sup>\*</sup>Work done while in internship at RTE

industrial stakeholders often lack precise knowledge regarding the PV fleet [5]. This concern is especially true in the case of rooftop PV, for which no centralized and disaggregated registry are generally available [6]. The lack of reliable information on the PV installed capacity can yield unreliable rooftop PV power generation estimates. For instance, for transmission system operators (TSOs), the lack of reliable rooftop PV measurements increases the flexibility needs, i.e., the ability of the grid to compensate for load or supply variability [7, 8, 9, 10]. Therefore, there is a growing need for reliable and more accurate rooftop PV mapping and characterization, i.e., estimation of the technical characteristics of the installation.

Numerous efforts were carried out to acquire information regarding PV installations in general and distributed PV in particular. For instance, Dunnett et al. [11] proposed a harmonized dataset using publicly available data from OpenStreetMap. Stowell et al. [12] leveraged crowdsourcing to map approximately 86% of the United Kingdom’s distributed PV installed capacity. In their approach, users were asked to delineate PV panels in their neighborhood. Alternatively, several works leveraged deep learning and overhead imagery to map PV installations quickly [13, 14, 15, 16]. Usually, a model is trained on a training dataset (e.g., [17, 5, 18]) and then deployed on a larger area. Overhead imagery comes as large tiles, so they are cut into small patches and passed into the trained model, which will return the probability that each image pixel depicts a PV panel. One finally transforms the so-called resulting segmentation map into a set of geolocalized polygons of the same format as the crowdsourced works previously mentioned.

Although valuable, PV polygons are insufficient for many applications (e.g., PV power generation estimation [19]). In order to provide installation characteristics, several methods have been proposed. Edun et al. [20] leveraged the Hough transform for azimuth estimation. Mayer et al. [21] used heuristics and LiDAR surface models to create PV registries containing tilt, azimuth, and installed capacity. So et al. [22] showed that installed capacity could be derived from the surface using a linear model. Recently, Perry et al. [23] introduced a python package to extract information such as the mounting configuration of PV panels. The main limitation of these works is that they have different data requirements, which limits their reproducibility.

To standardize the estimation of rooftop PV systems characteristics, we introduce **PyPVRoof**. This Python package is designed to estimate the technical characteristics (tilt and azimuth angles, installed capacity, localization, surface) from its geolocalized polygon and additional input data, i.e., preexisting PV registries and/or digital surface models (DSMs). **PyPVRoof** combines characteristics extraction methods chosen after a careful benchmark on the training dataset BDAPPV [5], which contains PV segmentation masks and installation characteristics. We designed this package to work under the most common scenarios of data availability.

To the best of our knowledge, **PyPVRoof** is the first attempt towards a standardized package for the extraction of PV systems characteristics. It works with polygons, the usual way of encoding PV systems localization, and handles

different additional data sources to work even if the user does not have digital surface models (DSMs) or existing data sources.

The remainder is organized as follows: in section 2, we review existing works focusing on PV mapping and characteristics extraction. In section 3 we present how our package works, its methods and metrics used for evaluation. In section 4, we review the data that we use for our benchmark and that the package requires to work. In section 5, we present our benchmark results and discuss the final choice of methods included in `PyPVRoof`. Section 6 concludes.

## 2 Related works

### 2.1 Inventories of PV systems

Keeping track of the deployment of PV installation is challenging as it is growing at an unprecedented pace. It is especially true in the case of distributed PV since the latter is aggregated at the city or census scale in current available public registries, with no information regarding the characteristics of the underlying installations [24, 6]. As a response, numerous initiatives emerged to map PV installations, ranging from power plants [25, 11, 26, 27] to rooftop PV installations [13, 14, 15, 16, 6, 21, 12]. When released, these registries take the form of geolocalized polygons, usually in the `.geojson` format. These polygons can be accessed through online platforms such as OpenInfraMap [28]. These inventories come from automatic detection from overhead imagery ([15, 6]) or directly through crowdsourcing efforts ([12]).

### 2.2 PV systems characteristics extraction

In addition to detecting and mapping PV installations, extracting the characteristics has greatly interested researchers and public authorities. Indeed, the installed capacity, expressed in kWp, is the primary indicator to keep track of the deployment of PV [29]. As a response, several methods were proposed to extract PV characteristics in addition to geolocalized polygons. So et al. [22] estimated the installed capacity from the polygon’s surface using a linear regression model. Rausch et al. [30] and Mayer et al. [21] further refined this approach to propose a method for automatically constructing a detailed registry using overhead imagery and 3D data. More recently, Kasmi et al. [6] proposed a method for mapping and characterizing PV installations without needing surface models. A few works also proposed methods to extract the azimuth angle of the installations ([20]) or the tilt and azimuth category (i.e., angle and azimuth ranges, [31]). More recently [23] proposed their package for extracting PV systems characteristics using deep learning and satellite imagery. Their package extracts the azimuth angle using the method of Edun et al. [20] and the configuration type (mounted, rooftop, tracker). However as their approach only relies on satellite images, they do not extract other characteristics, in particular the tilt angle of the installation, leading to an incomplete characterization of the

PV installations. We show that a comprehensive characterization requires additional data sources (i.e., auxiliary data or digital surface models), in particular for the tilt angle and installed capacity estimation.

### 3 Methods

#### 3.1 PyPVRoof: a Python package for extracting PV installations characteristics from geolocalized polygons

##### 3.1.1 Overview

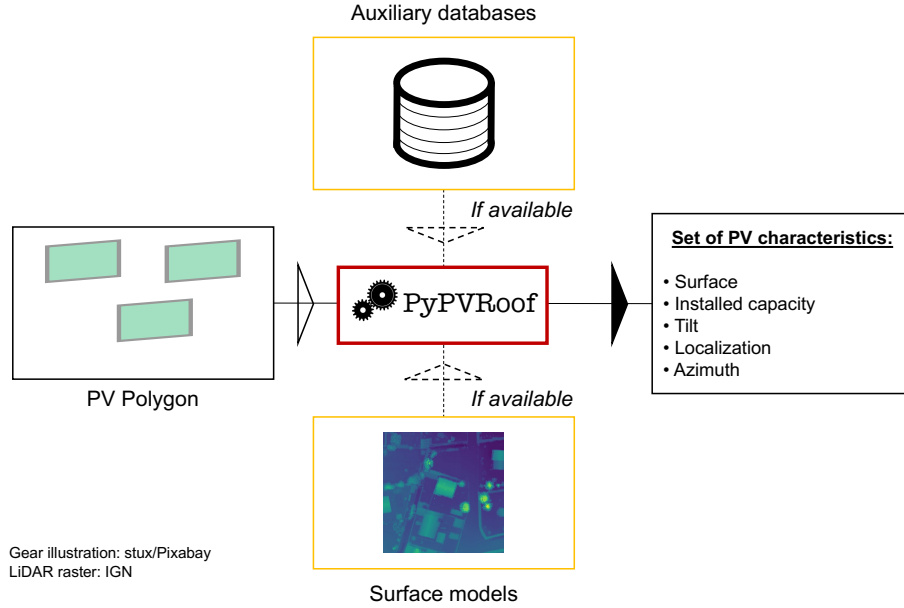


Figure 1: Flowchart of the proposed method to extract installations' characteristics

Figure 1 summarizes the workflow of PyPVRoof. PyPVRoof extracts PV characteristics from geolocalized polygons. It accommodates additional data sources, such as preexisting registries (i.e., auxiliary data) or digital surface models (DSM), depending on their availability for the user. The list of characteristics that we extract is the following:

- Localization (latitude and longitude)
- Tilt angle (in degrees)
- Azimuth angle (in degrees, relative to North)

- Surface (in  $\text{m}^2$ ). Estimating the surface requires knowing the tilt, as only the *projected* surface is derived from the input polygon.
- Installed capacity (in kWp). The surface is needed to estimate the installed capacity as its first-order approximation is the surface multiplied by an efficiency factor [22].

This set of characteristics covers various use cases, ranging from PV systems inventories to PV power forecasting. It is sufficient to describe an installation using physical models [32]. To extract all characteristics mentioned above or only some of them, the user only has to specify a method (provided that the data requirements are satisfied) and pass the polygon as input. Further details and tutorials are accessible on the public repository, accessible at this URL: <https://github.com/gabrielkasmi/pypvroof>.

### 3.1.2 PyPVRoof combines best-in-their-class methods

PyPVRoof combines methods for characteristics extraction based on a review of existing works in the field. We reviewed the main methods used, evaluated their data requirements, and benchmarked them to keep the best-performing methods. Further details about our benchmark results are provided in section 5. These methods were chosen based on accuracy, simplicity, and efficiency. In particular, we retained the most simple between two equally performing methods (e.g., the look-up table over the random forest). Finally, we restricted ourselves to methods that require as few additional inputs as possible. These methods reflect the current state-of-the-art for characteristics extraction.

### 3.1.3 An approach that adapts to the available data

As characteristics extraction is usually part of a larger pipeline (e.g., remote PV mapping such as in Mayer et al. [21], or Kasmi et al. [6]), resulting in multiple use cases depending on the approach proposed and the data available to the authors. Some models only rely on overhead imagery and only focus on surface estimation ([14]), while other approaches are more comprehensive but also require more data ([21]). PyPVRoof was developed to standardize existing approaches in both inputs and outputs. We also accommodate that additional data (e.g., auxiliary registries and/or surface models) can be considered during the process. Auxiliary data correspond to a registry containing PV characteristics, such as the characteristics file of [5]. The primary purpose of the auxiliary data file is to provide a first guess on the tilt angle and the panel efficiency value. Therefore, it can only be a sample of PV characteristics rather than a comprehensive registry. Surface models, i.e., digital surface models (DSM) correspond to a type of geographical information system (GIS) that delivers information about the height of objects (natural and built) on the ground. These models are often used to infer the tilt angle of buildings [33, 34] or PV panels [21].

**Use-case 1: auxiliary data** Figure 2 presents the flowchart and the associated methods to extract PV characteristics if the user has only access to auxiliary data. In this case, PyPVRoof leverages this data to calibrate the panel efficiency module coefficient to correlate the installation’s surface with an installed capacity. A look-up table (LUT) is also computed from this input data for the tilt angle estimation. Finally, we apply a bounding box algorithm to estimate the azimuth angle.

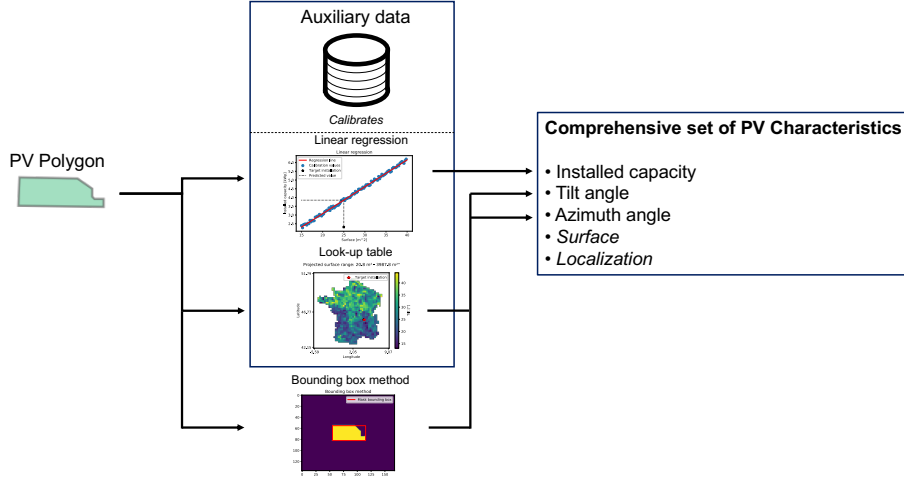


Figure 2: PyPVRoof flowchart if only auxiliary data is available

**Use-case 2: DSM data** Figure 3 presents the flowchart and the associated methods to extract PV characteristics if the user can only access digital surface models. Using a Theil-Sen estimator, we leverage the DSM to estimate the tilt *and* azimuth angles. On the other hand, the panel efficiency coefficient is a parameter set by the user.

**Use-case 3: no data** Figure 4 presents the flowchart and the associated methods to extract PV characteristics if no auxiliary data is available. In this case, the user imputes the panel efficiency coefficient and an average tilt angle, but the azimuth angle is estimated using the bounding-box algorithm.

In the project repository, we provide a tutorial that enables the user to try various methods to extract characteristics from polygons using different methods.

### 3.2 Detailed presentation of the methods

In this section, we review the methods that we kept in PyPVRoof. We refer the reader to the appendix A for a description of the remaining benchmarked methods.

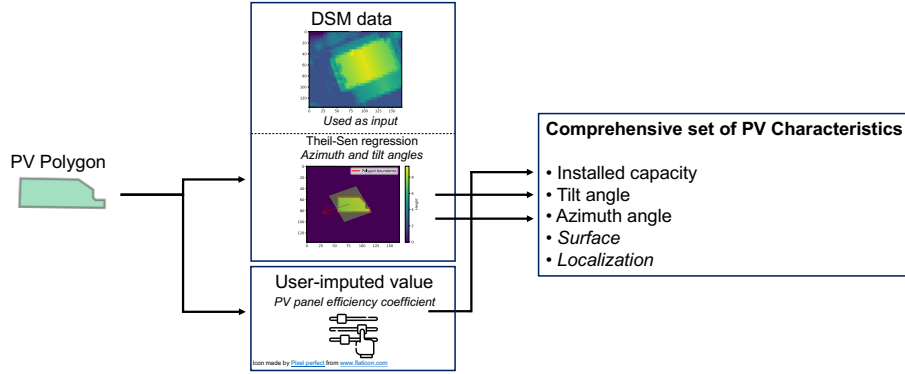


Figure 3: PyPVRoof flowchart if only digital surface models (DSMs) are available

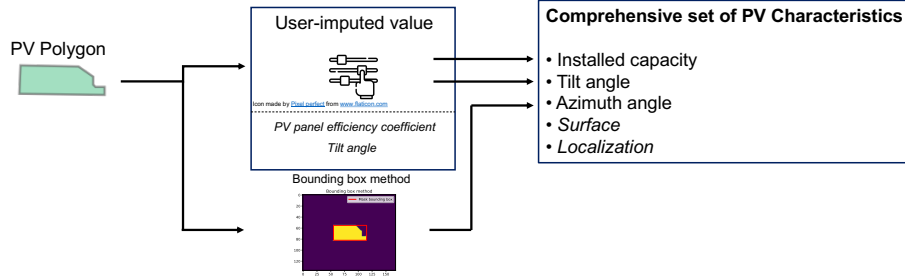


Figure 4: PyPVRoof flowchart if no complementary data is available

### 3.2.1 Direct computation

Overhead imagery is usually orthorectified (i.e., with a uniform scale). One can only compute the *projected* surface from the polygon. The computation is straightforward, and the only requirement is considering the projection. Parhar et al. [35] describe the Mercator case. In practice, packages such as `area` [36] estimate the surface of `geojson` polygons, taking into account the deformation induced by the projection system.

Once the projected surface is known, one needs the tilt angle to compute the real surface. Denoting  $S_{proj}$  the projected surface and  $\theta$  the tilt angle of the installation (in degrees), the real surface is given by equation (1).

$$S = S_{proj} / \cos\left(\theta \times \frac{\pi}{180}\right) \quad (1)$$

### 3.2.2 Constant parameters

**Constant tilt** Tilt is necessary to compute the real surface of the installation. When neither registries nor surface models are available, it is still possible to infer a tilt angle from the remaining data (i.e., the PV polygon). However, in



practice, the optimal tilt angle of an installation is known. Typically, a tilt angle of around 30 degrees is optimal in most European countries. Regional models estimating the PV yield of solar plants consider this value by default [37, 38]. In our case, we allow the user to input a default coefficient if necessary. This case can be seen as a worst-case situation if no surface models nor auxiliary data is available.

**Constant efficiency** An efficiency factor relates the surface of a PV installation and its installed capacity. The PV panel efficiency increased due to the cell efficiency increase over the last couple of decades [39]. This efficiency is usually measured in  $\text{kWp}/\text{m}^2$ . The efficiency depends on many criteria (e.g., module technology of the panel, aging, manufacturer), which are not necessarily publicly available. However, average efficiencies can be used. For instance, [30, 21] used a value of  $6 \text{ kWp}/\text{m}^2$  as a reference value to estimate the installed capacity from the surface. As for the tilt angle, we allow the user to input this efficiency value.

### 3.2.3 Theil-Sen estimation

The Theil-Sen estimator (TSE) initially proposed by Theil [40] and Sen [41] is a robust regression method. It consists in considering the median of the slopes of all lines (or planes in higher dimensions) through pairs of points. This method is more robust to outliers than ordinary least squares.

We use this method to fit a plane  $z(x, y) = ax + by + c$  parameterized by only three parameters,  $a, b$  and  $c$ , to a set of points corresponding to altitudes. These altitudes come from the digital surface model (DSM) passed as input. Figure 5 depicts an example of LiDAR DSM provided by the IGN. Lighter areas correspond to higher altitudes.

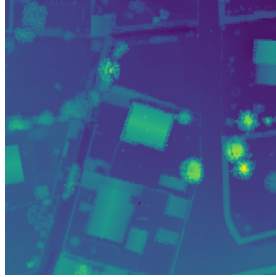


Figure 5: Example of DSM: the rasterization of the LiDAR from the IGN

The direction of the gradient of the plane gives the azimuth angle  $\varphi$ . The slope value along this gradient corresponds to the tilt angle  $\theta$ . The gradient of the plane  $\nabla z(x, y)$  is given in equation (2):

$$\nabla z(x, y) = \left( \frac{\partial z}{\partial x}(x, y), \frac{\partial z}{\partial y}(x, y) \right) = (a, b) \quad (2)$$

and

$$\varphi = \arctan\left(\frac{a}{b}\right), \theta = \arctan\left(\frac{h}{d}\right) \quad (3)$$

where  $h = a^2 + b^2$  and  $d = \sqrt{a^2 + b^2}$ . Figure 6 depicts the principle of the Theil-Sen method to compute the tilt and azimuth angles.

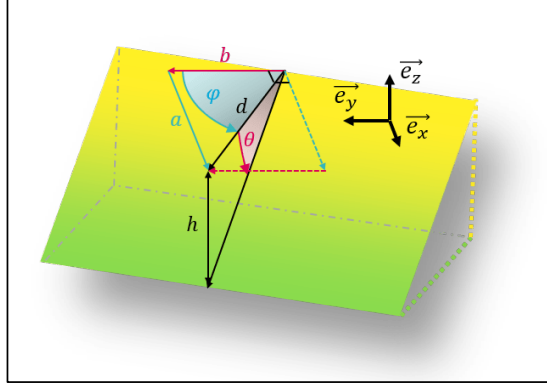


Figure 6: Theil-Sen method principle. The plane is deduced from the raster and is parameterized as  $z(x, y) = ax + by + c$ .  $\varphi$  corresponds to the azimuth angle and  $\theta$  to the tilt angle.  $\vec{e}_x, \vec{e}_y, \vec{e}_z$  correspond to the canonical basis of  $\mathbb{R}^3$ .

### 3.2.4 Look-up table

If surface models are unavailable, we can still recover a tilt angle more accurately than with direct computation. To achieve this, we only need a sample of tilt angles for the desired area, e.g., a smaller PV database or a building database. We can then reflect the spatial variability of the tilt angle by computing an average tilt angle per grid point. The reference value associated with the installation corresponds to the average of the existing installations located in this grid point. The lookup table requires that the auxiliary data frame span the complete area or interest (e.g., a region or a country).

We compute this so-called lookup table (LUT) only once, and the user can pass a precomputed LUT as input. Computation is done as follows: we first define the spatial extent by setting easternmost  $E$ , northernmost  $N$ , westernmost  $W$ , and southernmost  $S$  boundaries. These boundaries are expressed in geographical coordinates. We then define a grid by dividing the numerical intervals defined by  $E$  and  $W$  and  $S$  and  $N$  respectively. We end up with  $K$  longitude intervals and  $L$  latitude intervals. Besides, we cluster the auxiliary data frame by (projected) surface category to define  $T$  surface categories. After empirical investigations, defining intervals as quantiles yielded the best results.

We then aggregate all sample points  $x_1^{k,l,t}, \dots, x_n^{k,l,t}$  whose coordinates belong to the  $k \times l - th$  grid point and projected surface that belong to the  $t - th$  category.

We then compute the reference tilt angle for this  $(k, l, t)$ -th box, denoting  $\theta^{k,l,t}$  by averaging the tilt values of the  $n$  sample points falling into this bin. If no sample is available, we do not input a value.

Once this step is finished, we end up with a subset of grid points for which no reference value is available. We estimate a value by interpolating a  $\theta^{k,l,t}$  by interpolating the neighboring values. We do not interpolate across surface categories. Figure 7 displays the LUT obtained for the PV mapping algorithm of Kasmi et al. ([6]) using this method.

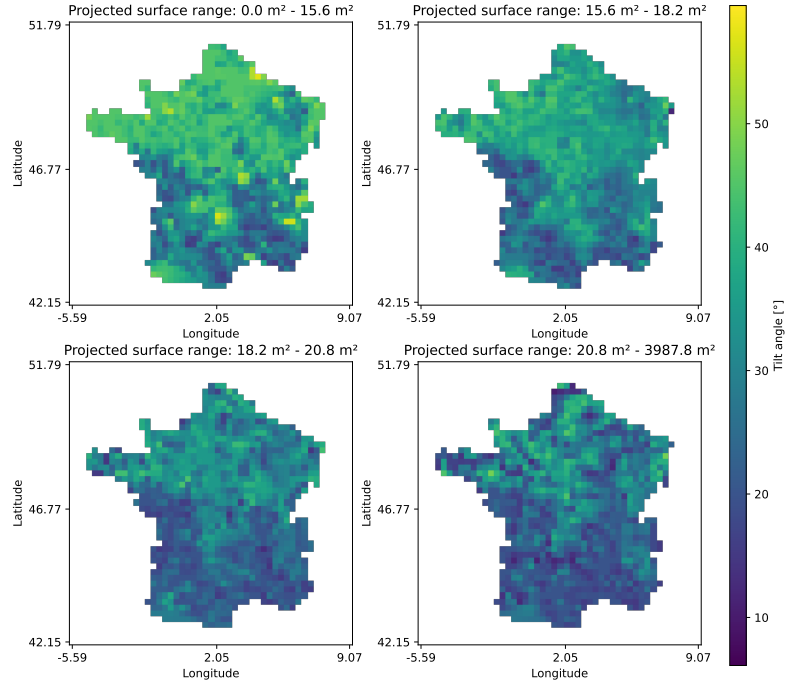


Figure 7: Lookup table for  $50 \times 50$  grid-points and four surface categories computed for the PV mapping algorithm of Kasmi et al.[6]. Surface categories correspond to quartiles of the distribution of the surface in the auxiliary data.

### 3.2.5 Bounding-box

The bounding box method only requires the polygon to compute the azimuth angle  $\varphi$ . The bounding-box method is an alternative when no surface models are accessible. We simplify the polygon's geometry by computing its bounding box.

Then, we compute the azimuth angles associated with the "long" and "short" sides of the rectangle. We input as azimuth angle the angle corresponding to the longest side. We implicitly assume that the PV panel tends to be wider than high. The main limitation of this method is that it cannot distinguish between a panel facing eastwards or westwards, northwards or southwards. In the latter case, however, we can assume that the PV panel should not point northwards (at least in the Northern Hemisphere). If our bounding-box heuristic estimates that the polygon points between -45 and 45 degrees (0 being the reference for the North), we correct the estimation by applying a horizontal symmetry.

### 3.2.6 Linear regressions

So et al. [22] showed that it is possible to accurately estimate the installed capacity by fitting a linear regression between the surface and the installed capacity. We build on this method. The linear model is given by equation (4).

$$c = \gamma_0 + \gamma S \quad (4)$$

Where  $S$  is the surface in  $\text{m}^2$  and  $c$  is the capacity in kWp of the installation. As pointed out by So et al., [22],  $\gamma_0$  is a bias coefficient; in the true model,  $\gamma_0$  should be equal to zero. In our case, we consider  $\gamma_0 = 0$  and estimate  $\gamma$  from BDAPPV.

Efficiencies can differ depending on the PV installation's surface [39]. To accommodate this, we introduce another estimation for the installed capacity, namely the clustered linear regression. Clusters are defined depending on the surface of the installation. The goal is to reflect the different efficiencies while keeping the number of parameters as low as possible. This approach is inspired by the second model of So et al. [22], which estimated a panel-wise coefficient  $\gamma$ . Their approach, however, required additional unobservable information, such as the manufacturer's design.

Figure 8 represents the linear regression of the installed capacity on the surface and shows the relatively low dispersion of points around this mean. The leftmost plot shows the different coefficients depending on the surface cluster. We focus on surfaces lower than  $200 \text{ m}^2$ , where the density of installations is the highest. We can see that the efficiencies recorded in our reference registry are higher for smaller installations.

## 3.3 Evaluation criteria of the methods

We evaluate each method based on metrics and the execution time. The performance metrics are the following:

- Mean error (bias) (ME):  $\frac{1}{n} \sum_{i=1}^n \hat{x}_i - x_i$
- Mean absolute error (MAE):  $\frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i|$

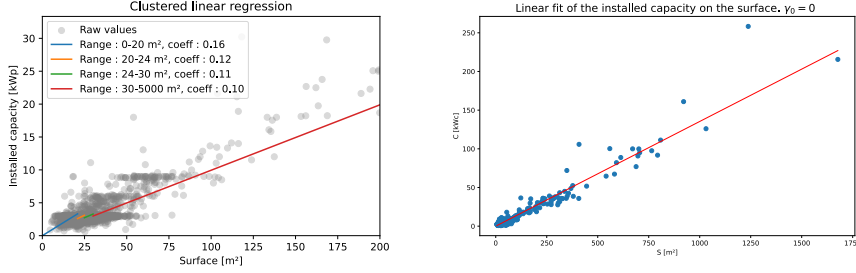


Figure 8: Left: clustered linear regression. Right: linear regression with a single coefficient.

- Root Mean Square Error (RMSE):  $\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2}$
- Mean absolute percentile error (MAPE):  $\frac{1}{n} \sum_{i=1}^n \frac{|\hat{x}_i - x_i|}{x_i}$

We evaluate our methods on the BDAPPV dataset, introduced in further detail in section 4. When training is required (e.g., random forests or lookup table), we consider an independent subset of the dataset so that the methods are all evaluated on unseen data samples.

## 4 Data

### 4.1 PV installations training database

PyPVRoof was developed on the PV characteristics dataset "BDAPPV" introduced by Kasmi et al. [5]. This dataset contains ground truth segmentation masks, images, and installation characteristics for more than 28000 installations in France and Western Europe, along with an explicit link between the segmentation masks and the installation characteristics for a subset of 8000 of them.

### 4.2 Geographical information system (GIS) data

We also rely on the surface models provided by the Institut Géographique National (IGN). These digital surface models (DSM) leverage two methods, the "photogrammetry" DSM and the LiDAR. Photogrammetry uses parallax to get the altitude points associated with each coordinate. Indeed, altitudes over an area are determined from different pictures from different points of view as nearer objects (from the aircraft carrying the aerial camera) move faster than distant objects. Such data is available almost everywhere in France, with a

ground resolution of around 20 cm/pixel and an altimetric precision of around 150 cm. The photogrammetry DSM is accessible at IGN’s Géoservices portal, accessible at this URL: <https://geoservices.ign.fr/>. The second technique, called Light Detection and Ranging (LiDAR), calculates distances from the reflection of a light beam on a surface. This technique has an altimetric resolution of 10 cm/pixels. Even though LiDAR raw data is composed of point clouds with around 10 points/m<sup>2</sup>, we have decided to interpolate and rasterize it to a 20 cm/pixel resolution to use the same developed methods to infer tilts and azimuths. By the time this article was written, the LiDAR DSM did not cover all of France and is only accessible for demonstration purposes on the IGN’s dedicated webpage, accessible at this URL: <https://geoservices.ign.fr/lidarhd>.

## 5 Results of the benchmark between methods

The results provided in this section can be replicated using the data available on our public Zenodo repository [1]. **These results are preliminary results and we do not draw any conclusions from them as we need a better baseline for a fair comparison of our methods.**

### 5.1 Summarized results

In this section, we report the accuracy results for each method and each characteristic. Our selected methods offer a snapshot of the best accuracy currently attainable.

Table 1 summarizes the accuracy results of these methods. We can see a large improvement gain when using LiDAR data, which we suspect can be even larger if the Theil-Sen algorithm is applied directly on the point cloud. We focused on rasters for a fair comparison with the photogrammetry DEM. Besides, a surprising result is that LiDAR data is better for azimuth angle than tilt estimation. An explanation for this is that azimuth estimation is less sensitive to noisy data points in the ( $z$ ) elevation direction than tilt estimation.

We highly recommend using the Theil-Sen method if the DSM is precise enough (e.g., coming from LiDAR data). Otherwise, the bounding-box method is competitive at a much lower computational cost.

Characteristic	Method	Accuracy (RMSE) [unit]
Surface	Direct computation	6.86 [m <sup>2</sup> ]
Tilt	LUT	10.29 [°]
	Theil-Sen (LiDAR)	14.69 [°]
Azimuth	Bounding-box	32.76 [°]
	Theil-Sen (LiDAR)	4.38 [°]
Surface	Linear regression	0.687 [kWp]

Table 1: Accuracy results of PyPVRoof’s methods for PV panels characteristics extraction

## 5.2 Detailed results

**Surface estimation** Table 2 reports the results. We observe small differences between annotated and predicted masks which assess the overall quality of the predicted masks. However, we give particular attention to the positive bias between masks and the surface reported in the characteristics file of BDAPPV, highlighting a tendency to overestimate the referenced surface area. Such bias is irrelevant since BDAPPV’s surface area values cannot be assessed: for instance, such an overrepresentation of installations of 20 m<sup>2</sup> could result from a systematic roundup.

Method	ME [m <sup>2</sup> ]	MAE [m <sup>2</sup> ]	RMSE [m <sup>2</sup> ]	Runtime [sec]
Direct computation	3.62	5.01	6.86	(-)

Table 2: Performance metrics for the estimation of the projected surface. The mean surface area is 20 m<sup>2</sup>.

**Tilt angle estimation** Table 3 presents preliminary results. For tilt estimation, it turned out that the LUT was a surprisingly strong baseline over the other methods: the random forest yielded only minor improvements, but the runtime is an order of magnitude larger. Although not significant for a single installation, such a difference in runtime is significant when scaling the method to thousands of PV polygons. As for the methods that require surface models, we can see that their accuracy relies on the quality of the input data. We tested the Theil-Sen method on photogrammetry-based surface models and LiDAR surface models. We can see a noticeable improvement when shifting from photogrammetry to LiDAR. **We cannot yet compare the LUT and the Theil-Sen methods. The results displayed in table 3 are only preliminary.**

Method	ME [°]	MAE [°]	RMSE [°]	Runtime [sec]
Random Forest	6e-4	5.34	7.03	0.28
Look-up table	-2.40	7.68	10.29	6e-6
Theil-Sen (Photogrammetry)	3.99	14.10	17.50	0.09
Theil-Sen (LiDAR)	2.06	11.08	14.69	0.09
Hough with DSM	2.90	13.45	16.62	2.47

Table 3: Performance metrics for the estimation of the tilt angle. The two lines for the Theil-Sen method report the accuracy results whether photogrammetry DSM or LiDAR DSM are passed as inputs.

**Azimuth angle estimation** For azimuth estimation, we replicated the method of [20] using the Hough algorithm. They report MAEs ranging from 15.62 to 30.53 degrees depending on the type of panel considered, the largest errors being associated with rooftop panels and the smallest with ground panels. Our replication is, therefore, in line with theirs, as we report an MAE of 22.70 degrees. Surprisingly, we see very few improvements brought by the Hough method with surface models. On the other end, the bounding-box method, which solely relies on the PV polygon, is a very accurate approach, even outperforming the Theil-Sen algorithm (in the case of photogrammetry DSM). The Theil-Sen method with LiDAR data is the most accurate, as shown in table 4.

Method	ME [°]	MAE [°]	RMSE [°]	Runtime [sec]
Hough[20]	2.10	22.70	40.26	0.04
Hough with DSM	-0.73	23.78	43.66	2.50
Theil-Sen (Photogrammetry)	-6.65	15.54	35.64	0.09
Theil-Sen (LiDAR)	-0.08	3.10	4.38	0.09
Bounding-box	-1.39	12.90	32.76	0.02

Table 4: Performance metrics for the estimation of the azimuth angle. The two lines for the Theil-Sen method report the accuracy results whether photogrammetry DSM or LiDAR DSM are passed as inputs.

**Installed capacity estimation** Estimating the installed capacity requires the tilt angle. Indeed, we use the real surface rather than the projected surface as input to estimate the installed capacity. Rausch et al. [30] reported a 9 percentage point increase in the median absolute percentage error (MedAPE) for estimating the installed capacity when considering the tilt angle. We compared variants of the random forest estimator, with  $\theta$  coming from different methods, to see how potential errors propagated. As it can be seen from table 5, all random forests perform equally. We can also see that these methods are only slightly better than the clustered linear regression, which improves over [22]. [22] reported mean squared errors ranging from 1.64 to 1.69, corresponding to an RMSE of 1.28-1.30. We slightly improved over their baseline with our clustered linear regression approach.

### 5.3 Choice of the methods

Based on the results of table 1 and the comprehensive comparison reported in section 5, we restricted ourselves to the following methods for each characteristic:

- Surface: direct computation.



Method	$\theta$	ME [kWp]	MAE [kWp]	RMSE [kWp]	MAPE [%]	Runtime [sec]
Random forest (with $S_{est}$ )	RF	0.022	0.328	0.750	9.37	1.1e-1
Random forest (with $S_{proj}$ and $\theta$ )	TS	0.061	0.393	0.848	11.48	4.4e-4
Random forest (with $S_{proj}$ and $\theta$ )	RF	0.079	0.379	0.921	10.66	4.3e-2
Clustered linear regression	RF	-0.015	0.376	0.687	11.57	7.2e-7

Table 5: Performance metrics for the estimation of the installed capacity. Column  $\theta$  indicates the method used to derive the tilt necessary to compute the estimated surface  $S_{est}$ , taken as input to estimate the installed capacity.

- Tilt angle: constant imputation, a look-up table, and Theil-Sen estimation. The constant imputation works in all cases, the LUT turned out to be very competitive compared to the random forests, and Theil-Sen is competitive when surface models are available.
- Azimuth angle: we keep the bounding-box method and the Theil-Sen estimation to be used when surface models are available.
- Installed capacity: we keep the constant imputation and the linear models, as it turned out to be very competitive with the random forests.

## 6 Conclusion

This work describes a method for the comprehensive and automated extraction of PV systems characteristics. This method takes as input PV polygons and returns the set of associated characteristics using the current best methods available and taking as input as few auxiliary data sources as possible. The user can pick his preferred method depending on the auxiliary data available in his case. Overall, we designed our approach to cover most use cases encountered when characterizing PV installation. In the context of growing PV and the increasing availability of large-scale inventories containing PV polygons, such methods will be beneficial for integrating these data into official registries or power forecasting models. Moreover, by enabling better characterization of the rooftop PV fleet, this package can contribute to producing statistics on PV installations characteristics.

## References

- [1] Y. Trémenbert, G. Kasmi, L. Dubus, Y.-M. Saint-Drenan, P. Blanc, PyPVRoof: a Python package for extracting the metadata of rooftop PV in-

- stallations from their polygon (Jan. 2023). doi:10.5281/zenodo.7586879. URL <https://doi.org/10.5281/zenodo.7586879>
- [2] RTE France, Bilan électrique 2021, <https://bilan-electrique-2021.rte-france.com/>, [Online; accessed 15-April-2022] (2021).
  - [3] IEA, Solar PV, <https://www.iea.org/reports/solar-pv>, [Online; accessed 15-April-2022] (2021).
  - [4] N. M. Haegel, R. Margolis, T. Buonassisi, D. Feldman, A. Froitzheim, R. Garabedian, M. Green, S. Glunz, H.-M. Henning, B. Holder, et al., Terawatt-scale photovoltaics: Trajectories and challenges, *Science* 356 (6334) (2017) 141–143.
  - [5] G. Kasmi, Y.-M. Saint-Drenan, D. Trebosc, R. Jolivet, J. Leloux, B. Sarr, L. Dubus, A crowdsourced dataset of aerial images with annotated solar photovoltaic arrays and installation metadata, *Scientific Data* 10 (1) (2023) 59.
  - [6] G. Kasmi, L. Dubus, P. Blanc, Y.-M. Saint-Drenan, Towards unsupervised assessment with open-source data of the accuracy of deep learning-based distributed pv mapping, in: Workshop on Machine Learning for Earth Observation (MACLEAN), in Conjunction with the ECML/PKDD 2022, 2022.
  - [7] H. Kazmi, Z. Tao, How good are TSO load and renewable generation forecasts: Learning curves, challenges, and the road ahead, *Applied Energy* 323 (2022) 119565.
  - [8] Y.-M. Saint-Drenan, G. H. Good, M. Braun, T. Freisinger, Analysis of the uncertainty in the estimates of regional PV power generation evaluated with the upscaling method, *Solar Energy* 135 (2016) 536–550.
  - [9] Y.-M. Saint-Drenan, S. Vogt, S. Killinger, J. M. Bright, R. Fritz, R. Potthast, Bayesian parameterisation of a regional photovoltaic model—Application to forecasting, *Solar Energy* 188 (2019) 760–774.
  - [10] M. Huber, D. Dimkova, T. Hamacher, Integration of wind and solar power in Europe: Assessment of flexibility requirements, *Energy* 69 (2014) 236–246.
  - [11] S. Dunnett, A. Sorichetta, G. Taylor, F. Eigenbrod, Harmonised global datasets of wind and solar farm locations and power, *Scientific data* 7 (1) (2020) 130.
  - [12] D. Stowell, J. Kelly, D. Tanner, J. Taylor, E. Jones, J. Geddes, E. Chalstrey, A harmonised, high-coverage, open dataset of solar photovoltaic installations in the uk, *Scientific Data* 7 (1) (2020) 1–15.

- [13] J. M. Malof, B. Li, B. Huang, K. Bradbury, A. Stretslov, Mapping solar array location, size, and capacity using deep learning and overhead imagery, arXiv preprint arXiv:1902.10895 (2019).
- [14] J. Yu, Z. Wang, A. Majumdar, R. Rajagopal, DeepSolar: A machine learning framework to efficiently construct a solar deployment database in the United States, *Joule* 2 (12) (2018) 2605–2617.
- [15] K. Mayer, Z. Wang, M.-L. Arlt, D. Neumann, R. Rajagopal, DeepSolar for Germany: A deep learning framework for PV system mapping from aerial imagery, in: 2020 International Conference on Smart Energy Systems and Technologies (SEST), IEEE, 2020, pp. 1–6.
- [16] W. Hu, K. Bradbury, J. M. Malof, B. Li, B. Huang, A. Streltsov, K. S. Fujita, B. Hoen, What you get is not always what you see: pitfalls in solar array assessment using overhead imagery, arXiv preprint arXiv:1902.10895 (2019).
- [17] K. Bradbury, R. Saboo, T. L. Johnson, J. M. Malof, A. Devarajan, W. Zhang, L. M. Collins, R. G. Newell, Distributed solar photovoltaic array location and extent dataset for remote sensing object identification, *Scientific data* 3 (1) (2016) 1–9.
- [18] M. Khomiakov, J. H. Radzikowski, C. A. Schmidt, M. B. Sørensen, M. Andersen, M. R. Andersen, J. Frellsen, Solardk: A high-resolution urban solar panel image classification and localization dataset, arXiv preprint arXiv:2212.01260 (2022).
- [19] Y.-M. Saint-Drenan, G. H. Good, M. Braun, A probabilistic approach to the estimation of regional photovoltaic power production, *Solar Energy* 147 (2017) 257–276.
- [20] A. S. Edun, K. Perry, J. B. Harley, C. Deline, Unsupervised azimuth estimation of solar arrays in low-resolution satellite imagery through semantic segmentation and hough transform, *Applied Energy* 298 (2021) 117273.
- [21] K. Mayer, B. Rausch, M.-L. Arlt, G. Gust, Z. Wang, D. Neumann, R. Rajagopal, 3D-PV-Locator: Large-scale detection of rooftop-mounted photovoltaic systems in 3D, *Applied Energy* 310 (2022) 118469.
- [22] B. So, C. Nezin, V. Kaimal, S. Keene, L. Collins, K. Bradbury, J. M. Malof, Estimating the electricity generation capacity of solar photovoltaic arrays using only color aerial imagery, in: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2017, pp. 1603–1606.
- [23] K. Perry, C. Campos, Panel segmentation: A python package for automated solar array metadata extraction using satellite imagery, *IEEE Journal of Photovoltaics* (2023).

- [24] T. De Jong, S. Bromuri, X. Chang, M. Debusschere, N. Rosenski, C. Scharfner, K. Strauch, M. Boehmer, L. Curier, Monitoring Spatial Sustainable Development: semi-automated analysis of Satellite and Aerial Images for Energy Transition and Sustainability Indicators, arXiv preprint arXiv:2009.05738 (2020).
- [25] L. Kruitwagen, K. Story, J. Friedrich, L. Byers, S. Skillman, C. Hepburn, A global inventory of photovoltaic solar energy generating units, *Nature* 598 (7882) (2021) 604–610.
- [26] X. Hou, B. Wang, W. Hu, L. Yin, H. Wu, Solarnet: a deep learning framework to map solar power plants in china from satellite imagery, arXiv preprint arXiv:1912.03685 (2019).
- [27] V. Plakman, J. Rosier, J. van Vliet, Solar park detection from publicly available satellite imagery, *GIScience & Remote Sensing* 59 (1) (2022) 461–480.
- [28] Open Infrastructure map, [Online; accessed 27-Feb-2023].  
URL <https://openinframap.org>
- [29] Q. Chen, X. Li, Z. Zhang, C. Zhou, Z. Guo, Z. Liu, H. Zhang, Remote sensing of photovoltaic scenarios: Techniques, applications and future directions, *Applied Energy* 333 (2023) 120579.
- [30] B. Rausch, K. Mayer, M.-L. Arlt, G. Gust, P. Staudt, C. Weinhardt, D. Neumann, R. Rajagopal, An Enriched Automated PV Registry: Combining Image Recognition and 3D Building Data, arXiv preprint arXiv:2012.03690 (2020).
- [31] A. Memari, V. C. Dam, L. Nolle, Deep learning for detecting tilt angle and orientation of photovoltaic panels on satellite imagery, in: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Springer, 2022, pp. 255–266.
- [32] S. Killinger, D. Lingfors, Y.-M. Saint-Drenan, P. Moraitis, W. van Sark, J. Taylor, N. A. Engerer, J. M. Bright, On the search for representative characteristics of PV systems: Data collection and analysis of PV system azimuth, tilt, capacity, yield and shading, *Solar Energy* 173 (2018) 1087–1106.
- [33] T. N. de Vries, J. Bronkhorst, M. Vermeer, J. C. Donker, S. A. Briels, H. Ziar, M. Zeman, O. Isabella, A quick-scan method to assess photovoltaic rooftop potential based on aerial imagery and lidar, *Solar Energy* 209 (2020) 96–107.
- [34] J. Martín-Jiménez, S. Del Pozo, M. Sánchez-Aparicio, S. Lagüela, Multi-scale roof characterization from lidar data and aerial orthoimagery: Automatic computation of building photovoltaic capacity, *Automation in Construction* 109 (2020) 102965.

- [35] P. Parhar, R. Sawasaki, A. Todeschini, C. Reed, H. Vahabi, N. Nusaputra, F. Vergara, HyperionSolarNet: Solar Panel Detection from Aerial Images, in: NeurIPS 2021 Workshop on Tackling Climate Change with Machine Learning, 2021.  
URL <https://www.climatechange.ai/papers/neurips2021/41>
- [36] Area Python Package, [Online; accessed 27-Feb-2023].  
URL <https://pypi.org/project/area/>
- [37] Joint Research Center, Photovoltaic Geographical Information System, [Online; accessed 06-Mar-2023].  
URL [https://re.jrc.ec.europa.eu/pvg\\_tools/fr/](https://re.jrc.ec.europa.eu/pvg_tools/fr/)
- [38] Y.-M. Saint-Drenan, L. Wald, T. Ranchin, L. Dubus, A. Troccoli, An approach for the estimation of the aggregated photovoltaic power generated in several european countries from meteorological data, *Advances in Science and Research* 15 (2018) 51–62.
- [39] National Renewable Energy Laboratory, Best Research-Cell Efficiency Chart, [Online; accessed 16-Mar-2023].  
URL <https://www.nrel.gov/pv/cell-efficiency.html>
- [40] H. Theil, A rank-invariant method of linear and polynomial regression analysis, *Henri Theil’s contributions to economics and econometrics: Econometric theory and methodology* (1992) 345–381.
- [41] P. K. Sen, Estimates of the regression coefficient based on kendall’s tau, *Journal of the American statistical association* 63 (324) (1968) 1379–1389.
- [42] J. Canny, A computational approach to edge detection, *IEEE Transactions on pattern analysis and machine intelligence* (6) (1986) 679–698.
- [43] P. V. Hough, Machine analysis of bubble chamber pictures, in: *International Conference on High Energy Accelerators and Instrumentation*, CERN, 1959, 1959, pp. 554–556.

## A Benchmarked characteristics extraction methods

### A.1 Random forests

A random forest is an ensemble learning method that can be used for classification or regression problems. The output of the random forest is the average of the prediction of each regression tree. Random forests perform better than individual trees and are less prone to overfitting. All random forest estimators are trained on the BDAPPV dataset [5], following a standard train/test split procedure.

**Tilt estimation** For the tilt estimation, the input variables are the localization of the PV panel polygon and its projected surface. We also tested the variant with additional input variables such as the length and width of the PV polygon, with no effect on the final accuracy. The dependent variable of the random forest, in this case, is the tilt angle of the installation.

**Installed capacity estimation** For the installed capacity, the dependent variables are the localization of the PV panel and the real surface, which corresponds to the projected surface corrected by the cosine of the tilt angle as defined in equation (1). We consider two variants for this estimator: one with the tilt and projected surface passed separately as input and another one where we directly input the real surface as input. As further discussed in table 5, the model with the real surface yields the best accuracy results.

## A.2 Hough algorithm

This method is based on [20]. We first apply a Canny Edge Detector [42] to the installation mask and dilate the output borders with a 5x5 pixels kernel. Such dilatation aims to simplify the line detection while applying the Hough Transform [43]. This transform detects straight lines in binary edge-detected images by counting the number of aligned points with each point of the image. A line is accounted for when this number is above some defined threshold. We then calculate the angle and the length of each individual line, but contrary to [20], we do not select the angle with the maximum frequency because such a choice could lead to a 90 degrees-error for installations that are taller than they are wide. Indeed, we prefer to store the cumulated length per 1-degree bin and select two angles following two rules. The first angle is associated with the maximum cumulated length. Then, the second angle is associated with the second cumulated length amongst angles that are at least 40 degrees away from the first one. This way, we avoid selecting the same main orientation twice with a 1-2-degree difference.

We also improve upon [20] by enhancing the method using the DSMs. DSMs can then be used to remove the ambiguity between the two main orientations given by the statistical analysis of cumulated lengths. We perform a quick test over the altitudes overlapping with the mask area to guess the approximate direction and value of the slope and select the closest angle given by the Hough method. Such a test, which can be a simple regression or the mean altitude difference along two opposite borders, allows estimating the tilt angle  $\theta$  at the same time. Figure 9 summarizes the process.

The leftmost image corresponds to the mask’s dilated edges with the corresponding altitude values. The Hough transform is applied to get the center image where every red segment corresponds to a detected line. Finally, the statistical analysis of lengths and angles is performed on the segments to get the two main orientations highlighted in the rightmost picture (the one corresponding to the highest drop in altitudes is in orange, and the remaining one is in yellow).

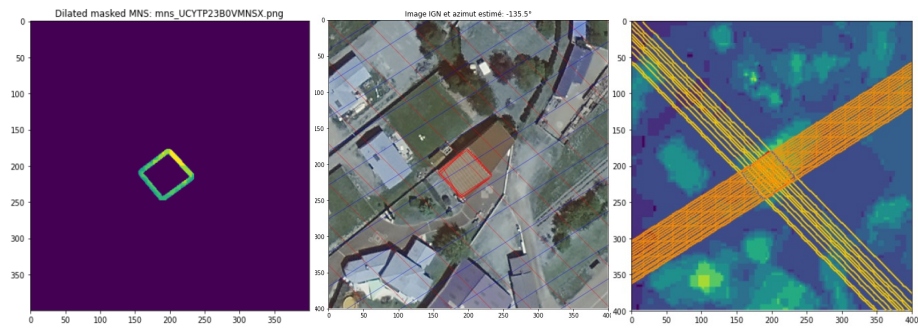


Figure 9: Estimation of the tilt angle using the Hough transform method