

# Panel Segmentation: A Python Package for Automated Solar Array Metadata Extraction Using Satellite Imagery

Kirsten Perry  and Christopher Campos

**Abstract**—The National Renewable Energy Laboratory (NREL) Python panel-segmentation package is a toolkit that automates the process of extracting accurate and valuable metadata related to solar array installations, using publicly available Google Maps satellite imagery. Previously published work includes automated azimuth estimation for individual solar installations in satellite images [1]. Our continued research focuses on automated detection and classification of solar installation mounting configuration (tracking or fixed-tilt; rooftop, ground, or carport). Specifically, a faster-region-based convolutional neural network Resnet-50 feature pyramid network model was trained and validated on 862 manually labeled satellite images. This model was used to perform object detection on satellite imagery, locating and classifying individual solar installations' mounting configuration and type. Model results showed a mean average precision score of 77.79%, with the model strongest at detecting fixed-tilt ground mount and fixed-tilt carport installations. The object detection model and its outputs have been incorporated into the panel-segmentation package's automated metadata extraction pipeline, which returns the mounting configuration and azimuth for individual solar arrays in satellite imagery [2]. The complete image dataset with labels has been released on the U.S. Department of Energy (DOE) DuraMAT DataHub, to encourage further research in this area [3].

**Index Terms**—Azimuth, metadata, mount, satellite imagery, solar.

## I. INTRODUCTION

THE United States solar industry has expanded rapidly over the past several years, with 23.6 GW of solar installed in the United States in 2021 alone, a 19% increase from 2020 [4]. This rise in investment has consequently led to a rise in solar industry acquisitions. According to a report released by Mercom Capital Group, over 34 corporate solar acquisitions were recorded in Q2 of 2021, an increase from 13 acquisitions in Q2 of 2020 [5]. During acquisitions, keeping accurate timely solar metadata, which includes installation tilt, azimuth, and mounting configuration,

is particularly an issue, as data for solar installations may be lost during transference between owners. Furthermore, solar fleet owners generally record site information manually, which can lead to transcription errors. Operators may rely on costly site inspections to keep their data accurate and up to date. Having this accurate metadata is paramount for accurately predicting solar yield and degradation results.

The National Renewable Energy Laboratory (NREL) panel-segmentation package was created to rectify missing or incorrect solar site information, by automating metadata extraction using preexisting, readily available Google Maps imagery [1]. Current package functionality includes automated detection and pixel-by-pixel segmentation of solar installations in satellite imagery via a region-based convolutional neural network (R-CNN) architecture. Additional package functionality includes calculating individual solar array azimuth via an unsupervised computer vision (CV) pipeline, which includes connected components clustering, Canny edge detection, and a Hough line transform. This work is further described in [1]. Our previous efforts build upon a body of published research, which focuses on solar array detection in satellite imagery [6], [7], [8], as well as array characterization following detection [9], [10], [11].

In spite of this growing body of research, no known attempts have been made to characterize the mounting configuration of solar arrays via satellite imagery analysis. Correct classification of mounting configuration is important, as previous research has shown that degradation rates may be affected by mounting configuration type [12]. Specifically, Jordan et al. [12] demonstrated that for a location with multiple installations of the same module type but varying mounting configurations, installations on large metal roof sections have higher degradation rates than installations with carport and rack mounting configurations. This occurs because metal roof types are correlated with higher operating temperatures, which can lead to faster degradation rates. Furthermore, knowing if an installation is fixed-tilt or single-axis tracking is paramount for modeling system performance accurately, as single-axis tracking systems have a higher expected energy yield, higher capacity factor, and different daily performance profile than fixed-tilt systems [13].

This research focuses specifically on the detection and classification of mounting type in satellite imagery via object detection, as well as integration of the resulting model into the metadata extraction pipeline. Not only do we present our object detection model results, we also outline the newly updated

Manuscript received 30 May 2022; accepted 12 December 2022. Date of publication 6 January 2023; date of current version 20 February 2023. This work was supported in part by the Alliance for Sustainable Energy, LLC, the manager and operator of the National Renewable Energy Laboratory for the U.S. Department of Energy (DOE) under Contract DE-AC36-08GO28308, and in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) under Agreement 38258. (Corresponding author: Kirsten Perry.)

The authors are with the NREL, Golden, CO 80401 USA (e-mail: kirsten.perry@nrel.gov; chris.acampos@yahoo.com).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JPHOTOV.2022.3230565>.

Digital Object Identifier 10.1109/JPHOTOV.2022.3230565

metadata extraction pipeline, which includes mounting configuration detection. These updates are publicly available via the NREL panel-segmentation package [2].

## II. METHODS

### A. Datasets

Satellite images at varying latitude–longitude coordinates were collected using the Google Maps application programming interface (API). The NREL photovoltaic (PV) fleet performance data initiative database, which contains data for over 1200 different solar sites across the United States, was used to build the dataset [14]. This database, which supports the DOE-funded PV fleet initiative, houses site metadata and time series data, which is then aggregated to perform fleet-scale degradation analysis. Available metadata includes site latitude–longitude coordinates, azimuth, tilt, and mounting configuration information (rooftop, carport, or ground mount; tracking or fixed tilt). Although most of this data are NDA-protected, a subset of these systems are publicly available via the DOE-funded open energy data initiative (OEDI).

Satellite images of site locations were generated using the same parameters outlined in [1] (zoom level of 18, 0.596-m/pixel resolution, and  $640 \times 640$  pixel dimensions). A total of 862 individual images from differing locations were collected. Of the 862 total images, 90 did not contain any solar installations. It is important to note that many of the satellite images with arrays contained multiple installations with differing mounting configurations.

The satellite image dataset was then labeled using the Python LabelImg object detection package [15]. Using LabelImg, boxes were drawn around solar installations in satellite imagery, and assigned specific mounting classifications. Solar installations were classified into one of four categories as follows.

- 1) Rooftop fixed-tilt mounts.
- 2) Carport fixed-tilt mounts.
- 3) Ground single-axis tracker mounts.
- 4) Ground fixed-tilt mounts.

Example labeled images for each of these categories are shown in Figs. 1–3.

After labeling, images were split into training, test, and validation sets, with 540 images in the training set, 216 images in the validation set, and 106 images in the test set. To balance the training dataset, which had an unbalanced class distribution, data augmentation was explored on images containing minority classes. The AutoAugment function in PyTorch was used for data augmentation, using the ImageNet policy [16]. The AutoAugment process is further described in [17]. However, no significant model improvements occurred when augmentation was performed on the dataset, so for simplicity only the original satellite imagery was used. Final training class representation was 16%, 22%, 44%, and 18% for the rooftop-fixed, ground-fixed, carport-fixed, and ground-single-axis tracker classes, respectively.



Fig. 1. Labeled satellite image, showing single axis tracker installations.

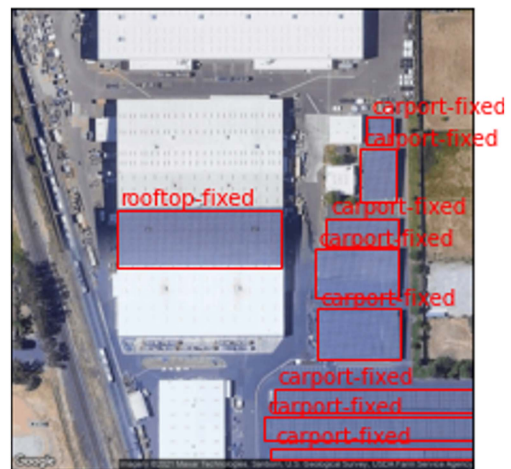


Fig. 2. Labeled satellite image, showing fixed tilt rooftop and carport installations.



Fig. 3. Labeled satellite image, showing a fixed-tilt ground mount installation.

## B. Object Detection Model

For the object detection task, a faster-RCNN Resnet-50 feature pyramid network (FPN) was used as the model backbone. This model, introduced in [18], was selected for a few key reasons. First, it is readily available via the Pytorch package, allowing for easy adoption and training via transfer learning [16]. Second, Resnet-50 FPN has been previously demonstrated to perform well at semantic segmentation tasks on high-resolution satellite imagery when detecting solar installations [19]. Although the goal in [19] is different from ours, the dataset used is similar (it is also satellite imagery with labeled solar arrays). Because the Resnet-50 FPN is a FPN, it is robust at identifying PV arrays at various scales [19]. In addition, the Resnet architecture ensures that the training accuracy is not diminished as convolutional layer size and subsequent model complexity is increased [20].

Transfer learning was used to train the model, with the original model trained on the Microsoft Common Objects in Context (MS-COCO) dataset [21]. This dataset contains 80 categories of approximately 160 K images, collected via the web [21]. The transfer learning model was trained and validated on the labeled satellite imagery datasets described in the datasets section above.

Images were preprocessed before model training. Specifically, each image was converted to a Pytorch tensor object, and its associated pixels were normalized.

Model hyperparameters were modified to optimize model performance on the test set. Final model hyperparameters were as follows.

- 1) Learning rate of 0.005
- 2) 10 epochs
- 3) Decay factor of the learning rate, gamma, of 0.1
- 4) Batch size of 2
- 5) Momentum of 0.9
- 6) Weight decay of 0.0005

The Python Detecto package was used as a wrapper for training the object detection model [22]. The Detecto package uses a Pytorch backbone and contains basic functionality for training object detection models using transfer learning, as well as generating model predictions.

Detecto functionality was used to generate predictions on the test dataset, with modifications. Specifically, the acceptance threshold for each class was set to 0.65, so any predictions with a probability score below 0.65 were discarded. The value 0.65 was empirically derived.

To aid in image visualization, overlapping object detection boxes were combined using nonmaximum suppression (NMS). NMS is a technique for selecting the optimal box in a series of overlapping boxes. Using NMS, lower scoring object detection boxes are iteratively removed when their intersection-over-union (IoU) score with other, higher scoring boxes is above a selected threshold. IoU is the area of overlap divided by the area of union between two boxes. In this case, the IoU is calculated between two prediction boxes. For NMS, an IoU threshold of 0.4 was empirically derived. A preexisting NMS function in the Pytorch TorchVision package was used in the data pipeline, after generating the associated predictions [16]. An example satellite

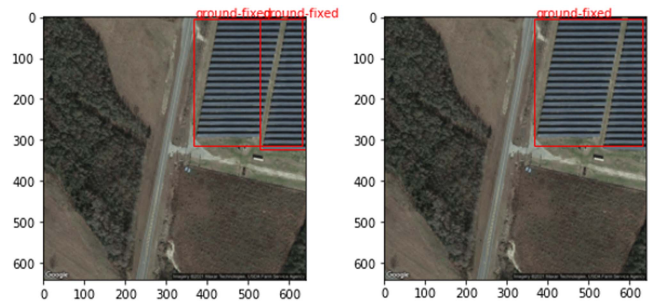


Fig. 4. Image before applying overlapping NMS, and image after applying NMS.

image without NMS and with NMS is shown in Fig. 4. As you can see, one of the boxes overlaps with the other box in the non-NMS image, and is removed when NMS is applied.

## III. RESULTS

### A. Mounting Configuration Model Performance

Performance on object detection models can be measured a variety of ways, most notably mean average precision (mAP) and average precision (AP) scores.

A few key metrics are used to generate AP scores and overall mAP score. First, the IoU between the predicted bounding box and the ground-truth bounding box must be calculated.

Precision and recall are calculated based on the IoU value, at a specific IoU threshold. For the results presented in this article, a standard IoU threshold of 0.5 is used. Precision is a measurement of the positive predictive value, or the number of predictions made by the model that are correct. Recall is a measure of model sensitivity, or the ability of the model to detect true positives. Average precision is defined as the area under the precision–recall curve. In this research, a 11-point interpolation for AP is used, where the shape of the precision–recall curve is the average of the maximum precision values at 11 equally spaced recall levels [23]. 11-point interpolated AP is defined in the following equation [23]:

$$AP_{11} = \frac{1}{11} * \sum_{R=(0..1, \dots, 1)} P(R) \quad (1)$$

where P is the maximum precision value at each value of R.

After calculating an AP score for all model classes, overall model performance can be evaluated via the mAP score. The mAP is the average of the AP across all classes, and is described via the following equation [23]:

$$mAP = \frac{1}{N} * \sum_{i=1}^N AP_i \quad (2)$$

where  $N$  is the total number of classes, and AP is the average precision of each class.

The calculated mAP scoring method established in [23] was used to benchmark overall model performance, and the AP score method was used to benchmark individual class performance.

TABLE I  
AP SCORES BY MODEL CLASS

Class	AP <sub>50</sub> (%)
Fixed-tilt ground mount	90.62
Fixed-tilt carport mount	93.54
Fixed-tilt rooftop mount	51.99
Single-axis tracking ground mount	75.01
Combined score (mAP)	77.79

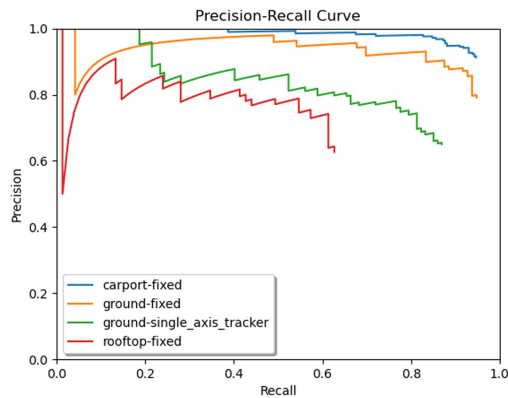


Fig. 5. Precision-recall curve for each of class in the object-detection model, run on the test data.

The associated Python implementation of [23] was used. These results are displayed in Table I, with each class’s precision–recall curve shown in Fig. 5. These results indicate the model’s strengths and weaknesses on a class-by-class basis. The model is best at detecting carport and fixed-tilt ground mount installations, and weakest at detecting rooftop installations. Poorer model performance when detecting rooftop installations is likely because of the significant variation in what these installations look like, particularly their shape.

The overall model mAP score is 77.79%. As a point of comparison, previously published literature on the MS-COCO dataset has several published models’ mAP scores ranging between 50% and 60%, with the current state-of-the-art measuring at 63.3% [24]. Because solar satellite imagery research has not previously focused on mounting configuration object detection, but rather on general solar array detection, a direct comparison is not possible. Still, the overall mAP score can be compared with previous literature that focuses on general solar array object detection, as a point of reference. He and Zhang [19] achieves a maximum mAP score of 95.66% and 80.31% using Resnet-50 FPN and Segnet architectures, respectively, for general solar array object detection. Although these are higher mAP scores than our model, it is important to note that [19] only has one object detection class, whereas our model has four. Consequently, our model has a more complex object detection task, where it needs to not only identify the solar array, but also classify what type of mounting configuration the array has. This added complexity makes our model subject to higher error.

### B. Model Drawbacks

In spite of high overall model performance, results on the test set showed repeated instances of the object detection model

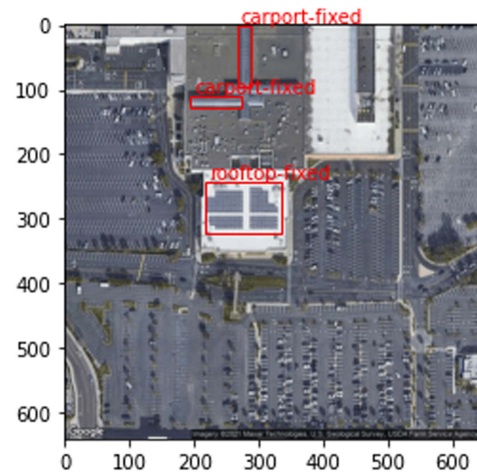


Fig. 6. Misclassified satellite image from the test set, where skylights on a mall are incorrectly classified as carport installations.

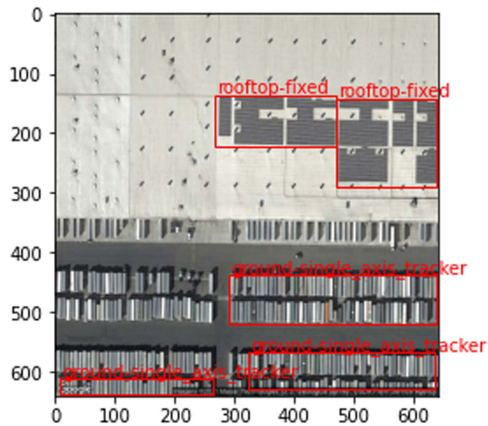


Fig. 7. Misclassified satellite image from the test set, where shipping containers are incorrectly classified as single-axis tracking installations.

misclassifying shipping containers and rooftop skylights as solar array installations. Both of these features highly resemble solar array installations, so these results make sense. Some examples of these misclassifications are shown in Figs. 6 and 7.

### C. Model Pipeline Integration and Dataset Release

Fig. 8 shows the pipeline for extracting solar metadata, available in the panel-segmentation package. The only required inputs are for running the pipeline are site latitude–longitude coordinates and a Google Maps API key. Using these inputs, a satellite image at the respective location is taken and saved locally. This satellite image is then used as an input for both the mounting configuration object detection model and the semantic segmentation model, which is described in [1]. The generated object detection model boxes are then used to cluster individual arrays in the semantic segmentation mask output. Each individual array is then run through the azimuth estimation routine, described in [1]. Final pipeline output is a dictionary, containing the mounting configuration and azimuth of each solar array detected in the satellite image.

This newly updated pipeline requires both the object detection model and the semantic segmentation to identify a solar array

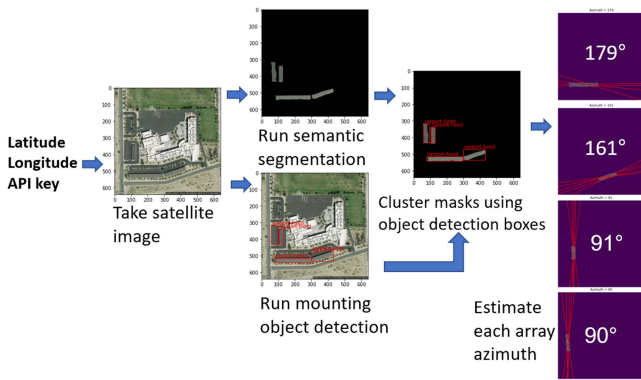


Fig. 8. Schematic of the updated panel-segmentation pipeline, where a location is run through a series of deep learning and computer vision models to estimate azimuth and mounting configuration.

in a specific region of a satellite image, ensuring fewer false positives in the metadata extraction pipeline.

To provide additional data transparency, as well as to promote further research in this space, the entire dataset used to train the model is available via the DOE DuraMAT Datahub [3]. All data have been anonymized, with no location metadata provided (including latitude–longitude coordinates). Each satellite image with its associated objection detection labels is provided.

#### IV. CONCLUSION

To supplement current panel-segmentation package functionality, we plan to add additional functions for calculating array size and expected energy output, ground-coverage ratio (GCR), and terrain classification. We are also investigating tilt estimation via 3-D data sources, such as Google OpenStreetMap and light detection and ranging technologies, using methods adapted from [9].

This research has been leveraged extensively by the NREL PV Fleets project, as many of the data sources provided by our industry fleet partners have metadata gaps. Quantifying the financial impact of these gaps has been difficult, as there is no known financial cost for having incorrect solar metadata. In addition to our current efforts to fill in these data gaps, our group plans to investigate the financial and operational costs of incorrect or incomplete solar fleet metadata, with the intent of drawing more attention to issue.

#### REFERENCES

[1] A. S. Edun, K. Perry, J. B. Harley, and C. Deline, “Unsupervised azimuth estimation of solar arrays in low-resolution satellite imagery through semantic segmentation and hough transform,” *Appl. Energy*, vol. 298, 2021, Art. no. 117273.

[2] A. S. Edun, K. Perry, and C. Deline, US DOE Office of Energy Efficiency and Renewable Energy, “Panel-segmentation,” 2020. [Online]. Available: <https://www.osti.gov/biblio/1726007>

[3] K. Perry, “Solar mounting configuration satellite imagery data,” 2022. [Online]. Available: <http://dx.doi.org/10.21948/1867572>

[4] “US solar market insight,” Solar Energy Industries Assoc., Washington, DC, USA, Tech. Rep., 03 2022.

[5] “Solar funding and M&A: 2021 first half report,” Mercom Capital Group, Austin, TX, USA, Tech. Rep., 2021.

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

[7] “Automatic detection of solar photovoltaic arrays in high resolution aerial imagery,” *Appl. Energy*, vol. 183, pp. 229–240, 2016.

[8] J. Camilo, R. Wang, L. M. Collins, K. Bradbury, and J. M. Malof, “Application of a semantic segmentation convolutional neural network for accurate automatic detection and mapping of solar photovoltaic arrays in aerial imagery,” 2018, *arXiv:abs/1801.04018*.

[9] K. Mayer et al., “3D-PV-locator: Large-scale detection of rooftop-mounted photovoltaic systems in 3 D,” *Appl. Energy*, vol. 310, 2022, Art. no. 118469.

[10] Q. Li, Y. Feng, Y. Leng, and D. Chen, “Solarfinder: Automatic detection of solar photovoltaic arrays,” in *Proc. 19th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2020, pp. 193–204.

[11] K. Mainzer, S. Killinger, R. McKenna, and W. Fichtner, “Assessment of rooftop photovoltaic potentials at the urban level using publicly available geodata and image recognition techniques,” *Sol. Energy*, vol. 155, pp. 561–573, 2017.

[12] D. Jordan, C. A. Deline, M. G. Deceglie, T. J. Silverman, and W. Luo, “PV degradation - mounting & temperature,” in *Proc. 46th IEEE Photovoltaic Specialists Conf.*, 2019, pp. 0673–0679. [Online]. Available: <https://www.osti.gov/biblio/1603896>

[13] A. Safieh, A. Elnoosh, E.-C. A. K. Kaiss, J. J. John, and A. Alnuaimi, “Field comparison study of fixed-tilted and single-axis tracking PV structures in the desert environment of Dubai, UAE,” in *Proc. 47th IEEE Photovoltaic Specialists Conf.*, 2020, pp. 2136–2139.

[14] C. Deline et al., “PV fleet performance data initiative,” Nat. Renewable Energy Lab., Golden, CO, USA, Methodol. Rep. NREL/TP-5K00-76687, Mar. 2020 [Online]. Available: <https://www.osti.gov/biblio/1659786>

[15] T. Lin, “LabelImg,” 2015. [Online]. Available: <https://github.com/heartexlabs/labelImg>

[16] A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach et al., Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>

[17] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, “Autoaugment: Learning augmentation strategies from data,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 113–123.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[19] K. He and L. Zhang, “Automatic detection and mapping of solar photovoltaic arrays with deep convolutional neural networks in high resolution satellite images,” in *Proc. IEEE 4th Conf. Energy Internet Energy Syst. Integration*, 2020, pp. 3068–3073.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015, *arXiv:abs/1512.03385*.

[21] T. Y. Lin et al., “Microsoft COCO: Common objects in context,” 2014, *arXiv:abs/1405.0312*.

[22] A. Bi, “Detecto,” 2021. [Online]. Available: <https://github.com/alankbi/detecto>

[23] R. Padilla, S. L. Netto, and E. A. B. da Silva, “A survey on performance metrics for object-detection algorithms,” in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, 2020, pp. 237–242.

[24] H. Zhang et al., “DINO: DETR with improved denoising anchor boxes for end-to-end object detection,” 2022, *arXiv:abs/2203.03605*.