

Identifying small decentralized solar systems in aerial images using deep learning

Âzeddine Frimane^{a,*}, Robert Johansson^b, Joakim Munkhammar^a, David Lingfors^a, Johan Lindahl^b

^a Department of Engineering Sciences, Uppsala University, SE-751 21 Uppsala, Sweden

^b Becquerel Sweden AB, SE-741 42 Knivsta, Sweden

ARTICLE INFO

Keywords:

Solar energy systems
Photovoltaics
Solar Thermal
Aerial images
Deep learning
Segmentation

ABSTRACT

Statistics on installed solar energy systems (SES) play a crucial role in the solar energy industry, providing valuable information for a wide range of stakeholders, such as policy makers, authorities, and financial evaluators. For example, grid operators rely on accurate data on photovoltaic penetration levels to ensure the quality and stability of the power supply. In this research, we present an automatic approach helping generate these statistics using deep learning and image processing techniques. Our proposed model is a machine learning approach that utilizes a specific architecture of convolutional neural networks (CNN) called the “U-net” to detect SES from aerial images. We experimented different network settings to enhance the SES identification performance.

In this study, the model was evaluated using two datasets from different locations, one from Sweden and one from Germany. Additionally, the model was trained and tested on a combination of both datasets. The impact of image resolution was also examined. The experimental results show that this architecture performs better than many recent CNN models that have been proposed in the literature for the task of SES identification from aerial images. To make it easy for others to replicate our findings, We have shared all the scripts, software, and dependencies required for running the model in this paper, along with instructions on how to use it in Appendix A.

1. Introduction

The global solar energy market is currently growing rapidly [1], and has generally outpaced expectations over the past decade [2–4]. For example, at the end of 2021, the global installed PV capacity was 945.4 GW, which contributed to an estimated 1 229 TWh in 2021, about 5% of the global electricity demand [1]. The scalable low-technological complexity, and modular characteristics of the PV technology have lead to lower absolute unit costs and investment risks [5], which have made it possible for a wide range of stakeholders to purchase, install and utilize PV worldwide. About 45% of the global PV market is made up of small-scale distributed systems [1].

However, the wide range in system sizes makes it difficult for stakeholders to track current and historical solar energy systems (SES) deployment, especially the large number of small decentralized systems. For example, the administrative capacity to follow the installation of PV systems is lacking in most countries in the world [6], and especially in developing countries (i.e. sovereign states with a less developed industrial base and a low Human Development Index (HDI)

relative to other countries [7]). A potential solution to register SES, and keep track of the market development when registration processes are failing or lacking in a country or region, is to locate stationary systems and estimate their size by the area through remote sensing aerial imagery and machine learning using convolutional neural networks (CNN) [8–20]. The benefit of this approach is that it has the potential to generate databases that include the number of SES, sizes of systems, exact location of systems and combine this information with socioeconomic data [9,19,21]. However, the drawback is that the SES needs to be visible from the sky and have the clear characteristics of regular systems to be registered by this method. This method (so far) have also the downside that smaller SES have lower precision and are more difficult to detect [13–15,22]. For example, The authors of Kruitwagen et al. [15], only focused on non-residential PV systems and as a result, a significant portion of the global market was not taken into account. Looking at for example Australia or Japan, the study of Kruitwagen et al. [15] reported that they identified 3 160 MW and 18 005 MW at the end of 2018, respectively, while the official

* Corresponding author.

E-mail address: Azeddine.frimane@angstrom.uu.se (Â. Frimane).

government numbers at the end of the same year are 11 586 MW for Australia and 56 162 MW for Japan [1]. Hence, a large share of the actually installed PV capacity is overlooked if distributed small scale PV systems are neglected. However, detecting also small PV systems have been addressed lately, in terms of refined methodologies and increased accuracy [9–11,17].

One may wonder why image detection algorithms are necessary when there are existing databases from installers that document the location and capacity of deployed SES. However, aerial image detection algorithms can be valuable in identifying SES installed on rooftops or in remote locations where data may be incomplete [18,20]. These algorithms can provide for example detailed information on the orientation and tilt angle of PV panels [17,23], which can aid in estimating their energy output. In addition, these algorithms can be employed to verify the accuracy of the reported number of installed SES [20]. They can also identify any safety or quality issues with the installations. Furthermore, aerial image detection algorithms can monitor changes in SES installations over time, such as decommissioning of SES installations [24], which can assist in tracking the growth of the solar industry [19] and identifying areas where additional solar installations may be required.

In this study we continue the work of identifying small decentralized SES by aerial imagery analysis using deep learning. We implemented a U-net model, originally introduced in Ronneberger et al. [25] for an approved segmentation and area estimation of SES [10,12,13,15,26]. The U-Net model is a convolutional architecture with encoding and decoding paths. It takes the image as input and produce the segmentation feature maps with respective pixel classes. Thus, a class label is assigned to each pixel with the U-Net model. A key strength of the U-net model compared to several architectures proposed in the literature is that it allows for the simultaneous use of global location and context in images, saving many coding steps and hardware usage. Furthermore, this model shows in practice that it can operate with a smaller number of input data compared to other models and provides better performance [25].

The utilization of the U-net model for SES detection provides a data-driven and automated solution with enhanced complexity, enabling precise detection. Its accurate segmentation and identification of SES from aerial images hold substantial practical value, facilitating efficient assessment of panel performance, maintenance requirements, and energy production estimation. Compared to other approaches, the U-net model demonstrates superiority in handling challenges such as partial shading and irregular shapes, making it a promising technique for advancing renewable energy analysis and solar asset management in SES detection.

Moreover, this paper’s findings can easily be replicated by other researchers, since all necessary software and dependencies have been compiled into a single Docker container. The R-script of the U-net model can be run on top of this container. Interested readers can find more information about replicating this work in Appendix A.

The structure of this paper is as follows. Section 2 provides the description of the used aerial image databases in this study. The methodology is fully detailed in Section 3. Section 4 presents the experimental results of the SES identification. In Section 5 we compare our model with recent findings in the literature. Weaknesses and future perspectives of this work are discussed in Section 6. Closing remarks are presented in Section 7. For replication of this work, Appendix A gives details on how to run and use the proposed model. Supplementary materials to this paper are in Appendix B.

2. Data

Two datasets of aerial images were used to train and test the method described in this paper. The first dataset is from Germany and the second from Sweden. Table 1 summarizes the data splitting process for the model training, validation and testing.

Table 1
Summarizes of the datasets splitting for the U-net model training, validation and testing.

	German data	Swedish data	Total
Training	3221	1241	4462
Validation	403	266	669
Test	403	266	669
Total	4027	1773	5800

2.1. German dataset

The German dataset used in this paper is the same one used for image segmentation in Mayer et al. [17]. Instructions for downloading this dataset can be found at <https://github.com/kdmayer/3D-PV-Locator>.

This dataset was created from randomly drawn instances that had been positively classified in the image classification step used in that study. The native spatial resolution of the images is 0.10 m/pixel. Each image consists of 320 × 320 pixels and is downloaded at an upsampled spatial resolution of 0.05 m/pixel. The dataset contains 4027 images in total and is split in an 8:1:1 ratio for training (3221 images), validation (403 images), and testing (403 images). The same split between training, validation and test is used in this paper.

2.2. Swedish dataset

The orthophotos used in this study were provided by the Swedish Land Survey and are 4-channel (Red, Green, Blue, Infrared), taken at a flight altitude of 3 000 m with a native resolution of 0.16 m/pixel in form of TIF images. However, the infrared part is not used in this study. Each orthophoto cover 2.5 × 2.5 km which corresponds to 15 625 × 15 625 pixels.

The actual images used for segmentation are upsampled from tiles of size 115 × 115 pixels to 299 × 299. The spatial resolution of the Swedish images is 0.062 m/pixel. Complete orthophoto images of two municipalities in Sweden (Falun and Knivsta) where scanned using a slightly modified version of the open access DeepSolar aerial imagery classification algorithm from Yu et al. [9]. The orthophoto images where divided into smaller tiles, which the classification algorithm of Yu et al. [9] could be applied to. The results from the scanning, supported with data over registered grid-connected PV systems from the Distribution System Operators in the two municipalities, where used to manually create a complete database of all the solar systems in the area, including both PV and solar thermal systems [24]. Polygons describing their exact shape were created manually, representing the ground truth. Using this list of all solar systems, tiles that contained some part of a solar system where selected to form the Swedish dataset.

This Swedish dataset contains 1773 images in total and is split in an 70:15:15 ratio for training (1241 images), validation (266 images), and testing (266 images). 1207 images are from the municipality of Falun and 566 images are from the municipality of Knivsta. It should be noted, that the mean, so called, solar panel proportion distribution (introduced by Li et al. [16]) is lower in the Swedish training set compared to the German.

2.3. Data augmentation process

Data augmentation is a process of datasets amplification. It artificially increases the size of a dataset by applying various operations to the existing data, creating new datapoints. In this study, the data augmentation process consists of two processes: first, the images are randomly flipped and then randomly rotated by an angle between –20 and 20. We only chose these two processes because we have a sufficient dataset to learn from, and to avoid high memory usage. From a modeling perspective, the U-net model is a powerful model that can learn from small datasets, and the augmentation process we used merely serves to better detect the geometric invariances of the target

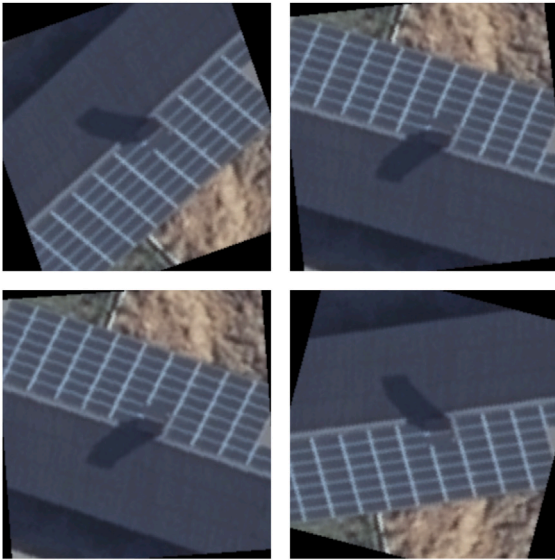


Fig. 1. Augmentation possibilities for a random image drawn from the Swedish dataset.

object. The data augmentation process is uniformly applied across all datasets.

An example of some possible augmentation operations on a random image from the Swedish dataset is shown in Fig. 1.

3. Methodology

To segregate the foreground (SES) from the background (everything else) in our aerial images, the U-Net model performs a fine-grained classification instead of coarse-grained classification of typical convolutional networks for the classification tasks. Accordingly, a class label is assigned to each pixel with the U-Net model.

3.1. U-net architecture

The distinctive feature of the U-net model is the symmetrical U-shaped encoder–decoder cascade architecture, stabilized by horizontal “bridges” across all blocks (or levels). The encoder calls the contracting path (or down-sizing path) where a number of convolutional processes are performed in order to capture the contextual features of the objects. On the other hand, the decoder calls the expansive path (or up-sizing path) where the image is up-sized to its original size allowing the objects localization.

The parameterization of the neural network is always a difficult task to setup. During the implementation of our U-net model, we experimented with different network settings such as varying the number of layers, filters and adjust the learning rate and hyperparameters. Ultimately, we decided on the following architecture: the contracting path consists of 5 convolutional blocks. Each block consists of 2 zero-padding 3×3 convolutions, each followed by a rectified linear unit (ReLU) and a dropout layer. For the expansive path in each block, we start by a 2×2 transposed-convolution followed by two unpadded 3×3 convolutions, each rectified by ReLU process. An important step of the expansive path is the concatenation process, where the input to an upsampling layer is a concatenation of the corresponding downsampling layer’s output. Since information is lost during the downsampling process of the encoder side, this helps preserve features from the encoder side of the network to the decoder side. At the final layer, a 1×1 convolution is used to map each component feature vector to the desired number of classes. An R script of the U-net model is shared with the paper, see Appendix A.

3.2. Optimization

For best performance, and to avoid training instabilities and class imbalance problem, the energy function is calculated by a convex combination of the pixel-wise categorical cross-entropy (CE) and dice loss (DL) measures. By using both DL and CE, the model will be able to proficiently manage class imbalance and generate segmentation results that are more precise. The DL prioritizes optimizing the overlap between the predicted and actual segmentation maps, whereas the CE prioritizes minimizing the variation between the predicted and actual pixel values. Therefore, through the combination of these two functions, the model can successfully distribute the significance of each class and decrease the influence of class imbalance on the ultimate segmentation outcomes.

The CE loss is a scalar value ranges between 0 and infinity, and increases as the predicted probability deviates from the actual label. Hence, a perfect model would have a logarithmic loss of zero. In our case, the number of class labels to predict is two: 0 for the background or 1 for the foreground. Thus the CE score can be computed using the following equation:

$$CE(x) = y(x) \times \log(p(x)) + (1 - y(x)) \times \log(1 - p(x)), \quad (1)$$

where $p(x)$ represents the softmax output at the pixel position x , and $y(x)$ is the actual label.

The DL ranges between 0 and 1, and to combine it with the CE metric in a single equation, it is common to use the complement of the DL, which is represented as $1-DL$, and is also referred to as the dice loss. The DL can be calculated using the following equation:

$$DL(x) = 1 - \frac{2p(x) \times y(x)}{p(x) + y(x)}. \quad (2)$$

The weighted loss (WL) is a convex combination of both previous metrics. It is calculated as:

$$WL(x) = a \times CE(x) + (1 - a) \times DL(x), \quad (3)$$

where a is a real scalar between 0 and 1. It is chosen in this study to be 0.8, giving higher importance to the CE in the optimization step. This scalar is not arbitrarily selected, and it is typically set to assign a higher weight to the CE loss. This decision is based on the fact that the CE loss is more sensitive to the misclassification of background pixels, whereas the DL is more sensitive to the misclassification of foreground pixels. By assigning more weight to the CE loss, the model is encouraged to accurately classify more background pixels, which can mitigate the detrimental effects of class imbalance on the overall performance of the model.

3.3. Benchmark metrics

To evaluate the performance of the model presented in this paper, and compare it to other state-of-the-art methods, four different metrics were employed. Firstly, the mean absolute percentage error (MAPE) was used, which is defined as;

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{S_i - \hat{S}_i}{S_i} \right|, \quad (4)$$

where S_i is the ground truth area of the solar system in the manually labeled mask of the image i , \hat{S}_i is the predicted solar system area with respect to the same image i and N is the number of images.

Secondly, the mean absolute error (MAE) was used, defined as;

$$MAE = \frac{1}{N} \sum_{i=1}^N |S_i - \hat{S}_i|. \quad (5)$$

Thirdly, we used the bias (B), which represents the under- or over-estimation of the segmentation process, which can be written as;

$$B = \frac{\sum_{i=1}^N S_i - \hat{S}_i}{\sum_{i=1}^N S_i}. \quad (6)$$

Lastly, we employed the mean intersection over union (mIoU), which is calculated by taking the ratio of the overlap between the predicted segmentation and the ground truth mask to the total area of both. For two classes, it can be represented as;

$$mIoU = \frac{1}{2} \sum_{k=1}^2 \frac{TP_k}{FN_k + FP_k + TP_k}, \quad (7)$$

where TP (*True positives*) represents the correctly predicted area of intersection between the ground truth mask and the predicted segmentation, FP (*False positives*) represents the area that the model predicted incorrectly outside of the ground truth mask, and FN (*False negatives*) represents the area within the ground truth mask that the model did not correctly predict.

3.4. Software and hardware architecture

The U-net model presented in this paper was implemented using the R programming language, especially the R-Torch package, which support fast tensor computation with GPU acceleration. All the necessary software and dependencies are packaged into a single Docker container, which can be accessed at <https://hub.docker.com/u/frimane>. This allows anyone to run the script consistently and uniformly on his infrastructure, without the need for prior setup of an R environment. Refer to Appendix A for instructions on how to use the model.

The training of the model was conducted on the Alvis cluster, which is part of the Chalmers centre for computational science and engineering (C3SE) and is dedicated to Artificial Intelligence and Machine Learning research. The C3SE is one of six centers that form the national metacenter Swedish National Infrastructure for Computing (SNIC).

4. Results

The U-net model is trained and tested on the Swedish and German datasets with six different batch sizes: 2, 4, 8, 16, 32 and 64. We have also trained the model on the mixture of both datasets for higher generalizability of the model. Furthermore, all previous settings are tested over two different image resolutions of images: 128×128 and 256×256 pixels. The aim here is to evaluate the effect of the resolution on the model's performance, and then make recommendations on how to choose it. As the hardware usage for CNN models is intensive, it would be beneficial if a lower resolution could be used given that the results are sufficiently accurate.

4.1. Model convergence

As part of the validation of the U-net model convergence graphs are provided. But, as the number of plots involved in this work is too large for a single journal paper, we decided to separate out the convergence plots in a supplementary paper published alongside the main document, see Appendix B.

4.2. Model validation

Fig. 2 visually shows the performance of the trained and tested model for all datasets, their mixture and with all different image resolution combinations. For a detailed overview of the model's performance, see Table 2. Fig. 2 clearly shows that the U-net model performs better when applied to the German dataset than to the Swedish. This is especially clear when the DL score is computed. As explained in Section 3.2, the DL score is a measure of the relative overlap between predicted segments of the SES and manually created masks. It predicts both large and small systems with the same values and is therefore characterized by large variance. For example, with the DL score, missing a single pixel in a tiny SES segment in the corner of the image can have the same effect as missing almost an entire large SES segment and thus depends

highly on the used dataset. Since the Swedish dataset contains a large number of images that contain only small parts of PV systems [24], and hence has a lower mean *SES proportion distribution* [16], which results in a class imbalance effect, the logic results is that DL values for the foreground class are higher for the Swedish data than the German. In contrast, CI score is evaluated on individual pixels, thus large segments of SES. In fact, this was our purpose when we used the WL between CI and DL to assess the goodness of fit of the model without ignoring any dimension and overcoming the imbalance class issue.

From Fig. 2 and Table 2, training the model with the mixture of the two datasets gives greater generalizability of the model and allows prediction of SES from both locations with a better trade-off in accuracy. Besides, one can clearly see that the difference in accuracy is not significant between trained models on images with the resolution of 256×256 or 128×128 (also tested on the same resolution). The accuracy is actually better with 256×256 resolution, but the difference is not significant. Hence, it can be beneficial to use the 128×128 from the hardware usage point of view, as the model will have to learn more parameters with more pixels being used.

4.3. Model cross-validation

Cross-validation is the procedure when the model is trained on one dataset and then tested on another. Fig. 3 and Table 3 show the performance of the model when trained with Swedish data and tested on German, and vice versa. Fig. 3 and Table 3 also demonstrate the performance of the model when trained with mixed data and tested with both Swedish and German datasets, respectively.

A little surprisingly is that the U-net model trained on Swedish data, while tested on the German data, became the more efficient. This is due to the fact that the German dataset is more carefully curated and selected. The German dataset specifically includes highly localized systems that are prominently featured in the center of the images, with fewer images containing partial systems. In other words, a model that is trained on a challenging dataset, such as the Swedish dataset, is likely to have learned to capture complex patterns and generalize to new data. If this model is then tested on a curated and selected dataset, such as the German dataset, the evaluation is likely to be more indicative of the model's true performance.

We can also see that training the model on the mixed dataset makes the model highly generalizable and more widely applicable. Finally, it is again here confirmed that the use of higher resolution does not significantly help the accuracy of the model, but rather consumes more memory and computational capacity.

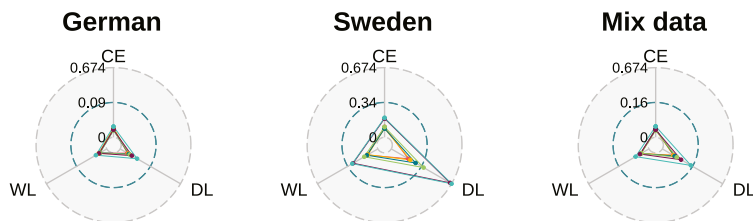
5. Model benchmarking

In order to benchmark our proposed modeling for SES identification with the existing literature, we calculated the MAPE, MAE, mIoU and the B scores for all datasets, trained and tested with all different resolution combinations. Table 4 summarizes the results, which vary considerably depending on dataset and resolution. Scores are always better if using the German data and slightly better when using a higher image resolution. In addition, training and testing the model on different image resolutions yields worse scores, especially when the training resolution is lower than the test resolution.

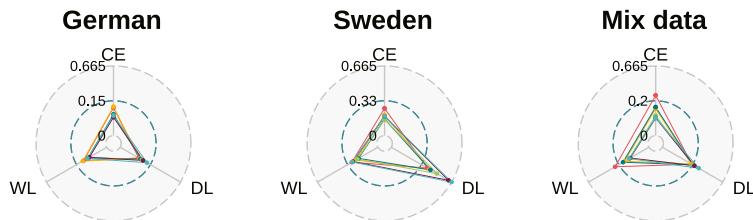
In the same way, to test the cross validation of the model, Table 5 shows MAPE, MAE, mIoU and the B scores for all datasets, trained on one data set and tested on another. The results here again surprisingly show that training the model on the Swedish data set and test it on the German yields better results. Also, training with mixed data provides better performance as usual.

Compared to other CNN architectures, the U-Net model offers distinct advantages for image segmentation tasks. Notably, it employs a contracting path to capture high-level features and an expansive path to produce a segmentation map at the original resolution. This

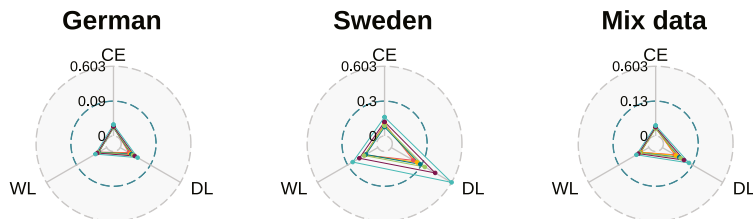
Train: 256×256; Test: 256×256



Train: 256×256; Test: 128×128



Train: 128×128; Test: 128×128



Train: 128×128; Test: 256×256

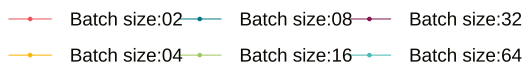
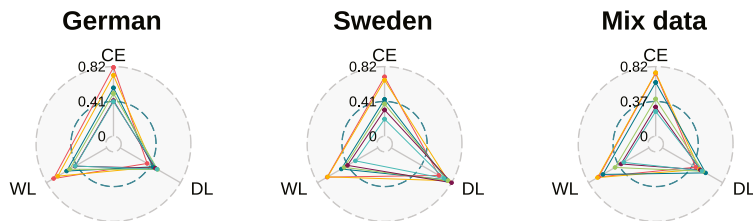


Fig. 2. Radar plots showing the cross entropy (CE) and the dice loss (DL) scores, along with their weighted loss (WL) for all datasets, trained and tested with all different resolution combinations. Note that the scale of the radial axes differs slightly between the rows.

unique design enables U-Net to effectively capture both local and global context information, which is crucial for accurate segmentation. Additionally, U-Net uses skip connections between the contracting and expansive paths to reuse information from earlier layers, resulting in more detailed segmentation maps. Moreover, U-Net’s use of a pixel-wise cross-entropy loss function during training allows it to compare predicted segmentation masks to ground truth segmentation masks, facilitating accurate segmentation.

Furthermore, U-Net’s relatively small number of parameters makes it easier to train on limited amounts of data, unlike other neural network architectures that may have many more parameters and be prone to overfitting [27].

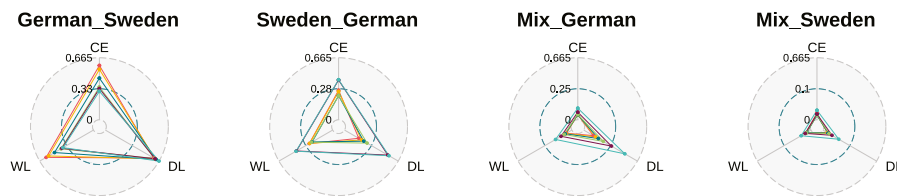
Based on our validation results and benchmarking scores, we determined that the model with a batch size of 8 strikes the best balance between performance and generalization. Table 6 illustrates a comparison between our U-Net model and recent literature, revealing that

Table 2

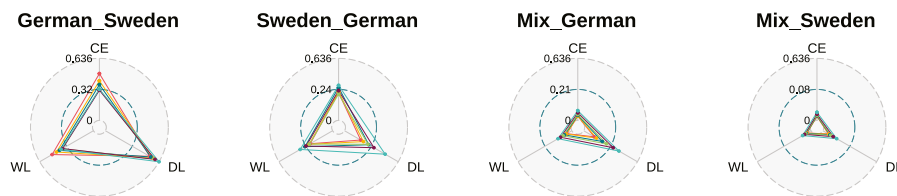
The table shows the cross entropy (CE) and the dice loss (DL) scores, along with their weighted loss (WL) for all datasets, trained and tested with all different resolution combinations.

Train resolution	Test resolution	Batch size	German			Swedish			Mix data		
			CE	DL	WL	CE	DL	WL	CE	DL	WL
256 × 256	256 × 256	02	0.08	0.08	0.08	0.10	0.20	0.13	0.08	0.13	0.09
		04	0.10	0.09	0.09	0.09	0.22	0.12	0.08	0.12	0.09
		08	0.07	0.10	0.08	0.09	0.28	0.14	0.08	0.14	0.10
		16	0.07	0.11	0.08	0.10	0.36	0.18	0.07	0.16	0.09
		32	0.07	0.13	0.09	0.18	0.66	0.32	0.07	0.21	0.11
		64	0.10	0.19	0.13	0.19	0.67	0.33	0.11	0.32	0.17
256 × 256	128 × 128	02	0.28	0.21	0.26	0.27	0.43	0.32	0.39	0.33	0.37
		04	0.28	0.22	0.26	0.20	0.43	0.27	0.27	0.32	0.29
		08	0.21	0.23	0.21	0.16	0.45	0.25	0.27	0.34	0.29
		16	0.18	0.24	0.20	0.15	0.50	0.26	0.19	0.34	0.24
		32	0.18	0.25	0.20	0.18	0.63	0.32	0.17	0.36	0.23
		64	0.19	0.29	0.23	0.18	0.66	0.33	0.17	0.40	0.24
128 × 128	128 × 128	02	0.08	0.09	0.08	0.10	0.21	0.14	0.07	0.12	0.09
		04	0.08	0.10	0.09	0.09	0.24	0.14	0.07	0.14	0.09
		08	0.08	0.12	0.09	0.08	0.29	0.14	0.07	0.17	0.10
		16	0.07	0.13	0.09	0.09	0.34	0.17	0.07	0.18	0.10
		32	0.08	0.15	0.10	0.12	0.44	0.22	0.08	0.21	0.12
		64	0.10	0.17	0.12	0.16	0.60	0.29	0.09	0.27	0.14
128 × 128	256 × 256	02	0.82	0.37	0.68	0.72	0.66	0.70	0.75	0.46	0.66
		04	0.71	0.45	0.64	0.66	0.77	0.69	0.74	0.50	0.67
		08	0.57	0.45	0.53	0.44	0.76	0.53	0.64	0.59	0.62
		16	0.50	0.51	0.51	0.38	0.78	0.50	0.44	0.55	0.47
		32	0.42	0.48	0.44	0.31	0.82	0.46	0.34	0.52	0.40
		64	0.41	0.50	0.43	0.20	0.72	0.36	0.29	0.54	0.37

Train: 256×256; Test: 256×256



Train: 128×128; Test: 128×128



— Batch size:02 — Batch size:08 — Batch size:32
— Batch size:04 — Batch size:16 — Batch size:64

Fig. 3. Radar plots showing the cross entropy (CE) and the dice loss (DL) scores, along with their weighted loss (WL) if trained on one data set (e.g., German) and tested on another (e.g., Swedish) for four combinations of data sets (e.g., German–Swedish) and two different resolutions. Note that the scale of the radial axes differs slightly between the rows.

our model’s segmentation accuracy surpasses previous reported performances. This reinforces U-Net’s effectiveness as an image segmentation tool and highlights its suitability for SES detection.

Fig. 4 shows some uses of the 8 batch-sized U-net model to generate segmentation of some SES from aerial images. The first row contains the example images, the second row contains the manually created masks from [24] over the same example images and the third row contains the result of the 8 batch-sized U-net model application on the same example images. These images have been selected to showcase different aspects

of the model’s performance, including cases where it accurately predicts pixel values and instances where its pixel predictions may be incorrect. However, the U-net architecture excels at capturing fine details, such as panel edges and textures, enabling precise segmentation. In successful cases, the model demonstrates its ability to distinguish SES from the surrounding environment, even in challenging scenarios with varying lighting conditions and complex backgrounds. However, there are instances where the model may struggle, particularly when panels are partially shaded or exhibit irregular shapes. False positives can occur

Table 3

The table shows the cross entropy (CE) and the dice loss (DL) scores, along with their weighted loss (WL) if trained on one data set (e.g., German) and tested on another (e.g., Swedish) for four combinations of data sets (e.g., German–Swedish).

Train resolution	Test resolution	Batch size	German–Swedish			Swedish–German			Mix data-German			Mix data-Swedish		
			CE	DL	WL	CE	DL	WL	CE	DL	WL	CE	DL	WL
256 × 256	256 × 256	02	0.57	0.58	0.57	0.32	0.18	0.28	0.05	0.12	0.07	0.08	0.06	0.07
		04	0.55	0.62	0.57	0.29	0.21	0.27	0.06	0.14	0.08	0.05	0.06	0.05
		08	0.44	0.62	0.50	0.26	0.24	0.25	0.05	0.18	0.09	0.05	0.06	0.05
		16	0.35	0.61	0.42	0.25	0.28	0.26	0.05	0.23	0.11	0.05	0.08	0.06
		64	0.34	0.62	0.42	0.43	0.54	0.46	0.08	0.34	0.16	0.06	0.11	0.07
128 × 128	128 × 128	02	0.47	0.53	0.49	0.28	0.18	0.25	0.05	0.13	0.07	0.05	0.07	0.05
		04	0.40	0.55	0.44	0.29	0.22	0.27	0.04	0.15	0.07	0.05	0.07	0.05
		08	0.36	0.55	0.42	0.34	0.28	0.32	0.05	0.22	0.10	0.05	0.09	0.06
		16	0.32	0.58	0.40	0.27	0.28	0.28	0.06	0.27	0.12	0.05	0.10	0.07
		64	0.31	0.58	0.39	0.30	0.35	0.32	0.08	0.35	0.16	0.06	0.12	0.08
		64	0.32	0.64	0.41	0.36	0.48	0.40	0.10	0.42	0.20	0.08	0.16	0.11

Table 4

The table shows the mean absolute percentage error (MAPE), the mean absolute error (MAE), the mean Intersection-over-Union (mIoU) and the bias (B) scores for all datasets, trained and tested with all different resolution combinations.

Train resolution	Test resolution	Batch size	German				Swedish				Mix data			
			MAPE	MAE	mIoU	B	MAPE	MAE	mIoU	B	MAPE	MAE	mIoU	B
256 × 256	256 × 256	02	0.08	638.81	0.92	0.01	0.21	645.49	0.86	0.06	0.14	624.68	0.90	0.01
		04	0.09	778.45	0.91	0.01	0.27	796.39	0.85	0.04	0.14	609.35	0.90	0.01
		08	0.09	787.62	0.91	0.01	0.35	998.83	0.81	0.04	0.16	705.84	0.89	0.01
		16	0.12	883.77	0.90	0.01	0.43	1487.14	0.74	0.06	0.19	785.79	0.87	-0.00
		32	0.15	1096.09	0.88	0.01	1.18	3194.51	0.57	-0.09	0.24	1025.04	0.83	-0.00
		64	0.24	1725.68	0.83	0.01	1.25	3315.18	0.56	-0.11	0.40	1731.00	0.76	0.00
256 × 256	128 × 128	02	0.23	551.36	0.83	0.03	0.53	483.10	0.73	0.12	0.35	551.54	0.78	0.09
		04	0.23	637.67	0.81	0.06	0.55	472.51	0.72	0.10	0.35	560.18	0.77	0.09
		08	0.24	620.86	0.81	0.06	0.64	513.70	0.70	0.02	0.35	618.96	0.76	0.12
		16	0.27	633.56	0.79	0.06	0.75	571.11	0.66	-0.03	0.36	619.19	0.75	0.10
		32	0.30	670.52	0.78	0.05	1.54	920.18	0.57	-0.32	0.37	610.31	0.74	0.07
		64	0.40	823.68	0.74	0.02	1.39	867.58	0.56	-0.19	0.49	700.53	0.70	0.03
128 × 128	128 × 128	02	0.08	152.49	0.91	0.00	0.26	198.82	0.84	0.07	0.14	147.61	0.89	0.01
		04	0.10	180.90	0.90	0.01	0.31	215.25	0.82	0.04	0.14	159.81	0.88	0.01
		08	0.11	208.16	0.89	0.01	0.35	248.31	0.79	0.04	0.19	193.63	0.86	0.00
		16	0.14	233.27	0.88	0.00	0.46	322.48	0.76	0.01	0.19	207.31	0.85	0.00
		32	0.16	285.99	0.86	0.01	0.58	502.65	0.69	0.03	0.24	265.42	0.83	-0.01
		64	0.20	363.49	0.84	0.00	0.98	712.52	0.60	-0.02	0.33	343.82	0.79	-0.02
128 × 128	256 × 256	02	0.44	5695.65	0.71	0.42	0.68	3980.89	0.60	0.72	0.51	5324.13	0.67	0.51
		04	0.53	6783.89	0.66	0.51	0.79	4431.84	0.54	0.80	0.57	5875.94	0.64	0.56
		08	0.51	6732.88	0.66	0.51	0.75	4293.79	0.55	0.77	0.63	6570.71	0.60	0.63
		16	0.58	7794.21	0.62	0.59	0.79	4550.40	0.54	0.82	0.59	6143.63	0.61	0.59
		32	0.54	7225.34	0.63	0.54	0.75	4510.06	0.51	0.81	0.56	5566.12	0.63	0.53
		64	0.56	7446.43	0.61	0.55	0.75	3171.14	0.55	0.34	0.57	5231.05	0.61	0.49

in such cases. To address these limitations, potential improvements include augmenting the training data with diverse shading patterns, refining the model through fine-tuning techniques, and exploring recent advanced U-net architectures.

6. Discussion and future perspectives

CNN-based semantic segmentation techniques have become the state-of-the-art approach for automatic identification of solar energy systems (SES) from aerial images [18]. Among them, the U-Net model has emerged as one of the most effective architectures for global localization and context mapping, making it a logical choice for the solar energy realm. However, class imbalance and data sparsity pose significant challenges for applying such models. In this study, we addressed these challenges by using a weighted loss function that combines cross-entropy and Dice loss to overcome the problem of class imbalance.

To further improve the accuracy of our model in future work, we propose exploring the addition of recurrent convolutional layers or recurrent residual convolutional layers to the U-Net model. Additionally,

exploring different loss functions could also help increase the model’s accuracy; e.g., Jaccard loss.

Furthermore, in future work, we plan to extend our study to a multi-class U-Net architecture where different solar energy systems are labeled differently, such as PV or solar thermal systems. This is a challenging task as the two technologies share similar texture and color appearance. However, we believe that with the right modifications and improvements, the U-Net model can be effectively adapted for multi-class segmentation of SES.

A crucial first step in determining the capacity of existing SESs is to estimate the area of the identified SESs. This study have proven that a U-net model can assess the area in an aerial imagery with high accuracy. However, the tilt of the modules in a SES is also needed for a correct area estimation. Calculating the tilt can be done either from 3D building data [17] or high/low-resolution LiDAR data [30,31]. Combining the later with the method of this study is the planned next step, with the purpose of estimating the nameplate DC-power of PV systems by assuming a reasonable conversion efficiency of the modules.

Table 5

The table shows the mean absolute percentage error (MAPE), the mean absolute error (MAE), the mean Intersection-over-Union (mIoU) and the bias (B) scores for all datasets, trained on one data set and tested on another.

Train resolution	Test resolution	Batch size	German–Swedish				Swedish–German				Mix data-German				Mix data-Swedish			
			MAPE (%)	MAE	mIoU	B	MAPE (%)	MAE	mIoU	B	MAPE (%)	MAE	mIoU	B	MAPE (%)	MAE	mIoU	B
256 × 256	256 × 256	02	0.61	2504.53	0.66	0.44	0.31	1953.95	0.85	-0.01	0.14	408.62	0.91	0.03	0.05	384.22	0.94	-0.00
		04	0.64	2725.84	0.65	0.48	0.37	2544.75	0.82	0.01	0.18	468.90	0.89	0.04	0.05	392.63	0.94	-0.00
		08	0.65	3127.68	0.63	0.56	0.44	2920.97	0.80	-0.02	0.22	560.97	0.86	0.03	0.05	427.03	0.93	-0.00
		16	0.63	2966.54	0.64	0.53	0.67	3875.22	0.75	-0.15	0.29	739.74	0.82	0.04	0.07	492.12	0.92	-0.00
		32	0.61	3177.75	0.63	0.56	1.65	7030.18	0.55	-0.24	0.39	1276.17	0.76	0.08	0.13	799.45	0.89	-0.01
		64	0.65	3260.47	0.60	0.57	1.54	6823.66	0.54	-0.18	0.53	2037.68	0.66	0.20	0.27	1741.14	0.82	-0.02
128 × 128	128 × 128	021	0.56	559.36	0.69	0.37	0.26	454.36	0.83	-0.00	0.13	91.94	0.90	0.03	0.05	95.73	0.93	-0.01
		04	0.59	590.68	0.67	0.41	0.41	607.99	0.81	-0.01	0.18	111.25	0.88	0.03	0.06	98.21	0.92	-0.01
		08	0.55	635.79	0.67	0.45	0.54	886.73	0.76	0.04	0.24	160.43	0.84	0.04	0.07	114.90	0.91	-0.01
		16	0.60	702.88	0.65	0.49	0.69	935.87	0.76	-0.15	0.27	201.68	0.81	0.06	0.09	147.46	0.90	-0.02
		32	0.57	694.81	0.64	0.48	1.06	1288.80	0.69	-0.27	0.38	287.86	0.76	0.08	0.12	190.71	0.88	-0.02
		64	0.60	789.22	0.62	0.54	1.35	1507.14	0.60	-0.21	0.45	387.75	0.71	0.11	0.22	314.82	0.84	-0.04

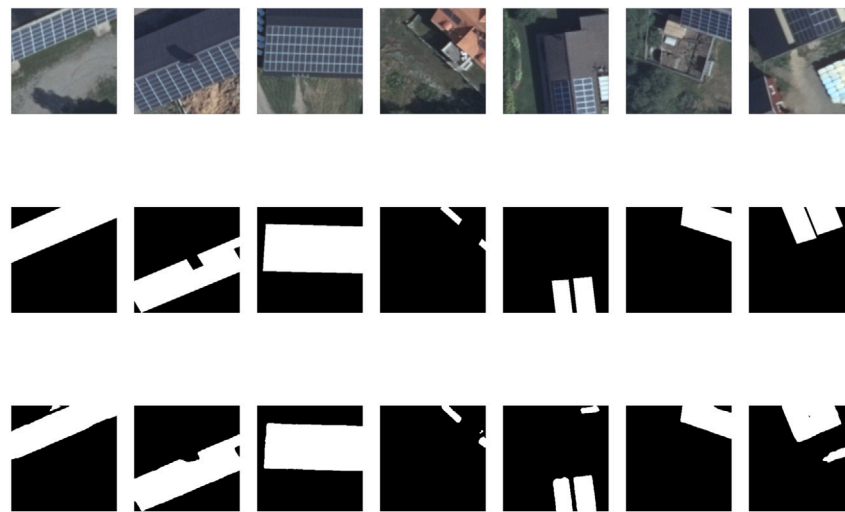


Fig. 4. Example of images segmentation (From the Swedish dataset). First row shows examples images. Second row shows manually created masks from the same example images. Third row shows the result of the selected U-net model application on the same example images. The selected cases demonstrate instances where the model’s pixel predictions are precise, as well as cases where they may deviate from the expected values.

Table 6

Comparison between our model with 8 as a batch size and some recent articles in the literature.

Document	MAPE (%)	MAE	mIoU (%)	B (%)
Yu et al. [9]	24.60	-	-	3.00
Castello et al. [10]	-	-	64.00	-
Zhuang et al. [28]	-	-	75.00	-
Parhar et al. [26]	-	-	82.00	-
Kasmi et al. [29]	-	-	86.00	-
Mayer et al. [17]	18.50	-	74.00	3.90
Our_German_256 × 256	9.00	787.62	91.00	1.00
Our_German_128 × 128	10.00	180.90	90.00	1.00
Our_Swedish_256 × 256	35.00	998.83	81.00	4.00
Our_Swedish_128 × 128	35.00	248.31	79.00	4.00
Our_Mix_256 × 256	16.00	705.84	89.00	1.00
Our_Mix_128 × 128	19.00	193.63	86.00	0.10

7. Conclusion

In this paper, we propose a U-Net model architecture for SES localization from aerial images. The model is trained using two different datasets, the first subset is from Sweden and the second one is from Germany. For more generalizability, the model is trained and tested on the mixture of both datasets.

In conclusion, the study demonstrates that the proposed model achieves superior performance on segmentation tasks when compared

to other recent models used as benchmarks. Furthermore, mixing the two datasets, the German and Swedish, gives a better trade-off and works accurately in both datasets. Additionally, it was found that while higher resolution did not significantly improve segmentation results, it did result in increased hardware usage. Further research is needed to fully realize the potential of these results, such as using 3D building data or high/low-resolution LiDAR data to estimate the exact identified SES area, orientation and shading and to optimize the utilization of the U-net results.

To re-use this algorithm, all needed software and dependencies to run the U-net code are freely available. See [Appendix A](#) of this paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the financing from the Swedish Energy Agency, Sweden through the research project “Automatic mapping of solar panels and generation of solar forecasts through aerial imagery and machine learning” (50265-1) and Solar Electricity Research Center Sweden (SOLVE). The work also forms part of the Swedish strategic research programme StandUp for Energy.

The computations handling were enabled by resources in project “PV-semantic-segmentation of satellite images (SNIC 2022/22-145)” provided by the Swedish National Infrastructure for Computing (SNIC), partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Appendix A

This section outlines the practical steps involved in using the U-net model described in this paper.

First of all, for ease of use, one can get our [Docker container](#) which comes with all the necessary dependencies to run the GPU version of the U-net model under R language in any machine with a compatible GPU device (It runs on the CPU if no GPU is found). However, training a U-Net model on a CPU takes considerably longer than training on a GPU, as GPUs offer parallel processing capabilities. The actual training time can vary greatly depending on the hardware specifications, model size and complexity, machine configuration being used, etc.

To do so, execute the following command:

```
# Install Docker if it is not:
sudo apt-get update
sudo apt-get install docker.io
# Then pull our container:
docker pull frimane/azfunetr:latest
```

It is important to note that we designed this Docker container to be separate from the U-net model script, allowing it to be utilized for running other R scripts with GPU-efficiency.

To run the U-net model on top of this Docker container, files and directories need to have a given hierarchy. This was only our design, but anyone can modify it as desired by changing the paths from the source code scripts. Fig. A.5 shows this hierarchy.

Firstly, the U-net script is written in 5 R-files: **optimization.R**, **pvSegDsClass.R**, **training.R**, **UNetContainers.R** and **UnetTorch.R**. These script files are held in a directory called **Rscript**. The datasets split into **test**, **train** and **validation** directories. Each one of them is divided into **image** and **mask** directories. The whole dataset should be kept in a directory named **pvData**. The final results will be stored in a directory called **output** when the U-net execution ends.

Finally, to start the U-net model, the following script can be used:

```
#!/bin/bash
echo "Starting Docker service"
systemctl restart docker
echo "Starting U-net model"
docker run -it --gpus all --rm -v
    Working_directory/Rscripts:/Rscripts -v
    Working_directory/pvData:/pvData -v
    Working_directory/output:/output
    path_to/azfunetr:latest
# Working_directory: the folder where all
# the work is performed
# path_to: the path to the directory where
# the container is saved
```

Listing 1: Command lines needed to run the U-net model on the top of the Docker container.

Singularity software can also be used instead of the Docker container. Simply convert Docker into a Singularity file.

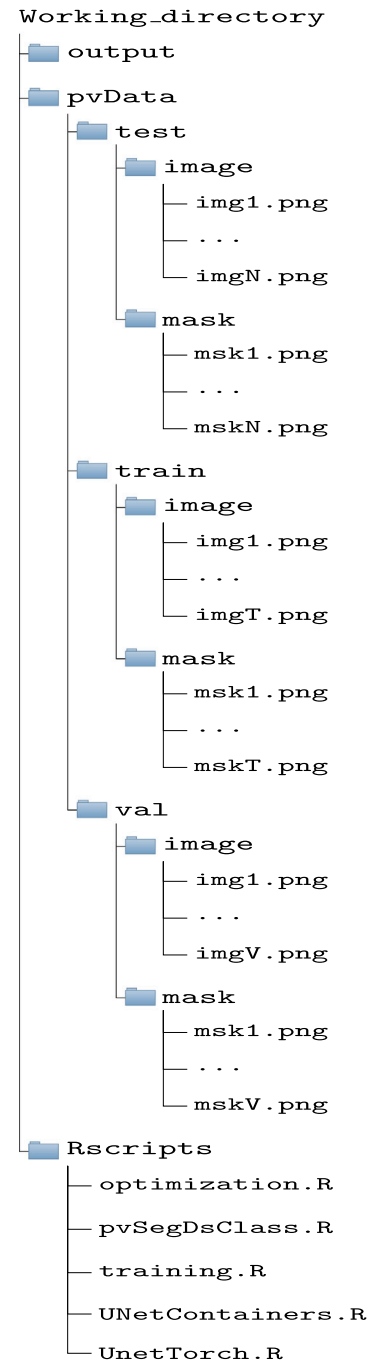


Fig. A.5. File tree showing the hierarchy needed to run the U-net model on top of the Docker container in our case.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.solener.2023.111822>.

This paper comes with supplementary materials, see the online version, at *doi place holder*. This supplementary material is an accompanying paper providing results and plots for the convergence and number of epochs during the U-net model execution. The plots are segregated by location and resolution.

In addition, the supplementary material contains the U-net script files zipped in file called **U_net_code.zip**.

References

[1] IEA PVPS task 1, G. Masson, I. Kaizuka, E. Bosch, C. Plaza, A. Scognamiglio, A. Jäger-Waldau, J. Lindahl, E. Blokken, Trends in Photovoltaic Applications — 2022, 2022.

[2] F. Creutzig, P. Agoston, J.C. Goldschmidt, G. Luderer, G. Nemet, R.C. Pietzcker, The underestimated potential of solar energy to mitigate climate change, *Nature Energy* 2 (2017) 17140, <http://dx.doi.org/10.1038/nenergy.2017.140>.

[3] M. Jaxa-Rozen, E. Trutnevte, Sources of uncertainty in long-term global scenarios of solar photovoltaic technology, *Nature Clim. Change* 11 (2021) 266–273, <http://dx.doi.org/10.1038/s41558-021-00998-8>, URL: <https://www.nature.com/articles/nenergy2017140>.

[4] M. Victoria, N. Haegel, I.M. Peters, R. Sinton, A. Jäger-Waldau, C. del Cañizo, C. Breyer, M. Stocks, A. Blakers, I. Kaizuka, K. Komoto, A. Smets, Solar photovoltaics is ready to power a sustainable future, *Joule* 5 (2021) 1041–1056, <http://dx.doi.org/10.1016/j.joule.2021.03.005>, URL: <https://www.sciencedirect.com/science/article/pii/S2542435121001008>.

[5] C. Wilson, A. Grubler, N. Bento, S. Healey, S.D. Stercke, C. Zimm, Granular technologies to accelerate decarbonization, *Science* 368 (2020) 36–39, <http://dx.doi.org/10.1126/science.aaz8060>.

[6] IEA PVPS task 1, Data Model for PV Systems — Data Model and Data Acquisition for PV Registration Schemes and Grid Connection Evaluations, 2020.

[7] UNDP, The Next Frontier: Human Development and the Anthropocene, 2020.

[8] J. Yuan, H.H.L. Yang, O.A. Omिताomu, B.L. Bhaduri, Large-scale solar panel mapping from aerial images using deep convolutional networks, in: Proceedings of the 2016 IEEE International Conference on Big Data, 2016, pp. 2703–2708, <http://dx.doi.org/10.1109/BigData.2016.7840915>.

[9] 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 (2018) 2605–2617, <http://dx.doi.org/10.1016/j.joule.2018.11.021>, URL: <https://www.sciencedirect.com/science/article/pii/S2542435118305701>.

[10] R. Castello, S. Roquette, M. Esguerra, A. Guerra, J.L. Scartezzini, Deep learning in the built environment: Automatic detection of rooftop solar panels using convolutional neural networks, *J. Phys. Conf. Ser.* 1343 (2019) 1–6, <http://dx.doi.org/10.1088/1742-6596/1343/1/012034>.

[11] 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: Proceedings of the 3rd International Conference on Smart Energy Systems and Technologies, SEST, 2020, pp. 11–16, <http://dx.doi.org/10.1109/SEST48500.2020.9203258>.

[12] X. Hou, B. Wang, W. Hu, I. Yin, A. Huang, H. Wu, SolarNet: A deep learning framework to map solar plants in China from satellite imagery, in: ICLR 2020 Workshop on Tackling Climate Change with Machine Learning, 2020.

[13] M.V.C.V.d. Costa, O.L.F.d. Carvalho, A.G. Orlandi, I. Hirata, A.O.d. Albuquerque, F.V.e. Silva, R.F. Guimarães, R.A.T. Gomes, O.A.d.C. Júnior, Remote sensing for monitoring photovoltaic solar plants in Brazil using deep semantic segmentation, 14 (2021) URL: <https://www.mdpi.com/1996-1073/14/10/2960>.

[14] X. Zhang, M. Zeraatpisheh, M.M. Rahman, S. Wang, M. Xu, Texture is important in improving the accuracy of mapping photovoltaic power plants: A case study of Ningxia Autonomous Region, China, *Remote Sens.* 13 (2021) <http://dx.doi.org/10.3390/rs13193909>.

[15] L. Kruitwagen, K.T. Story, J. Friedrich, L. Byers, S. Skillman, C. Hepburn, A global inventory of photovoltaic solar energy generating units, *Nature* 598 (2021) 604–610, <http://dx.doi.org/10.1038/s41586-021-03957>.

[16] P. Li, H. Zhang, Z. Guo, S. Lyu, J. Chen, W. Li, X. Song, R. Shibasaki, J. Yan, Understanding rooftop PV panel semantic segmentation of satellite and aerial images for better using machine learning, *Adv. Appl. Energy* 4 (2021) 100057, <http://dx.doi.org/10.1016/j.adapen.2021.100057>, URL: <https://www.sciencedirect.com/science/article/pii/S2666792421000494>.

[17] 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, *Appl. Energy* 310 (2022) 118469, <http://dx.doi.org/10.1016/j.apenergy.2021.118469>, URL: <https://www.sciencedirect.com/science/article/pii/S0306261921016937>.

[18] S. Ren, W. Hu, K. Bradbury, D. Harrison-Atlas, L. Malaguzzi Valeri, B. Murray, J.M. Malof, Automated extraction of energy systems information from remotely sensed data: A review and analysis, *Appl. Energy* 326 (2022) 119876, <http://dx.doi.org/10.1016/j.apenergy.2022.119876>.

[19] Z. Wang, M.L. Arlt, C. Zanocco, A. Majumdar, R. Rajagopal, DeepSolar++: Understanding residential solar adoption trajectories with computer vision and technology diffusion models, *Joule* 6 (2022) 2611–2625, <http://dx.doi.org/10.1016/j.joule.2022.09.011>, URL: <https://www.sciencedirect.com/science/article/pii/S2542435122004779>.

[20] W. Hu, K. Bradbury, J.M. Malof, B. Li, B. Huang, A. Streltsov, K. Sydney Fujita, B. Hoen, What you get is not always what you see—pitfalls in solar array assessment using overhead imagery, *Appl. Energy* 327 (2022) 120143, <http://dx.doi.org/10.1016/j.apenergy.2022.120143>, URL: <https://www.sciencedirect.com/science/article/pii/S0306261922014003>.

[21] J. Lindahl, S. Ekbring, R. Johansson, D. Lingfors, J. Munkhammar, Socioeconomic and demographic factors behind the deployment of domestic photovoltaic and solar thermal systems in three Swedish municipalities, in: Proceedings of the 8th World Conference on Photovoltaic Energy Conversion, 2022, pp. 1530–1540.

[22] Z. Xia, Y. Li, R. Chen, D. Sengupta, X. Guo, B. Xiong, Y. Niu, Mapping the rapid development of photovoltaic power stations in northwestern China using remote sensing, *Energy Rep.* 8 (2022) 4117–4127, <http://dx.doi.org/10.1016/j.egy.2022.03.039>, URL: <https://www.sciencedirect.com/science/article/pii/S2352484722005935>.

[23] 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, 2020, [arXiv:2012.03690](https://arxiv.org/abs/2012.03690).

[24] J. Lindahl, R. Johansson, D. Lingfors, Mapping of decentralised photovoltaic and solar thermal systems by remote sensing aerial imagery and deep machine learning for statistic generation, 2023, Unpublished manuscript.

[25] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention, MICCAI 2015*, Springer International Publishing, Cham, 2015, pp. 234–241.

[26] P. Parhar, R. Sawasaki, A. Todeschini, C. Reed, H. Vahabi, N. Nusaputra, F. Vergara, HyperionSolarNet: Solar panel detection from aerial images, 2022, [CoRR URL: https://arxiv.org/abs/2201.02107](https://arxiv.org/abs/2201.02107).

[27] J.M. Bright, Åzeddine Frimane, J. Munkhammar, J. Widén, The future of synthetic solar irradiance, in: *Synthetic Solar Irradiance*, 2021, p. 28, http://dx.doi.org/10.1063/9780735421820_006.

[28] L. Zhuang, Z. Zhang, L. Wang, The automatic segmentation of residential solar panels based on satellite images: A cross learning driven U-net method, *Appl. Soft Comput.* 92 (2020) 106283, <http://dx.doi.org/10.1016/j.asoc.2020.106283>, URL: <https://www.sciencedirect.com/science/article/pii/S1568494620302234>.

[29] 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, 2022, <http://dx.doi.org/10.48550/ARXIV.2207.07466>, [arXiv](https://arxiv.org/abs/2207.07466).

[30] D. Lingfors, J. Bright, N. Engerer, J. Ahlberg, S. Killinger, J. Widén, Comparing the capability of low- and high-resolution LiDAR data with application to solar resource assessment, roof type classification and shading analysis, *Appl. Energy* 205 (2017) 1216–1230, <http://dx.doi.org/10.1016/j.apenergy.2017.08.045>, URL: <https://www.sciencedirect.com/science/article/pii/S0306261917310619>.

[31] D. Lingfors, S. Killinger, N.A. Engerer, J. Widén, J.M. Bright, Identification of PV system shading using a LiDAR-based solar resource assessment model: An evaluation and cross-validation, *Sol. Energy* 159 (2018) 157–172, <http://dx.doi.org/10.1016/j.solener.2017.10.061>, URL: <https://www.sciencedirect.com/science/article/pii/S0038092X17309416>.