SolarDetector: Automatic Solar PV Array Identification using Big Satellite Imagery Data

Qi Li
Colorado School of Mines
Golden, Colorado, USA

Sander Schott
Colorado School of Mines
Golden, Colorado, USA

Dong Chen
Colorado School of Mines
Golden, Colorado, USA

ABSTRACT

Due to the intermittent nature of solar energy, it has been increasingly challenging for the utilities, third-parties, and government agencies to integrate distributed energy resources generated by rooftop solar photovoltaic (PV) arrays into smart grids. Recently, there is a rising interest in automatically collecting solar installation information in a geospatial region that are necessary to manage this stochastic green energy, including the quantity and locations of solar PV deployments, and their profiling information. Most recent work focuses on using big aerial or satellite imagery data to train machine learning or deep learning models to automatically detect solar PV arrays. Unfortunately, these approaches are suffering low detection accuracy due to the insufficient sample and feature learning when building their models, and the separation of rooftop object segmentation and identification during their detection process. In addition, most recent approaches cannot report accurate multi-panel detection results.

To address these problems, we design a new approach—SolarDetector that can automatically detect and profile distributed solar photovoltaic arrays in a given geospatial region without any extra cost. SolarDetector first leverages data augmentation techniques and Generative adversarial networks (GANs) to automatically learn accurate features for rooftop objects. Then, SolarDetector employs Mask R-CNN algorithm to accurately identify rooftop solar arrays and also learn the detailed installation information for each solar array simultaneously. In addition, SolarDetector could also integrate with large-scale data processing engine—Apache Spark and graphics processing units (GPUs) to further improve its training cost. We evaluate SolarDetector using 263,430 public satellite images from 11 geospatial regions in the U.S. We find that pre-trained SolarDetector yields an average MCC of 0.76 to detect solar PV arrays over two big datasets, which is $\sim$ 50% better than the most notable approach—SolarFinder. In addition, unlike prior work, we show that SolarDetector can also accurately report the profiling information for the detected rooftop objects.

CCS CONCEPTS

• **Computing methodologies** → *Model development and analysis*; *Machine learning approaches*; *Neural networks*; *Classification and regression trees.*

IoTDI '23, May 09–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0037-8/23/05.
https://doi.org/10.1145/3576842.3582384

KEYWORDS

Big Data, Data Analytics, Solar Detection, Solar Segmentation, Deep Learning, Image Processing

ACM Reference Format:
Qi Li, Sander Schott, and Dong Chen. 2023. SolarDetector: Automatic Solar PV Array Identification using Big Satellite Imagery Data. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3576842.3582384

1 INTRODUCTION

Smart Grid, as a networked system that consists of more than 500 million sensors, advanced smart meters, and actuators, is the foundation of modern society and one of the largest Internet of Things (IoT) deployments in the world [2]. However, with electricity demand and variable renewable energy—solar generated energy adoption increasing across the U.S., the current smart grid is being stretched to its capacity to provide reliable transmission and distribution of power [33]. To help manage increasing penetration of solar energy generation while maintaining reliability, the grid could be updated to a "smart grid", a modernization which has accurate distributed solar generation resources to help automate and control the complex electricity needs of the 21st Century.

Due to the intermittent nature of solar energy, it has been increasingly challenging for the utilities, third-parties, and government agencies to integrate distributed energy resources generated by rooftop solar photovoltaic (PV) arrays into smart grids. The number of solar-powered homes in the U.S. is rapidly increasing due to a steep decline in solar module prices. The U.S. installed 4.6 gigawatts (GWdc) of solar PV capacity in Q2 2022 to reach 130.9 GWdc of total installed capacity, enough to power 23 million American homes. And the U.S. officially surpassed 3 million installations across all market segments. In particular, over 70% of solar deployments in the U.S. are continuously small-scale photovoltaic (PV) arrays from residential rooftops. For instance, government agencies (e.g., Massachusetts Applications for Cap Allocation [26]) started to place limits on the amount of solar PV arrays that can be installed in a geospatial region. The current management process highly relies on accurate regional statistics of solar deployment generation capacity. Thus, recently, there is a rising interest in automatically collecting solar installation information in an area, including the quantity and locations of solar PV deployments, and their profiling information.

Most recent work [7, 8, 12, 18–21, 28, 31] focuses on using big aerial or satellite imagery data to train machine learning or deep learning models to automatically detect solar PV arrays. The rooftop satellite and aerial images in publicly accessible maps APIs are taken by sensors and cameras in visible wavelengths on satellites and aircraft, which collect each image at a specific date and time.

However, these prior approaches typically require a significant amount of very high resolution (VHR) images (0.3∼0.8 per pixel) and human handcrafted solar PV array templates to train a reasonably accurate model. But, this kind of VHR data may cost as \$15 per km^2, and is not available at every location in the U.S. To mitigate these issues, most recent work [15] proposed a hybrid approach that can automatically detect solar PV arrays using only regular satellite imagery data. However, the two-step detection process of the hybrid approach has limited its detection performance. The process is built on top of insufficient samples and thus cannot capture all the features when building its models. Specially, the separation of rooftop object segmentation and detection processes has caused the missing of rooftop contextual information, which is critical for final solar PV arrays detection. Thus, this hybrid approach is still suffering low detection accuracy. Since most prior approaches [7, 8, 12, 15, 18–21, 28, 31] are detecting solar arrays at contour level, none of them can accurately and reliably identify multi-panel solar deployments.

To address the problem, we design a new automatic system—SolarDetector that can accurately detect and profile distributed solar photovoltaic arrays in a given region without any extra cost. Specially, SolarDetector can integrate with large-scale data processing engine—Apache Spark, and leverage graphics processing units (GPUs) to further improve deep learning model (re)training cost. Our hypothesis is that the new system—SolarDetector is capable of detecting rooftop solar PV arrays more accurately and efficiently when it combines the benefits from Deep Convolutional Generative Adversarial Networks (DCGANs)-based data augmentation, Mask R-CNN based solar PV array instance segmentation approaches, and large-scale data processing engines(and hardware). In evaluating our hypothesis, this paper makes the following contributions.

Identifying Gaps in the State-of-the-art. As reference points for solar PV arrays detection, we examine prior approaches. We find that prior ML-based and DL-based approaches are usually trained using VHR images which are costly and not available at every location, and thus cannot scale up. The separation of rooftop object segmentation and detection processes in hybrid approaches has caused the missing of rooftop contextual information, which is critical for final solar PV arrays detection. Also, since most prior ML-based, DL-based and hybrid approaches are detecting solar PV arrays at image contour level, none of them can accurately and reliably report multi-panel solar deployments.

Detection Challenge. We highlight the major challenges that we met when designing for our SolarDetector that can automatically detect and profile. Current approaches are suffering low detection accuracy and low (re)training performance mainly due to the challenges from insufficient feature learning, inaccurate multiple-panel detection,the separation of segmentation and detection, and non-integrated data processing engines.

SolarDetector Design. We design a solar PV array detection system—SolarDetector, which can automatically detect and profile distributed solar photovoltaic arrays in a given geospatial region with low (re)training costs. First, SolarDetector leverages Google Maps API and OpenStreet Maps API to download and preprocess the rooftop solar PV arrays in a given region. Second, SolarDetector leverages data augmentation techniques and Generative adversarial networks (GANs) to build large rooftop solar PV array satellite images that can enable us to learn the features and parameters of solar PV array detection models more accurately. Then, SolarDetector employs Mask R-CNN-based algorithm to accurately identify rooftop solar arrays and also learn the detailed installation information for each rooftop solar panel deployment simultaneously. Eventually, SolarDetector integrates its deep learning models with data processing engines—Apache Spark and GPU to further optimize detection and (re)training performance.

Implementation and Evaluation. We implement SolarDetector in `python` using widely available open-source frameworks, including OpenCV, Scikit-learn, PyCUDA, TensorFlow and PyTorch. We evaluate SolarDetector using $\sim 63,430$ public satellite images from 11 geospatial regions in the U.S. We find that pre-trained SolarDetector yields an average MCC of 0.76 over two big datasets, which is $\sim$ 50% better than the most notable approach—SolarFinder. We evaluate our new approach–SolarDetector using multiple ways: (1) We compare SolarDetector's results with the groundtruth data from 260,000 sites and show that it can accurately detect rooftop solar installations and also learn installation characteristics of each solar site. (2) We validate SolarDetector's detection results using ground truth data from 3,430 publicly-available solar PV array rooftop images using Google Maps API. (3) We validate SolarDetector's accuracy for profiling local physical characteristics for ∼4,000 solar sites by examining the accuracy of rooftop object profiling(e.g., the size and orientation of solar panels, the size of shadows on rooftops, and multi-panel deployment). (4) We compare SolarDetector's model training costs using Spark and multiple GPUs acceleration.

Releasing Datasets and Code. We release all the datasets that are comprised of over 63,430 satellite images and the source code of SolarDetector on our website [29] so that researchers can use SolarDetector to benchmark their future work. In addition to utilities and their third parities, SolarDector can also provide supplemental more up-to-date information for solar installers, smart city managers, and other third parties who do not directly have the access to electric gird smart meter data.

2 BACKGROUND AND RELATED WORK

Problem statement: Given a geospatial region, we first want to build a new, low-cost approach that can automatically extract rooftop satellite images from publicly-available low or standard resolution satellite imagery APIs. We then present a new approach that can detect rooftop solar arrays accurately and effectively. Moreover, for each of the detected rooftop solar arrays, we want to learn the size, orientation, shading condition, and other physical characteristics that are critical to predicting solar generation capacity per solar site. We also want to learn the other objects on the rooftops, such as trees, chimney, and shading. Note that, we do not assume our system users have fully access to electric grid energy meter data. In real practice, there is always a delay when the new solar site can become online on the electric grid due to the waiting time to get city permit and pass inspection process. Our work can be used as a supplemental toolkit to help users to better inform the new deployments.

We outline the design alternatives for detecting distributed rooftop solar PV arrays using net meter data and big satellite imagery data, including machine learning (ML)-based approaches,

Model	True Positives	True Negatives	False Positives	False Negatives	MCC
SVMs	95.49%	55.50%	44.50%	4.50%	0.42
Logistic Regression	99.55%	31.12%	68.80%	0.45%	0.29
Random Forest	99.55%	31.20%	68.80%	0.45%	0.29
CNNs	14.41%	96.98%	3.02 %	85.59%	0.21
SolarFinder	90.09 %	80.77%	19.23%	9.91%	0.61

Table 1: The comparison of detection accuracy when employing different prior solar array detection approaches.

deep learning (DL)-based approaches, and a hybrid approach which combines the benefits from both ML-based and DL-based approaches. In doing so, we review a wide range of the most recent sophisticated solar array detection approaches based on logical regression (LR), support vector machines (SVMs) and random forest (RF) [18, 20, 21], convolutional neural networks (CNNs) [7, 19, 28], and hybrid approach [15]. Table 1 quantifies the effectiveness of the five approaches by showing the percentage of the approaches that yield true positives (detect solar array and the rooftop does have one), true negatives (detects no solar array and the rooftop does not have one), false positives (detect solar array but the rooftop does not have one), false negatives (detect no solar array but the rooftop does have one). We also report the Matthews Correlation Coefficient (MCC) metric for each approach, a standard measure of a binary classifier's performance, where its values are in the range of -1.0 and 1.0, with 1.0 being a perfect solar array detection, 0.0 being random solar array prediction, and -1.0 indicating an always wrong solar array detection. To report Table 1, we used a dataset that has 3,430 satellite images using Google Maps APIs [10]. We use the same dataset to benchmark the five different solar array detection approaches. Also, we trained all the models with a 70∼30% split of the training dataset to the testing dataset. In doing so, Table 1 shows the solar array detection accuracy comparison of 5 different recent approaches in a "hold-out" manner.

2.1 Net Meter-based Approach

Prior approaches [13, 17, 25, 26, 26] typically leverage statistical learning, machine learning and other data analytical techniques to train accurate machine learning classifiers. These approaches require significant amount of historical pure solar generation data, which may not be available due to the new solar sites become online, to calibrate their models, and also have the limitation of distinguishing solar PV array profiling from other solar degradation, e.g., shading, dust, snow, cloudy, and etc. In addition, smart net meter traces are typically not publicly available and do not actually scale up to all the locations. In real practice, there is always a delay when new solar site become "online" due to the waiting time for the city permit and inspection process.

2.2 Machine Learning-based Approach

Prior approaches [18, 21] leveraged ML models to identify solar PV arrays from very high resolution (VHR) rooftop satellite images. These VHR images typically have a resolution of 0.3 meters per pixel and 8-bit in each RGB color channel. The insight of these ML-based approaches is that solar PV arrays have unique physical shape features that allow us to train a ML classifier to predict the existence of solar PV arrays in a VHR image. The major challenge of these approaches is to empirically identify these unique shape features using VHR images. The researchers in work [21] used 100 manually selected VHR images and empirically extracted principal features, including: prescreened confidence in foreground color, color histogram of background pixels, the ratio of $area$ to $perimeters^2$, and the mean, variance and Kurtosis of the grayscale pixels per region. Then, the researchers leveraged the support vector machine (SVM) classifier to perform the binary classification. A later work in [18] demonstrated a detection approach based on the similar insight as in work [21], but leveraged random forest (RF) classier to train their classification model.

Observation: Our results show that the prior ML-based approach based on kernel of logical regression (LR), support vector machines (SVMs) and random forest (RF) yields a very low MCC of 0.29, 0.42 and 0.29, respectively. These ML-based approaches are reporting better True Positives (TP) and worse True Negatives (TN) than the CNNs approaches. This is mainly due to the fact that although physical color and shape features are very effective when describing solar arrays, these features cannot enable significant distinguishability between solar arrays and trees, shades or shadows.

2.3 Deep Learning-based Approach

Significant recent research focused on using visual geometry group networks (VGGNets) based deep convolutional neural networks (CNNs) techniques [7, 19] to automatically detect rooftop solar arrays using satellite images. The VGGNet architecture [27] is designed to significantly increase the depth of the existing architectures of CNNs with 16∼19 layers using very small 3×3 convolution filters. Since VGGNet is substantially deeper than the other CNN models, the VGGNet is more susceptible to the vanishing gradient problem and applicable to other image recognition datasets [4, 7, 19]. Broadly, these techniques all require a significant amount of training data including very high resolution (VHR) imagery (0.3∼0.8m/pixel) and human handcrafted image templates to train their models.

For instance, the authors in [7] proposed a five layers CNN that includes three convolutional layers and two fully connected layers. The inputs are 3,347 three-channel satellite images with a size of 200×200 pixels. The first convolutional layer is comprised of neurons that connect to the subregions of the input images and has 96 maps and each map contains 64×64 neurons. The second convolutional layer contains 256 feature maps with 17×17 neurons on each map. The third convolutional layer has 120 feature maps with the size of 13×13 neurons per map. The major problem with this approach is that the input rooftop satellite images have many "outliers" rather than solar arrays, and the CNN model is not able to reliably identify them.

The work in [19] is an improvement to the approach in [7]. Their architecture has two different modules, including VGG(x) modules, and fully connected neuron FC(y) modules. The CNN is comprised of two consecutive convolutional networks, and each has x filters that are 3×3 pixels in size. And each convolutional layer is followed by a rectified linear unit (ReLU) activation. The last part of the CNN model is a 3×3 pixels max-pooling layer, with a stride of 2 pixels. The training dataset encompasses 135 km^2 of surface and has 2,794 PV array annotations. All the training data and ground truth data are handcrafted by human annotators. Another work [20] is a variant of the CNN model in the work in [19] and employs Random Forest modeling to benchmark the performance of the proposed CNN model.

Observation: Our results show that the VGGnet-based CNN approach yields a MCC of 0.21 as reported in Table 1. The CNN approach is reporting the True Positives percentage of 14.41% which is ~80% worse than the ML-based approach using random forest (RF) classifier, yielding at a True Positives percentage of 99.55%. This is mainly due to the CNN approach possibly not being able to reliably distinguish solar arrays from other rooftop objects (e.g. shading generated by nearby tall buildings and trees) that have similar features as solar arrays. Another significant drawback is the general CNN approach requires very high resolution (VHR) satellite images that are not available at every location.

2.4 Hybrid Approach

The most notable prior approach research—SolarFinder [15] leveraged a linear regression model to combine the benefits from both ML-based and DL-based approaches. The key insight of this "hybrid" approach is that ML-based approaches are more accurate when identifying solar PV arrays, while, DL-based approaches are performing better to detect other rooftop outliers. In particular, this hybrid approach requires two-step operations to identify solar PV arrays, including rooftop image object segmentation and contour level solar PV array detection. SolarFinder's detection accuracy is highly limited by the accuracy of the KMeans-based rooftop object segmentation process. SolarFinder also assumes $k = 5$ clustering is working for all the rooftops. As shown in Figure 1 (a) and (b), some shadow generated by near trees is recognized as solar PV array by mistake. In addition, Figure 1 (c) and (d) show that both of shades and ridges are wrongly classified as solar PV arrays in the "clustering" process. This is mainly due to the fact that the RGB grayscales and shapes of solar PV arrays, shades, and trees are quite similar and thus KMeans-based segmentation cannot accurately and reliably segment the rooftop objects. To make the detection performance even worse, rooftop object segmentation removes the relationships between different objects and the rooftop. For example, people typically cannot deploy solar PV arrays on the edges of a rooftop. In contrast, shades generated by trees are more likely aligning well with the rooftop edges. The final linear regression model is not able to pick up these critical missing information at contour level to detect solar PV arrays. Also, as shown in Figure 2, since this hybrid approach detects solar PV arrays in the separated contours, it is extremely difficult to accurately report multiple solar PV arrays on one rooftop.

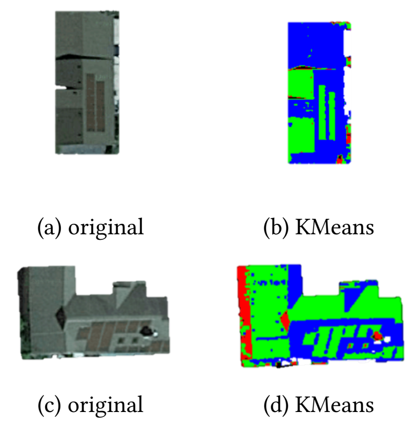


Figure 1: Illustration of inaccurate Rooftop segmentation when applying the most notable approach—SolarFinder [15]. Each color represents one cluster.

Observation: Our results show that the hybrid approach—SolarFinder [15] yields a MCC of 0.61 as reported in Table 1, which is ~2 times better than the most recent DL-based approach yielding at a MCC of 0.21. However, this hybrid approach still shows very low detection accuracy. This is mainly due to the fact that the separation of rooftop object segmentation and detection processes in SolarFinder has resulted in the missing of rooftop contextual information, which is critical when detecting solar PV arrays. In addition, this hybrid approach cannot accurately report multiple solar PV arrays cases since its final linear regression detection is fully performed at image contours.

2.5 Summary

In summary, prior ML-based approaches [18, 20, 21] show high True Positives which indicate the potentials to identify solar PV arrays, while, prior DL-based approaches [7, 19, 28] demonstrate high True Negatives, which illustrate their "promises" in detecting other rooftop objects rather solar PV arrays. In addition, prior ML-based and DL-based approaches are usually trained using VHR images which are costly and not available at every location. Thus, both of ML-based and DL-based approaches cannot scale up in real practice. The hybrid approach which aims at combining the benefits from both ML-based and DL-based approaches yields the highest MCC, which demonstrates the best solar PV array detection performance. However, the two-step detection process of the hybrid approach has limited its detection performance. Specially, the separation of rooftop object segmentation and detection processes has caused the missing of rooftop contextual information, which is critical for final solar PV arrays detection. Thus, this hybrid approach is still suffering low detection accuracy. Also, since most prior ML-based, DL-based and hybrid approaches are detecting solar PV arrays at the image contour level, none of them can accurately and reliably report multi-panel solar deployments. In addition, all the prior work are suffering low efficiency in their training and retraining costs. We also quantify the detection model training time. We find that SVMs

Figure 2: An example of multiple panels on a rooftop.

require 45.23 seconds per Megabyte (MB) training data to learn a reasonably accurate detection model. While it takes CNNs-based approach 45.29 seconds per MB to train a solar PV array detection model. Hybrid approach requires the longest training time of 90.69 seconds per MB to train a solar PV array detection model due to its combining nature. These valuable insights will guide our design of SolarDetector.

3 CHALLENGES

In this section, we describe the major challenges that we met when designing for our new approach -SolarDetector that can automatically detect and profile distributed solar photovoltaic arrays in a given region.

Insufficient features learning. Prior work mainly developed ML/DL/hybrid classifiers using unbalanced and insufficient rooftop satellite image samples. Thus, these ML-based, DL-based and hybrid approaches cannot observe and identify all the principal features to train their models. In particular, as shown in Figure 1, SolarFinder and other prior approaches that rely on K-Means clustering algorithm to segment rooftop objects cannot reliably distinguish solar panels from shady roofs and ridges. This is due to the fact that the extacted features of solar panels and other roof objects are quite similar at RGB gray-scale levels. Also, the shape and RGB gray-scale characteristics of rooftop solar PV arrays may vary from different manufacturers. To address this issue, SolarDetector leverages multiple data augmentation techniques (including DCGANs) to build a big balanced rooftop satellite image dataset, which can help to extract more significant features and learn more accurate classifiers.

Multiple panel detection. Another challenge is to report the multi-panel solar PV array deployment, as shown in Figure 2. Since recent approaches performed their solar PV array detection on contours rather than images, they cannot (accurately) report multi-panel solar PV array deployments. However, the profiling information of multi-panel solar PV deployments has become more and more critical for government agencies and utilities to accurately manage the solar PV installation cap in a given region. To address this issue, SolarDetector leverages Mask R-CNN based approaches, which can report multiple solar PV array instances simultaneously, to accurately report multi-panel solar PV array deployments. More details can be found in our evaluation section.

Separation of segmentation and detection. Most recent ML-based and DL-based approaches rely on rooftop object segmentation as their first step to detect rooftop solar PV arrays. However, as shown in Figure 1, this segmentation process is not reliable. The outputs of this rooftop object segmentation process are contours, which are the small pieces of rooftop images. The second step of these approaches are typically examining each contour to see if its features are similar to the previously learned principal features. The final solar PV array detection process is built on top of the rooftop object segmentation errors. In addition, the separation of rooftop object segmentation and detection processes has resulted in the missing of rooftop contextual information (e.g., distance from an object to roof edges), which is critical when detecting solar PV arrays. To address this issue, SolarDetector performs rooftop object segmentation and solar PV array detection at image level simultaneously to reserve the rooftop contextual information. More details can be found in our design section.

Non-integrated data processing engines. Recent approaches are suffering low performance on their machine learning or deep learning model training and retraining. To address this issue, SolarDetector integrates with large-scale data processing engine—Apache Spark and leverages graphics processing units(GPUs) to further improve deep learning and machine learning model training and retraining costs.

4 DESIGN

In addressing the above-mentioned challenges, we design a new system approach—SolarDetector that can accurately detect distributed solar arrays automatically in a given geospatial region without any extra cost. SolarDetector first leverages data augmentation techniques and Generative adversarial networks (GANs) to build a large rooftop solar PV array satellite images that can enable us to learn features and parameters of solar PV array detection models more accurately. Then, SolarDetector employs Mask R-CNN-based algorithm to accurately identify rooftop solar arrays and also learn the detailed installation information for each rooftop solar panel deployment simultaneously. SolarDetector will integrate with Spark and GPUs to improve the training cost of different models. Figure 3 shows the SolarDetector pipeline of operations.

4.1 Extracting Rooftop Satellite Images

As shown in Figure 4, some houses and trees are predicted as solar arrays. Thus, we may not able to accurately distinguish solar PV arrays from other objects directly from the unprocessed regional satellite image using empirically extracted features. This is due to the fact that the extracted features of solar arrays, trees, and houses are quite similar at RGB gray-scale levels. Also, the shapes of houses are similar to the shapes of solar arrays, most of them are rectangular.

To address this issue, we propose first to segment rooftops in a given satellite image. We leverage the reversed satellite image fetching API approach that was presented in [15]. In essence, we use publicly-available maps APIs, including Google Maps [10] and OpenStreetMap [22] to fetch and extract the rooftop satellite images. Similar to [15], the input of the approach is a region satellite and output is the segmented rooftop satellite images. Given a set of target regions, we first collect all the residential building rooftop polygon information using OpenStreetMap API. The return of OpenStreetMap API is the OSM file that contains the profiling information for all the objects. Then, we recover the whole rooftop polygons using those nodes' information and then feed them into the Google Maps API that returns satellite imagery. Note that, unlike prior work in which rooftop segmentation are performed either per house or per contour levels, SolarFramework could extract all

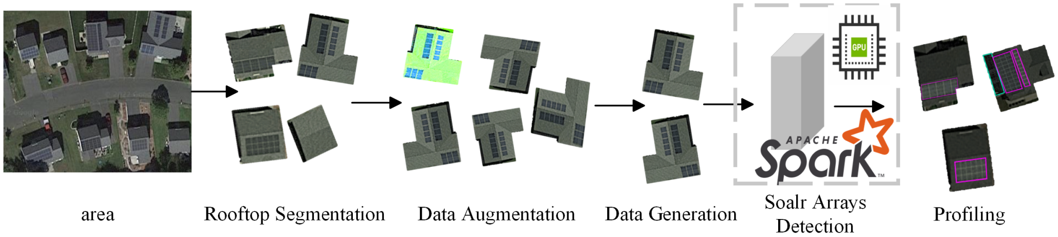
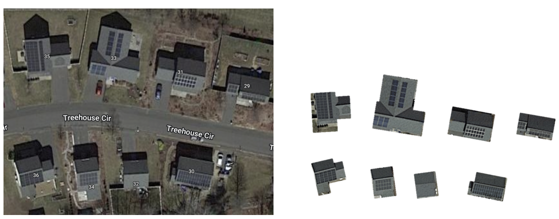


Figure 3: The pipeline of operations SolarDetector.

Figure 4: Detecting solar arrays from multi-houses



(a) Before Segmentation (b) After Segmentation

Figure 5: Sample of a regional Rooftop Extraction.

the rooftops within the region simultaneously. Figure 5 shows an example of this whole region rooftop extraction.

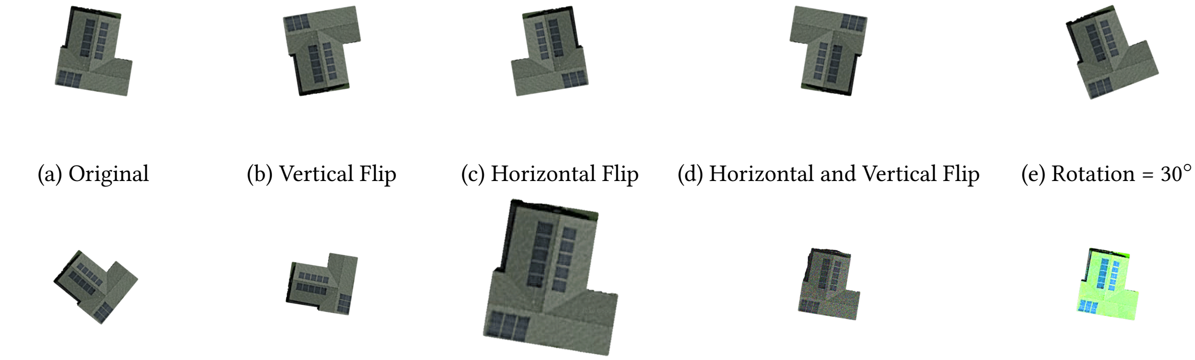
4.2 Preprocessing Rooftop Satellite Images

As discussed in the Related Work section, prior approaches are suffering low detection accuracy due to insufficient sample and feature learning when building their models. To address this challenge, we leverage data augmentation and Generative Adversarial Networks (GANs)-based approach to automatically generate more samples that can enable us to learn features and parameters of solar PV array detection models more accurately. The key insight for this rooftop image augmentation is that different homes may have different solar PV array installation characteristics, including size, orientation, tilt and shade. To mimic different solar PV array deployments, we propose to use multiple augmentation approaches to further process the extracted rooftop images. Figure 6 shows the detailed multiple data augmentation techniques, including vertical and horizontal flipping, multi-degree (including $5^{\circ}\sim 90^{\circ}$) rotating, cropping, adding noise and increasing brightness, which enable SolarDetector to more efficiently identify principal features for ML/DL models. In addition, throughout these image data augmentation approaches, we also exaggerate solar PV array principal features that can better train our final solar PV array detection models. Note that, the major reason that we design DC-GAN model is to generate a larger dataset which enables us to identify unseen samples and features of different solar site configurations (e.g., different tilts, orientations, sizes). In addition, data augmentation approaches that SolarDetector uses is orthogonal to the other aspects of the technique and is thus "pluggable," such that we could use other computer vision approaches to perform data augmentation operations here. We will include new becoming online models in our near future work.

The above-mentioned image augmentation has provided more solar PV array deployment samples to train solar PV array detection models. However, these additional samples may still not be sufficient to train reasonably accurate DL-based approaches that typically require significant amount of well-balanced image samples to calibrate their models and optimizations. To address this issue, we propose to leverage Deep Convolutional Generative Adversarial Networks (DCGANs) to further generate a large solar PV array training dataset.

To present the rooftop image more effectively, we leverage generative adversarial networks(GANs) [9] architecture to build our solar PV array image generator. However, recent work [32] has shown that GANs model has some performance limitations. For instance, GANs might be unstable to train and thus resulting in generators that produce nonsensical, noisy, and incomprehensible new artificial images. The recent work [24] presented a new GANs—Deep Convolutional GANs (DCGANs) that has mitigated these issues by replacing the deterministic pooling function to strided convolution and using Rectified Linear Units (ReLU) activation in all generator layers, and leaky rectified linear unit (Leaky ReLU) activation in all discriminator layers. DCGANs has become the standard architecture to solve image generation problems. Figure 7 shows the design of our DCGANs module. The inputs of our module are solar PV array satellite images after applying those different data augmentation approaches. The outputs are the generated big artificial satellite image dataset. More importantly, users cannot reliably distinguish the generated satellite images and the original satellite images. This will ensure the model learning accuracy using our generated big satellite image dataset.

Our DCGANs module has two models—Generator and Discriminator. The generator output is connected directly to the discriminator input. The two models almost have the same architecture, but reflected. Our DCGANs architecture is composed of convolutional layer without max pooling or fully connected layers. Our DCGANs model leverages convolutional stride and transposed convolution for downsampling and upsampling, respectively. The generator model uses a $1 * 100$ noise vector, which are followed by five convolutional layers. For the generator activation function, we use Tanh activation function for the last layer. For the rest of the layers, we



(a) Original (b) Vertical Flip (c) Horizontal Flip (d) Horizontal and Vertical Flip (e) Rotation = 30°

(f) Rotation = 60° (g) Rotation = 90° (h) Crop (i) Noise (j) Brightness

Figure 6: Data augmentation samples

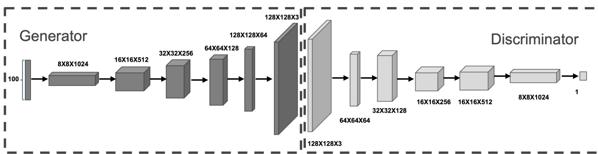


Figure 7: The pipeline of our DCGANs system structure.

leverage ReLU activation function. For the discriminator activation function, we use the Sigmoid activation function for the last layer. For the rest layers, we leverage Leaky ReLU activation function.

The rooftop satellite image preprocessing enables SolarDetecor to learn more accurate solar PV array principal features and train solar PV array detection models using a well-balanced big training satellite image dataset. In doing so, we can observe substantially more positive and negative exaggerated samples than we could have had access to. Figure 8 shows some samples of the generated rooftop satellite images using our DCGANs at 4,100 epochs.

4.3 Detecting Rooftop Solar PV Arrays

As we discussed in Section 3, the separation of rooftop object segmentation and detection processes has resulted in the missing of rooftop contextual information, which is critical when detecting solar PV arrays. To address this issue, SolarDetector leverages Mask R-CNN [11] to perform rooftop object segmentation and solar PV array detection at image level simultaneously to reserve the rooftop contextual information.

Figure 9 shows the architecture of our Mask R-CNN design. The input of the Mask R-CNN model is the whole rooftop satellite images. Our mask R-CNN has two stages. The first stage is to build region proposal network, which can generate region proposals of solar PV arrays. The second stage is building the process for solar PV arrays detection and segmentation. In addition to object classification, we can also identify bounding box locations and generate grayscale solar PV array masks. This additional information will also enable us to profile rooftop objects. Next, we will explain the detailed processes that we use Mask R-CNN to detect and profile solar PV array deployments.

4.3.1 Extracting Feature Maps. Our Mask R-CNN model first extracts feature maps from rooftop satellite images. In essence, we leverage ResNet50 Feature Pyramid Networks (FPN) to extract satellite images feature maps. ResNet50 is a 50-layer deep convolutional neural network (DCNN), it stacks residual blocks on top of each other to form a network. This ResNet50 FPN architecture can help us extract accurate solar PV array features at multiple layers and scales. Meanwhile, ResNet50 FPM may be suffering vanishing or exploding gradient issues. We address this issue by using residual blocks that can skip connections. A skip connection is an alternate shortcut between multiple stacked layers, which can mitigate the vanishing or exploding gradient problems of ResNet50 FPM.

4.3.2 Extracting Region Proposals. After extracting satellite images feature maps, SolarDetector then focuses on extracting region proposals. As shown in Figure 9, SolarDector feeds the extracted features into Region Proposal Networks (RPN). The RPN can classify multiple rooftop objects and probability of the objects classification simultaneously. More importantly, the RPN can also predict region proposals for different objects at multiple scales and aspect ratios, which will ensure rooftop objects of different sizes are accurately classified. The output of RPN network are the generated candidate region proposals for all rooftop objects. In particular, those high-probability region proposals—Region of Interest(RoIs) will be forwarded to the next stage in SolarDetector's system pipeline.

4.3.3 Detecting and Segmenting Instances. As shown in Figure 9, all the ROIs will be fed into the Region of Interest Align (RoIAlign) layer, which is an operation for extracting a small feature map from each RoI in detection and segmentation. RoIAlign can align the extracted features with the input rooftop satellite images. The RoIAlign layer is followed by three additional layers, including rooftop objects classification layer, bounding box layer, and mask

Figure 8: Sample generated by DCGANs

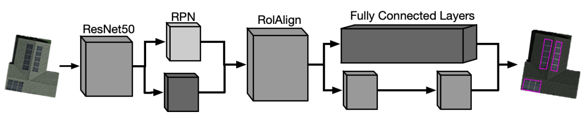


Figure 9: The architecture of our Mask R-CNN design

prediction layer. Eventually, rooftop object classification and bounding box generation are both achieved by connecting the fully connected layer. The object mask prediction is connected through convolutional layer and deconvolutional layer.

4.3.4 Multi-thread and multi-process detection models. In addition, we design the detection models of SolarDetector in a multi-thread and multi-process manner, which can enable SolarDetector to cooperate with big data processing engines, including both Apache Spark and GPUs, to further accelerate the (re)traing process. Note that, although we have demonstrated that SolarDetector(with Mask R-CNN) yields a higher MCC and much shorter (re)traing time, SolarDetectorcan also work with other new detecting models.

4.4 Post-processing Solar PV Array Detection

In addition to detecting solar PV arrays and other rooftop objects, SolarDetector can also profile them. The solar PV array profiling information may include size, orientation, multi-panel status, shade situation, etc. For instance, to report solar PV array size, SolarDetector examines the number of pixels that are included in the identified solar PV arrays. Since each pixel denotes an area with a size of S km^2, where S can be derived from satellite image zoom level (typically 20) and its location on rooftop. SolarDetector first simply multiplies the pixel size by the number of pixels in a solar PV array instance. Then, SolarDetector performs a union operation to add up all the solar PV array instances on the same rooftop to report the final solar PV array size. Similarly, SolarDetector can also get the number of solar PV arrays on a rooftop by using the masks prediction results from Mask R-CNN model (shown in Figure 9). Similarly, SolarDetector can also detect and profile other rooftop objects, such as trees, chimneys, windows, and shadows. Note that the shadow, tree, or window information can not be learned from the energy meter data directly. These additional profiling information will enable the utilities (and their third parties) and the government agencies to better predict and manage the distributed rooftop solar generated energy.

5 IMPLEMENTATION

We implement SolarDetector in `python` using widely available open-source frameworks, including OpenCV, Scikit-learn, PyCUDA, TensorFlow, and PyTorch. SolarDetector leverages OpenStreet Maps API [22] to fetch the rooftop location and Google Maps API [10] to fetch satellite rooftop images. We use OpenCV, NumPy and Pandas for images processing based data augmentation. We leverage PyTorch framework to implement DCGAN generator to generate rooftop images. We implement Mask R-CNN detection and segmentation models on top of PyTorch frameworks. We leverage Opencv and Numpy to further profile rooftop object information. Eventually, we schedule the batch jobs on our GPU servers to compare the MCC accuracy of different solar PV array detecting approaches using CUDA. The server that we use to get all the benchmarking and evaluation results has resources as follows: 1) CPU: Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz GPU: nVidia TITAN V RAM: 128GB, 4) OS: Ubuntu 18.04.4 LTS.

6 EXPERIMENTAL EVALUATION

Below we describe our dataset, experimental setup, metrics used to evaluate our approach, and evaluation results.

6.1 Datasets

Dataset A. We downloaded ~260,000 publicly available satellite images from 13 geospatial regions of 9 different states in the U.S., including Colorado, California, Massachusetts, Minnesota, Arizona, Maryland, Wisconsin and Washington from SolarFinder Dataset [30]. The dataset includes 13 regions with a radius ranging from 5km to 20km. The ratio of solar array rooftop to non-solar array rooftop is ~ 1:100. The dataset also has groundtruth of each satellite rooftop image, including whether there is solar arrays on the rooftop and polygon outlier information for solar PV arrays.
Dataset B. We collected ~3,430 publicly-available solar PV array rooftop images using Google Maps API. The ratio of the solar array rooftop to non-solar array rooftop is 1:5. Given a rooftop listed in the datasets, we prepared the groundtruth for rooftop objects, including windows, shadow, chimneys, trees and solar PV arrays. Besides the objects class information, we also prepared the polygon outlier information for each object using OpenStreet Maps API [22]. Then we augmented the training dataset B using image processing based techniques to 5,725 satellite rooftop images. To collect solar PV arrays of all orientations, we leverage OpenCV to randomly rotate satellite rooftop images clockwise.

6.2 Experimental Setup

We implement and evaluate four different solar PV array detection approaches in two different manners, including cross validation and hold-out validation.

Cross Validation. In this case, all of the SVMs, Random Forest, Logistic Regression, CNNs, SolarFinder and SolarDetector approaches can access to the satellite rooftop images from their testing sites. In this way, we are benchmarking the best performance of these 6 different approaches.

Hold-out Validation. In this case, all of the SVMs, Random Forest, Logistic Regression, CNNs, SolarFinder and SolarDetector approaches can not access to the satellite rooftop images from their testing sites. In this way, we are benchmarking the real performance of the 6 different approaches.

6.3 Evaluating Metrics

Matthews Correlation Coefficient (MCC). To quantify the accuracy of different solar PV array detection approaches, we note that the standard evaluating metrics, e.g, accuracy, F1, would not work well on our highly imbalanced data. And this observation has been studied by researchers in work [1]. Most solar PV array dataset is highly imbalanced, the ratio of solar array rooftop to non-solar arrays rooftop is 1:100. Based on the recommendation from prior work [1, 23], we use the Matthews Correlation Coefficient (MCC), a standard measure of a binary classifier's performance, where values are in the range −1.0 to 1.0. With 1.0 being perfect solar PV arrays detection, 0.0 being random solar PV arrays prediction, and −1.0 indicating solar PV arrays detection is always wrong. The expression for computing MCC is below, where TP is the fraction of true positives, FP is the fraction of false positives, TN is the fraction of true negatives, and FN is the fraction of false negatives, such that TP+FP+TN+FN= 1. The MCC is a more reliable statistical rate that produces a high score if the prediction obtained good results in all of the four confusion matrix categories: true positives, false negatives, true negatives, and false positives. MCC is preferred over other scores (e.g., F1 score) as it is a more "balanced" assessment of classifiers, no matter which class is positive.

$$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{1}$$

Intersection of Union (IoU). To quantify the accuracy of SolarDetector to predict size for solar PV arrays, we use Intersection of Union (IoU) which is widely used in prior work to measure the similarity between the detected region and the groundtruth region. As a measure of similarity for the two sets of pixel data, with a range from 0% to 100%. The higher the percentage, the more precise predictions that SolarDetector can do. It can be defined as follows,

$$IoU = \frac{x \bigcap r_g}{r_d \bigcup r_g} \tag{2}$$

where r_d denotes the detected region for a solar PV array, and r_g indicates the groundtruth region for a solar array.

Mean Orientation Error (MOE). To quantify the accuracy of SolarDetector to predict orientations for solar PV arrays, we employ the mean orientation error (MOE) that is introduced in a recent work [14]. The MOE captures the per-pixel error between the predicted and the actual azimuth angle. It is defined as follows,

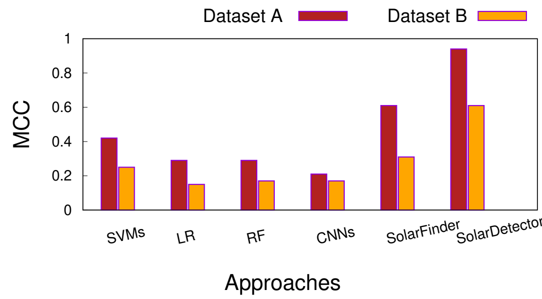


Figure 10: The comparison of detection accuracy when employing SVMs, Logistic Regression, Random forest, CNNs, SolarFinder, and SolarDetector approaches using Dataset A and Dataset B.

$$MOE = \frac{1}{C} \cdot \sum_i \frac{\sum_j p_{ij} \cdot Azimuths\_differ(o_i, o_j)}{t_i} \tag{3}$$

where C is the total number of classes (i.e., azimuths), o_i and o_j is the azimuth angles, and p_{ij} indicates the number of pixels of azimuth j reported as azimuth i, and t_i is the total number of pixels in class i. In addtion, $Azimuths\_differ$ is a function that return the difference between two azimuth angles. The MOE should return a value between 0° (perfect estimation) and 180° (opposite estimation).

6.4 Experimental Results

6.4.1 Quantifying Solar PV Array Detection Accuracy. We first compare SolarDetector with SVMs, Random Forest, Logistic Regression, CNNs, SolarFinder, and our SolarDetector approaches using two satellite images datasets—Dataset A and Dataset B. Unsurprisingly, as shown in Figure 10, SolarDetector is the best performing solar PV arrays detection approach on both datasets. For Dataset A, we can observe that SVMs approach, Logistic Regression approach and Random Forest approach yields a MCC of 0.42, 0.29, and 0.29, respectively, this is due to the fact that these ML-based approaches report very low True Negative percentages as shown in Table 1. We can also find that CNNs yields MCC of 0.21, this is due to CNNs report very low True Positive percentages. The hybrid approach—SolarFinder which combines the benefits of both machine learning approach and CNNs yields a better MCC of 0.61. Our new SolarDetector yields the best MCC of 0.96. which is ∼ 50% higher than SolarFinder. This is mainly due to the fact that SolarDetector addresses the three major challenges (in Section 3) that SolarFinder and other ML/DL approaches are currently suffering.

For Dataset B, we can observe that Logistic Regression approach reports the worst MCC of 0.17, this is mainly due to that Logistic Regression approach typically reports very low True Negative percentages. While, SVMs approach, Random Forest approach, and CNNs approach yields a MCC of 0.25, 0.17, and 0.17, respectively. SolarFinder yields a MCC of 0.31, which is better than ML-based approaches and CNNs approaches. Unsurprisingly, SolarDetector yields the best MCC of 0.61, which is constantly ∼ 50% better than the most notable approach—SolarFinder.

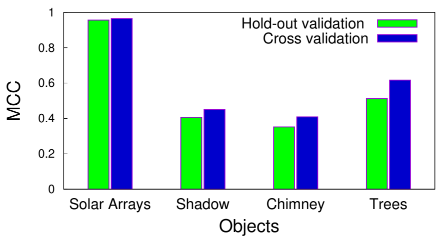


Figure 11: The accuracy comparison of rooftop object detection when SolarDetector applying cross validation and hold-out validation.

Results: Comparing with SVMs, Random Forest, Logistic Regression, CNNs approaches and SolarFinder approaches, SolarDetector is the always best detecting approach. In particular, SolarDetector yields an average MCC of 0.76, which is ∼ 50% better than the most notable approach—SolarFinder.

6.4.2 Comparing cross validation and hold-out validation. Next, we will examine the completeness and robustness of SolarDetector. Figure 11 shows that SolarDetector with cross validation yields overall slightly better detection accuracy than SolarDetector with hold-out validation. This is mainly due to the fact SolarDetector can access to information from the training satellite imagery dataset and thus can better calibrate its modeling learning process. For all the hold-out evaluations, SolarDetector does not have this access to training data to fine-tune its model weights, and thus reports slightly lower MCCs for each object. In particular, among all the rooftop objects, SolarDetector report almost the same MCC—0.96 for cross validation and hold-out validation. This is mainly because SolarDetector leverages multiple data augmentation approaches to more comprehensively learn principal features. This will enable SolarDetector to be trained using existing datasets and then detect solar PV arrays in new regions.
Results: SolarDetector with cross validation yields overall slightly better detection accuracy than SolarDetector with hold-out validation. For solar PV array detection, SolarDetector demonstrates strong completeness and robustness of its modeling process and thus reports almost the same MCC for both cross validation and hold-out validation.

6.4.3 Quantifying Other Rooftop Object Detection Accuracy. Next, we compare SolarDetector detection accuracy of other rooftop objects. As shown in Figure 11, similar to solar PV arrays detection, SolarDetector can also accurately detect other rooftop objects. For the cross validation evaluations, solar PV array detection yields the best MCC of 0.956. SolarDetector can also detect other objects such as shadows, chimney, trees with MCCs of 0.407, 0.352, and 0.512, respectively. Note that, most of prior approaches are not able to detect or report these objects. SolarDetector reports slightly worse MCCs when detecting trees, shadows and chimneys. This is mainly due the fact that the shape of shadows generated by tall trees or buildings is a function of the solar PV deployment's location, Sun's position in the sky , time of a day, and also local weather conditions (e.g., wind). And this information currently is not collected or provided with Dataset A and B. We plan to combine this information with our model learning in our future work, which is out of scope of this work. In addition, another reason for the slightly worse detection on chimneys is that SolarDetector does not have sufficient samples to learn the principal features to identify chimney from rooftop satellite images in Dataset A and B.
Results: In addition to detecting solar PV arrays, SolarDetector is able to detect other rooftop objects (e.g. chimney, shadow, trees) in satellite images with reasonable accuracy.

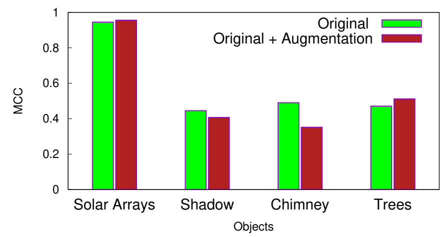


Figure 12: The detection accuracy comparison of SolarDetector (with hold-out validation) with and without applying data augmentation.

Model	Objects	MCC	IoU	Orientation
SolarDetector	Solar PV arrays	0.956	0.713	1.68
SolarDetector	Shadows	0.407	0.603	10.72
SolarDetector	Trees	0.512	0.712	3.94
Fast RCNN	Solar PV arrays	0.932	0.568	2.13
Fast RCNN	Shadows	0.442	0.463	12.37
Fast RCNN	Trees	0.456	0.511	4.26
Faster RCNN	Solar PV arrays	0.935	0.572	2.13
Faster RCNN	Shadows	0.389	0.265	15.42
Faster RCNN	Trees	0.459	0.513	4.31
SegNet	Solar PV arrays	0.714	0.306	3.36
SegNet	Shadows	0.103	0.174	17.42
SegNet	Trees	0.246	0.192	7.78

Table 2: The categorized comparison of detection and profiling accuracy of different objects.

6.4.4 Quantifying Data Augmentation Performance. Figure 12 shows the rooftop object average detection accuracy of SolarDetector with and without applying data augmentation approaches (discussed in Section 4) over 10 different regions with slightly improvements. Unsurprisingly, with applying data augmentation approaches, SolarDetector reports better MCCs when identifying solar PV arrays and tress. This is mainly because data augmentations help SolarDetector to address insufficient training samples and principal feature learning issues. While, for shadows and chimneys, SolarDetector is reporting worse MCCs when applying data augmentation approaches. This is mainly because our current data augmentation approach does not fully consider the rooftop contextual information. For instance, chimneys typically is built on either left or right side of a rooftop. Another example, shadow shapes are changing all the times and can be aligning well with the edges of roofs. We plan to explore these improvement opportunities in our future work.
Results: SolarDetector yields better accuracy for detecting solar PV arrays and trees when applying data augmentation approaches.

Augmentation	MCC
No	0.701
Yes	0.704

Table 3: The comparison of multi-panels accuracy.

6.4.5 Profiling Rooftop Objects. Next, we will examine SolarDetector's ability to profile detected rooftop objects. We use Intersection of Union (IoU) as the evaluation metric to examine the reported size of solar PV arrays, shadows, and trees. The shadow from nearby trees or tall buildings, which is the key parameter of solar performance and forecasting models, cannot be directly learned from the energy meter data. And recent research [3, 5, 6, 16] has shown that these information could significantly affect solar PV deployment generation performance and site surveys. Table 2 shows the MCC and IoU results when detecting solar PV arrays, shadows, and trees using SolarDetector, Fast RCNN, Faster RCNN, and SegNet, respectively. SolarDectector yields IoTs as 0.713, 0.602 and 0.712, respectively. Interestingly, although SolarDetector only reports MCCs as 0.407 and 0.512 for shadows and trees detection, SolarDetetor yields very high IoUs. This is mainly because grayscale features of shadows and tress are so significant and have less variance for SolarDetector to extract them. We also observe that SolarDetector can yield the MOE as 1.68, which can accurately report orientation of solar PV arrays. These profiling information is critical input information for solar energy generation prediction models and solar generated energy cap managements.
Results: SolarDetect can also accurately profile detected rooftop objects, such as the size and orientation of solar PV arrays and the size of trees and shadows.

6.4.6 Quantifying SolarDetection multi-panels detection accuracy. Eventually, we employ IoU to quantify the accuracy for SolarDetector to predict the multi-panel rooftop solar PV array deployments. Since our Mask R-CNN model can generate pixel level solar PV array masks, SolarDetector can learn the number of panels by the number of the masks. As shown in table 3, with data augmentation multi-panel prediction average accuracy gets slightly better than without augmentation with a IoU of 0.704 over 10 different regions.
Results: SolarDetector can also accurately report multi-panel rooftop deployments.

6.4.7 Quantifying Solar PV Array Detection Training Cost using Spark. We first compare the training costs of SVMs, CNNs, and Hybrid approach (Linear Regression) using Spark. Compared with the training cost in Table 4, after applying Spark (with 10 cores CPU server) on each model, we find that SVMs, CNNs and Hybrid approach yield a training cost as of 1.372, 0.568 and 3.118 (measured in second per MB training data), respectively.
Results: After using Spark, we find that SolarDetector significantly cut the training cost (equivalent to $\sim$ 100 times faster) for SVMs, CNNs, and Hybrid solar PV array detecting approaches.

6.4.8 Quantifying Solar PV Array Training Cost using GPUs. We next examine the training cost using multiple GPUs acceleration. We implement different models using multiple threads programming. Note that, SVMs approach can not be schedule on GPUs due to their model design nature. Unsurprisingly, SolarDetector (with

Model	Training Time (seconds/MB)
SVMs (Spark)	1.372
Pure CNNs (Spark)	0.568
Hybrid (Spark)	3.118
CNNs (1 GPU)	12.804
CNNs (2 GPUs)	8.968
Mask R-CNN (CPU)	109.742
Mask R-CNN (GPU)	26.832

Table 4: The performance comparison of different models using SolarFramework

.

GPU) accelerates the training time of Mask R-CNN by 5 times as of SolarDetector (with CPU).
Results: After using GPUs, we find that SolarFramework significantly cut the training cost (up to $\sim$ 5 times faster) for CNNs-based and Mask R-CNN-based detecting approaches.

6.4.9 Quantifying the Cost of using Spark and GPUs. We are planning to host SolarDetector using our public GPU server using public APIs in near future. In addition, we also examine the cost if users would like to run SolarDetector on their own (cloud) servers. We compare against the cost of hardware resourese we used on SolarDetector on Amazon AWS and Google Colab platform. For Amazon AWS EC2 market, users can select the size, memory, CPUs, GPUs, and storage. With 1 GPU, 16 vCPUs, 64G RAM, the cost will be $1.204 per hour. With 2 GPUs, 32 vCPUs, 128G RAM, the cost will be $1.734 per hour. In comparison, Google Colab, which is widely being used by AI researchers community to train machine learning and deep learning models, is free with K-80 GPU and 12GB of RAM in total. To access faster GPU like NVIDIA TESLA T4 or P100, users can subscribe to Google Colab Pro for only $9.99 per month.
Results: SolarDetector is a low-cost and highly effective framework that can detect and profile rooftop solar PV arrays simultaneously.

6.5 Real-world Applications

Marketing for Solar Installation. The information about solar panels in an area can be used by solar panel companies (e.g., Sun Power, Vivint, SunLux Energy, Sungevity) that offer solar panel systems lease, loan, and purchase options or homes and community properties to better market their products and offers. Solar panel installers usually cannot access residential energy meter data in a community and don't share information about solar panel installations. Thus, a prominent solar panel detection system such as our SolarDetector is highly desired for them.
Solar Panel Performance Diagnostics. Homeowners are increasingly deploying rooftop solar photovoltaic (PV) arrays due to the rapid decline in solar module prices. However, homeowners may have to spend up to ~$375 to diagnose their damaged rooftop solar PV system [16]. The profile information that SolarDetector could learn from a solar site can be used to assist the solar degradation diagnostics process. In particular, SolarDetector can help to evaluate inverter performance, which cannot be accurately learned from only net meter energy traces.
Solar Panel Installation CAP Management. The output of SolarDetector can be easily integrated with public maps APIs to create

a detailed visualization system for solar PV array deployments in an area. The administrative offices of smart cities and the utilities can use the new solar panel visualization system to better inform allocation decisions.

7 CONCLUSION AND FUTURE WORK

We design a new approach—SolarDetector that can automatically detect and profile distributed solar photovoltaic arrays in a given geospatial region without any extra cost. SolarDetector first leverages data augmentation techniques and Generative adversarial networks (GANs) to automatically learn accurate features for rooftop objects. Then, SolarDetector employs Mask R-CNN algorithm to accurately identify rooftop solar arrays and also learn the detailed installation information for each solar array simultaneously. We evaluate SolarDetector using 263,430 public satellite images from 11 geospatial regions in the U.S. We find that SolarDetector yields an average MCC of 0.76 to detect solar PV arrays over two big datasets, which is $\sim$ 50% better than the most notable approach—SolarFinder. Unlike prior work, we show that SolarDetector can also accurately report the profiling information for the detected rooftop objects. In addition, SolarDetector could also integrate with large-scale data processing engine—Apache Spark and graphics processing units(GPUs) to further improve its training cost.

We plan to collect more satellite solar array rooftop images to improve the accuracy of the detection model. We plan to improve the DCGANs model to include rooftop contextual information in data augmentation process. In addition to conventional data argumentation ways, we also plan to investigate other potentially better data augmentation approaches (e.g., mixup: Beyond Empirical Risk Minimization).

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and our shepherd—Dr. Venkatesha Prasad for providing us their insightful comments and valuable feedback, which significantly improved the quality of this paper.

REFERENCES

[1] Josephine Akosa. [n. d.]. Predictive accuracy: a misleading performance measure for highly imbalanced data.

[2] Fadi Al-Turjman and Mohammad Abujubbeh. 2019. IoT-enabled smart grid via SM: An overview. *Future Generation Computer Systems* 96 (2019), 579–590.

[3] Noman Bashir, Dong Chen, David Irwin, and Prashant Shenoy. 2019. Solar-TK: A Data-driven Toolkit for Solar PV Performance Modeling and Forecasting. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 456–466.

[4] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[5] Dong Chen, Joseph Breda, and David Irwin. 2018. Staring at the sun: a physical black-box solar performance model. In *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 53–62.

[6] Abderrahim Derdar, Nadia Bensiali, Mohamed Adjabi, Nadir Boutasseta, Mohammed Salah Bouakkaz, Issam Attoui, Nadir Fergani, and Ahmed Bouraiou. 2022. Photovoltaic energy generation systems monitoring and performance optimization using wireless sensors network and metaheuristics. *Sustainable Computing: Informatics and Systems* 35 (2022), 100684.

[7] Vladimir Golovko, Sergei Bezobrazov, Alexander Kroshchanka, Anatoliy Sachenko, Myroslav Komar, and Andriy Karachka. 2017. Convolutional neural network based solar photovoltaic panel detection in satellite photos. In *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 1. IEEE, 14–19.

[8] Vladimir Golovko, Alexander Kroshchanka, Sergei Bezobrazov, Anatoliy Sachenko, Myroslav Komar, and Oleksandr Novosad. 2018. Development of Solar Panels Detector. In *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*. IEEE, 761–764.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[10] googlemaps [n. d.]. googlemaps. https://developers.google.com/maps/documentation/maps-static/intro.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2980–2988. https://doi.org/10.1109/ICCV.2017.322

[12] Yuji HIGUCHI and Tadatoshi BABASAKI. 2018. Failure detection of solar panels using thermographic images captured by drone. In *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE, 391–396.

[13] Srinivasan Iyengar, Stephen Lee, Daniel Sheldon, and Prashant Shenoy. 2018. Solarclique: Detecting anomalies in residential solar arrays. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*. 1–10.

[14] Stephen Lee, Srinivasan Iyengar, Menghong Feng, Prashant Shenoy, and Subhransu Maji. 2019. DeepRoof: A Data-driven Approach For Solar Potential Estimation Using Rooftop Imagery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. ACM, New York, NY, USA, 2105–2113. https://doi.org/10.1145/3292500.3330741

[15] Qi Li, Yuzhou Feng, Yuyang Leng, and Dong Chen. 2020. SolarFinder: Automatic Detection of Solar Photovoltaic Arrays. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 193–204. https://doi.org/10.1109/IPSN48710.2020.00024

[16] Qi Li, Keyang Yu, and Dong Chen. 2021. SolarDiagnostics: Automatic damage detection on rooftop solar photovoltaic arrays. *Sustainable Computing: Informatics and Systems* 32 (2021), 100595.

[17] Andreas Livera, Marios Theristis, George Makrides, and George E Georghiou. 2019. Recent advances in failure diagnosis techniques based on performance data analysis for grid-connected photovoltaic systems. *Renewable energy* 133 (2019), 126–143.

[18] Jordan M Malof, Kyle Bradbury, Leslie M Collins, Richard G Newell, Alexander Serrano, Hetian Wu, and Sam Keene. 2016. Image features for pixel-wise detection of solar photovoltaic arrays in aerial imagery using a random forest classifier. In *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE, 799–803.

[19] Jordan M Malof, Leslie M Collins, and Kyle Bradbury. 2017. A deep convolutional neural network, with pre-training, for solar photovoltaic array detection in aerial imagery. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 874–877.

[20] Jordan M Malof, Leslie M Collins, Kyle Bradbury, and Richard G Newell. 2016. A deep convolutional neural network and a random forest classifier for solar photovoltaic array detection in aerial imagery. In *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE, 650–654.

[21] Jordan M Malof, Rui Hou, Leslie M Collins, Kyle Bradbury, and Richard Newell. 2015. Automatic solar photovoltaic panel detection in satellite imagery. In *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE, 1428–1431.

[22] OpenStreetMap [n. d.]. OpenStreetMap. https://www.openstreetmap.org/.

[23] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. 2018. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. (2018).

[24] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[25] Sunil Rao, Sameeksha Katoch, Pavan Turaga, Andreas Spanias, Cihan Tepedelenlioglu, Raja Ayyanar, Henry Braun, Jongmin Lee, U Shanthamallu, M Banavar, et al. 2017. A cyber-physical system approach for photovoltaic array monitoring and control. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 1–6.

[26] Salima Sarikh, Mustapha Raoufi, Amin Bennouna, Ahmed Benlarabi, and Badr Ikken. 2019. Photovoltaic system fault identification methodology based on IV characteristics analysis. In *AIP Conference Proceedings*, Vol. 2123. AIP Publishing LLC, 020037.

[27] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[28] Brenda So, Cory Nezin, Vishnu Kaimal, Sam Keene, Leslie Collins, Kyle Bradbury, and Jordan M Malof. 2017. 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, 1603–1606.

[29] SolarDetector [n. d.]. SolarDetector. https://github.com/cyber-physical-systems/solardetector.

[30] SolarFinder Dataset [n. d.]. SolarFinder. https://www.kaggle.com/datasets/qli027/solar-finder/.
[31] Rui Wang, Joseph Camilo, Leslie M Collins, Kyle Bradbury, and Jordan M Malof. 2017. The poor generalization of deep convolutional networks to aerial imagery from new geographic locations: an empirical study with solar array detection. In *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, 1–8.
[32] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. 2018. Transferring GANs: generating images from limited data. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 218–234.
[33] Yaosuo Xue, Michael Starke, Jin Dong, Mohammed Olama, Teja Kuruganti, Jeffrey Taft, and Mallikarjun Shankar. 2018. On a future for smart inverters with integrated system functions. In *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*. IEEE, 1–8.