SolarFinder: Automatic Detection of Solar Photovoltaic Arrays

Qi Li, Yuzhou Feng, Yuyang Leng and Dong Chen
Florida International University

ABSTRACT

Smart cities, utilities, third-parties, and government agencies are having pressure on managing stochastic power generation from distributed rooftop solar photovoltaic (PV) arrays, such as predicting and reacting to the variations in electric grid. Recently, there is a rising interest to identify solar PV arrays automatically and passively. Traditional approaches such as online assessment and utilities interconnection filings are time consuming and costly, and limited in geospatial resolution, and thus do not scale up to every location. Significant recent work focuses on using aerial imagery to train machine learning or deep learning models to automatically detect solar PV arrays. Unfortunately, these approaches typically require Very High Resolution (VHR) images and human handcrafted solar PV array templates for training, which have a minimum cost of \$15 per km^2 and are not always available at every location.

To address the problem, we design a new system—SolarFinder that can automatically detect distributed solar PV arrays in a given geospatial region without any extra cost. SolarFinder first automatically fetches regular resolution satellite images within the region using publicly-available imagery APIs. Then, SolarFinder leverages multi-dimensional K-means algorithm to automatically segment solar arrays on rooftop images. Eventually, SolarFinder employs hybrid linear regression approach that integrates support vector machine (SVM) modeling with a deep convolutional neural networks (CNNs) approach to accurately identify solar PV arrays and characterize each solar deployment. We evaluate SolarFinder using 269,632 satellite images that include 1,143,636 contours from 13 geospatial regions in U.S. We find that pre-trained SolarFinder yields a Matthews Correlation Coefficient (MCC) of 0.17, which is 3 times better than the most recent pre-trained CNNs approach and the same as a re-trained CNNs approach.

CCS CONCEPTS

•Computing methodologies → Model development and analysis; Model verification and validation;

KEYWORDS

Solar Detection, Deep Learning, Image Processing, Energy

1 INTRODUCTION

Smart grid, as a network system that is comprised of over 500 million sensors and actuators, is the foundation of modern society and actually one of the largest Internet of Things (IoT) deployments in the world [8]. The number of solar-powered homes is rapidly increasing due to a steep decline in solar module prices. In the first quarter of 2019, over ~70% of solar deployments in the U.S. are continuously small-scale photovoltaic (PV) arrays from residential rooftops. These distributed rooftop solar PV arrays are grid-tied deployments that are feeding excess solar power generated back into smart grid.

Smart grid is having challenges in managing and controlling this intermittent solar generated energy. For instance, the increasing penetration of rooftop solar PV arrays is decreasing the accuracy of net load predictions for utilities. Meanwhile, utilities are losing revenue from homeowners generating their own solar power during the day, but still maintaining the same generating capacity to provide these homeowners electricity when their solar PV arrays are not able to generate power. As a result, government agencies (e.g., Massachusetts Applications for Cap Allocation [26]) usually place limit on the amount of solar PV arrays that can be installed in a geospatial region. The current process relies on accurate statistics of solar deployment generation capacity within a region. Thus, recently, there is a strong interest from utilities, third-parties, and government agencies in passively identify and characterize rooftop solar PV arrays in the electric grid at scale and learn their configuration information, such as size, orientation and shading, which is critical and valuable for solar forecasting and operation management in smart grid.

Traditional approaches such as online assessment and utilities interconnection filings are time consuming and costly [9]. In addition, they are typically limited in some geospatial resolution and thus do not actually scale up to all the locations. Significant recent work [15, 16, 18, 22–25, 33, 36] focuses on using aerial imagery to train machine learning (ML) or deep learning (DL) models to automatically detect solar deployments. The key insight here is that solar arrays are visually identifiable, as shown in Figure 1. Broadly, these techniques all require training data that includes very high resolution (VHR) images (0.3~0.8m per pixel) and human handcrafted image templates to calibrate their models. Unfortunately, these data are very costly (may cost as \$15 per km^2 [31]) and not available at every location, and thus new techniques are necessary.

To address the problem, we design a new approach—SolarFinder that can automatically detect distributed solar PV arrays in a given geospatial region without any extra cost. Our hypothesis is that the new hybrid approach—SolarFinder is capable of detecting rooftop solar PV arrays more accurately when it combining the benefits from both of the machine learning approaches and the deep learning approaches. In evaluating our hypothesis, this paper makes the following contributions.

Pure Approaches Comparison. As reference points for solar PV arrays detection, we first discuss both of the prior pure machine learning (ML)-based solar arrays detection approaches and the most recent pure deep learning (DL)-based approaches. We benchmarked prior solar detection approaches and studied their benefits and drawbacks. We find that ML-based approaches typically report better True Positives, while, the DL-based approaches usually report better True Negatives.

Detection Challenge. We highlight numerous challenges that we met to detect solar arrays automatically from low or regular resolution satellite imagery data. Rooftop solar array identification is affected by numerous unknown variables, including the physical

Figure 1: Solar PV arrays are visually identified in publicly-available Google satellite imagery.

characteristics of a home's rooftop solar PV arrays, e.g., shading generated by nearby tall buildings and trees, size, orientation, and other outliers on rooftop, e.g., window, chimney, etc.

SolarFinder Design. We design a new hybrid approach—SolarFinder, which detects solar PV arrays in a given region without any extra cost. SolarFinder first automatically fetches satellite images within each region using publicly-available maps APIs. Then, SolarFinder applies K-means to automatically segment rooftop images into contours. Finally, SolarFinder leverages hybrid linear regression model that integrates a support vector machine (SVM) classifier with a Convolutional Neural Networks (CNNs) model to accurately identify solar PV arrays, and also learn the detailed installation information for each solar deployment simultaneously. This information is critical for utilities, third parties, and government agencies to manage solar resources.

Implementation and Evaluation. We implement and evaluate SolarFinder using 269,632 publicly-available satellite images that include 1,143,636 contours from 13 geospatial regions in the U.S. We find that supervised SolarFinder is able to detect solar PV arrays with the Matthews Correlation Coefficient (MCC)—0.31, which is 2 times better than the most recent CNNs approach yielding at a MCC of 0.17. Interestingly, pre-trained (or unsupervised) SolarFinder yields a MCC of 0.17, which is 3 times better than the most recent pre-trained CNNs-based approach and is the same as a supervised CNNs-based approach. Thus, SolarFinder achieves similar accuracy without access to any training data from testing sites as a fully supervised approach with complete access to such training data. We evaluate our new approach–SolarFinder using multiple ways:

(1) We compare SolarFinder's results with the groundtruth labeled data for 269,632 sites and show that it can accurately detect rooftop solar installations and also learn the local physical characteristics for each solar site.

(2) We validate SolarFinder's detection results using the groundtruth data for 500 sites from a government agency—Massachusetts Applications for Cap Allocation (MassACA) [26].

(3) We validate SolarFinder's accuracy for profiling local physical characteristics for 10 solar sites by integrating with most recent solar generation capacity prediction work [11, 12], and the evaluation shows that SolarFinder-assisted solar forecasting models can help utilities to better predict solar generation in the grid.

Releasing Datasets and Code. We release all the datasets that are comprised of over 200,000 satellite images and the source code of SolarFinder on our website [34] such that other researchers may use SolarFinder to benchmark their future work.

2 BACKGROUND

Problem statement: Given a geospatial region, we first want to build a new approach that can automatically search and extract rooftop images from publicly-available low or regular resolution satellite imagery. We then present a new approach that can accurately segment objects in each rooftop image. We further seek to build a new model to accurately identify the solar arrays among all the objects from each rooftop image. Moreover, for each detected rooftop solar array, we want to learn its size, orientation and shading situation, and other physical characteristics that are critical to predict solar generation capacity at each solar site. In doing these, we can perform statistical learning for solar installations within a given region, such as how many homes have solar arrays installed on their rooftops, and how many solar arrays are facing towards south and north and so on. Formally, given a target geospatial region a_i, we need to segment all the rooftop images r_i of all the buildings b_i with region a_i. And for each rooftop r_i, we then need to segment all the objects b_i ($1 <= i < N$) into small "contours" c_i. Eventually, we will identify each c_i that is solar array and report its estimated size, orientation and shading situation. Further, using this information, we need to predict solar generation capacity for all the buildings b_i with region a_i.

We outline the design alternatives for detecting distributed rooftop solar PV arrays using satellite images, including Machine Learning (ML)-based approaches, and deep Convolutional Neural Networks (CNNs)-based approaches. In doing so, we review a wide range of the most recent sophisticated solar PV array detection approaches based on Logical Regression (LR), Support Vector Machines (SVMs) and Random Forest (RF) [22, 24, 25], and (CNNs) [15, 23, 33]. Table 1 quantifies the effectiveness of the three approaches by showing the percentage of the approaches yield True Positives (detect solar array and the rooftop does have one), True Negatives (detect 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). The accuracy is then the sum of the true positive and true negative percentages. We report the Matthews Correlation Coefficient (MCC) [27] 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 detection, 0.0 being random detection, and -1.0 indicating an always wrong detection.

To generate Table 1, we prepared a dataset that has 1,000 satellite images using Google Maps Image APIs [17]. We use the same dataset to benchmark the performance of different approaches in Table 1. The dataset has a class ratio that is the ratio of positives to negatives as 1:1. We also preprocessed images for the different approaches based on their own requirements. Especially, for the CNNs-based approach, we handcrafted solar PV array masks for its training data in the way as described in [23]. For the ML-based approaches, we train all the models with a 70-30% split of training data to test data.

2.1 Machine Learning Approaches

Prior research in [22, 25] leverages ML models to identify solar PV arrays using very high resolution (VHR) rooftop satellite images that have a resolution of 0.3 meter per pixel and 8-bit in each RGB

Model	True Positives	False Negatives	True Negatives	False Positives	MCC
Logical Regression	74.06%	25.94%	74.54%	25.46%	0.15
SVMs (linear)	75.77%	24.23%	75.37%	24.62%	0.16
Random Forest	94.88%	5.11%	45.95%	54.05%	0.11
VGGnet-based CNNs	54.49%	45.51%	89.11%	10.89%	0.18

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

(a) solar arrays (b) structure (c) structure

(d) shade (e) ridge (f) trees

Figure 2: The objects that may exist in rooftop images.

color channel. The idea of these approaches is that solar PV arrays have some unique physical shape features that allow us to train ML classifiers to predict the existence of solar PV arrays in the VHR images. The major challenge of these approaches is to empirically identify unique shape features from those VHR images that can be used to train the classifiers. The researchers in work [25] use 100 manually selected VHR images, with 50 images having solar PV arrays installed and the other 50 images having no solar arrays deployed on their building rooftops. The features extracted empirically include: 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 constructed a training dataset that includes all the above shape features and has 18 dimensions. Eventually, a SVM classifier is employed to perform the binary classification. A later work in [22] demonstrates a detection approach based on the similar insight as in work [25], but uses more features and leverages RF classifier to train their model.

Observation: Our results in Table 1 show that the prior ML-based approach using the kernel of LR, SVMs and RF yields a MCC of 0.15, 0.11 and 0.16, respectively. The accuracy of these ML-based solar PV arrays detection approaches is slightly worse than the CNNs-based approaches that typically yield a MMC of 0.18. But, the ML-based approaches are reporting better True Positives (TP) than the CNNs-based approaches. This is mainly due to the fact that physical color and shape features are very effective for ML models to identify solar PV arrays.

2.2 Deep Learning Approaches

Significant recent research focuses on leveraging visual geometry group networks (VGGNets) [32] based CNNs techniques [15, 23] to automatically detect rooftop solar PV arrays using satellite images. 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 solar PV array templates to train their models. For instance, the authors in [15] proposed a five layers CNNs-based approach that includes three convolutional layers and two fully connected layers. The inputs are 3,347 three-channel satellite images with the size of 200×200 pixels. The training data of their CNNs model is a well balanced dataset with a sample ratio of 1,643 positive samples to 1,704 negative samples. The major problem with this approach is that the input rooftop images have many "outliers" rather than solar arrays, and the CNNs model is not able to reliably identify them. The work in [23] is an improvement to the approach in [15]. The authors presented a CNNs architecture that has two different modules, including VGG(x) modules, and fully connected neuron FC(y) modules. The CNNs are 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 CNNs 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 solar PV array annotations. All the training data and groundtruth data are labeled by human annotators. The work in [23] has better accuracy to identify solar PV arrays due to two reasons: 1). The CNNs-based approach is trained using VHR images that have spatial resolution of 0.3 meter per pixel; 2). The solar PV array masks are handcrafted by human annotators, and significantly help the CNNs-based approach to distinguish solar PV arrays from other objects on the same rooftop image. Another work [24] is a variant of the CNNs-based model in the work in [23] and employs RF modeling to benchmark performance.

Observation: Our results show that CNNs-based approach yields a MCC of 0.18 which is the best one among all the approaches reported in Table 1. However, the CNNs-based approach is reporting the TN percentage of 89.11% which is 43.16% better than the ML-based approach using RF classifier, yielding at a TN percentage of 45.95%. This is mainly due to the CNNs-based approach may not be able to reliably distinguish solar PV arrays from the other rooftop objects that have similar grayscale as solar PV arrays'.

2.3 Summary

In summary, the ML-based approaches [22, 24, 25] are more accurate when identifying solar PV arrays, while, the CNNs-based approaches [15, 23, 33] perform better for the identification of rooftop outliers than solar arrays. This is mainly due to the fact solar arrays have significant features that allow us to identify them. Training on these significant features allows ML classifiers to identify solar arrays accurately. In comparison, CNNs-based approaches are good at detecting those non-solar-array objects in rooftop images. The non-solar-array objects generally are hard to model manually or physically, for instance, the shades on the rooftops vary at different houses. For the same rooftop, the shape and size of the

Figure 3: Pipeline of operations SolarFinder uses to identify solar arrays within a target region.

shadings vary at different times of a day, and on different days of a year. It is very challenging for ML-based approaches to learn this effect and detect the shades efficiently. While, the CNNs-based approaches [22, 24, 25] that employ multiple convolutional network layers and fully-connected layers are able to learn the characteristics of those outliers. Therefore, to accurately detect rooftop solar PV arrays, it is desirable to employ a hybrid approach that can combine the benefits from both of the ML-based approaches and the CNNs-based approaches. That says, the new hybrid approach should be able to report higher True Positives percentages and True Negatives percentages simultaneously, and thus a better MCC value. The insights above lead to the design of SolarFinder's approach, which integrates physical modeling, ML and DL techniques to detect solar PV arrays in a region more accurately and more efficiently without any cost.

3 DETECTION CHALLENGES

In this section, we describe the major challenges that we met when designing for our new approach—SolarFinder that accurately detects distributed solar arrays in a target geospatial region.

Low resolution satellite images. Unlike prior approaches [10, 15, 15, 23, 23] that have access to VHR satellite images for a target geospatial region, SolarFinder only uses the publicly-available satellite images that are typically in low or regular resolution. Thus, the shape features that are exacted empirically using VHR satellite images may not be able to accurately describe the characteristics of solar PV arrays using these satellite images.

Automatic building rooftop segmentation. Prior approaches [10, 15, 15, 23, 23] rely on human annotators to segment rooftop images in a region. However, this segmentation approach only applies on solar array detections in small regions and it does not scale up to every location in the world. In addition, it is impractical for SolarFinder to employ human annotators or Amazon Mechanical Turk to handcraft building rooftop images due to the arbitrary searching region and its area.

Objects segmentation. Current DL-based approaches require handcrafted solar PV array image templates to train a reasonably accurate model. While, within a small region that has area of 78.54 km^2 and the ratio of solar-powered homes to regular homes as 0.57, the total amount of satellite images that SolarFinder has to preprocess can be as many as 41,995. That says, SolarFinder has to automatically segment rooftop objects on each rooftop imagery.

Inaccurate shape and color features. In addition to solar panels, many other objects may exist on the rooftops. As show in Figure 2, these may include ridges, structures, trees, and shades. Especially, the physical shape features of ridges, structures and shades have significant overlaps on their statistical characteristics, e.g., mean,

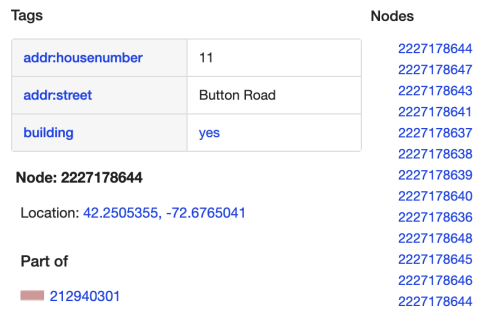


Figure 4: The overview of the OSM file that includes building tag, nodes, and location for a two-story private house.

variance, range, standard deviation, etc. We will discuss more in our shape features extracting section later.

Highly imbalanced data. The prior work mainly prepares their training using a 1:1 ratio which indicates a well-balanced VHR image dataset such that 50% of the dataset are positives samples (have solar PV arrays) and the other 50% images have no solar PV arrays included. However, in the actual "searching" of solar arrays within an arbitrary region, as shown later in Section 6, the satellite image datasets are typically highly imbalanced. The current ML-based approaches may need more "negations" to achieve more accurate true negatives when detecting solar arrays. We identify new physical features to mitigate this issue. In addition, prior research [19] has shown that many accuracy metrics such as ACC, F1, precision, recall and others are very sensitive to the ratio of negative examples to positive samples in their datasets. Thus, prior approaches' results may not be able to reflect actual binary classification results when applying on datasets that have different ratios. We use MCC to report all accuracy results in this paper.

4 SOLARFINDER DESIGN

In addressing these issues, we design a new system—SolarFinder that accurately detects distributed solar array automatically without any extra cost. SolarFinder works by first automatically segment satellite images to rooftop images. After identifying the building rooftops from satellite images, SolarFinder leverages K-means to automatically segment rooftop images into object contours. Then, SolarFinder employs a hybrid approach that integrates the SVM-RBF model with the CNNs model to detect solar PV arrays. Finally, SolarFinder applies solar array size, orientation and other characteristics estimators to further profile each solar installation. Figure 3 shows the SolarFinder's pipeline of operations.

4.1 Preprocessing Satellite Images

Segmenting Building Satellite Images. As discussed in Section 3, it is impractical to leverage human annotators or Amazon Mechanical Turk to handcraft building rooftop images for SolarFinder. To address this problem, instead of focusing on segmenting building images directly from satellite images, we present a reversed image fetching approach that leverages publicly-available maps APIs, e.g., Google Maps [17], OpenStreetMap [28], etc. The input of the approach is a region and outputs are the segmented rooftop satellite images. Given a set of target regions r_i, where $0 < i < n+1$,

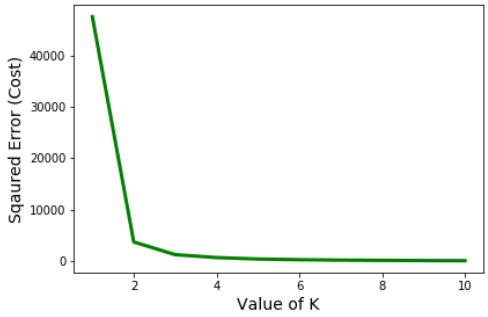


Figure 5: The relationship between K and WCSS errors for K-Means clustering.

SolarFinder first collects all the building b_i rooftop polygons' information using OpenStreetMap API. The return of OpenStreetMap API is the OSM file that contains profiling information for all the objects. Thus, SolarFinder filters out those "outliers" that are not buildings. By doing this, as shown in Figure 4, SolarFinder fetches the location information of all the nodes for each building. Eventually, SolarFinder recovers rooftop polygons using those nodes information, and feeds them into the Google Maps API that returns the satellite imagery r_i when a region is specified in the requests.

Note that, OpenStreetMap has the rooftop polygon information for most of the buildings. SolarFinder's approach to estimating the polygon information of rooftops is orthogonal to the other aspects of the technique and we could use other approaches to profile rooftops. The data from OSM has been used in various ways including production of paper maps and electronic maps (similar to Google Maps, for example), geocoding of address and place names, and route planning. However, we are not aware of any other work that used OpenStreetMap to identify the rooftop polygon shapes.

Segmenting Rooftop Satellite Images. After segmentation of building satellite images, we now have all the rooftop images. SolarFinder then leverages unsupervised multi-dimensional k-Means algorithm [20] to automatically segment each rooftop image r_i into a set of contours c_i such that objects on the rooftop r_i are isolated. K-means clustering finds the best centroids by alternating between assigning grayscale data point per pixel to clusters using current centroids or selecting the centroids using current assignment of grayscale data point per pixel to clusters. Thus, given a rooftop image r_i, our goal is to assign each pixel based on its grayscale value into the best cluster. The key to apply K-means clustering is to determine the optimal K. We leverage elbow method [3] to determine the optimal number of clusters—K. The elbow method is using within-cluster sum of square (WCSS) as the metric to benchmark each possible K. As shown in Figure 5, we find that when choosing $K = 5$, the K-means algorithm yields at the minimum WCSS. The outputs of this segment process are the contours that potentially have all the rooftop objects reside.

4.2 Detecting Rooftop Solar PV Arrays

SolarFinder leverages a hybrid approach that integrates ML model with CNNs-based modeling to accurately identify solar PV arrays in each rooftop images, and thus achieve the benefits from both. Below, we first introduce how we identify principle features and ML classifiers. After that, we discuss the design of our CNNs modeling. Eventually, we introduce the design of our hybrid approach that combines ML classifier and CNNs modeling.

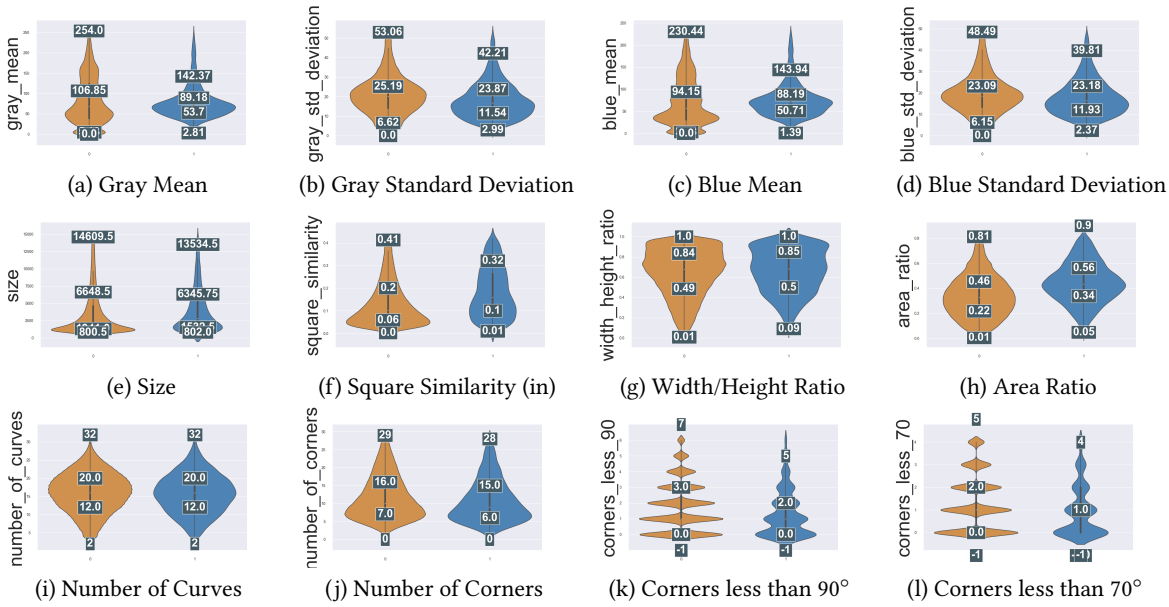
To identify principle features of solar PV arrays, as shown in Figure 6, we first build a large solar array satellite image dataset that has 269,632 solar rooftop images and then physically examine 12 features based on shape, RGB channels, and grayscale in all the contours of solar arrays, including the mean and standard deviation of grayscale, mean and standard deviation of blue channels, size, square similarity, ratio of width to height, area ratio, number of edges, number of corners, and corner degrees. We include the shapes of solar PV arrays as an important feature in our ML model. In addition, as we observed in satellite images, the shapes of solar PV arrays may look like squares, rhombus, diamonds, or parallelograms rather than always uniform rectangles due to their tilt, orientation and shadings from nearby trees and tall buildings. Thus, as shown in Figure 6, we identified additional features based this observation, including number of curves, number of corners, and square similarity. Note that, SolarFinder's approach to estimating the shapes of solar arrays is orthogonal to the other aspects of the technique and is thus "pluggable," such that we could use other computer vision approaches to estimate object shapes on rooftops.

For each feature in Figure 6, we report two statistical analysis results, the left one shows the results when we analyzing on non-solar-panel resident contours, and the right one shows the results when we use solar panel resident contours. Thus, for a sensitive feature that contributes to identifying solar arrays, it is expected that the feature can show a significantly different pattern between the statistical analysis results in the subgraph. For example, the results in Figure 6 show that grayscale mean (Figure 6(a)) and blue channel mean (Figure 6(c)) are sensitive metrics that can be used to identify solar PV arrays in a contour. Interestingly, as shown in Figure 6(j), the feature—the number of corners is not a very sensitive metric to identify solar arrays since we can observe there is not a significant change on this value for solar panel resident contours and non-solar panel resident contours. Similarly, another feature—the number of edges (shown in Figure 6(j)) has 100% overlap in statistical analytics using the contours from two different categories.

However, it is difficult to empirically extract the principle features for ML classifiers directly from this statistical analysis. Thus, we plot a scatter grid in Figure 7 that shows the correlations between each feature with all other features. Some features may highly correlate with other features, although they may show significant sensitivity in statistical reporting results as shown in Figure 6. For instance, the shape feature—shape factor (a.k.a, Square Similarity in Figure 6) that indicates as to the shape of the object. It can be defined as follows,

$$\frac{4 \cdot \pi \cdot a_i}{{p_i}^2} \tag{1}$$

where a_i denotes the area of the $i-th$ contour, and p_i indicates the perimeter of the $i-th$ contour. Circles have the greatest area to perimeter ratio and this feature will approach a value of 1 for a perfect circle. Squares like solar panels are around 0.78. A thin thread-like object would have the lowest shape factor approaching 0. The shape factor (Figure 6(f)) highly correlates with other features, such as the ratio of width to height (Figure 6(g)) and area ratio (Figure 6(h)) for a contour. However, we are not able to simply choose these features since they may overfit our ML models.



(a) Gray Mean (b) Gray Standard Deviation (c) Blue Mean (d) Blue Standard Deviation

(e) Size (f) Square Similarity (in) (g) Width/Height Ratio (h) Area Ratio

(i) Number of Curves (j) Number of Corners (k) Corners less than 90° (l) Corners less than 70°

Figure 6: The statistical analysis of 12 features that can be used to detect solar PV arrays.

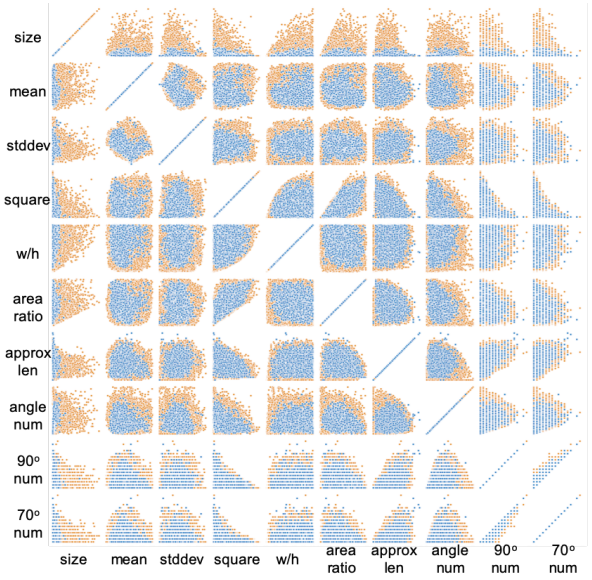
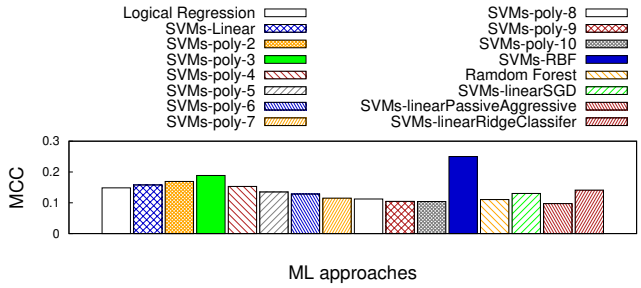


Figure 7: The scatter grid plot shows the correlations between shape features and color features using 125,672 rooftop contours.

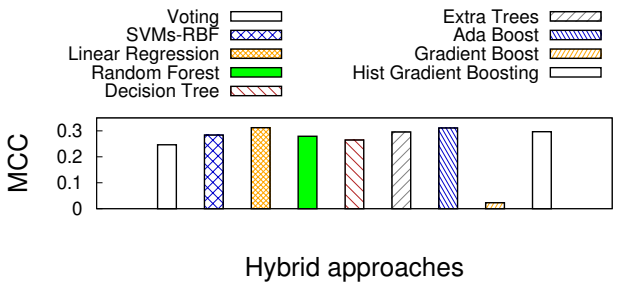
To address this problem, we leverage principal component analysis (PCA) which simplifies the complexity in high-dimensional data while retaining trends and patterns. PCA provides dimensionality reduction that has been used to optimize the training time and solve part of the overfitting problem. We employ PCA to automatically learn the principle features for solar PV arrays detection. When applying PCA on our dataset, the input is the whole dataset ignoring the binary class labels. PCA computes the covariance matrix, and eigenvectors and corresponding eigenvalues. Eventually, we transform our samples onto new subspace. We find that starting from 8 components, and the PCA's variance is closing to 1.0. That says, SolarFinder is able to reduce dimensions from 17 to 8. We use these 8 principle components as the inputs for our ML classifier.

4.2.1 Training Machine Learning Classifier. We find the principle features in Section 4.2. Next, we focus on selecting the optimal ML classifier that has the best accuracy for solar arrays detection. We investigate the most widely used machine learning models in prior solar detection work, including logical regression, support vector machines (SVMs), and random forest. In particular, we also benchmark the different kernels for SVMs, including linear, linear passive-aggressive, linear ridge, polynomial with 1~10 degrees, and radial basis function (RBF). The results in Figure 8(a) show that the SVMs classifier with RBF kernel yields the best MCC as 0.25, which is 2 times better than the random forest classifier. Note that, the prior research in [22] relies on random forest model to identify solar PV arrays. This is mainly due to their observation is based on a very limited size of image dataset. Thus, by intergrading our approach which is built based on a significantly larger dataset, the prior work [22, 24, 25] may achieve better accuracy.

4.2.2 Training Deep Learning Classifier. As we already discussed in Section 2, we find that ML-based approaches are typically reporting better true positives percentages, while, CNNs-based approaches are usually reporting slightly better true negatives percentages. Based on this insight, in addition to the ML-based approach—SVMs classifier with RBF kernel, we also design a CNNs-based based DL approach to detect solar PV arrays using their satellite images. Below, we describe the design of our CNNs architecture. As shown in Figure 9, our CNN architecture is comprised of input, convolutional layers (ReLU), max pooling, fully-connected layers (with and without ReLU) and output. The inputs are 150x150 contour images, and the first two layers are convolutional layers that have 150x150 neurons with a rectified linear unit (ReLU). Then, we have another two convolutional layers that have 75x75 neurons



(a) Detection accuracy when use different ML models.



(b) Detection accuracy when use different hybrid approaches.

Figure 8: The comparison of detection accuracy when using different ML models and kernels and hybrid fusion approaches.

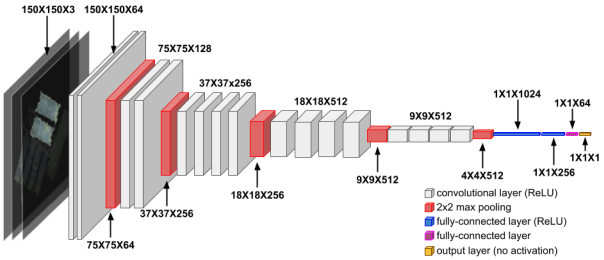


Figure 9: The overview of our CNNs architecture design.

with ReLUs. After these, we employ another 4 convolutional layers with ReLUs, and these layers all have 18x18 neurons. Finally, we leverage 4 convolutional layers with ReLUs, and these layers all have 9x9 neurons. Among the different groups of convolutional layers, we have 2x2 max pooling which is used to down-sample the input images and reduce its dimensionality. In addition, two fully-connected layers with ReLU and another fully-connected layer (without ReLU) are added to process SolarFinder's outputs.

4.2.3 Building Hybrid Approach. Our key insights are 1) SVMs-RBF model achieves a better True Positives (reporting solar PV arrays), 2) CNNs model has a better True Negatives (reporting non-solar-arrays objects on rooftops), 3) to get the benefit of both, we design the hybrid approach—SolarFinder. The SVMs-RBF model, as a part of our hybrid approach leverages the new features that we developed based our observation of physical features from solar PV arrays (shown in Figure 6) to identify solar PV arrays when using true positive/negative bias data.

Below, we describe the design of this hybrid approach. We first study the performance of a set of potential hybrid fusion approaches that are using both the averaging and boosting methods. These approaches include Voting, SVMs, and Linear Regression (LR), Random Forest, Decision Tree, Extra Trees, Ada Boost, Gradient Boost, and Hist Gradient Boosting that have been widely used in prior assembling learnings. Note that, SolarFinder's approach to combine the predictions of the two different estimators/classifiers is orthogonal to the other aspects of the techniques, such that we could use other fusion methods to combine the predictions. The inputs are both the outputs from the SVMs-RBF prediction (in Section 4.2.1) and the outputs from the CNNs approach prediction (in Section 4.2.2). In doing so, SolarFinder can inherently combine the benefits from both. The comparison results that we use different models to build SolarFinder's hybrid approach are shown in Figure 8(b). We split the dataset into training dataset and testing dataset using a ratio of 7:3. The results indicate that LR yields at the best MCC as 0.24. Thus, by leveraging LR to combine the SVMs-RBF modeling and CNNs modeling, we achieve a better MCC and thus can more accurately identify solar PV arrays from rooftop images than prior pure SVMs and pure CNNs approaches. The details of our LR modeling is described as follows,

$$Y(i) = 0.6443 \cdot Y_{CNNs}(i) + 1.6638 \cdot Y_{SVMs}(i) - 1.4677 \quad (2)$$

where $Y(i)$ denotes the final output of SolarFinder for contour i, $Y_{CNNs}(i)$ indicates the prediction output using pure CNNs approach, and $Y_{SVMs}(i)$ denotes the prediction output using pure SVMs-RBF approach. LR fits a linear model with coefficients—$C = (C1, C_2)$ to minimize the residual sum of squares between the observed results in the dataset, and the results are predicted by the linear approximation. For the LR model here, we learned the coefficients as (0.6443, 1.6638) with a lost/bias function as -1.4677.

4.3 Profiling Rooftop Solar Arrays

In addition to detecting solar PV arrays, SolarFinder can also profile each reported solar PV array. The profiling information may include size, orientation, shade, window, chimney, etc. For instance, to report the size, SolarFinder examines the number of pixels that are included in the identified solar arrays. Since each pixel denotes an area with a size of S km^2, where S can be derived based on the image zoom level—20 and its location. SolarFinder first simply multiplies the pixel size by the number of pixels in a solar array resident contour. Then, SolarFinder performs a union operation to add up all the contours for the same rooftop to report its solar PV array size. Similarly, SolarFinder can report the size of shading generated by nearby tall trees or buildings. To learn the orientation for a rooftop solar deployment, SolarFinder measures the angle difference between the minimum bounding rectangle (MBR) and the minimum area rectangle for each contour. SolarFinder reports the orientation by estimating the difference between them.

4.4 Preprocessing and Training Overhead

As we had discussed in Section 2, prior techniques typically require a significant amount of training data including VHR imagery (0.3~0.8m/pixel) and human handcrafted solar panel image templates to train their models. The preprocessing is time consuming, our hybrid system—SolarFinder leverages OpenStreetMap [28] API to fetch the rooftop images, and then uses the unsupervised rooftop object clustering approach to automatically segment solar arrays from rooftop images. In addition, our CNNs model architecture has fewer layers than prior CNNs-based approaches. As we shown in Section 6, SolarFinder is the best "unsupervised" (pre-trained and without repeated training) yields the same accuracy as "supervised" (repeated training) pure CNNs-based approach. In doing so, SolarFinder reduces the preprocessing overhead and training overhead.

5 IMPLEMENTATION

We implement SolarFinder in `python` using widely available open-source frameworks, including Pandas [29], OpenCV [4], Scikit-learn [5] and PyCUDA [1, 2]. SolarFinder leverages a number of Maps APIs, e.g., Google Maps API [17], and OpenStreetMap API [28]. Our current implementation fetches satellite images (800x800 pixels) within a target region as described in Section 4. We use OpenCV, NumPy and Pandas for grayscale and RGB channel image data processing. We use the Scikit-learn [5] machine learning library in python to build our pure ML-based approaches. The library supports multiple techniques including SVMs with different kernel functions, multiple linear regression models and PCA. In particular, to report the results in Table 3, we implement the pure ML-based models as specified in prior work [22, 24, 25] using the same input features, dependent output variable, and the same kernels. However, for the results in Table 1 and all other comparison and evaluation results in figures, we use our own extracted features Section 4 that are identified and extracted by applying PCA. For the CNNs-based approaches, we implement them based on the framework from VGGnet [6]. To implement the hybrid approaches, we use Scikit-learn [5], OpenCV and VGGnet. Finally, we schedule the batch jobs on our GPU servers to compare the MCC accuracy of 8 different approaches using CUDA. The server that we use to get all the benchmarking and evaluation results has resources as follows: 1) CPU: 2x Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 2) GPU: nVidia TITAN X (Pascal) (x8), 3) RAM: 128GB, 4) OS: Linux CentOS 7.

6 EXPERIMENTAL EVALUATION

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

6.1 Datasets

Dataset 1. We collect publicly-available satellite images using Google Maps API [17] and OpenStreetMap API [28] from 13 regions of 9 different states in U.S., including Colorado, Wisconsin, California, Massachusetts, Minnesota, Arizona, Maryland, and Washington. Table 2 shows the detailed profiling information for each region. The regions are sorted by the quantity ratio of solar-powered homes to non-solar-powered homes. We randomly selected these 13 regions using Google Maps API [17] with a radius ranging from 5 km to 20 km. We chose to start with 5 km radius areas because many other related research such as [13, 14] is interested in 5 km radius areas, for instance, solar localization, given "anonymous" solar generation usage data, prior work can localize the source site that produces these data to an interested area with 5 km radius. Then, the prior work can integrate our approach—SolarFinder to further narrow down to a specific home. In addition, we developed a toolkit that allows us to visually label solar PV arrays and their groundtruth profiling information in rooftop satellite images. This toolkit is already included in SolarFinder release [34].

Dataset 2. We download 500 private houses satellite images using Google Maps API [17] and OpenStreetMap API [28] based on the locations and groundtruth data that are provided on the website of a government agency — Massachusetts Applications for Cap Allocation (MassACA) [26]. Given a solar-powered home listed in this dataset, we can access to its grundtruth data, including street address, solar generation capacity and installation details.

Dataset 3. We also download 1-minute level solar generation capacity data for 1 year from the dataset which is released by the most recent solar forecasting work [11, 12]. This dataset has three dimensions including timestamps, groundtruth solar generation, predicted solar generation using the models in work [11, 12].

Note that, the groundtruth data may change over the time. Our approach will work as long as new solar PV arrays are updated in the publicly-available satellite images. In particular, as shown in Section 6.4.3, SolarFinder does not need to be re-trained when more solar PV arrays become online. In addition, the penetration of increasing PV adoption may improve the accuracy of SolarFinder.

6.2 Experimental Setup

We implement and compare two different categories—supervised and unsupervised of solar PV arrays detection approaches. To better understand and analyze the benefits of different approaches, we also implement a naive thresholding approach, which leverages the insight—some statistical features allow us to distinguish solar arrays from other outliers on rooftops from Figure 6. For the pure SVM-based approach, we employ the best performance kernel—the Radial Basis Function (RBF) that is evaluated in Figure 8. For the pure CNNs-based approach, we use the VGGnet [6] based CNNs architecture which is shown in Figure 9. Finally, for the hybrid approach which SolarFinder employs, we leverage the Linear Regression (LR) model that is designed in Equation 2. Thus, we have 4 different solar PV array detecting approaches per category.

Supervised Approaches. In this case, all of the naive thresholding, pure SVMs, pure CNNs, and hybrid approaches can access to the solar array satellite images from their testing sites. For the pure CNNs and hybrid approaches, we also fine-tune the VGGnet using the information from the testing sites. In doing so, we are benchmarking the best performance of these 4 different approaches.

Unsupervised Approaches. In contrast, in this case, all of the naive thresholding, pure SVMs, pure CNNs, and hybrid approaches can not access to satellite images from their testing sites. For pure CNNs and hybrid approaches, we do not fine-tune the VGGnet using the information from the testing sites. In doing so, we are benchmarking the practical performance of the 4 different approaches.

Regions	State	Centroid Location	Radius (km)	Houses	Solar-power Houses	Solar Deployed Ratio
#1	WI	43.084961,-88.317162	5	12	0	0.00%
#2	WA	47.313595,-121.99985	5	2,110	11	0.52%
#3	MN	44.926191,-93.213728	5	6,655	37	0.56%
#4	CA	37.438949,-122.18969	5	8,339	473	0.57%
#5	MD	39.371454,-76.738717	5	7,158	84	1.17%
#6	AZ	33.322122,-111.94023	5	526	7	1.33%
#7	MA	42.250448,-72.676531	15	53,491	1,193	2.01%
#8	MA	42.250448,-72.676531	20	185,486	3,795	2.05%
#9	MA	42.250448,-72.676531	10	11,874	531	2.23%
#10	CA	36.764751,-119.80308	5	305	8	2.62%
#11	CO	39.881184,-104.96045	5	3,063	88	2.87%
#12	MA	42.250448,-72.676531	5	6,296	380	6.00%
#13	CA	37.309364,-122.06914	5	7,021	852	12.13%

Table 2: The profiling information for 13 different regions in 8 different states of U.S. we use in the evaluations.

In real applications, SolarFinder works in this unsupervised way such that *no* groundtruth data is required from testing rooftops in a new region to detecting distributed solar PV arrays.

6.3 Evaluating Metrics

Blow we describe the metrics that we use to evaluate SolarFinder and other related approaches.

Matthews Correlation Coefficient (MCC). To quantify the accuracy of different detection approaches, we find that the standard evaluating metrics, e.g, accuracy, F1, would not work well on highly imbalanced data. And this observation has been studied by researches in work [7, 30]. Based on the recommendation from prior work [7, 30], we use the MCC [27], 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.

$$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3}$$

Jaccard Similarity Index (JSI). To quantify the accuracy of SolarFinder to predict size for solar PV arrays, we use JSI which is widely used in prior work to measure the similarity between detected regions and groundtruth regions. JSI measures the similarity for the two sets of pixel data, with a range from 0% to 100%. The higher the percentage, the more precise prediction that SolarFinder can do. It can be defined as follows,

$$JSI = \frac{r_d \bigcap r_g}{r_d \bigcup r_g} \tag{4}$$

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

Mean Orientation Error (MOE). To quantify the accuracy of SolarFinder to predict orientations of solar PV arrays, we employ the MOE that is introduced in a recent work [21]. The MOE captures the per-pixel error between the predicted and the actual azimuth angle. It is defined as follows,

$$MOE = \frac{1}{C} \cdot \sum_i \frac{\sum_j p_{ij} \cdot Azimuths\_differ(o_i, o_j)}{t_i} \tag{5}$$

where C is the total number of classes (i.e., azimuths), o_i and o_j are 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 addition, $Azimuths\_differ$ returns the difference between two azimuth angles. The MOE should return a value between 0° (perfect estimation) and 180° (opposite estimation).

Mean Absolute Percentage Error (MAPE). To quantify the accuracy of SolarFinder-assisted solar forecasting models, we compute the MAPE, as follows, between the ground truth solar energy and the solar energy that SolarFinder-assisted infers over all time intervals t. A lower MAPE indicates higher accuracy with a 0% MAPE being perfectly accurate solar PV array detecting and profiling.

$$MAPE = \frac{100}{n} \sum_{t=0}^{n} |\frac{S_t - P_t}{S_t}| \tag{6}$$

where n describes the duration of the solar prediction, S_t denotes the actual solar generation capacity, and P_t indicates the predicted solar generation capacity at the moment t.

6.4 Experimental Results

6.4.1 Comparing Supervised Approaches. As we explained in the section 6.2, we first compare SolarFinder with fully supervised pure SVM and pure CNNs approaches that have access to satellite images from testing sites. In this case, the three approaches split the dataset into training and testing dataset using a ratio of 7:3 after cross-validation. Unsurprisingly, as shown in Table 3, SolarFinder yields the best MCC—0.31, and is the best performing and the most sophisticated solar PV arrays detection approach. We can also see that the pure CNNs approach and the pure SVM approach yields a MCC of 0.17 and 0.25, respectively. However, the pure SVM approach reports significant better True Positives percentages than that of pure CNNs approach. Interestingly, although naive thresholding approach yields the worst MCC, it does yield the best True Negatives percentage. That says, we can leverage the naive thresholding approach to label groundtruth satellite images, and it will significantly reduce the time that human annotators spend to collect groundtruth data.

Model	True Positives	True Negatives	False Positives	False Negatives	MCC
Pure thresholding (supervised)	15.47%	94.62%	5.38%	84.53%	0.06
Pure SVMs (supervised)	84.87%	84.51%	15.49%	15.13%	0.25
Pure CNNs (supervised)	54.49%	89.11%	10.89%	45.51%	0.17
SolarFinder (supervised)	79.41%	91.01%	8.99%	20.59%	0.31
Pure thresholding (unsupervised)	23.37%	94.84%	5.16%	76.63%	0.06
Pure SVMs (unsupervised)	84.78%	76.29%	23.71%	15.22%	0.11
Pure CNNs (unsupervised)	53.26%	78.33%	21.67%	46.74%	0.06
SolarFinder (unsupervised)	71.74%	91.98%	8.02%	28.26%	0.17

Table 3: The comparison of detection accuracy when employing naive thresholding, pure SVMs, pure CNNs and hybrid approaches.

Results: *Comparing with the supervised ML and CNNs approaches, SolarFinder is the best performing approach and it yields the best MCC as 0.31, which is 5 times better than pure thresholding approach and ~2 times better than pure CNNs approaches.*

6.4.2 Comparing Unsupervised Approaches. We then compare the accuracy of unsupervised pure SVM, pure CNNs and SolarFinder approaches that do not have access to any satellite images from testing sites. In this case, the three approaches split the dataset into training dataset and testing dataset using a ratio of 7:3 without cross-validation between the two datasets. Note that, the pure CNNs approach does not use any information from testing sites to fine-tune its CNNs model at this time. As shown in Table 3, as we expected, SolarFinder yields the best MCC—0.17, which is 3 times better than prior pure CNNs approach, yielding at a MCC of 0.06. Similar to the comparison results of supervised approaches, naive thresholding approach still yields the worst MCC. Again, both of the pure CNNs approach and pure SVM approach have the similar MCC. However, the pure SVM approach reports significant (~31.52%) better True Positives percentages than that of pure CNNs approach. Interestingly, naive thresholding still yields the best True Negatives percentage.

Results: *Comparing with the unsupervised ML-based and CNNs-based approaches, SolarFinder is the best performing approach and it yields the best MCC as 0.17, which is 3 times better than both of the pure thresholding approach and the pure CNNs approach.*

6.4.3 Unsupervised VS Supervised Approaches. Table 3 shows that unsupervised SolarFinder yields the same MCC (~0.17) as supervised pure CNNs approach. In addition, the MCC reported by the unsupervised CNNs-based approach is significantly worse than that of the supervised CNNs-based approach, decreasing by 3x (from 0.18 to 0.06). This is mainly due to the fact that the unsupervised CNNs-based approach can not leverage any information from testing satellite images to fine-tune its neural networks. Interestingly, the unsupervised CNNs-based approach performs significantly worse and yields exactly the same MCC as the naive thresholding approach.

Results: *Comparing with both of the supervised and unsupervised ML-based and CNNs-based approaches, SolarFinder is the best unsupervised performing approach and it yields the best MCC as 0.17, which is the same as supervised pure CNNs approach.*

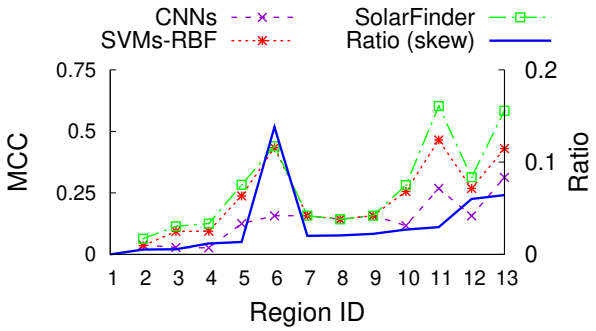


Figure 10: Higher ratios of solar-powered homes to non-solar-powered homes results in higher accuracy. SolarFinder yields the best accuracy across 13 different locations (sorted by the ratios in ascending order).

6.4.4 Quantifying SolarFinder's Accuracy. We next evaluate the detection effect on pre-trained or unsupervised SolarFinder using different regions that have different ratios of solar-powered homes to no-solar installed homes. By doing this, we can examine SolarFinder's accuracy when searching over the regions having imbalanced satellite images data. As shown in Figure 11, the MMC curves reported by all the solar array detection approaches including our SolarFinder are having similar pattern as the curve of the ratios of solar-powered buildings to non-solar-powered buildings. This is mainly because when the ratio goes up, we have more solar arrays satellite images in the dataset, thus SolarFinder is able to yield a better MCC. In addition, SolarFinder's MCC is always the top of all other approaches' MMC curves. Thus, the highly imbalanced satellite image datasets do not affect SolarFinder's accuracy. Solar PV adoption depends on the politics and incentives in a state. As shown in Figure 11, this course incorporation of this information that results in a higher adoption rate in a region should yield a better detection accuracy of SolarFinder. Note that, the 13 regions have the ratio of solar-powered homes to no-solar installed homes ranging from 0%~12.13% and the radius ranging from 5~20km. The regions that have different areas but the same ratios, e.g., Region #7, #8 and #9, yield the similar MCC.

Results: *Higher ratios of solar-powered homes to non-solar-powered homes results in higher accuracy of SolarFinder. In addition, SolarFinder consistently achieves the best accuracy across 13 locations that have different ratios.*

We next plot the receiver operating characteristic (ROC) curves for pure SVM, pure CNNs, and SolarFinder approaches. The goal

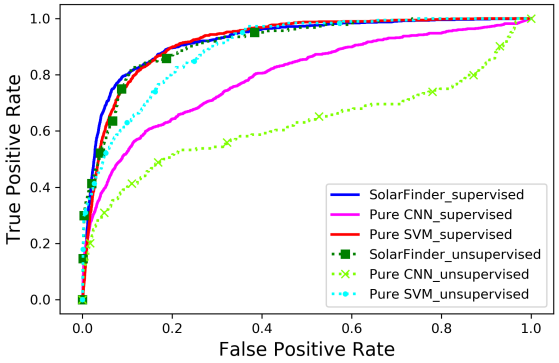


Figure 11: The comparison of receiver operating characteristic (ROC) curves when applying supervised (solid) and unsupervised (dashed) classifiers, including pure SVM approach, pure CNNs approach and SolarFinder.

of this examination is to evaluate the output quality for these 3 different approaches. ROC curves feature true positives rate on the Y-axis, and false positives rate on the X-axis. Thus, that says, the top left corner of the plot is the "ideal"—a false positive rate of zero, and a true positive charge of one. In addition, a larger area under the curve (AUC) is typically better. As shown in Figure 11, for the supervised comparison (solid), SolarFinder stays at the top left corner and overlaps with the pure SVMs approach. While, in the unsupervised comparison (dashed), SolarFinder is the only one that stays at the top left corner quickly and stably yields a true positives rate as ~1.0. In addition, the AUC under SolarFinder curve has the largest area. Therefore, among all the approaches examined in Figure 11, SolarFinder is the best binary classier when detecting solar PV arrays using satellite images.

Results: *SolarFinder's ROC curve stays on the top of the left corner and has the largest AUC. Thus, comparing with prior ML-based and CNNs-based approaches, SolarFinder is the best binary classier for solar PV arrays detection.*

6.4.5 Profiling Detected Solar PV Arrays. Next, we examine the accuracy of SolarFinder when predicting solar array size and orientation using Dataset 2 in section 6.1. SolarFinder first fetches the 500 homes rooftop images and segments them into contours. SolarFinder then applies unsupervised SolarFinder approach over those contours to identify solar panels and learning their characteristics, e.g., size, orientation, shade, etc.

Predicting the sizes of solar PV arrays. We employ the metric—JSI (in Section 6.3) to report the accuracy. As discussed in Section 4, to report the size of solar arrays, SolarFinder first examines the number of pixels that are included in the identified solar PV array contours. Then, SolarFinder performs a union operation to add up all the contours for the same rooftop to report the solar array size. We find that SolarFinder is able to report a JSI as 63.5% using only low resolution satellite imagery.

Predicting the orientations of solar PV arrays. We employ the metric—MOE explained section 6.3 to quantify the accuracy for SolarFinder to predict orientations for solar PV arrays. SolarFinder learns the orientation by analyzing the difference in degrees between the minimum bounding rectangle (MBR), and the minimum area rectangle for a counter and its rooftop. We find that SolarFinder yields a MOE as 3.72 °.

Results: *In addition to accurately detect solar PV arrays, SolarFinder is able to report accurate physical characteristics, e.g., size, orientation, shade, etc. simultaneously.*

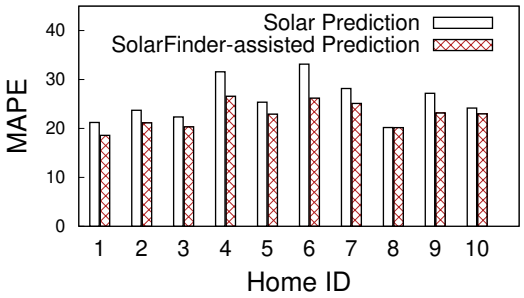


Figure 12: The comparison of solar generation prediction before and after integrating with SolarFinder.

6.4.6 Integrating with Solar Forecasting. Eventually, we integrate SolarFinder's output with the most recent solar generation capacity prediction work [11, 12]. In this case, we apply the SolarFinder-assisted solar predicting modeling on Dataset 3 as discussed in prior section 6.1. Figure 12 shows the MAPEs of 10 homes solar generation prediction before and after calibrating forecasting models using SolarFinder's results. As shown in Figure 12, with fine-tuning using SolarFinder's results, the solar generation prediction models in [11, 12] are able to report small MAPEs over all the 10 solar-powered homes. This is because SolarFinder learns accurate solar installation characteristics using the hybrid approach from satellite images as shown in the prior section, and these characteristics (parameters) are helpful to improve the calibration of solar generation prediction models.

Results: *SolarFinder-assisted solar forecasting models have smaller MAPE, and yields better solar generation prediction accuracy.*

7 RELATED WORK

There is significant prior work on detecting solar PV arrays using satellite images. The prior research employs either ML classifiers [22, 24, 25] or deep learning models [15, 23, 33] to predict the existence of solar PV arrays on the rooftop images. The most notable recent work[15, 22, 24, 33] all use VHR satellite images and evaluate their approaches using very limited dataset due to the high expense to buy or download those VHR images. In addition, prior work [15, 22–25, 33] are also limited to some specific regions and do not scale up. Instead, our new hybrid approach—SolarFinder is built on top of the insights from a larger free publicly-available regular or low resolution satellite image dataset across 13 different regions from 8 states of U.S. SolarFinder shows that we can build a hybrid approach that combines benefits from both of ML models and CNNs models. By doing this, we build a general approach that achieves a better accuracy as 3 times better MCC than the most notable work [15, 33]. In addition, SolarFinder can profile each solar deployment by learning its physical characteristics (i.e., size, orientation, and shade). And our evaluation results show that

SolarFinder-assisted solar generation model has better accuracy than the original models [11, 12].

8 CONCLUSION

We design a new hybrid approach—SolarFinder to automatically detect solar PV arrays using publicly-available satellite images without any extra cost. For a given region, SolarFinder works by first automatically segment satellite images to rooftop images. Second, SolarFinder leverages K-means to automatically segment rooftop images into object contours. Then, SolarFinder employs a linear regression hybrid approach that integrates SVM-RBF model with CNNs model to detect solar PV arrays in each contour. Finally, SolarFinder applies solar array size, orientation and other characteristics estimators to further profile each solar site. We evaluate SolarFinder using 269,632 public satellite images that include 1,143,636 contours from 13 geospatial regions in the U.S. We find that pre-trained (or unsupervised) SolarFinder yields a MCC of 0.17, which is 3 times better than the most recent pre-trained CNNs approach and is the same as a supervised CNNs approach.

We plan to implement the optimization for SolarFinder's post-processing module to report more accurate profiling information, for instance, shade area, which is a critical factor that affect solar generation. We also plan to learn the accuracy of SolarFinder under new situations, such as 1) home owners install Tesla roof shingles [35] rather than regular solar PV arrays, 2) the adoption rate of solar PV arrays increases in a region over time. To detect Tesla roof shingles, SolarFinder needs to include new features that can identify the new Tesla roof shingles. Overall, SolarFinder should achieve higher accuracy in a region when the ratio of solar-powered homes to non-solar powered homes increasing.

Acknowledgements. We would like to thank the anonymous reviewers and our shepherd for providing us their insightful comments and valuable feedback, which significantly improved the quality of this paper. We thank Florida International University School of Computing and Information Sciences for the travel award to present this work.

REFERENCES

[1] 2017. An Even Easier Introduction to CUDA. https://devblogs.nvidia.com/even-easier-introduction-cuda/. (2017).
[2] 2017. PyCUDA. https://mathema.tician.de/software/pycuda/. (2017).
[3] 2020. Elbow Method. https://www.scikit-yb.org/en/latest/api/cluster/elbow.html. (2020).
[4] 2020. OpenCV. https://opencv.org/. (2020).
[5] 2020. Scikit-learn Machine Learning in Python. https://scikit-learn.org/stable/. (2020).
[6] 2020. Very Deep Convolutional Networks for Large-Scale Visual Recognition. https://www.robots.ox.ac.uk/~vgg/research/very_deep/. (2020).
[7] Josephine Akosa. Predictive accuracy: a misleading performance measure for highly imbalanced data.
[8] Fadi Al-Turjman and Mohammad Abujubbeh. 2019. IoT-enabled smart grid via SM: An overview. *Future Generation Computer Systems* 96 (2019), 579–590.
[9] MJE Alam, KM Muttaqi, and D Sutanto. 2013. An approach for online assessment of rooftop solar PV impacts on low-voltage distribution networks. *IEEE Transactions on Sustainable Energy* 5, 2 (2013), 663–672.
[10] 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.
[11] 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.
[12] Dong Chen and David Irwin. 2017. Black-box solar performance modeling: Comparing physical, machine learning, and hybrid approaches. *ACM SIGMETRICS Performance Evaluation Review* 45, 2 (2017), 79–84.
[13] Dong Chen and David Irwin. 2017. Weatherman: Exposing weather-based privacy threats in big energy data. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 1079–1086.
[14] Dong Chen, Srinivasan Iyengar, David Irwin, and Prashant Shenoy. 2016. Sunspot: Exposing the location of anonymous solar-powered homes. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. 85–94.
[15] 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*, Vol. 1. IEEE, 14–19.
[16] 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.
[17] googlemaps 2020. Googlemaps. https://developers.google.com/maps/documentation/maps-static/intro. (2020).
[18] 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.
[19] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. 2013. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 245–251.
[20] k-means 2020. K-means. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html/. (2020).
[21] 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 (KDD '19)*. ACM, New York, NY, USA, 2105–2113. https://doi.org/10.1145/3292500.3330741
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] massana 2020. Massachusetts System of Assurance of Net Metering Eligibility. http://www.massaca.org/. (2020).
[27] mcc 2020. Matthews Correlation Coefficient. https://en.wikipedia.org/wiki/Matthews_correlation_coefficient/. (2020).
[28] OpenStreetMap 2020. OpenStreetMap. https://www.openstreetmap.org/#map=4/38.01/-95.84. (2020).
[29] pandas 2020. Pandas. https://pandas.pydata.org/. (2020).
[30] 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).
[31] satellite-imagery 2019. Global Earth Imaging Pricing Plans. https://geocento.com/imagery-pricing-plans/. (2019).
[32] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[33] 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.
[34] solarfinder 2020. SolarFinder. https://github.com/cyber-physical-systems/SolarFinder. (Feb 2020).
[35] Tesla 2020. Tesla Roof Shingles. https://iroofing.org/tesla-roof-shingles/. (2020).
[36] 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.