Large-Scale Solar Panel Mapping from Aerial Images Using Deep Convolutional Networks

Jiangye Yuan
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: yuanj@ornl.gov

Hsiu-Han Lexie Yang
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: yangh@ornl.gov

Olufemi A. Omitaomu
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: omitaomuoa@ornl.gov

Budhendra L. Bhaduri
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: bhaduribl@ornl.gov

***Abstract*—Up-to-date maps of installed solar photovoltaic panels are a critical input for policy and financial assessment of solar distributed generation. However, such maps for large areas are not available. With high coverage and low cost, aerial images enable large-scale mapping, but it is highly difficult to automatically identify solar panels from images, which are small objects with varying appearances dispersed in complex scenes. We introduce a new approach based on deep convolutional networks, which effectively learns to delineate solar panels in aerial scenes. The approach is applied to mapping solar panels in imagery covering 200 square kilometers in two cities, using only 12 square kilometers of training data that are manually labeled. Results are generated efficiently with an accuracy comparable to manual mapping, demonstrating the effectiveness and scalability of our approach.**

***Index Terms*—Solar PV panel, convolutional network, mapping**

1. Introduction

Solar photovoltaic (PV) is the fastest growing source of distributed generation. There has been a significant increase in the number of installed solar panels in recent years. In U.S. solar installations are expected to reach 16 gigawatts, doubling installations in 2015[1]. However, the actual distribution of installed solar panels is not available on a large scale. The detailed information about installed solar panels is only available to installers and utility companies, who usually are reluctant to share the data. Therefore, a reliable and scalable solution to solar panel mapping is highly desired, which will greatly benefit applications related to energy policy making, power systems, and solar PV market analysis.

In this paper, we aim to map solar panels using aerial images. Thanks to advances made in remote sensing capabilities, aerial images with high spatial and temporal resolution are widely available, which provide an ideal data source for mapping solar panels. Infrared images are recently utilized for solar panel detection [1], which, however, are much less available.

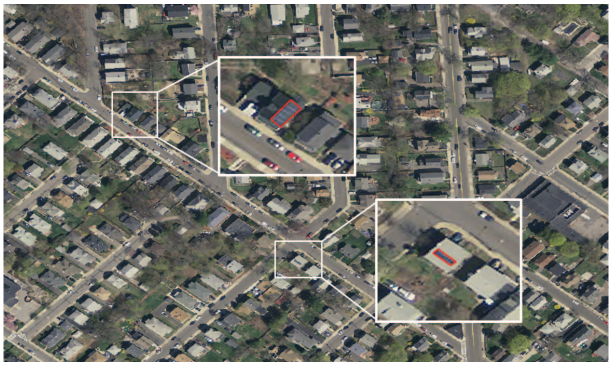

Figure 1. An aerial image containing solar panels. Zoom-in views are provided to show installed solar panels.

Automatic detection of solar panels in aerial images is a challenging task. Solar panels are very small objects scattered in complex scenes. As shown in Fig. 1, solar panels are considerably smaller than objects that are often targeted in aerial image analysis, such as roads and buildings. We will use images with a spatial resolution of 0.3 meters. A large number of solar panels occupy less than one hundred pixels, which provide very little image features and hence can be easily confused with other objects. Moreover, the appearances of solar panels in images vary vastly. In addition to image variations caused by differences in acquisition conditions, solar panels have a variety of types, sizes, and shapes.

To the best of our knowledge, this is the first work dealing with large-scale solar panel detection from images. From the technical aspect, this problem is related to object

[1] http://www.seia.org/news/us-solar-market-set-grow-119-2016-installations-reach-16-gw

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

instance detection, which have been studied in the computer vision community. A typical framework is to design features that can be computed from local windows and train a detection system with such features as input. Given an image, the detection system is applied with a window sliding over the image to locate objects [2], [3]. Success of such a framework highly depends on the discriminative power of designed features, which often takes domain experts tremendous effort to achieve. More importantly, features from fixed size windows have inherent limitations when dealing with small objects – a small size causes the loss of contextual information and a large size results in features capturing too much irrelevant information. A recent method has been proposed that aims specifically at small objects [4]. The method has been shown to work well on objects with similar sizes in homogeneous background, while our task deals with variously sized objects in extremely diverse background.

Deep convolutional networks (ConvNets) trained with massive labeled data have shown to be very powerful to capture the hierarchical nature of features in images and generalize beyond training samples [5]. The capabilities lead to great success in challenging image classification problems [6]. Built upon this success, ConvNet based object localization and segmentation have also been actively studied [7], [8]. However, there is little work reported for detecting very small objects for two reasons. First, since in a ConvNet input images go through a series of downsampling operations to achieve high level representations, small objects are likely to be discarded during the process and hard to recover. The second reason lies in the fact that most work focuses on natural image analysis, where it is generally acceptable to miss very small objects, as long as more salient objects are correctly identified.

In this paper, we present a solution for accurately extracting solar panels from aerial images. Our solution not only detects solar panel locations but their spatial extent. We employ a recently proposed ConvNet method [9], and show that the design of the method is well suited for our task. In order to cope with unique challenges of the task, we take a number of special strategies of network training. The performance of the trained system is demonstrated on very large images containing complex urban scenes.

2. ConvNet for object extraction

We utilize the ConvNet approach proposed in [9]. Here we provide a brief overview and explain the rationale of applying the method to this task. The network architecture is illustrated in Fig. 2. There are seven regular ConvNet stages, each of which consists of a convolutional layer and optionally a max-pooling layer. The network takes 3 band input images. Convolutional layers of the seven stages have 50 filters of size $5 \times 5 \times 3$, 70 filters of size $5 \times 5 \times 50$, 100 filters of size $3 \times 3 \times 70$, 150 filters of size $3 \times 3 \times 100$, 100 filters of size of $3 \times 3 \times 150$, 70 filters of size of $3 \times 3 \times 100$, and 70 filters of size of $3 \times 3 \times 70$, respectively. Each of the first four stages has a max-pooling over a 2×2 unit region. Because all the operations are performed locally, input images can be in arbitrary sizes.

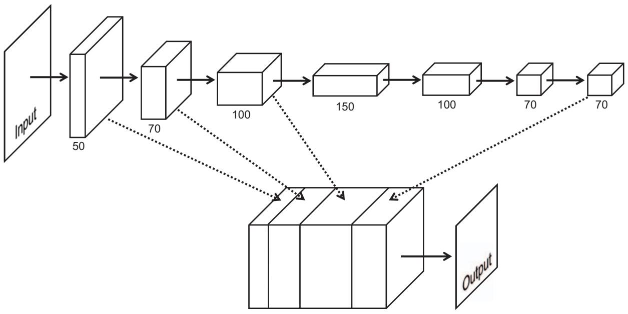

Figure 2. Network architecture. Solid arrows indicate convolutional operations. Each cube represents output feature maps from a stage. Dotted arrows indicate upsampling.

Different from regular ConvNets, there is an integration stage, which upsamples feature maps from the first, second, third, and seventh stages and generates a feature stack. In the feature stack, each unit location has a feature vector consisting of neuron activation at different layers. The feature vectors are fed into a single-layer perceptron classifier to produce a prediction map, which is implemented as a convolutional layer with a $1 \times 1 \times 290$ filter. The prediction map is half the size of input image.

This network has two appealing properties for solar panel detection. First, the network outputs pixel-level predictions, which makes it possible for detecting very small objects. Second, prediction is based on multi-stage features capturing information at different semantic levels. Features from early stages capture low-level information at fine resolutions, such as corners and edges, which are useful for precise localization, while features from late stages capture high-level information from large image patches, such as whether an area is on a roof or a road.

The signed distance function of boundaries are introduced to represent labeled data for training [9]. The signed distance value of each pixel is the distance from the pixel to its closest boundary pixel, and positive/negative signs indicate insider/outside of objects. Since solar panels are sparse, straightforward forms of labeled data including boundary maps and region maps lead to highly imbalanced classes. The signed distance transform essentially converts two-class labels (object/non-object or boundary/non-boundary) into labels with many fine-grained classes, which have a more balanced sample distribution. Meanwhile, compared to boundary maps and region maps, signed distance labels contain more information of spatial layout that can be learned. For example, the signed distance value of -5 (5 pixels away from solar panels) generally corresponds to rooftop-like pixels, while -50 usually roads or parking lots. Such information helps the network better learn to identify solar panels.

In the final stage, we apply 128 filters of size $1 \times 1 \times 290$ to the feature stack, resulting in a prediction vector for each pixel. This is similar to a class distribution in multi-class classification, which is then normalized by the softmax function. Each element of the normalized vector indicates the probability of the pixel within a certain distance range. The cost function is defined as the cross entropy with labeled data rescaled to the 128 integers from -64 to 63. Compared to the cost function built on single-value prediction, this cost function leads to smaller training errors. When applying the network, we take the sum of the 128 integers at each pixel weighted by the normalized prediction vector.

3. Network training

We use orthorectified aerial images with RGB bands covering Washington D.C., San Francisco, CA and Boston, MA, which are respectively taken in 2012, 2012, and 2013. Images of different cities are collected by different sensors at different times and therefore exhibit distinct image characteristics. The image resolution is 0.3 meter, which provides necessary details for detecting most solar panels and has a sufficiently high coverage. Although there exist images with higher spatial resolution, they have lower availability and require more computation for mapping a given region.

To obtain labeled data, three image analysts are assigned to manually delineate solar panels on images, with help of existing databases recording locations of installed solar panels. Around 4000 solar panels were labeled in three days. Based on the labels and images, we compile training data by cropping images around labeled solar panels.

Initially, we create 500×500 image tiles at the original resolution (0.3 meters). We found that the network converges very slowly when trained with such data. The main reason is that the portion of solar panel pixels among all pixels is minimal. Although the signed distance representation yields more balanced class distributions than two-class representations (e.g., object/non-object), the number of solar panel pixels at the original resolution is too small to generate sufficient backpropagation errors after a certain period of training. We take a simple approach to address this problem. We upsample the image through interpolation to achieve half the original resolution, and generate 500×500 image tiles. By doing so, the percentage of solar panel pixels in training data is considerably increased. We create 2040 training images and the corresponding label maps.

Training is based on stochastic gradient descent with 5 images as a mini-batch. We adopt the weight update rule in [6] with learning rate 0.02, momentum 0.9, and weight decay 5^{-4}. We randomly select 1800 images for training and valid on the rest with the metric of average misclassification rate. The system is implemented using Theano [10]. The network is trained using a single NVIDIA Tesla 12GB GPU. The training stops after 140 epochs.

We visually inspect results from the trained network, where most of solar panels are accurately extracted. However, there exist a noticeable amount of false alarms. Given the way we create training images, each image corresponds to an area of 75×75 square meters around solar panels, and images are sparsely distributed. As a result, a multitude of objects are never seen by the network. Although the network is able to correctly reject most of them, it detects some patterns that are similar to solar panels and rarely occur in training data, such as tree shadows on rooftops and zebra crossings. To improve results, we add images with such false alarms to training data. We collect 110 images and label all pixels as -64. The network is initialized with the previously trained parameters and trained on the new dataset for 60 epochs. Despite the small number of extra images, the new model produces a much lower false alarm rate. The two rounds of training in total took roughly four days.

4. Results

We apply the trained network to two images, each of which has $40,000 \times 30,000$ pixels, converting to 108 square kilometers. One image covers the entire San Francisco area, and the other an area in north Boston. Test images cover 18 times the area of training images (12 square kilometers). The San Francisco image has a 4% overlap with training images, and the Boston image is completely separate from training images.

The network processes a 1000×1000 tile at each time. Note that since the network works with any input size, using large input images reduces the overhead for dealing with border effects. Tiles are $2\times$ upsampled, and output maps are downsampled back. In output, pixels with positive values are considered solar panels, except for those forming connected regions with less than 50 pixels. The network takes less than 3 seconds to process a 1000×1000 tile. Each image is completed in one hour using a single GPU.

Extraction results are presented in Figs. 3 and 4, where solar panels are marked in red and overlaid with images. Although the network is exposed to a very limited amount of labeled data, especially negative samples, it works reliably for the two images that have different characteristics caused by imaging sensor properties, illumination, geographic features, etc. In total, around 4,500 solar panels are extracted in San Francisco and around 1,300 in Boston. For a better visual assessment, Fig. 5 shows image patches (not covered by training data) with detected solar panels. As can be seen, solar panels are identified with well-localized boundaries regardless of varying sizes and patterns. Most of errors in the results are false alarms, which are special ground objects that happen to be similar to solar panels in terms of both object patterns and their surroundings. A few examples are shown in Fig. 6. Incorporating more negative samples should further reduce such errors.

For quantitative evaluation, we select an image tile of 5000×5000 pixels in each city that is not within training data and manually generate ground truth. Measuring per pixel accuracy is not practically meaningful in our task. Due to the small size of solar panels, a few pixel mismatches can significantly affect pixel based measurements but may be negligible in applications. We use the following procedure to compute two performance scores, completeness and correctness, which are often used to compare road vectors [11]. We compute centers of detected solar panels and dilate

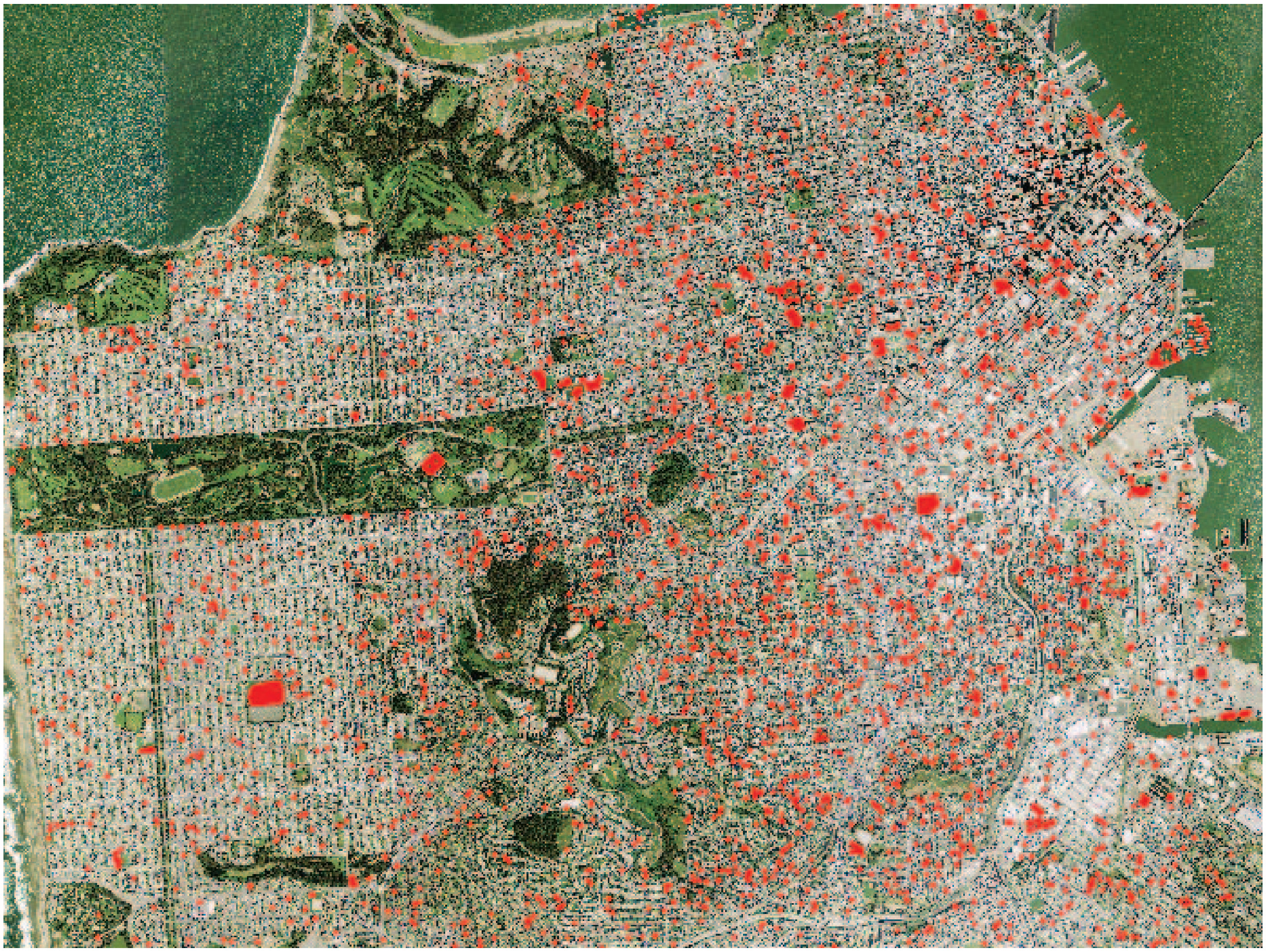

Figure 3. Solar panel mapping result for the image covering San Francisco.

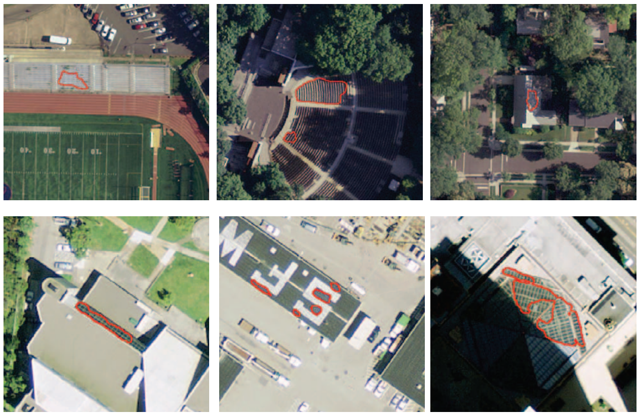

Figure 6. Examples of incorrect extractions.

manual labels by 1 meter. Completeness is defined as the number of manual labels containing center points divided by the total number of manual labels. Correctness is the number of center points inside manual labels divided by the total number of center points. The scores of results are presented in Table 1.

TABLE 1. QUANTITATIVE EVALUATION

Image	Completeness	Correctness
San Francisco	0.873	0.855
Boston	0.840	0.812

The results can further improved by utilizing ancillary data. For example, knowing installed solar panels are generally away from roads, we use road data to filter out incorrect detection. We overlay road vector data from OpenStreetMap[2] with the results. A buffer area is defined around roads. Solar panels overlapped with buffer areas are removed. This simple process increases the correctness rate by 3% and 2% respectively for two images while maintaining almost the same completeness.

[2] https://www.openstreetmap.org/

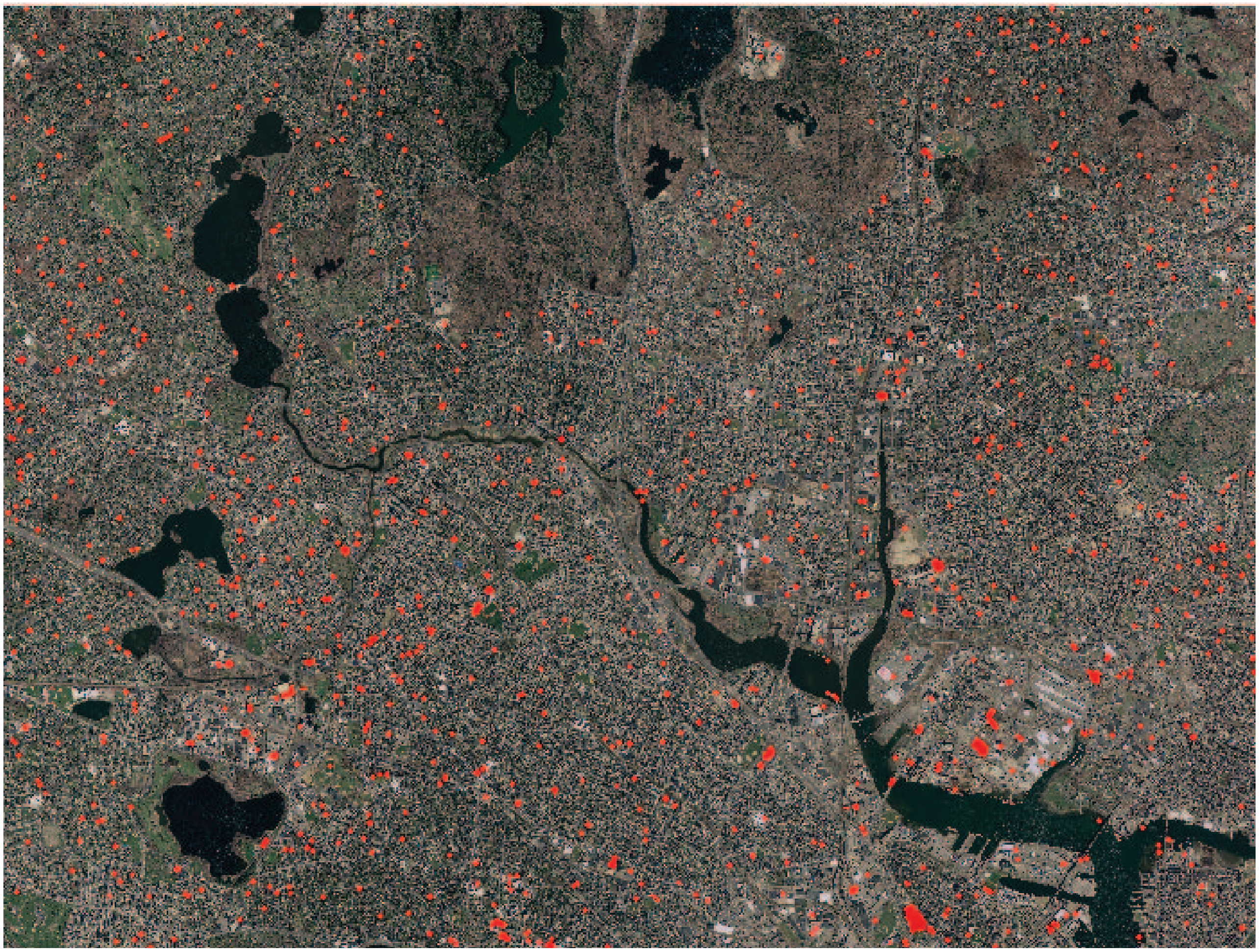

Figure 4. Solar panel mapping result for the image covering Boston.

5. Conclusions

This paper addresses for the first time large-scale solar panel mapping from aerial images. A special ConvNet is utilized, and new training strategies are designed to tackle particular challenges in this task. The trained system is applied to high resolution images covering large areas in two cites. Results show the promise of our approach. Moreover, this work demonstrates that semantic objects captured by a small number of pixels in aerial scenes can be reliably extracted in an automatic way, and our approach can be readily extended to a wide range of objects (e.g., vehicles, road markings, swimming pools, etc.), which will significantly enhance current mapping capabilities.

Acknowledgments

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

The authors would like to thank Thomaz Carvalhaes, Melanie Laverdiere, and Brian Moore for their contribution on manual labeling. Support for this work was provided by the Office of Energy Policy and Systems Analysis at The Department of Energy.

References

[1] S. Dotenco, M. Dalsass, L. Winkler, C. Brabec, A. Maier, F. Gallwitz *et al.*, “Automatic detection and analysis of photovoltaic modules in aerial infrared imagery,” in *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[2] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[3] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, “Learning to detect partially overlapping instances,” in *IEEE Conference on Computer Vision And Pattern Recognition*, 2013.

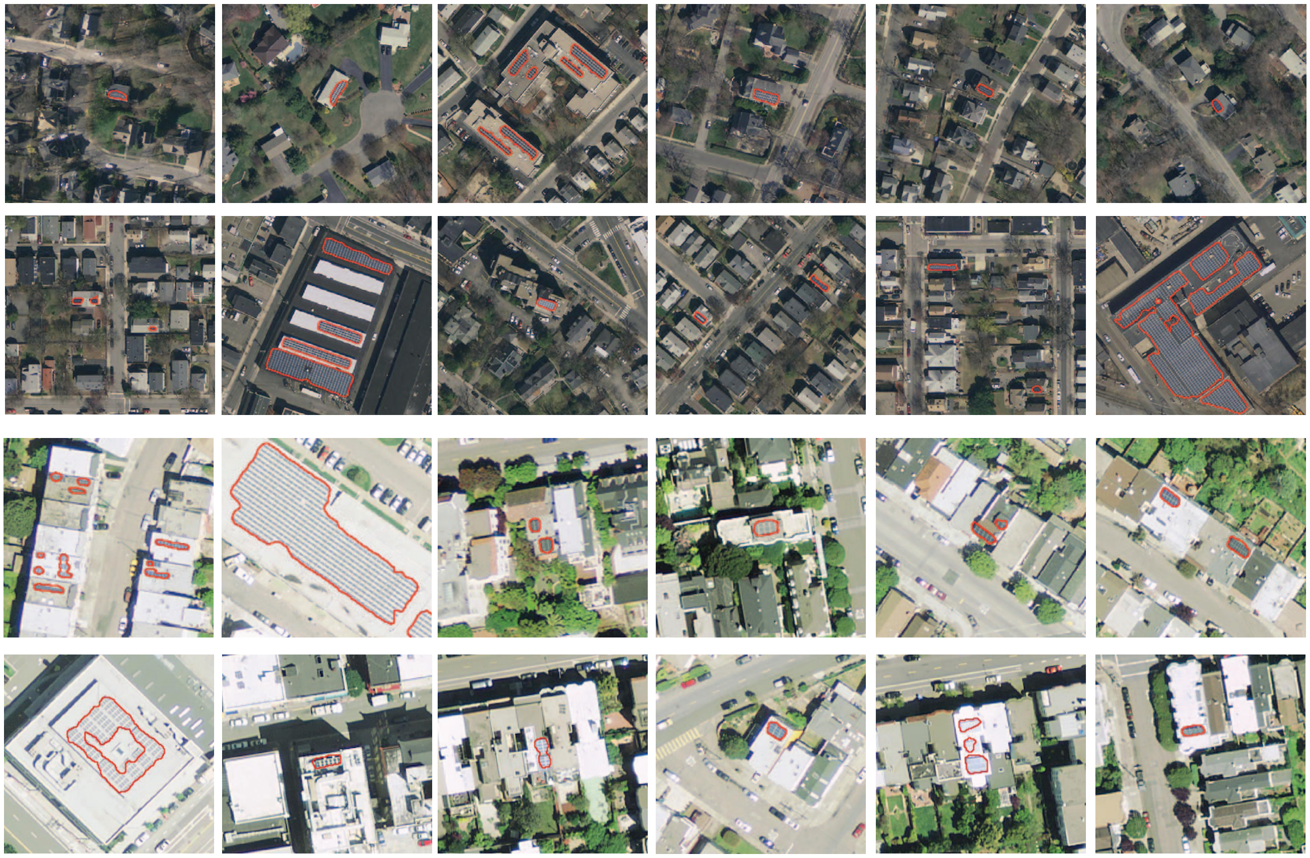

Figure 5. Example results. Boundaries of extracted solar panels are marked in red. Top two rows show images from Boston, and bottom two rows San Francisco.

[4] Z. Ma, L. Yu, and A. B. Chan, “Small instance detection by integer programming on object density maps,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in *International Conference on Learning Representations*, 2014.

[8] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[9] Yuan, “Automatic building extraction in aerial scenes using convolutional networks,” *arXiv:1602.06564*, 2016.

[10] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, “Theano: new features and speed improvements,” *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2012.

[11] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet, “Empirical evaluation of automatically extracted road axes,” in *Empirical Evaluation Techniques in Computer Vision*, 1998.