Identify Solar Panels in Low Resolution Satellite Imagery with Siamese Architecture and Cross-Correlation

Zhengcheng Wang[1, 2]*, **Zhecheng Wang**[1]*, **Arun Majumdar**[1], **Ram Rajagopal**[1]
[1]Stanford University, [2]Tsinghua University
zhengche16@mails.tsinghua.edu.cn, {zhecheng, amajumdar, ramr}@stanford.edu

Abstract

Understanding solar adoption trends and their underlying dynamics requires a comprehensive and granular time-series solar installation database which is unavailable today and expensive to create manually. To this end, we leverage a deep siamese network that automatically identifies solar panels in *historical* low-resolution (LR) satellite images by comparing the target image with its high-resolution exemplar at the same location. To resolve the potential displacement between solar panels in the exemplar image and that in the target image, we use a cross-correlation module to collate the spatial features learned from each input and measure their similarity. Experimental result shows that our model significantly outperforms baseline methods on a dataset of historical LR images collected in California.

1 Introduction

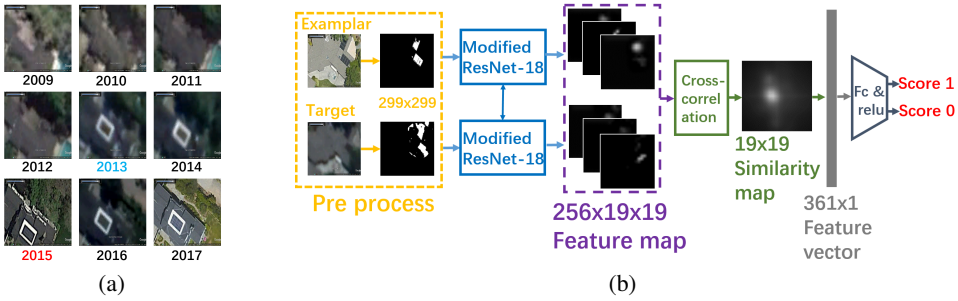
Solar installations are growing worldwide at a rapid pace [1], which contributes to the decarbonization of the energy sector and ultimately helps our world address climate change. However, a high-fidelity *time-series* solar installation database is still unavailable at a granular level (e.g. neighborhood or borough level), which prohibits the comprehensive analysis of solar adoption behaviors and impedes the further promotion and policymaking on solar energy across the world. OpenPV [2] reports incomplete, course-grained, and inefficient-to-update solar installation records in the U.S. based on voluntary contribution. DeepSolar [3] used a deep learning method to automatically detect solar installation in high-resolution (HR) satellite images and thus constructed a nationwide solar installation database efficiently, yet without any information of installation trends. To bridge this gap, one option is to apply machine learning on historical imagery to identify the installation time of existing solar panels. However, historical satellite imagery such as NAIP and OneAtlas all feature: (1) low resolution (usually $>$1m), which is difficult even for human eyes to recognize a solar panel, (2) object displacement across images of different years at the same location, and (3) lack of datapoints in some years requiring combining multiple image sources with different resolution, color distribution, and noise distribution to cover a whole time series. An image sequence example shown in Figure 1(a) demonstrates the challenge in solving this problem with machine learning methods.

To tackle such issues, we propose a deep siamese network to identify solar panels in historical LR "target" image which takes both the target image and a HR "exemplar" image at the same location as inputs. The identification is applied only on locations where existing solar installation has been spotted and thus a HR exemplar image *with* solar installation must be available. Our method is inspired by the fact that in many cases even humans cannot identify solar panels in a LR image, but can compare it with HR image at the same location to make decisions. We demonstrate our method on a real-world historical LR image dataset, and hope that our method will enable the construction of

*Equal contribution

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

a highly granular time-series solar installation database that facilitates: (1) researchers for conducting fine-grained solar adoption trend analysis, (2) solar developer for promoting PV sales with smarter strategies, (3) policymakers for designing and estimating the effects of renewable energy incentives, and eventually enhances the positive impact of solar energy on fighting climate change.



(a) (b)

Figure 1: (a) Image sequence. Red: HR exemplar image. Blue: First LR image with solar installation. (b) Framework architecture.

2 Methods

As is shown in Figure 1(b), our framework takes a target image and its corresponding exemplar as inputs and outputs the softmax score indicating whether the target image contains solar panel or not. The framework consists of: (1) A preprocessing module that binarizes both exemplar and target images highlighting the potential regions of solar panel. (2) A siamese architecture with two identical convolutional neural networks (CNNs). Each is a modified ResNet-18 that extracts visual features from the binary input of either exemplar or target image, and project it into a 3D feature map. (3) A cross-correlation module which integrates the feature maps from both exemplar and target images. The output is then mapped to the final score using a fully connected network.

Preprocessing Module. We leverage hand-crafted features rather than an end-to-end CNN to extract discriminative regions from both exemplar and target images, because the huge variance of resolution, color, and noise in LR images makes CNN difficult to capture the discriminative features of solar panels. By contrast, feature engineering with domain knowledge can remove irrelevant features while preserving the most discriminative regions for downstream neural network. Specifically, we firstly generate a Class Activation Map (CAM) [4] for exemplar image which outlines the discriminative region of solar panels. The CAM is further binarized, dilated, and applied as a mask on both exemplar and target images to highlight the potential region of solar panels. For both images, pixels inside potential regions are binarized to highlight the potential solar panel boundaries. We also utilize color histogram to remove irrelevant colors like green and brown. Detailed steps see Appendix A. After preprocessing, a positive target image should share similar patterns with the exemplar image (Examples in Figure 1(b)), while a negative target image should have different patterns.

Siamese Network. Siamese neural network has been widely used in similarity-related computer vision tasks such as signature verification [5], face verification [6, 7], and one-shot image recognition [8]. It leverages two identical neural networks with shared weights to learn the feature representations of two different inputs. In our work, we use siamese network to process the exemplar-target binary image pair with the aim to determine the similarity between their patterns. Specifically, we use ResNet-18 [9] as feature extractor that projects each binary image into a $256\times19\times19$ feature map. We do not apply deeper CNN and even remove the final convolutional layer of ResNet-18 because the inputs are binary with simple patterns and our dataset is relatively small.

Cross-Correlation Module. Cross-correlation is an operation to measure the similarity of two signals considering displacement, and has been used for object tracking in videos [10, 11, 12]. The inputs of cross-correlation layer are two 3D feature maps and the output is a 2D similarity map which measures the similarity between two feature maps under different displacement. It is mathematically equivalent to sliding one feature map across different positions of the other and calculating the inner product at every position (See illustration in Figure 2). In our work, to tackle the displacement between solar panels in exemplar and that potentially in target, we apply cross-correlation on the two

$256 \times 19 \times 19$ feature maps outputted by the siamese network. Zero-padding size on target feature map is 9. The similarity map outputted by cross-correlation is 19×19 and then projected into the final softmax score with two fully connected layers, each followed by a ReLU.

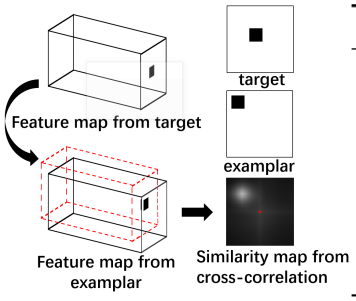


Figure 2: Cross-correlation.

Model	Sens. (%)	Spec. (%)	HMSS(%)
ResNet(RGB)	79.87	68.24	73.60
SN(RGB) +CC	89.83	47.15	61.84
PSN(RGB)+CC	83.12	71.22	76.71
Pair(BI)+CC+HSV	82.82	75.53	79.01
SN(BI)+CC	67.70	96.34	79.47
SN(BI)+HSV	71.13	90.03	79.52
PSN(BI)+CC+HSV	73.54	80.06	76.66
SN(BI)+CC+HSV	84.19	88.82	**86.44**

Table 1: Classification performances on test set.

3 Experiments

We construct a dataset by aggregating historical images from different sources available on Google Earth. Each target image is paired with a HR *positive* exemplar image at same location (Dataset details see Appendix C). We use sensitivity (true positive rate), specificity (true negative rate), and their harmonic mean (HMSS) as metrics to evaluate the classification performance, as the year identification goal requires both sensitivity and specificity to be high (see details in Appendix D).

Table 1 shows the results. **ResNet** means using only target image as input to ResNet-18 without exemplar; **RGB** means inputting raw RGB images while **BI** means using preprocessed binary images; **HSV** means using HSV histogram to remove irrelevant colors; **CC** means cross-correlation module; **SN** means siamese network while **PSN** means pseudo siamese network which is a variant of siamese network with two neural networks using different weights. As a baseline model, **ResNet(RGB)** takes only raw target image as input and yields unsatisfactory performance, indicating that capturing information from LR image alone is insufficient. **SN(RGB)+CC** performs even worse while **PSN(RGB)+CC** is better, suggesting that: (1) siamese network with identical weights cannot process raw HR and LR images together because of their huge discrepancy; (2) including exemplar image for comparison with PSN can indeed enhance the performance. **SN(BI)+CC** outperforms **PSN(RGB)+CC** by 2.8% in HMSS, validating the effect of binarization. **SN(BI)+HSV** concatenates the features of exemplar and target for prediction, and its HMSS is 6.9% less than that of **SN(BI)+CC+HSV**, demonstrating that cross-correlation can significantly improve the performance because of its translation-insensitive property. **SN(BI)+CC+HSV** outperforms **SN(BI)+CC** by 7.0% in HMSS, indicating the positive effect of removing irrelevant colors from inputs as a denoising process. **Pair(BI)+CC+HSV** applies cross-correlation directly on preprocessed binary images without siamese network, and its inferior HMSS shows that siamese network can extract higher-level features from the preprocessed binary images and increase the performance. Comparing the performances of **SN(BI)+CC+HSV** and **PSN(BI)+CC+HSV**, we find that for binary images with similar style, siamese network is better than pseudo siamese network since it contains less trainable parameters thus it is less likely to overfit. To summarize, our proposed model **SN(BI)+CC+HSV** significantly outperforms all other models, demonstrating its strength in LR solar panel identification. In time-series task that identifies installation year, this model can achieve 53.2% zero-year deviation rate and 75.1% one-year deviation rate (See metric explanation and comparison in Appendix D).

4 Conclusion

In this paper, we propose a framework which combines hand-crafted feature extraction, siamese architecture, and cross-correlation to identify solar panels in the historical satellite images that suffer from low resolution, object displacement, and heterogeneous image styles. In the future, we will further improve its performance in determining installation years for solar panels in image sequences, and eventually apply this method to construct a large-scale, fine-grained solar installation database with time-series information supporting both predictive and causal analysis of solar adoption.

Acknowledgements

We thank Amazon Web Service for offering cloud computing credits. Zhengcheng Wang was supported in part by the Tsinghua Academic Fund for Undergraduate Overseas Studies. Zhecheng Wang was supported by the Stanford Interdisciplinary Graduate Fellowship as Satre Family Fellow.

References

[1] IEA, "Renewables 2018: Analysis and forecasts to 2023," tech. rep., 2018.

[2] National Renewable Energy Laboratory, "The Open PV Project." https://openpv.nrel.gov.

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

[4] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, pp. 737–744, 1994.

[6] S. Chopra, R. Hadsell, Y. LeCun, *et al.*, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR (1)*, pp. 539–546, 2005.

[7] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

[8] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865, Springer, 2016.

[11] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813, 2017.

[12] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291, 2019.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

Appendix A: Preprocessing

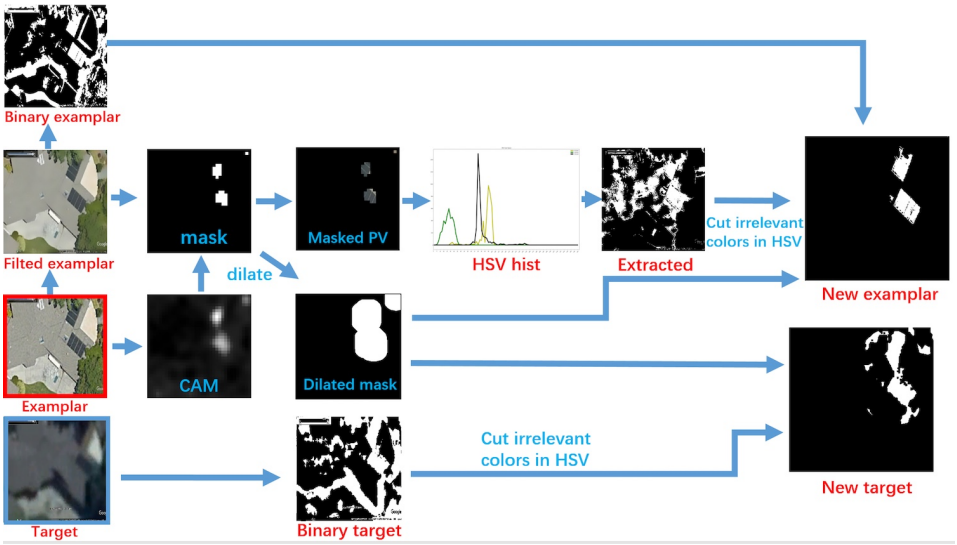


Figure 3: Flow chart of preprocessing

Figure 3 shows the flow chart of preprocessing module:

For exemplar image: The preprocessing of exemplar image is comprised of 6 steps:
1) Obtain a dilated mask: Apply DeepSolar model [3] on exemplar image to generate its Class Acitvation Map (CAM) showing the activated region of solar panel, and then binarize it to get a mask which shows the specific region of solar panel. In order to extract complete regions of solar panel in latter steps and consider the displacement issue, we dilate the mask to obtain a larger mask.
2) Binarization: We perform bilateral filtering on the exemplar image and then perform binarization with adaptive threshold on filtered exemplar image to enhance the sharpness of solar panel boundary, generating a binary exemplar.
3) Extract similar color of solar panels: Overlay the pre-dilated mask (from step 1) on filtered exemplar to extract the solar panel region. After that, convert the pixels in this region from RGB space to HSV space, which is often used for color extraction in the image processing field. Calculate the histogram of H,S,V channels, which roughly represents the HSV distribution of solar panels. Then extract those pixels whose HSV value is in the distribution region. Finally, convert extracted image to binary (set extracted pixels to white and others to black)
4) Combine images from step 1-3 together: Perform logic "and" on: dilated mask from step 1, binary exemplar from step 2, and extracted binary image from step 3.
5) Remove irrelevant colors: Some colors are not likely to appear in a solar panel in both exemplar and target image, such as brown, green, black, and white. Therefore, we remove the pixels of these colors in HSV space by setting these pixels to be black.

For target image: The preprocessing of target image is comprised of 3 steps:
1) Binarization: Perform binarization with adaptive threshold on target image and get a binary target.
2) Dilated mask: Apply dilated mask extracted from exemplar on binary target.
3) Remove irrelevant colors: Subtract area with irrelevant colors using the same steps as in exemplar preprocessing.
After preprocessing, we finally obtain a pair of binary exemplar and target in which the potential pixels of solar panel area are extracted while others are removed as much as possible. This will facilitate the downstream neural network for extracting more discriminative features.

Appendix B: Examples of similarity maps outputted by cross-correlation

See some examples of similarity maps outputted by cross-correlation in Figure 5.

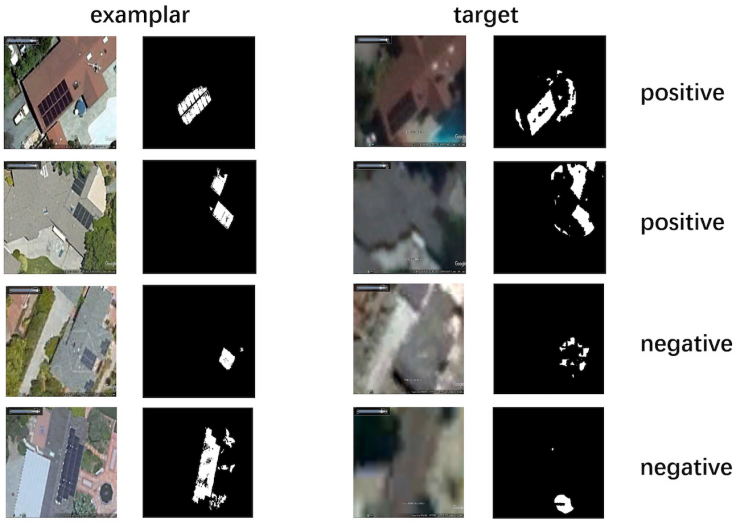


Figure 4: Some examples before and after preprocessing.

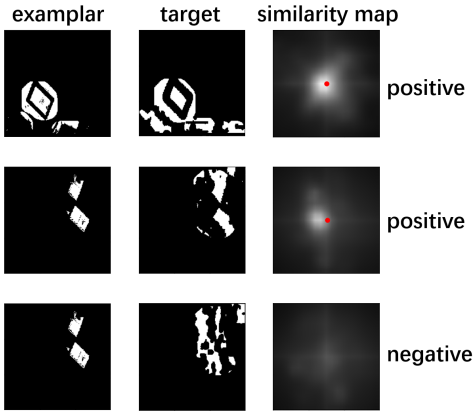


Figure 5: Some examples of the similarity maps outputted by cross-correlation. Left column: binary exemplars. Middle column: binary targets. Right column: similarity maps. First row: positive sample with no displacement. Second row: positive sample with displacement. Third row: negative sample.

Appendix C: Dataset information

We collect historical satellite images in California from multiple sources available on GoogleEarth as our dataset. See dataset statistics in Table 2. Whether an image is counted as high-resolution (HR) or low-resolution (LR) is determined by structural similarity (SSIM) index [13]. We run the original DeepSolar model on all HR images in an image sequence to determine the *positive* HR exemplar image of that sequence. We manually label the installation year for the image sequences in our dataset. For each image sequence, any LR image taken before the installation year is counted as negative, and any LR image after that year is counted as positive.

Table 2: Dataset statistics

Type	Positive	Negative
Train	2880	6822
Validation	321	759
Test	291	331

Appendix D: Identifying installation year in an image sequence

We ignore the rare cases that solar panel can be uninstalled later after installation. Under this assumption, if we use "0" to denote negative sample and "1" to denote positive sample, our image sequence is a sequence with all "0" in the first part and all "1" in the last part. There cannot be "1" between "0" such as "00100".

For a single image whose groundtruth label is 0, its probability of being predicted correctly is:

$$P_0 = \frac{\text{TN}}{\text{TN} + \text{FP}} = \text{specificity} \tag{1}$$

And for a single image whose groundtruth label is 1, its probability of being predicted correctly is:

$$P_1 = \frac{\text{TP}}{\text{TP} + \text{FN}} = \text{sensitivity} \tag{2}$$

If we have an image sequence with first n images as negative and last m images as positive, then the probability of predicting the whole sequence correctly is:

$$P_{sequence} = {P_0}^n {P_1}^m = (\text{specificity})^n (\text{sensitivity})^m \tag{3}$$

We apply our proposed model **SN(BI)+CC+HSV** on all image sequences in the test set and it achieves 53.2% zero-year deviation rate and 75.1% one-year deviation rate for installation year identification. Here zero-year deviation rate is the ratio of sequence samples where the installation year estimated by the model is exactly the same as the labeled one, and one-year deviation rate is the ratio of sequence samples where the installation year estimated by the model is *no more than one year earlier or later* than the labeled year. By contrast, the original DeepSolar model can only achieve 29.8% zero-year deviation rate and 52.1% one-year deviation rate in the same task. We aim to improve installation year estimation accuracy further in the future work.