A season independent U-net model for robust mapping of solar arrays using Sentinel-2 imagery

Michael Evans[1,2], Taylor Minich[1,3], Rachel Soobitsky[4], Kumar Mainali[4,5]

1.Center for Conservation Innovation, Defenders of Wildlife, Washington DC

2.Environmental Science and Policy Dept., George Mason University, Fairfax VA

3. Environmental Protection Agency, Washington DC

4. Conservation Innovation Center, Chesapeake Conservancy, Annapolis MD

5. Department of Biology, University of Maryland, College Park, MD

corresponding author(s): Michael Evans (mevans@defenders.org)

Keywords

Computer Vision; Deep Learning; Earth Engine; Remote Sensing; Renewable Energy; Tensorflow

Abstract

We have an unprecedented ability to map the Earth's surface as deep learning technologies are applied to an abundance of high-frequency Earth observation data. Simple, free, and effective methods are needed to enable a variety of stakeholders to use these tools to improve scientific knowledge and decision making. Here we present a trained U-Net model that can map and delineate ground mounted solar arrays using publicly available Sentinel-2 imagery, and that requires minimal data pre-processing and no feature engineering. By using label overloading and image augmentation during training, the model is robust to temporal and spatial variation in imagery. The trained model achieved a precision and recall of ~90% each and an intersection over union of 84.3% on independent validation data from two distinct geographies. This generalizability in space and time makes the model useful for repeatedly mapping solar arrays. We use this model to delineate all ground mounted solar arrays in North Carolina and the Chesapeake Bay watershed to illustrate how these methods can be used to quickly and easily produce accurate maps of solar infrastructure.

Background & Summary

The proliferation of Earth observation (EO) data has enabled the mapping of features on the Earth's surface at an extraordinary temporal frequency and level of detail. Publicly available images of the entire Earth are collected every 5 days[1] and global images with ≤ 1 m resolution are collected daily by private companies. With these data, researchers are mapping the distribution and state of forest loss[2], urban growth[3], water resources[4] and other changes to the Earth's surface with unprecedented accuracy. These endeavours are critical to better understand and conserve natural resources[5]. Even greater advances in mapping capabilities arise from the combination of EO data with deep learning approaches[6–8]. Yet the ability to apply these tools to environmental challenges remains out of reach for many organizations, as early progress has largely focused on high-resolution imagery and large training datasets that require substantial financial and computing resources[8]. Thus, accessible approaches are needed to unlock the impacts possible from the combination of EO data and deep learning techniques for the purposes of conservation.

The application of computer vision to EO data has expanded the precision with which researchers are able to map the Earth's surface. Computer vision describes the automated identification of objects in images, often using deep learning models to recognize, locate, and delineate objects like cats, cars, and faces in photographs [9]. More recently, these same models have been applied to satellite and aerial imagery to identify and classify land cover [7,10]. In terms of generating geospatial data, the most important development has been the application of convolutional neural networks (CNNs) – deep learning architectures that can take advantage of the shape and context of an object – marking a transition away from traditional pixel-based classification approaches like those historically used to create land cover maps[11]. Incorporating information about spatial context can be critical for distinguishing features with similar spectral characteristics. For instance, landslides may be spectrally similar to anthropogenic land clearing, but the two are distinguishable based on shape, configuration, and landscape context[12].

Given the ability of CNNs to delineate specific objects, a process known as image segmentation, many variants of this framework have been applied to various EO data sources[7,13]. Thus far, many of these advances have focused on model development and training feature engineering, often in specific use cases taken at a snapshot in time. In this paper, we focus on advancing the breadth of applicability of computer vision and remote sensing to conservation challenges relative to historical approaches that can be resource intensive and specialized. To be of maximal use to conservation, computer vision models using EO data need to:

1. Use publicly available data
2. Use data that is updated regularly
3. Be free or of low cost to implement
4. Achieve sufficient accuracy with limited training datasets
5. Generalize to work in a variety of geographical temporal contexts

Methods and models must meet these five criteria to enable conservationists to regularly use them to create actionable outputs. To address these needs, we sought to develop simple, replicable computer vision models that delineate features of interest using free, publicly available earth observation data. This has been made possible with the launch of Google Earth Engine[14] and its integration with the Tensorflow machine learning library[15], as well as the Microsoft Planetary Computer and it's integration withing the Microsoft Azure platform. As a demonstration of concept, and to inform ongoing conservation analyses, we develop and present deep learning workflows that can be used to map ground mounted solar arrays.

As solar energy proliferates, conservationists are interested in understanding the impacts of solar arrays on natural ecosystems and in promoting development in environmentally conscious manner. Such an approach requires understanding the current extent of renewable energy infrastructure, as well as quantifying past patterns of development. Solar energy production has grown rapidly in the United States over the past decade, experiencing a 35-fold increase from 2008 to 2018[16]. Decision makers and researchers need updated maps of solar arrays to understand their impact on habitat availability and fragmentation, and to identify opportunities in which solar can be simultaneously used for habitat restoration benefiting native species[17,18]. Additionally, maps of solar arrays at past time points can facilitate an understanding of patterns in the growth and development of this infrastructure.

Here we present a trained U-Net – a well-studied fully-convolutional architecture for image segmentation [19] – model to delineate ground-mounted solar arrays at landscape scales using public data and low-cost resources. We demonstrate the model within North Carolina and the Chesapeake Bay watershed, which includes parts of New York, Pennsylvania, Maryland, Delaware, and Virginia. North Carolina produces the second most solar energy in the United States (after California), generating more than 4% of its annual energy from solar arrays following substantial growth over the past decade[16] – and there is interest in understanding the current configuration of arrays relative to wildlife habitat. In the Chesapeake Bay, there is interest in quantifying how changes in impervious surface may affect hydrological dynamics. The output data are being used in ongoing conservation and land use research related to the growth of solar energy, and the robustness of the model indicates utility in other geographic areas.

Methods

Model Structure

Our implementation of U-Net consisted of 5 consecutive encoder blocks, which increase the feature space of the data while reducing spatial resolution, and 5 decoder blocks that restore spatial detail[20]. Each encoder block was comprised of two sequences of a convolutional layer, a batch normalization layer and a rectified linear unit activation, followed by a max pooling step reducing spatial resolution. Decoder blocks consisted of a deconvolution layer that increase spatial resolution, the output of which was concatenated with output from the reciprocal decoder layer, followed by two sequences of convolution, batch normalization, and rectified linear unit activation. The final layer contained a convolutional layer with sigmoid activation function constraining output values to [0-1]. This model structure takes in three-dimensional image data, in which the last dimension contains features used to make predictions, and outputs an array of probabilities with first and second dimensions equal to the input image.

Training Labels

Our analyses focused on mapping ground mounted solar arrays in the state of North Carolina and within the Chesapeake Bay watershed. We hereafter refer to ground mounted solar arrays as solar arrays. We hand digitized 663 solar arrays within North Carolina and 395 arrays from within the Chesapeake Bay watershed as geospatially referenced polygons[20] using a combination of Sentinel-2 and basemap imagery available in Google Earth Engine. We converted these polygons into a single band 'label' raster with pixel values equal to 1 in areas covered by a solar array and 0 elsewhere (Fig. 1).

Earth Observation Data

We trained our U-Net model to delineate solar arrays using top of atmosphere reflectance data collected by the Sentinel-2 imaging system[1] as input. Sentinel-2 data are publicly available and contain 13 multispectral bands collected at 10, 20, or 60 m resolution. Using Google Earth Engine, we accessed all Sentinel-2 images intersecting the boundaries of North Carolina, Virginia, Maryland, Delaware, Pennsylvania, and New York collected between 2020-01-01 and 2020-12-31 in which less than 20 percent of pixels were labelled as cloudy in the image metadata. We then masked clouds from these images using the included quality assurance band, which flags cloudy pixels. Finally, we selected values from the three visible red, green, blue (RGB), near infrared (NIR), and two short-wave infrared bands (SWIR1 & SWIR2) as input variables.

To make the model robust to phenological variability, we applied label overloading[21], by first separating the collection of 2020 Sentinel-2 images into four seasonal collections: Spring (01Mar20 - 31May20); Summer (01Jun20 – 31Aug20); Fall (01Sep20 – 30Nov20); and Winter (01Dec20 – 28Feb20). Each of these four collections were then reduced to a single image per season by taking the median value at each pixel among all images in the collection. Thus, we generated four images containing six bands of Sentinel-2 reflectance data covering the study areas. We stacked the label image onto each of these four 6-band Sentinel-2 images to create four 7-band rasters with input variables and labels for each season (Fig. 1). We then sampled image chips - 256×256x7 arrays - from each of these four images.

Because solar arrays were relatively sparse features on a landscape, we took two steps to ensure our model had enough positive examples from which to learn to recognize these features. First, we sampled image chips at the centroids of each digitized solar array. We then generated a random sample of 1000 points within 5km of these features and sampled chips at each point. The resulting sets of chips were divided into 70% model training and 30% evaluation sets.

Model Training & Evaluation

We used these image chips to train the U-Net model. During training we rescaled image chips to standardize model input. For each incoming chip, we normalized the variable values contained in each pixel using per-pixel means and variances. Thus, the reflectance values among bands at each pixel had mean = 0 and st dev = 1. We additionally implemented morphological and spectral image augmentation to artificially inflate the variability of training data. At training time, we randomly applied a rotation of 0, 90, 180, or 270 degrees as well as random horizontal and vertical flips to image chips including multispectral training and label data[22]. We then separated the label band from multispectral bands and augmented the colour of the latter by randomly adjusting the brightness and contrast by +/- 5%.

We trained the U-Net model using Keras with Tensorflow backend using batches of 16 chips per training step for 100 epochs, optimizing a weighted binary cross entropy loss function[19,23] using the Adam optimizer with initial learning rate of 1e-4 and a decay rate of $\beta_1 = 0.9$, $\beta_2 = 0.999$. We set the positive sample weight to 20, based on the observed ratio of positive and negative pixels included in our training data. At the end of each training epoch, we evaluated model performance on the evaluation data set in terms of intersection over union (IoU) between predictions and labels. During model training we automatically saved the model weights that maximized IoU. We used Tensorboard[15] to track model performance during training and evaluation.

After training the model for 100 epochs, we visually inspected model predictions in four 1800 km^2 test areas in North Carolina to quickly assess whether the model was over- or under-predicting solar arrays. These four validation areas were selected to sample each of the four Sentinel orbit paths covering the state. Within each we performed a complete visual inspection to delineate by hand all solar arrays, yielding 146 polygons. We continued training the model for an additional 100 epochs, reducing the positive weight parameter of our binary cross entropy loss function to 0.8 to reduce a high occurrence of false positives observed in the initial prediction outputs.

After training the model for 200 total epochs, we tested the predictive performance of the resulting model by mapping all solar arrays in North Carolina and the Chesapeake Bay watershed, which included images that were used neither during training nor evaluation. First, we used the trained model to generate a raster of solar array probabilities using Sentinel-2 data from summer 2020 covering NY, PA, MD, DE, VA, and NC. To generate predicted probabilities, we split Sentinel-2 images into overlapping 512x512 pixel chips in which the outermost 128 pixels on each edge were shared between adjacent chips. These chips were passed to the trained U-Net model to obtain 512x512 probability array outputs. The 128-pixel edges were then discarded, yielding contiguously adjacent 256x256 patches. We converted these probabilities to predicted solar array polygons using a 0.9 probability threshold.

We calculated the recall (i.e., true positive rate) precision (i.e., false positive rate) and IoU of model output polygons relative to manually verified ground truth data. For predictions in North Carolina, we recorded each output polygon as either a true or false positive using 2020 Sentinel-2 imagery as a reference. We also used 1m-resolution NAIP imagery[24] to verify cases where the 10m-resolution Sentinel-2 imagery was unclear. We considered outputs that partially captured solar arrays visible in reference imagery as true positives. We used hand-digitized arrays in the four validation images to estimate false negative rates and calculate IoU. In the case of missing or incomplete output polygons, we include the ground truth polygons in the output polygon data[20].

For output polygons in the Chesapeake Bay watershed, we contracted Washington College's GIS program to perform a similar ground truth and accuracy estimation procedure. Four students were each assigned a different geographical area within the watershed to review. As in North Carolina, students labelled each output polygon as either a true or false positive, using 2020 Sentinel-2 imagery and the most recently available NAIP imagery in each state as reference. To estimate false negatives, the students used a solar array point dataset from the Solar Energy Industries Association (SEIA) and the Energy Information Administration (EIA) to identify and digitize missing solar arrays in the model outputs. We compiled the counts of true positive, false positive, and false negative and compiled these data in a confusion matrix, from which we calculated recall (i.e., true positive rate) and precision (i.e., false negative rate). We used hand-delineated polygons to calculate IoU.

All data preparation and sampling were performed using Google Collaboratory notebooks with a Python 3 runtime. Model training and prediction were run through Microsoft Azure Machine Learning Studio on a NC6 machine with 56 GB or RAM, and an NVIDIA Tesla K80 GPU.

Data Records

We provide the training data, trained U-Net model and weights files, and output solar array data in an Open Science Framework repository[20].

Sentinel-2 data used to train and evaluate the model are available in TFRecord format. Each file holds several 256x256x7 image chips containing seven features in float format: ‘B2’, ‘B3’, ‘B4’, ‘B8’, ‘B11’, ‘B12’, and ‘landcover,’ corresponding to the Sentinel-2 reflectance values and label. The polygons used to generate training data are also available in geojson files.

The trained U-Net model is provided in an h5 file, and model weights are provided in an hdf5 file, which can be directly loaded into Keras.

Output solar array data consist of georeferenced polygons stored in geojson and shapefile format. We provide a geojson file for each set of predictions in North Carolina in 2016, 2018, 2019, and 2020, and the Chesapeake Bay watershed in 2020. These spatial data use the EPSG 3857 web Mercator coordinate reference system.

Technical Validation

At the end of training, the U-Net solar array model had a mean IoU of 85.9% as calculated by Tensorflow on evaluation data. Using hand digitized ground truth data, the model predictions across North Carolina and the Chesapeake Bay watershed had a mean IoU of 84.3%, a recall/sensitivity rate of 89.9% and a precision score of 90.7% (Table 2).

Usage Notes

The model and weights file can be used in a Keras with Tensorflow backend framework to delineate solar arrays within Sentinel-2 imagery. The model expects 512 x 512 x 7 inputs containing Sentinel-2 bands 2,3,4,8,11,12. Collaboratory notebooks are included in a GitHub repository[25], a snapshot of which is also permanently stored in an OSF repository[20], that provide working examples demonstrating the model training process, the generation of prediction images using Google Earth Engine, and the usage of the model to generate solar array polygon outputs. We recognize that training deep-learning models is an energy intensive process with a measurable carbon footprint[26], and encourage researchers and decision makers to use this or other trained solar array segmentation models to generate maps of solar arrays, rather than re-training or training new models from scratch.

Code Availability

All code used to generate images, sample training data, train models, and run predictions is available in the GitHub repository [25]. A persistent snapshot of this repository is provided in an Open Science Framework repository[20].

Acknowledgements

We thank the Washington College GIS program for performing manual validation of model outputs. J. W. Malcom and S. Minnemeyer provided helpful review of the manuscript. Recent work in the field of conservation science and STEM at large has identified a bias in citation practices such that papers from women and other minorities are relatively under-cited. While we did not proactively choose references that reflect the diversity of the field in thought, form or contribution, gender and other factors for this work, we recognize the biases that may have been

unintentionally introduced. While we do not believe these biases influenced the development of our U-Net model, they may impact the interpretation and application of its outputs.

Author contributions

M. J. Evans conceptualized the study, developed image sampling and processing methodology, lead model training and predictions, and lead manuscript writing.

T. Minich digitized solar arrays in North Carolina and contributed to image processing and sampling to generate model training and evaluation data.

R. Soobitsky provided digitized solar array polygons from within the Chesapeake Bay watershed and organized and supervised the quality assessment process.

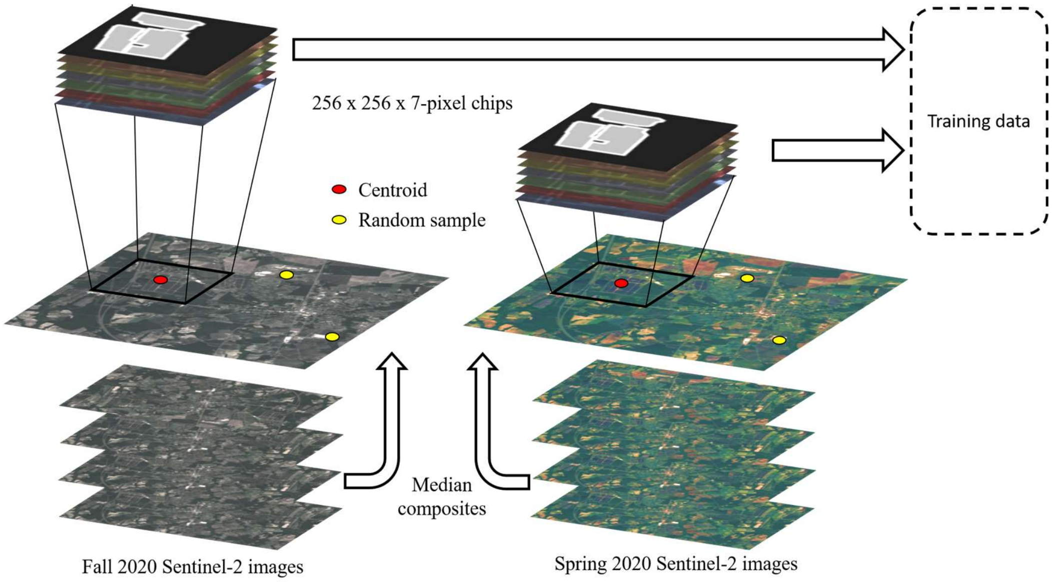
K. Mainali analysed quality assessment data and contributed to manuscript writing.

Competing interests

The authors declare no competing interests related to this work.

doi:10.20944/preprints202005.0345.v2

Figures


256 x 256 x 7-pixel chips
Training data
● Centroid
● Random sample
Median
composites
Fall 2020 Sentinel-2 images
Spring 2020 Sentinel-2 images

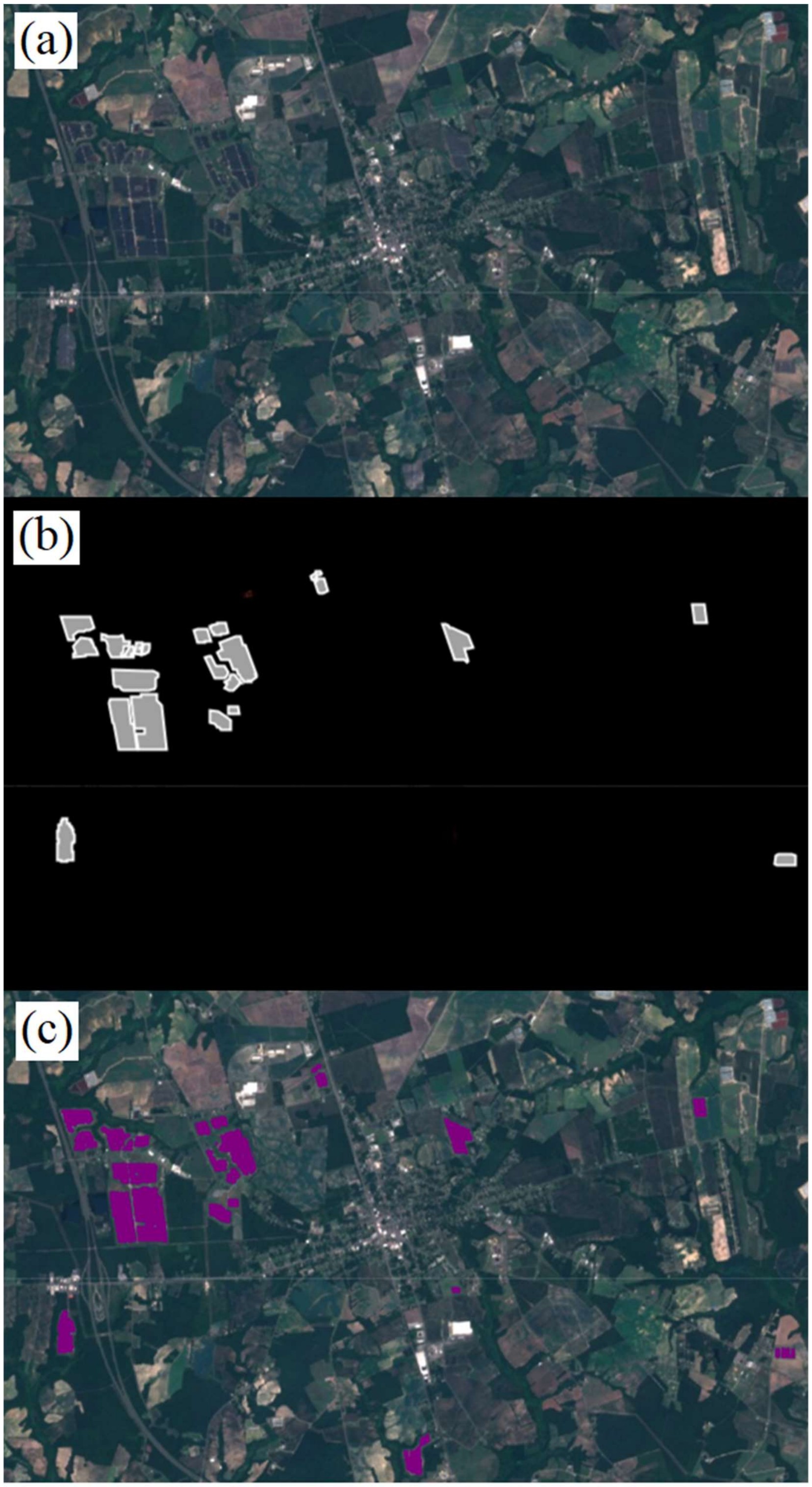

(a)
(b)
(c)

Figure Legends

Figure 1. Flow-chart showing data collection process used to train deep learning models delineating solar arrays and parking lots. Steps contained within the dashed border are performed using Google Earth Engine, and consist of dividing imagery into seasonal collections (when available), creating median composites per collection, and sampling these composites at target feature centroids and random locations.

Figure 2. Solar arrays delineated by a U-Net convolutional neural network using Sentinel-2 data. Panels flow from top to bottom showing the raw imagery used to generate training features (a), ground truth labels (b), and model output predictions (c).

Tables

Table 1. Post-hoc evaluation of model performance. The confusion matrix contains the frequency with which solar arrays delineated by the trained U-Net model in test areas of North Carolina and within the Chesapeake Bay watershed aligned with ground truth data in each region.

	Present			Absent		
	NC	Chesapeake	Total	NC	Chesapeake	Total
Present	123	482	**605**	23	45	**68**
Absent	17	45	**62**	-	-	-

References

1. Drusch, M. *et al.* Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ.* **120**, 25–36 (2012).

2. Song, X.-P. *et al.* Global land change from 1982 to 2016. *Nature* **560**, 639–643 (2018).

3. Watson, J. E. M. & Venter, O. Mapping the Continuum of Humanity's Footprint on Land. *One Earth* **1**, 175–180 (2019).

4. Pekel, J. F., Cottam, A., Gorelick, N. & Belward, A. S. High-resolution mapping of global surface water and its long-term changes. *Nature* **540**, 418–422 (2016).

5. Evans, M. J. & Malcom, J. W. Supporting habitat conservation with automated change detection in Google Earth Engine. *Conserv. Biol.* (2021) doi:10.1111/COBI.13680.

6. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* vol. 521 436–444 (2015).

7. Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F. & Zhang, Y. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sens.* **10**, 1119 (2018).

8. Li, J., Huang, X. & Gong, J. Deep neural network for remote-sensing image interpretation: Status and perspectives. *Natl. Sci. Rev.* **6**, 1082–1086 (2019).

9. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *3rd Int. Conf. Learn. Represent.* (2015).

10. Nogueira, K., Penatti, O. A. B. & dos Santos, J. A. Towards better exploiting

convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **61**, 539–556 (2017).

11. Jin, S. *et al.* Overall methodology design for the United States national land cover database 2016 products. *Remote Sens.* **11**, (2019).

12. Ghorbanzadeh, O. *et al.* Evaluation of Different Machine Learning Methods and Deep-Learning Convolutional Neural Networks for Landslide Detection. *Remote Sens.* **11**, 196 (2019).

13. Wiratama, W., Lee, J. & Sim, D. Change Detection on Multi-Spectral Images Based on Feature-level U-Net. *IEEE Access* **8**, 12279–12289 (2020).

14. Gorelick, N. *et al.* Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sens. Environ.* **202**, 18–27 (2017).

15. Abadi, M. *et al.* Tensorflow: Large-scale machine learning on heterogeneous systems. (2015).

16. Margolis, R., Feldman, D. & Boff, D. Solar Industry Update. *Natl. Renew. Energy Lab.* 1–83 (2018) doi:NREL/PR-6A20-68425.

17. Beatty, B., Macknick, J., Mccall, J., Braus, G. & Buckner, D. Native vegetation performance under a solar PV array at the National Wind Technology Center. 54 (2017) doi:Contract No. DE-AC36-08GO28308.

18. Sinha, P., Hoffman, B., Sakers, J. & Althouse, L. Best Practices in Responsible Land Use for Improving Biodiversity at a Utility-Scale Solar Facility. *Case Stud. Environ.* **2**, 1.16-12 (2018).

19. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in *arXiv* vol. 9351 234–241 (2015).

20. Evans, M. J., Minich, T., Soobitsky, R. & Mainali, K. Solar Array Computer Vision. (2021) doi:10.17605/OSF.IO/6RPHN.

21. Robinson, C. *et al.* Large scale high-tesolution land cover mapping with multi-resolution data. *2019 IEEE Conf. Comput. Vis. Pattern Recognit.* 12718–12727 (2019) doi:10.1109/CVPR.2019.01301.

22. Peng, D., Zhang, Y. & Guan, H. End-to-end change detection for high resolution satellite images using improved UNet++. *Remote Sens.* **11**, 1382 (2019).

23. Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Jorge Cardoso, M. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **10553 LNCS**, 240–248 (2017).

24. Boryan, C., Yang, Z., Mueller, R. & Craig, M. Monitoring US agriculture: the US Department of Agriculture, National Agricultural Statistics Service, Cropland Data Layer Program. *Geocarto Int.* **26**, 341–358 (2011).

25. Evans, M. J. Satellite_ComputerVision. https://github.com/mjevans26/Satellite_ComputerVision (2019).

26. Wolff Anthony, L. F., Kanding, B. & Selvan, R. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models.