

Predicting PV Areas in Aerial Images with Deep Learning

Matthias Zech

DLR Institute of Networked Energy Systems
Oldenburg, Germany
Matthias.Zech@dlr.de

Joseph Ranalli

Penn State Hazleton
Hazleton, PA
jar339@psu.edu

Abstract—Data on the location of distributed photovoltaic installations are valuable to a variety of research activities. We have trained and applied a Fully Convolutional Neural Network to identify PV sites from aerial images of Oldenburg, Germany acquired from Google Maps. The architecture used was U-net, which was trained on a set of manually labelled images, and verified against a test dataset. The model is able to accurately estimate location and shape of PV plants in the north European town of Oldenburg. In addition, the model is able to estimate its own uncertainty, breaking the black box assumption of Deep Learning.

Index Terms—Fully Convolutional Neural Network, model uncertainty, PV identification, Deep Learning

I. INTRODUCTION

Distributed photovoltaics (PV) represent an area of growth for renewable energy generation. Due to the diurnal cycle, clouds, and other factors, distributed PV have a highly variable generation. This means that distributed PV would be expected to have an impact on grid operations that is different from more traditional forms of electricity generation. It is important to develop a complete understanding of distributed renewable generation in order to empower energy planners and grid operators to accommodate growth of distributed PV generation, while continuing to provide reliable electricity to customers.

A great deal of research has focused on improving our ability to predict the impacts of distributed generation on the electricity grid. One shortcoming, however, is that data about the location and capacity of distributed PV installations may be limited or unavailable. Rooftop PV installations in particular may be difficult to find complete data for, due to customer privacy concerns and lack of availability of information at a granular level. Depending on the specific location, rooftop PV installations may represent a substantial share of installed capacity. For example, in Germany, single-site PV under 100 kW installed capacity accounts for 52% of installed capacity [1]. This accounts for most residential and commercial PV systems.

Previous studies have considered solving the problem of missing small scale PV data by utilizing computer vision and Neural Network (NN) approaches to identify PV sites from aerial images [2]–[7]. In this study, we train a Fully Convolutional Neural Network to identify PV systems from aerial images.

II. METHODOLOGY

The purpose of this study is to detect and localize areas covered by PV panels in aerial images. In Computer Vision, this kind of problem is called semantic segmentation, as it assigns each pixel of an input image to a respective class. In contrast to instance segmentation, the model does not distinguish between different PV panels, but rather, identifies all areas of an image thought to be PV. So-called *training* of a NN means use of an optimizer to minimize a loss function, which measures the distance of the model’s performance from a desired output. In this study a Supervised Learning approach is used which requires a set of labeled reference data on which the NN’s performance is trained and evaluated.

A. Image Data and Labeling

In order to meet the need for training data, a source of aerial or satellite images is necessary. Previous studies have used aerial imagery from Google Earth [7], aerial imagery from Bing [8], or ground level images from Google Street View [8]–[12] as the basis for computer vision analyses for a variety of purposes.

In this study Google Maps images were used for the region containing Oldenburg, Germany. A total of 13,345 image tiles were selected out of which 1,325 were labeled manually. Tiles had a raw size of 639 X 640 pixels and used a Web Mercator projection zoom level of 19. At the latitude of Oldenburg (around 51° N), This corresponds to a resolution of slightly less than 0.2 m per pixel, with each image showing a roughly 120 m x 120 m area.

Each of the 1,325 individual tiles was manually visualized and labeled using the python tool *labelme* [13]. The output of this process was a single, JSON format file corresponding to each image, containing polygons that surrounded pixels of the image that contained apparent PV installations. For our case, two classes were considered, namely, PV areas and background. The label polygons were converted to a binary mask format representing these two classes, using additional code included with *labelme*. The time consuming nature of manual labeling represents the bottleneck in this process. While the majority of systems were residential scale, a number of commercial-scale rooftop arrays were present, as well as one utility scale PV plant.

Several unique features were observed during the labeling process that bear further discussion. Most of the rooftop systems in the training dataset were on residential rooftops, which in Germany are often steeply pitched. In addition to PV, solar thermal systems are known to be common within the region, but at times were not visually distinct from PV systems in the aerial images used, even for a human labeler. Finally, homes in Oldenburg frequently have wintergartens, greenhouse-like rooms connected to a home. These wintergartens have the appearance of rectangular glass panes from above, which may be difficult to distinguish from PV systems. Examples of these features are shown in Fig. 1.



Fig. 1: Examples of labelling challenges in Oldenburg.

B. Model

With the availability of a labeled training dataset, the next step in the process is to identify a NN architecture and train the network for the task. As the model should be applicable on a larger scale at country or continent levels and to avoid the risk of overfitting, the least complex model should be found to match prediction time and model skill.

1) *Network Architecture*: Fully Convolutional Neural Networks [14] are NN architectures showing high promise for semantic segmentation. They predict the class to which each pixel of an input image belongs, leading to output segmentation masks. In contrast to other NN architectures, this method works for arbitrary image sizes.

The U-net architecture from [15] is based on the concept of Fully Convolutional Neural Networks. It was originally proposed for biomedical image segmentation, in which typically only a small training set is available, and which requires

highly accurate segmentation masks. The U-net architecture is divided into two symmetrical paths leading to the eponymous u-shaped form. On the left side, there is a contracting path which decreases the image resolutions while increasing the depth. On the right is a symmetrical expansive path with increasing image dimensions and decreasing depth, as visible in Fig. 2.

The contracting path is very similar to common NN architectures for object detection which use convolutional and max pooling layers and gradually reduce the image size. The idea behind this common architecture is to increase the receptive field in order to learn more complex features. This left-hand section of the U-net is referred to as the *encoder*, in that it creates a new mathematical representation of the source image pixels.

To localize the objects and to reconstruct the original image size, the expansive path, unique to U-net, upsamples the image back to the original resolution. This path is referred to as the *decoder*, and it is this path that allows the U-net to perform pixelwise predictions of the target objects. Further distinguishing the U-net architecture are the skip connections between the two paths which concatenate encoder and decoder of the same level to more precisely localize the masks, as stated by [15].

2) *Training*: The 1325 labeled masks were randomly divided into separate training, validation and test datasets, with a 80%/10%/10% ratio. The validation data was used to monitor the training progress to avoid overfitting. This process was repeated twice, using two separate random seeds (denoted sets A and B here) for the splitting, to ensure that no non-random factors influenced the data partitioning. Results for both seeds of the split will be presented below.

Images were pre-processed by resizing and normalization. The input 639x640 tiles were resized to 576x576 to match the image size used by the pretrained U-net model. The aerial images were formatted as three 8-bit color (Red/Green/Blue) channels. These images were normalized by the maximum possible value (255) such that each pixel was represented as a floating point value between zero and one for each color channel. The labels (represented by a binary mask) were converted to floating point representations with binary 0 or 1 values representing the two classes, background and PV, respectively.

In this study, the U-net model was trained by an equally weighted combination of the binary-cross entropy loss and the Jaccard loss. For a target t and prediction p these loss functions are respectively given by the Eqs. 1 and 2. The Jaccard loss is used as a robust function to avoid the class imbalance between background and PV. The combined loss function had stable training behaviour and the best results regarding detection rate and shape predictions.

$$BCE(t, p) = -[t \log(p) + (1 - t) \log(1 - p)] \quad (1)$$

$$J(t, p) = 1 - \frac{|t \cap p|}{|t \cup p|} \quad (2)$$

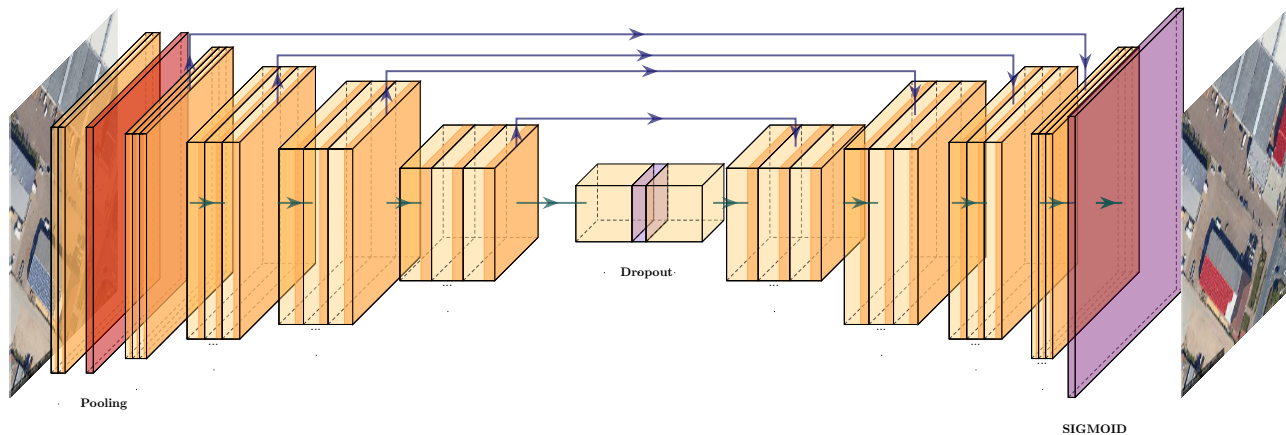


Fig. 2: U-net architecture used in this study. The left half of the network representing the contracting path is referred to as the encoder, while the expanding path is the decoder.

The model was trained using the Stochastic Gradient Descent optimizer with momentum [16]. Stochastic gradient descent is a stochastic approximation of the well-known gradient descent algorithm, which allows a model to be trained on batches of the original data. A comparatively small batch size of four images per epoch was chosen such that the images fit inside memory and to make optimal usage of the GPU. Because of the small batch size a high momentum as proposed by [15] was applied, which ensures that the training is less sensitive to single images.

To speed up convergence rates, the weights from the ImageNet competition [17] were used as a starting point for the optimization. This common practice in Deep Learning namely *Transfer Learning* significantly speeds up the training time, even if the weights are initially trained on different objects [16]. Additionally, the weights for the encoder portion of the U-net (see Fig. 2) were held fixed throughout the training process, meaning that only the decoder weights were varied. Iteration was performed for a total of 350 epochs, keeping only the weights for the epoch with the lowest value for the validation loss. An example of the loss function’s improvement throughout the training process is shown in Fig. 3. The divergence between the training and validation loss toward the end of the 350 epochs indicates that overfitting is beginning to occur.

During training, data augmentation was used to increase the effective size of the training dataset. Training data are augmented using rotation, width and height shifts, and zooming, using the *ImageDataGenerator* package built into [18]. We used a wrapper provided by *keras-unet* to actually code the augmentation [19]. The parameters used for the data augmentation are shown in Table I.

3) *Uncertainty quantification*: In addition to the evaluation metrics listed above, we also investigated the uncertainty of the network. NNs are often regarded as black box models whose decisions are difficult to understand. In the recent years however, there have been efforts to make Artificial Intelligence

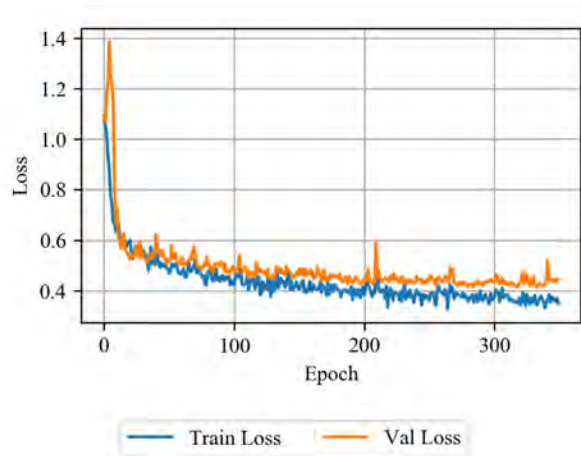


Fig. 3: Training and validation loss vs. epoch for ResNet-34, seed A.

TABLE I: Data Augmentation Parameters

Augmentation	Value
Rotation	$\pm 30^\circ$
Width Shift	$\pm 5\%$
Height Shift	$\pm 5\%$
Zoom	$\pm 20\%$

explainable to better understand Deep Learning models and their outputs. The concept of Bayesian Neural Networks follows this approach by regarding the NN parameters not as fixed values, but as random variables. Bayesian Neural Networks and uncertainty in Deep Learning are still a highly active research field as many common Bayesian techniques like Variational Inference or Markov Chain Monte Carlo are not tractable for the large number of parameters of Deep Neural Networks.

A computationally tractable method is introduced by [20],

who used the Dropout technique as an approximation of the uncertainty. Dropout is a regularization technique which randomly deactivates units and their connections in a NN. It is usually applied during model training to prevent overfitting. However, [20] shows that when Dropout is activated for predictions, uncertainty estimates can be derived. This technique, called *Monte-Carlo Dropout*, is applied in this study to quantify uncertainty. It has been used in other U-net studies, and has been proven to give reliable uncertainty estimates [21]. In the present study, we applied dropout in the bottleneck between the contracting and the expanding path in contrast to [21] who applied dropout in each layer. To estimate the output distribution, we applied a sample size of 100 model runs. The model uncertainty is approximated through the standard deviation of the model predictions.

III. RESULTS

The model was trained and evaluated in python using the *tensorflow* library [18]. An implementation of U-net called *segmentation models* was employed to quickly vary the model backbone and utilize the pre-trained ImageNet competition weights [22]. The model architecture was selected by testing different architectures for the U-net. The comparative tests were run with the *Snakemake* library [23]. It has been observed that the choice of the backbone model of the U-net has a major influence on the model performance. As a backbone model, the ResNet architecture was chosen as it has shown better accuracy than other backbone models, such as the VGG architecture.

The metrics tested for model evaluation were the loss function described previously, the Intersection Over Union (IOU) score, the Precision (P), the Recall (R) and the F1 score. Precision is defined as the fraction of true predictions that are correct, recall is the fraction of all target true values that were predicted as true, and the F1 score is the harmonic mean of Precision and Recall as shown in Eq. 3. For the loss metric, a lower score indicates better performance, while higher values indicate better performance for all other metrics. Excepting loss, all metrics have a maximum value of 1.0, which would indicate perfect performance.

$$F1 = \frac{2}{P^{-1} + R^{-1}} \quad (3)$$

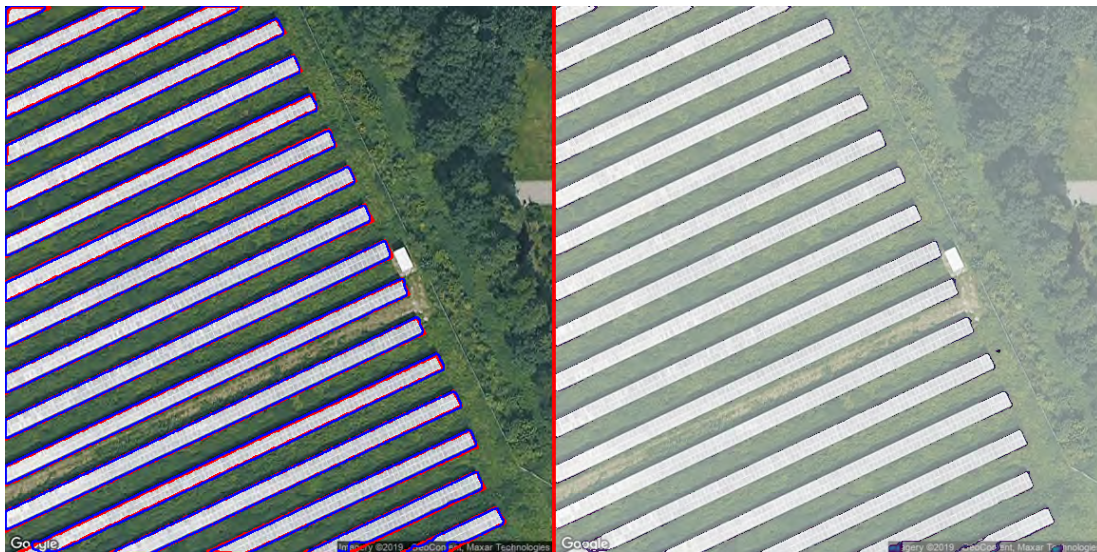
The results of the benchmark indicate that ResNet-50 exhibited the best performance with a validation loss function of 0.33-0.34, with ResNet-34 at a very close second place. The full results of the evaluation are shown in Table II. The ResNet-18 and ResNet-101 backbones both had slightly higher losses, indicating that ResNet-18 is perhaps too simple, while ResNet-101 may be too complex. We note some slight differences between the precision and recall metrics for the two seeds (A and B). Given the consistency that is observed for all the datasets (e.g. the precision of B is always better than the precision of A), we attribute this to some slightly non-random characteristics of the test dataset for these particular splits. Though this could perhaps be better resolved by including more test data, the backbones identified as "best"

are consistent across all the metrics, and as a result we have confidence in the quality of the trained model. To interpret the results for ResNet-50 in words, approximately 80% of the pixels of its predictions of PV coincide with PV in the labeled data. Likewise, it is able to identify approximately 80% of true pixels labeled as containing PV. As scaling from identified area to PV installed capacity would to first approximation be expected to be a linear operation, we can infer the same approximate performance with respect to translating these predictions to an estimate of PV capacity.

TABLE II: Model Evaluation Results

Model		Metric				
Backbone	Seed	Loss	IOU	Precision	Recall	F1 Score
ResNet-18	A	0.40	0.63	0.70	0.86	0.77
ResNet-18	B	0.37	0.65	0.83	0.76	0.79
ResNet-34	A	0.33	0.69	0.79	0.84	0.81
ResNet-34	B	0.36	0.68	0.84	0.77	0.80
ResNet-50	A	0.33	0.69	0.79	0.84	0.82
ResNet-50	B	0.34	0.69	0.84	0.79	0.81
ResNet-101	A	0.37	0.66	0.74	0.85	0.79
ResNet-101	B	0.35	0.68	0.86	0.76	0.81

Fig. 4 illustrates several cases that demonstrate the NN performance in identifying PV panels. In Fig. 4a, we see the model's successful identification of a utility scale PV plant. Uncertainty shown in the right half of the image shows high confidence over the main footprint of each row, with increased uncertainty at the edges, implying that the network has some variability in exactly which pixels constitute the system boundaries. In Fig. 4b we see several rooftops that contain obvious PV systems, which are well identified by the network. One notable case is the system in the top-right of the image which was not marked as a system in the labels, due to labeler error. Two false positives are also present in the predictions. So we demonstrate that in some cases, the trained network can help correct mistakes made by a user in labeling. It is notable that the false positive predictions are made with a high degree of confidence, indicating that uncertainty is not universally useful in interpreting incorrect predictions. Finally, though the model generally shows decent performance in distinguishing between skylights/wintergartens, an example of the network performing poorly on these types of surfaces is shown in Fig. 4c. The model identifies the array on the southern part of the rooftop with a high degree of confidence, but predicts a number of false positive regions on the covered walkway in the bottom right of the image, and on the building's glass rooftop and skylights. Notable in these cases is that the for most of the false positive areas, a high degree of uncertainty is indicated. Further analysis of the potential of uncertainty to serve as a useful interpretive tool for these predictions is still necessary. The NN uncertainty may be thought to be representative of the uncertainty a human labeler might experience, and may also indicate that the variance in the RGB values of the data might not be large enough to easily distinguish between



(a)



(b)



(c)

Fig. 4: Example results from ResNet-50. Red lines show boundary of labeled PV regions. Blue lines show predicted PV regions. Right half of each image demonstrates uncertainty.

certain types of rooftops and PV areas, such that additional training data showing these types of rooftops, or additional data inputs might help in discrimination. A common pattern over all images is a higher uncertainty at the edges of the PV panels which is probably due to less similar pixels in the boundary regions and inaccurate labeling.

In addition to comparing the performance using uncertainty, it is instructive to investigate some common examples of model error. Fig. 5 shows three examples of common errors to illustrate the situations when the NN fails to predict an existing PV array (False Negative) or when it assumes a PV array which does not exist (False Positive). A main source of errors are erroneous labels, as exact PV locations are not known and labels are based on visual inspection. Figure 5a illustrates an example in which the NN predicts the correct PV arrays whereas the labels are based on the outer frame of the PV arrays. The figure also shows that NN has difficulty detecting PV panels in the presence of strong reflections, as in some of the middle central PV rows. A similar error is made due to shadows that fall on the PV panels. Figure 5c shows the effect of shadow on the model output errors. The tree in the middle figure leads to a high False Negative rate, as the model does not label the shadowed pixels as PV panels. This is in contrast to the case of wintertartens, as shown in Fig. 5c, where the NN falsely predicts PV due to the similar grid looking structure of wintertartens and PV panels.

IV. SUMMARY

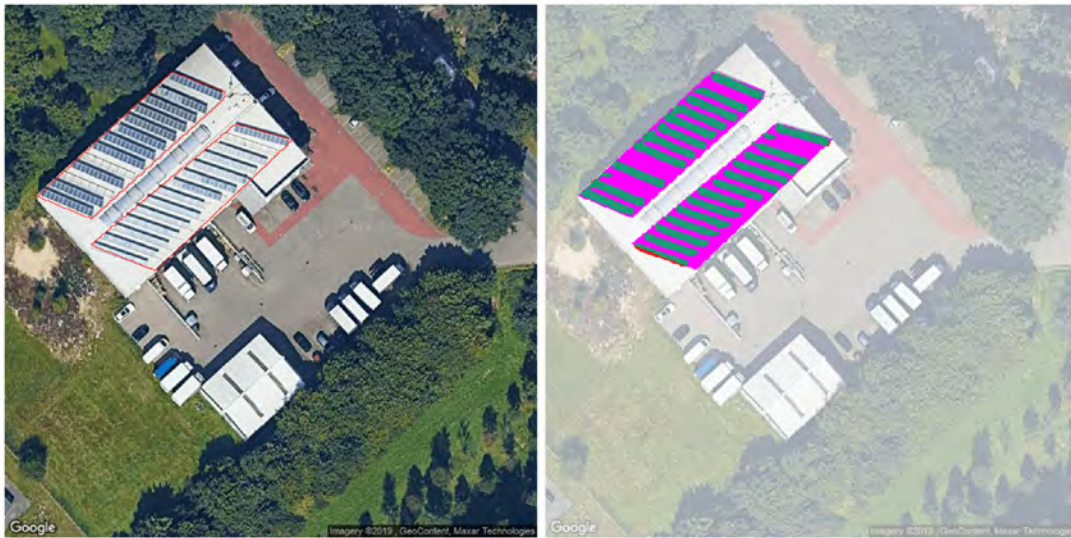
Fully Convolutional Neural Networks show promise in identifying PV arrays from satellite images. This type of analysis could be used to generate a database of PV installations in a region, and could enable a number of additional approaches to renewable energy modelling. By classifying PV systems on a pixel basis, the model used in this study also enables the possibility of estimating a system's rated capacity, which may be of interest to electrical models of distributed generation. Further development on this method is needed to improve its accuracy and assess its applicability in other environments with different patterns of building construction and rooftop configurations. This study shows some common sources for errors and uncertainty of the Neural Network output which affect the final model skill. Future work is still needed to investigate how the errors made by the Neural Network can be avoided through a more diverse set of training data, which for example needs to include sun reflectance and shadow effects. Furthermore, the study has shown that uncertainty often but not always relates to model errors. It may also be promising to link uncertainty and training explicitly, using it to enhance the labeling process and improve subsequent training. Nonetheless, Neural Network models show promise for identifying PV areas from aerial images and may serve as a useful tool for mapping distributed PV as an input to future distributed PV modelling efforts.

V. ACKNOWLEDGMENT

J.R. acknowledges financial support by Penn State Engineering Technology and Commonwealth Engineering and Penn State Hazleton, and appreciates the support of the Carl von Ossietzky University of Oldenburg and DLR Institute of Networked Energy Systems who hosted his visiting scientist appointment. We furthermore thank Madeleine Bischoff for supporting the labeling process.

REFERENCES

- [1] G. Lettner, H. Auer, A. Fleischhacker, D. Schwabeneder, D. Bettina, F. Moisl, E. Roman, D. Velte, and A. Huidobro, "EXISTING AND FUTURE PV PROSUMER CONCEPTS," PVP4Grid Consortium, Tech. Rep., May 2018. [Online]. Available: https://www.pvp4grid.eu/wp-content/uploads/2018/12/D2.1_Existing-future-prosumer-concepts_PVP4G-1.pdf
- [2] 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, Dec. 2018. [Online]. Available: [https://www.cell.com/joule/abstract/S2542-4351\(18\)30570-1](https://www.cell.com/joule/abstract/S2542-4351(18)30570-1)
- [3] J. M. Malof, K. Bradbury, L. M. Collins, R. G. Newell, A. Serrano, H. Wu, and S. Keene, "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)*, Nov. 2016, pp. 799–803, iSSN: null.
- [4] J. M. Malof, L. M. Collins, K. Bradbury, and R. G. Newell, "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)*, Nov. 2016, pp. 650–654, iSSN: null.
- [5] J. M. Malof, K. Bradbury, L. M. Collins, and R. G. Newell, "Automatic detection of solar photovoltaic arrays in high resolution aerial imagery," *Applied Energy*, vol. 183, pp. 229–240, Dec. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306261916313009>
- [6] M. Wang, Q. Cui, Y. Sun, and Q. Wang, "Photovoltaic panel extraction from very high-resolution aerial imagery using region-line primitive association analysis and template matching," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 141, pp. 100–111, Jul. 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271618301205>
- [7] Z. Wang, Z. Wang, A. Majumdar, and R. Rajagopal, "Identify Solar Panels in Low Resolution Satellite Imagery with Siamese Architecture and Cross-Correlation," Vancouver, Canada, Dec. 2019, p. 7.
- [8] S. Law, B. Paige, and C. Russell, "Take a Look Around: Using Street View and Satellite Images to Estimate House Prices," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 5, pp. 1–19, Sep. 2019, arXiv: 1807.07155. [Online]. Available: <http://arxiv.org/abs/1807.07155>
- [9] A. R. Zamir and M. Shah, "Accurate Image Localization Based on Google Maps Street View," in *Computer Vision – ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer, 2010, pp. 255–268.
- [10] N. Naik, S. D. Kominers, R. Raskar, E. L. Glaeser, and C. A. Hidalgo, "Computer vision uncovers predictors of physical urban change," *Proceedings of the National Academy of Sciences*, vol. 114, no. 29, pp. 7571–7576, Jul. 2017. [Online]. Available: <https://www.pnas.org/content/114/29/7571>
- [11] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, E. L. Aiden, and L. Fei-Fei, "Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States," *Proceedings of the National Academy of Sciences*, vol. 114, no. 50, pp. 13 108–13 113, Dec. 2017. [Online]. Available: <https://www.pnas.org/content/114/50/13108>
- [12] L. Najafizadeh and J. E. Froehlich, "A Feasibility Study of Using Google Street View and Computer Vision to Track the Evolution of Urban Accessibility," in *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '18*. Galway, Ireland: ACM Press, 2018, pp. 340–342. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3234695.3240999>



(a)



(b)



(c)

Fig. 5: Error analysis. Left column: Input images and convex hull of ground truth (*red*). Right column: positive predicted pixels (*green*), False Negative (*pink*) and False Positive (*red*)

- [13] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, May 2008. [Online]. Available: <http://link.springer.com/10.1007/s11263-007-0090-8>
- [14] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *arXiv:1411.4038*, Mar. 2015, arXiv: 1411.4038. [Online]. Available: <http://arxiv.org/abs/1411.4038>
- [15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241. [Online]. Available: <http://arxiv.org/abs/1505.04597>
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: <https://www.deeplearningbook.org>
- [17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, iSSN: 1063-6919.
- [18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Google, Tech. Rep., 2015. [Online]. Available: <https://tensorflow.org>
- [19] K. Zak, "karolzak/keras-unet," May 2020, original-date: 2019-03-13T07:50:21Z. [Online]. Available: <https://github.com/karolzak/keras-unet>
- [20] Y. Gal, "Uncertainty in Deep Learning," Doctoral, University of Cambridge, Cambridge, England, Sep. 2016. [Online]. Available: <http://mlg.eng.cam.ac.uk/yarin/thesis/thesis.pdf>
- [21] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation," in *1st Conference on Medical Imaging with Deep Learning*, Amsterdam, Netherlands, 2018, p. 13. [Online]. Available: https://openreview.net/pdf?id=Sk_P2Q9sG
- [22] P. Yakubovskiy, "Segmentation Models," 2019. [Online]. Available: https://github.com/qubvel/segmentation_models
- [23] J. Köster and S. Rahmann, "Snakemake—a scalable bioinformatics workflow engine," *Bioinformatics*, vol. 28, no. 19, pp. 2520–2522, Oct. 2012. [Online]. Available: <https://academic.oup.com/bioinformatics/article/28/19/2520/290322>