A Deep Convolutional Neural Network and a Random Forest Classifier for Solar Photovoltaic Array Detection in Aerial Imagery

Jordan M. Malof, Leslie M. Collins
Electrical and Computer Engineering
Duke University
Durham, NC, USA
jmmalo03@gmail.com

Kyle Bradbury, Richard G. Newell
Duke University Energy Initiative
Duke University
Durham, NC, USA
kyle.bradbury@duke.edu

***Abstract*—Power generation from distributed solar photovoltaic PV arrays has grown rapidly in recent years. As a result, there is interest in collecting information about the quantity, power capacity, and energy generated by such arrays; and to do so over small geo-spatial regions (e.g., counties, cities, or even smaller regions). Unfortunately, existing sources of such information are dispersed, limited in geospatial resolution, and otherwise incomplete or publically unavailable. As result, we recently proposed a new approach for collecting such distributed PV information that relies on computer algorithms to automatically detect PV arrays in high resolution aerial imagery** [1]**. Here we build on this work by investigating two machine learning algorithms for PV array detection: a Random Forest classifier (RF)** [2] **and a deep convolutional neural network (CNN)** [3]**. We use the RF algorithm as a benchmark, or baseline, for comparison with a CNN model. The two models are developed and tested using a large collection of publicly available** [4] **aerial imagery, covering $135\ \mathrm{km}^2$, and including over 2,700 manually annotated distributed PV array locations. The results indicate that the CNN substantially improves over the RF. The CNN is capable of excellent performance, detecting nearly 80% of true panels with a precision measure of 72%.**

Keywords-component; convolutional neural networks, deep learning, detection, solar, energy, photovoltaic

I. INTRODUCTION

The quantity of solar photovoltaic (PV) arrays has grown rapidly in the US in recent years [5], [6], and a large proportion of this growth is due to small-scale, or distributed, PV arrays [7], [8]. Distributed PV (DPV) offers many benefits [9], but integrating it into existing power grids is challenging. To aid in the integration of DPV, there is growing interest among government agencies, utilities, and third party decision makers in having access to detailed information about DPV; information such as the locations, power capacity, and the energy production of existing arrays. As a result, several organizations have begun collecting and/or publishing such information, including the Interstate Renewable Energy Council (IREC) [10], Greentech Media [11], and the Energy Information Initiative (EIA) of the U.S. Department of Energy [12][13].

Although available information on distributed PV is expanding, it is nonetheless difficult to obtain. Existing methods of obtaining this information, such as surveys and utility interconnection filings, are costly and time consuming to collect. They are also typically limited in spatial resolution to the state or national level [3],[6].

Recently, we proposed a new approach for collecting PV information [1], [14] that relies on using computer algorithms to automatically identify PV arrays in high resolution (≤ 0.3 meter) aerial imagery. Fig. 1a shows an example of 0.3 meter resolution aerial imagery in which the PV arrays have been annotated. This approach permits the collection of DPV information at a very high geo-spatial resolution. Also, because the approach is automated, it is inexpensive to apply, and to do so repeatedly as new imagery becomes available.

We build on this previous work by investigating two popular machine learning algorithms for PV array detection in aerial imagery: a Random Forest classifier (RF) [2] and a deep convolutional neural network (CNN) [3]. Both algorithms have shown success in image recognition tasks [15]–[18], including those related to aerial imagery [19]–[23]. The performance of both the RF and CNN are evaluated using a large dataset of aerial imagery, encompassing $135\ \mathrm{km}^2$ of surface area, and 2,700 PV arrays. The true locations of PV arrays in the imagery have been manually annotated. The experimental results demonstrate that the CNN substantial improves over the RF, and achieves excellent detection results. We note also that the full dataset, from which we use a subset of PV annotations, are publicly available for download [4] to encourage future algorithm development.

The remainder of this paper is organized as follows. Section II describes the aerial imagery dataset. Section III presents the RF and CNN detection algorithms. Section IV describes the experimental design and the results, and Section V presents our conclusions and ideas for future work.

II. COLOR ORTHOIMAGERY DATASET

All experiments were conducted on a dataset of color (RGB) aerial imagery, collected over the US city of Fresno, California. All of the imagery was collected in the same month in 2013, using ortho-rectified aerial photography, with a spatial resolution of 0.3 meters per pixel. An example of the imagery is shown in Fig. 1a, where the solar PV locations are annotated in red. The true locations of PV arrays in the aerial

This work was supported in part by the Alfred P. Sloan Foundation and by the Wells Fargo Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Alfred P. Sloan Foundation or the Wells Fargo Foundation.

978-1-5090-3388-1/16/$31.00 ©2016 IEEE

imagery were manually annotated with polygons by human annotators. The dataset used in the experiments in this work is a random subset of the larger set of imagery available in [4]. The dataset used in this work encompasses 135 km^2 of surface area, and 2,794 PV array annotations.

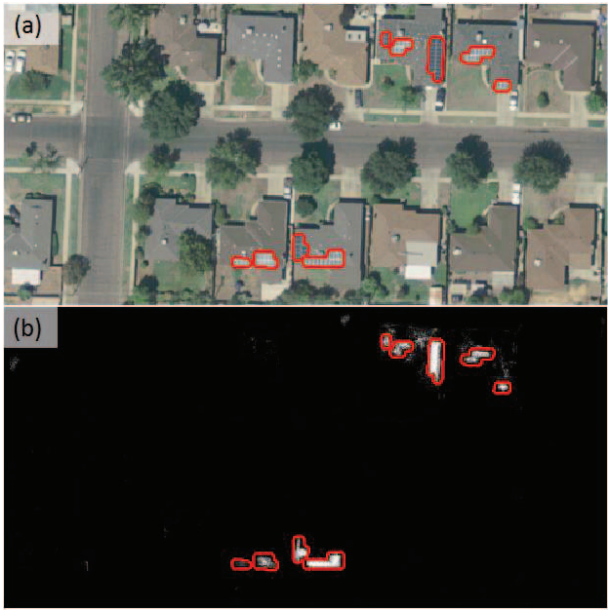

Fig. 1 (a) provides an example of the color ortho-imagery used in this work, along with several examples of solar PV annotations (red polygons). (b) is the pixel-wise output of the RF detector along with the same red polygons as in 1a. Bright locations indicate regions of high confidence, where PV arrays are likely to exist.

In order to avoid a positive bias in the performance evaluation of the proposed PV detection algorithms, we split our experimental dataset into two disjoint sets: Fresno Training and Fresno Testing. This is a common approach for evaluating supervised machine learning algorithms, such as those considered here. A summary of the imagery in each dataset is presented below in Table 1.

TABLE 1
SUMMARY OF FRESNO COLOR AERIAL IMAGERY DATASET

Designation	Area of Imagery	Number of PV Annotations
Fresno Training	90 km^2	1,780
Fresno Testing	45 km^2	1,014

III. THE SOLAR PV DETECTION ALGORITHMS

There are two detection algorithms investigated in this work: an RF classifier and a CNN. Both algorithms have been successfully applied for image recognition tasks, however, in recent years CNNs have achieved the best performance on many major image recognition benchmarks [15]–[18]. Despite their excellent performance, CNNs often require much more time to train than competing algorithms (e.g., the RF). This problem is exacerbated because CNNs are difficult to design, and often require testing of many models.

To address these computational challenges we adopt a cascade architecture where the RF operates first on all the aerial imagery. The RF identifies a smaller subset of candidate locations in the imagery where PV arrays are likely to exist. The CNN is then trained and tested only on locations identified by the RF. This dramatically reduces the amount of imagery for training and testing the CNN, while still leveraging its superior recognition performance. This cascade architecture is illustrated in Fig. 2. In this architecture the RF detection performance also acts as a baseline, or benchmark, for examining whether the CNN improves performance (by removing false detections).

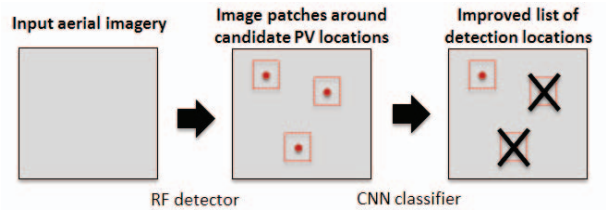

Fig. 2. An illustration of the cascade detection architecture employed in this work. The first stage of the cascade is an RF detector that identifies candidate PV array locations in the aerial imagery. Patches of imagery are extracted around each RF detection location and then provided to a CNN which identifies false detections.

A. The Random Forest prescreener

Random Forests (RF) [2] are a supervised statistical classification method that has been successfully applied to a variety of problems. Some examples include image processing [24], medical diagnosis [25], pose recognition [26], [27], and remote sensing [19], [20]. In this work, the RF performs pixel-wise detection by assigning a "confidence" to each pixel indicating how likely it is that the pixel corresponds to a solar PV array. This approach is similar to the way the RF has been used in related contexts [19], [27]. The result of this processing is a confidence map, indicating where PV arrays are likely to exist. An example image and its corresponding RF confidence map are shown in Fig. 1b. The RF obtains individual detection locations by identifying local maxima in the confidence maps (i.e., pixels whose confidence values that are greater than all adjacent pixels). The final output of the RF is a list of detection locations, and their respective confidence values. This algorithm is based on the algorithm presented in our earlier work in [14].

1) Aerial image features

The RF assigns a confidence value to each pixel based on its feature vector which is a vector of local image statistics. These statistics capture local image color and texture information that is useful for identifying PV arrays. In this work a feature vector is extracted for each pixel that consists of the means, μ, and variances, σ^2, computed in several 3x3 windows surrounding it. More precisely, for a given pixel location, p_0, the windows are organized into two rings surrounding p_0, and each ring consists of 9 windows. Each window can be characterized by its vertical offset, y, and horizontal offset, x, from p_0. This is illustrated in Fig. 3. A set

of 9 windows (one ring) is denoted here by S_r, where the subscript r parameterizes the locations of the windows in the ring. The locations of the windows in S_r are then given by

$$S_r = \{(x, y): x \in \{0, -r, r\}, y \in \{0, -r, r\}\}. \quad (1)$$

There are 9 windows in each ring, and each window yields 6 features (a μ and σ^2 for each color channel), resulting in 54 total features per ring. In this work, we extracted features in two rings, given by S_2 and S_4. Note that S_2 and S_4 share a window location at $(x, y) = (0,0)$. One of these duplicates is removed, leaving a total of 54+54-6=102 total features.

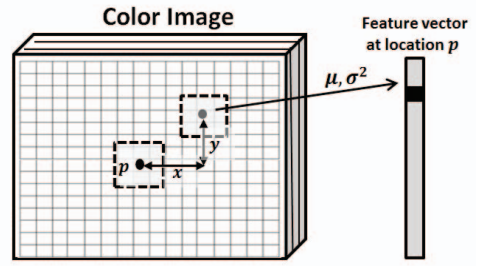

Fig. 3. An illustration of how a feature is extracted for a single pixel, located at p. Means, μ, and variances, σ^2, are extracted in 3x3 windows surrounding p. Each window is parameterized by its horizontal and vertical offset, given by x and y respectively.

Fig. 4 Examples of image patches extracted at RF detection locations. (a) shows twenty examples of true panel detections and (b) shows twenty

B. The Deep Convolutional Neural Network Classifier

CNNs are a type of machine learning classification model that have shown state-of-the-art performance on a variety of tasks, in particular, image recognition tasks. They have previously been applied to recognition of objects in satellite imagery, such as roads [21], or in scene classification [22]. Here the CNN is used to classify 40x40 patches of RGB aerial imagery as either corresponding to a solar PV panel or not. It operates only on patches detected by the RF prescreener. Some examples patches are shown in Fig. 4.

Deep CNNs consist of several processing blocks, or modules, that are applied sequentially to the data. The ordering of these modules, and their parameter settings, can have a significant influence on the classification performance of a network. As a result, the CNN architecture for a given application must be set carefully, and varies across applications. The CNN architecture chosen for this work is shown in Fig. 5. The architecture is characterized by the use of several consecutive "blocks", where each block consists of two small 3x3 convolutional filtering layers, followed by a max pooling layer. Max pooling [16] refers to an operation where a group of pixels is replaced by the maximum pixel intensity within the group. Max pooling was done in this work over 3x3 pixel windows, with a stride of 2 (i.e., max pooling was done on every 2nd pixel in the imagery). Note that, typical of recent CNNs [16], [17], every convolutional layer is followed by a rectified linear unit (ReLU) nonlinearity.

Input (40 x 40 RGB image)
conv3-32
conv3-32
maxpool
conv3-32
conv3-32
maxpool
conv3-64
conv3-64
maxpool
fully connected - 64
Output (2-way soft-max)

Fig. 5. Architecture of the CNN used in this work. Each row in the figure specifies a processing step in the network. Convolutional network layers have titles of the form "convA-B". A layer titled convA-B refers to a layer with A unique filters that are each AxA pixels in size. "maxpool" refers to a 3x3 pixel max pooling operation with a stride of 2 pixels. The last two layers of the network consist of a fully connected neural network classifier with 64 hidden units, and a two-way soft-max output layer, respectively.

The remainder of the network consists of a Neural Network classifier with 64 hidden neurons with a two-class output. In Fig. 5, this part of the CNN is referred to as a fully connected layer with a 2 unit softmax output: consistent with the CNN literature. Softmax refers to the form of the CNN output, which consists of a 2-unit vector, where each entry represents a probability (i.e., PV or not PV).

For training (i.e., inferring the parameters of the CNN) a standard stochastic gradient descent procedure was used, similar to [16], [17], with batch sizes of 200, momentum set to

0.9, and the weight decay set to 0.0001. The training data for the CNN was augmented by making rotated copies of each PV array image patch, using 45 degree increments. In similar settings, augmentation has been shown to improve results [16]. No augmentation was performed to non-PV patches in order to lower the training time.

IV. Experiments

The primary role of the Fresno training dataset was to train the RF and CNN classification models, as well as optimize other parameters associated with the detection algorithm. The Fresno Testing dataset was used to obtain an unbiased performance estimate for the detector. This is a common crossvalidation approach for supervised machine learning algorithms [28]. A total of five million pixels from the Fresno Training imagery were used to train the RF classifier. All of the available PV pixels were used (roughly 500,000), and the remaining training pixels were sampled randomly from the non-PV imagery. Running the RF on the training data yielded a total of 93,712 training patches (after augmentation) for the CNN.

The metric used to evaluate the performance of the algorithm is the precision recall (PR) curve. The PR curve is a popular performance metric for object detection in aerial imagery [21], [29]–[31], and therefore it is adopted here. PR curves measure the performance tradeoff between making correct detections and false detections, as the sensitivity of a detector, or classifier, is varied. The x-axis of a PR curve is the recall, R, which is the proportion of all true target objects (i.e., PV arrays) in the data that were returned by the algorithm as detections. The y-axis is the precision, P, which is the proportion of all detected objects (i.e., both true and false) which are true targets. An effective detector will tend towards the top right corner of the PR curve, thereby maximizing both recall and precision.

It is important to note that the RF detector often returns multiple detections over the same PV annotation. This can lead to a positive bias in the measured precision of the detectors because each unique detected PV array can (incorrectly) contribute multiple correct detections. In order to resolve this redundancy, and potentially misleading bias, only the maximum confidence detection over each PV array was considered during scoring.

A. Results

The results of applying the trained RF and CNN detectors to the Fresno Testing data are presented in Fig. 66. The results indicate that the precision of the RF detector begins high (equal to one), and steadily drops as the recall rate increases. The RF only reaches a maximum recall rate of roughly 0.9, reflecting the fact that the RF prescreener misses about 10% of the PV arrays. Because the CNN classifier only operates on the objects detected by the RF, the CNN also attains a maximum recall rate of 0.9. Therefore, the main objective of the CNN is to improve the precision of the detections, by lowering the confidence values of the false detections (compared to the true detections). The results indicate that indeed, the CNN classifier substantially improves the precision of the detections. For each recall rate, the CNN achieves a greater precision than the RF. The improvements are especially large at higher recall rates. For example, at a recall of 0.8, the precision of the RF is 0.1, while the CNN attains a precision of 0.7.

Further, the CNN can detect the large majority of panels (over 70%) with very few false alarms, as indicated by the precision of 0.9. This indicates that only one in ten of returned detections (at this recall rate) are false.

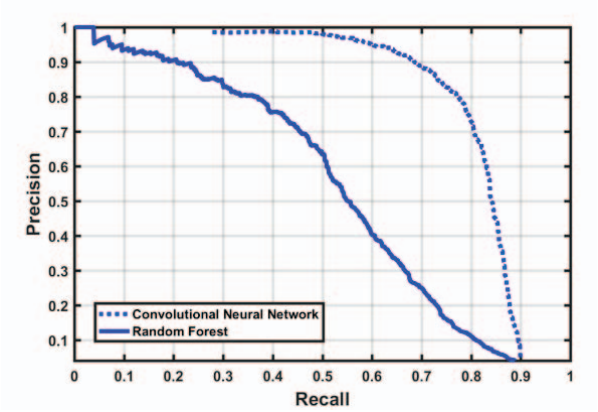

Fig. 6. PR curve for the RF and CNN PV array detection algorithms.

Further insight into the result can be gained by measuring the training error on the training data and testing data, which is shown in Fig. 7. The results indicate that the CNN eventually achieves nearly perfect performance on the training data, suggesting that the deep CNN architecture is powerful enough to learn to distinguish between general PV and non-PV imagery.

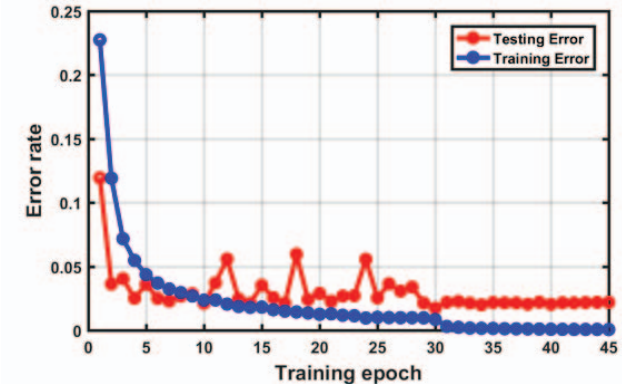

Fig. 7. Training error rate (blue) and validation error rate (red) for the CNN as a function of the training epoch. A single training epoch means that every training example was considered exactly once during stochastic gradient descent. The network was trained for 45 epochs.

The performance on the testing dataset however, stops improving beyond epoch 30. This suggests that there are some PV array variations (e.g., textures or shapes) in the testing imagery that are not represented in the training data, and therefore the CNN did not learn to recognize them. It is also important to note that the testing dataset error does not increase as training continues, which suggests that the CNN is

not overfitting to the training data (i.e., the CNN model generalizes well to previously unseen aerial imagery).

V. CONCLUSIONS

In this we investigated two algorithms for the detection of solar PV arrays in high resolution aerial imagery: a Random Forest (RF) and a deep convolutional neural network (CNN). For computational efficiency, the algorithms are employed in a cascade architecture where the CNN only operates on locations in the imagery that are first detected by the RF. The goal of the CNN is to improve detection performance over the RF by identifying and removing false detections. Both the RF and CNN were tested on a large collection of aerial imagery ($\geq 135km^2$) where the true PV array locations were indicated by manual annotations. The results indicate that the CNN substantially lowers the number of false detections made by the RF. The final detector (RF plus CNN) can detect nearly 80% of true panels, with a precision rate of 72%. Further, analysis of the training error of the CNN reveals that the network architecture used in this work is sufficiently powerful to learn to identify general PV array imagery, without overfitting to the training data (i.e., the CNN generalizes well to previously unseen aerial imagery).

Future work should include techniques to extract the precise shape and size of PV arrays, as well as estimate PV array capacity and energy generation based on aerial imagery, and other meta data.

REFERENCES

[1] J. M. Malof, R. Hou, L. M. Collins, K. Bradbury, and R. Newell, "Automatic solar photovoltaic panel detection in satellite imagery," in *International Conference on Renewable Energy Research and Applications (ICRERA)*, 2015, pp. 1428–1431.

[2] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[3] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *Handb. brain theory neural networks*, vol. 3361, no. April 2016, pp. 255–258, 1995.

[4] K. Bradbury, R. Saboo, J. Malof, T. Johnson, A. Devarajan, W. Zhang, L. Collins, and R. Newell, "Distributed Solar Photovoltaic Array Location and Extent Data Set for Remote Sensing Object Identification," *figshare*, 2016. [Online]. Available: https://dx.doi.org/10.6084/m9.figshare.3385780.v1. [Accessed: 01-Jun-2016].

[5] M. J. E. Alam, K. M. Muttaqi, and D. Sutanto, "An approach for online assessment of rooftop solar PV impacts on low-voltage distribution networks," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 663–672, 2014.

[6] A. Chersin, W. Ongsakul, J. Mitra, and S. Member, "Improving of Uncertain Power Generation of Rooftop Solar PV Using Battery Storage," in *International Conference and Utility Exhibition on Green Energy for Sustainable Development*, 2014, no. March, pp. 1–4.

[7] "Electric Power Monthly", The US Energy Information Administration, 2016.

[8] "Net Generation from Renewable Sources: Total (All Sectors), 2006-February 2016", US Energy Informaation Administration, 2016.

[9] G. K. Singh, "Solar power generation by PV (photovoltaic) technology: A review," *Energy*, vol. 53, pp. 1–13, 2013.

[10] L. Sherwood, "U.S. Solar Market Trends 2013," 2013.

[11] S. E. I. Association, "U.S. Solar Market Prepares for Biggest Quarter in History," 2015. [Online]. Available: http://www.seia.org/news/us-solar-market-prepares-biggest-quarter-history.

[12] "EIA electricity data now include estimated small-scale solar PV capacity and generation," *EIA (US Energy Information Administration)*, 2015. [Online]. Available: https://www.eia.gov/todayinenergy/detail.cfm?id=23972.

[13] "Electric Power Monthly: with data for January 2015," 2015.

[14] J. M. Malof, K. Bradbury, L. M. Collins, and R. G. Newell, "Automatic detection of solar photovoltaic arrays in high resolution aerial imagery," *Appl. Energy*, vol. 183, pp. 229–240, 2016.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, 2015.

[16] A. Krizhevsky and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," pp. 1–9.

[17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recoginition," *Intl. Conf. Learn. Represent.*, pp. 1–14, 2015.

[18] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013," *Comput. Vision–ECCV 2014*, vol. 8689, pp. 818–833, 2014.

[19] P. Tokarczyk, J. Montoya, and K. Schindler, "an Evaluation of Feature Learning Methods for High Resolution Image Classification," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. I–3, pp. 389–394, 2012.

[20] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, 2006.

[21] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6316 LNCS, no. PART 6, pp. 210–223, 2010.

[22] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSat-A learning framework for satellite imagery," *arXiv Prepr. arXiv1509.03602*, 2015.

[23] M. Längkvist, A. Kiselev, M. Alirezaie, and A. Loutfi, "Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks," *Remote Sens.*, vol. 8, no. 4, p. 329, 2016.

[24] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006.

[25] V. Lempitsky, M. Verhoek, J. A. Noble, and A. Blake, "Random forest classification for automatic delineation of myocardium in real-time 3D echocardiography," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5528, pp. 447–456, 2009.

[26] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 617–624, 2011.

[27] J. Shotton, a W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, a Kipman, and a Blake, "Real-time human pose recognition in parts from single depth images," *Cvpr*, pp. 1297–1304, 2011.

[28] C. M. Bishop, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[29] D. Chaudhuri, N. K. K. Kushwaha, A. Samal, and R. C. C. Agarwal, "Automatic Building Detection From High-Resolution Satellite Images Based on Morphology and Internal Gray Variance," *Sel. Top. Appl. Earth Obs. Remote Sensing, IEEE J.*, vol. PP, no. 99, pp. 1–13, 2015.

[30] a M. Cheriyadat, "Unsupervised Feature Learning for Aerial Scene Classification," *Geosci. Remote Sensing, IEEE Trans.*, vol. 52, no. 1, pp. 439–451, 2014.

[31] A. Manno-Kovacs and A. O. Ok, "Building Detection From Monocular VHR Images by Integrated Urban Area Knowledge," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2140–2144, 2015.