Surface Water Mapping by Deep Learning

Furkan Isikdogan, Alan C. Bovik, *Fellow, IEEE*, and Paola Passalacqua

***Abstract*—Mapping of surface water is useful in a variety of remote sensing applications, such as estimating the availability of water, measuring its change in time, and predicting droughts and floods. Using the imagery acquired by currently active Landsat missions, a surface water map can be generated from any selected region as often as every 8 days. Traditional Landsat water indices require carefully selected threshold values that vary depending on the region being imaged and on the atmospheric conditions. They also suffer from many false positives, arising mainly from snow and ice, and from terrain and cloud shadows being mistaken for water. Systems that produce high-quality water maps usually rely on ancillary data and complex rule-based expert systems to overcome these problems. Here, we instead adopt a data-driven, deep-learning-based approach to surface water mapping. We propose a fully convolutional neural network that is trained to segment water on Landsat imagery. Our proposed model, named DeepWaterMap, learns the characteristics of water bodies from data drawn from across the globe. The trained model separates water from land, snow, ice, clouds, and shadows using only Landsat bands as input. Our code and trained models are publicly available at http://live.ece.utexas.edu/research/deepwatermap/.**

***Index Terms*—Computer vision, convolutional neural networks, landsat, machine learning, remote sensing.**

I. Introduction

MAPPING surface water has been a common application of remote sensing. Automated and semiautomated surface water mapping methods generally rely on rule-based systems [1]–[6], machine learning models, or a combination of these two approaches [7]. Rule-based systems set specific thresholds on certain spectral bands or deploy multiband indices, whereas machine learning models tune trainable parameters on data to learn optimal separations between classes.

A simple and commonly adopted approach to classifying water bodies on Landsat images is to use a two-band water index, such as the normalized difference water index (NDWI) [8] or its modification MNDWI [9]. These water indices make use of the reflectance characteristics of water in visible and infrared bands to enhance water features. The enhanced results are then thresholded to classify water bodies. This process may be viewed as a simple rule-based system with a single rule.

Manuscript received March 31, 2017; revised May 20, 2017 and July 16, 2017; accepted July 30, 2017. The work of P. Passalacqua was supported by the National Science Foundation under Grant CAREER/EAR1350336, Grant FESD/EAR1135427, and Grant SEES/OCE-1600222. *(Corresponding author: Paola Passalacqua.)*

The authors are with the University of Texas at Austin, Austin, TX 78112 USA (e-mail: isikdogan@utexas.edu; bovik@ece.utexas.edu; paola@austin.utexas.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2017.2735443

One problem encountered when using existing water indices is that the optimal threshold values that separate water and non-water responses vary with the section of earth being imaged, hindering their global applicability. Although more sophisticated methods, such as the automated water extraction index [10], have improved the stability of methods that rely on optimal thresholds, these threshold values still vary by region. Furthermore, NDWI and MNDWI poorly differentiate between water, snow, and terrain shadows [2], even using an optimal threshold for a given region. More sophisticated rule-based systems that produce high-quality water maps [1], [2] rely on complex sets of rules and ancillary data to overcome these problems, such as the moderate resolution imaging spectroradiometer (MODIS) data, digital elevation models, and glacier inventory datasets.

Toward developing algorithms that can classify water bodies accurately, a wide variety of machine learning algorithms, including artificial neural networks, have been explored in the literature [7], [11], [12]. Many algorithms that rely on traditional artificial neural networks learn spectral characteristics of water pixels, without incorporating shape and texture information. Although these algorithms have been successful at regional scales, it has proved difficult to generalize them at the global scale, since the characteristics of water and land vary significantly across different regions [7].

Recent progress in artificial neural network research has shown the effectiveness of deep learning methods at solving different segmentation, identification, and classification problems. In particular, the use of convolutional neural networks has led to a leap forward in image recognition [13]–[15]. Recent methods have enabled per-pixel labeling of images by training end-to-end convolutional neural networks, thereby greatly advancing the state-of-the-art in semantic image segmentation [16]–[21]. The success of convolutional neural networks is a result of the culmination of novel network architectures that can learn hierarchies of features having high generalization capabilities, the availability of large datasets, and powerful hardware-accelerated computing. Large datasets that consist of pictures of everyday scenes, such as ImageNet [22] and Microsoft COCO [23], have been used in many image recognition applications [13], [15], [24], [25]. However, there has been little research conducted on applications of convolutional neural networks using large-scale remotely sensed image datasets, such as the Landsat archives. Despite some promising research on applications of convolutional neural networks for remote sensing [26]–[30], including some interesting classification approaches using deeper models [31]–[33], the potential of exploiting very large-scale Landsat imagery, even at a global scale, remains largely unexplored in deep learning applications.

Landsat archives contain remotely sensed imagery obtained with global coverage for over 40 years, which are publicly available free of charge. The currently active Landsat missions (Landsat 7 and 8) finish a complete pass around the Earth every 16 days, with an 8-day relative offset from each other. Therefore, it is possible to refresh the maps of the world's surface water every 8 days utilizing Landsat imagery. The limited availability of reliably labeled Landsat data, though, has hindered the applicability of deep learning models for water body mapping. Recently, a global inland water (GIW) dataset has been made publicly available by the global land cover facility (GLCF) [2]. The GLCF dataset has been developed using different types of methods and data, i.e., water and vegetation indices on Landsat data, terrain indices on digital elevation models, and a MODIS-based water mask. The dataset provides per-pixel labels for each Landsat image in the Global Land Survey 2000 (GLS2000) collection [34].

In this paper, we adapt convolutional neural network ideas that have been successfully applied to the semantic segmentation of everyday pictures, to the problem of surface water mapping of multispectral Landsat imagery. Specifically, we approach surface water mapping as an image segmentation problem. We utilize similarities between remotely sensed images and everyday photographs, while accounting for their differences in our neural network topology. We show that deep convolutional neural networks trained end-to-end on multispectral Landsat imagery, and on their corresponding per-pixel labels, can be used to accurately map water bodies at the global scale.

The main contributions of this paper are as follows: We designed a novel convolutional neural network architecture that is capable of learning land cover features at multiple scales from remotely sensed multispectral imagery. The model architecture (see Fig. 1) is mainly based on the well-known fully convolutional network architecture [16], [17], yet it has key differences that adapt our model to the targeted application, including a greatly reduced number of trainable parameters, the analysis at a larger number of scales, and the way the layers are connected. Using this architecture, we trained a deep-learning-based surface water model for Landsat images. Our proposed model embeds the characteristics of water bodies in context across the globe. These shape, texture, and spectral characteristics help distinguish water from snow, ice, cloud, and terrain shadows, without requiring a locally varying threshold. The model is straightforward to implement and is fast in application. As we show, the trained model delivers remarkable water mapping results.

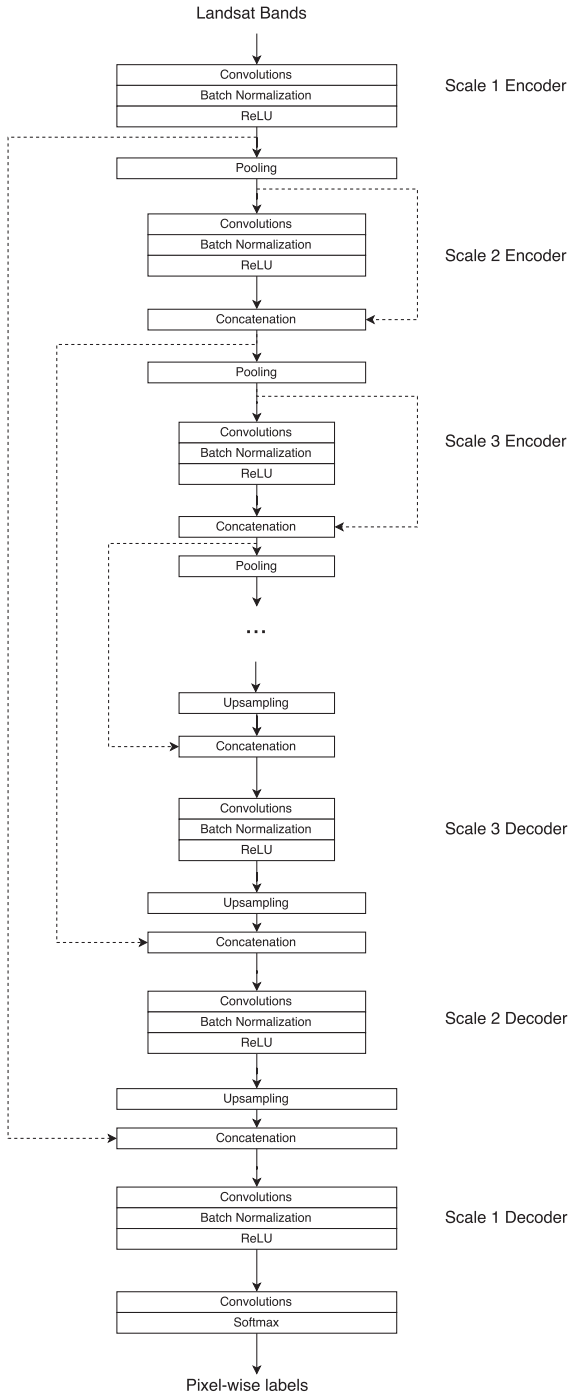


Fig. 1. Overall architecture of DeepWaterMap, which produces pixel-wise labels on a given Landsat scene. The skip connections are shown with dashed lines. The skip connections on the left combine fine and coarse layer activations. The skip connections on the right provide access to previous layer activations at each layer. This figure illustrates the simplest version of the model. More complex versions stack more convolutional blocks per scale.

II. Fully Convolutional Networks for Surface Water Mapping

A. Background

A convolutional neural network (CNN) is a type of artificial neural network that draws inspiration from the biological visual cortex. Like other types of artificial neural networks, CNNs consist of layers of interconnected neurons, which implement mathematical functions having trainable parameters. A key difference between a convolutional and an ordinary fully connected artificial neural network, such as a multilayer perceptron, is the local connectivity and weight sharing between neurons. In a fully connected layer, each neuron is connected to every neuron in the input. This is not feasible for high-dimensional inputs, such as images. Convolutional layers, on the other hand, connect each neuron to a local region of the input, where

neurons share weights. Essentially, local connectivity and weight sharing make a convolutional layer a set of image filters with trainable weights. This approach greatly reduces the number of parameters and enables learning features that provide better generalization and localization.

A typical CNN learns hierarchical image features by stacking convolutional layers with interleaved pooling layers that act as downsamplers. The outputs of a convolutional layer are passed through a nonlinear activation function before being fed into the next layer. As a simple example, a three-layer CNN block with no pooling layers may be denoted as

$$g(\mathbf{I}) = \sigma(\sigma(\sigma(\mathbf{I} * \mathbf{C_1}) * \mathbf{C_2}) * \mathbf{C_3}) \tag{1}$$

where $\mathbf{I}$ is the input image, σ is the activation function, C_n are the convolutional layer weights, and the function g represents activation at the end of the convolutional block. The convolutional layers are followed by fully connected layers at the end of the network that classify the data, given the convolutional layer activations.

Fully convolutional networks (FCNs) [16], [17] were proposed as a modification of CNN architectures that were previously designed for image classification. FCNs extend earlier models, such as AlexNet [13], GoogLeNet [14], and VGG net [15], by replacing the fully connected layers at the end of these networks with convolutional layers. This modification enables a model to accept images of arbitrary size as input and to make predictions at every pixel instead of producing a single label per image.

Layers that act like a downsampler in the models, such as pooling and convolution with a stride larger than one, limit the scale of detail in the final prediction. FCNs overcome this limitation by combining features at different resolution levels via skip connections that connect layers at different scales. Fusing fine and coarse layers makes it possible to recover fine spatial information discarded by the coarse layers, while preserving coarse structures.

FCNs have produced promising image segmentation results on everyday images [16], [17]. Everyday photographs and Landsat images greatly differ in the number of spectral bands and the range of image sizes. Everyday pictures consist of bands in the visible spectrum (e.g., RGB), while Landsat images also include infrared bands. The number of bands can easily be adjusted by modifying the number of nodes in the input layer. The range of image scales can be much larger in remotely sensed images (e.g., a 2000-m-wide river versus a 30-m-wide river) as compared to photographs normally taken from a human point of view (e.g., a bus versus a person). Generally, we have found that the development of remotely sensed image segmentation models greatly benefit from conducting the analysis over a larger number of scales.

B. DeepWaterMap: A Deep-Learning-Based Water Model

Our model, which we call DeepWaterMap, is a multiscale fully convolutional neural network that acts like an encoder–decoder network. DeepWaterMap has two types of skip connections that connect nonconsecutive layers (see Fig. 1). The first type of skip connection (Fig. 1 dashed lines on the left) is similar to those used in FCNs, where fine and upsampled coarse layers are fused together by summing the predictions made at different scales. Our network modifies this idea by replacing the summation operation by concatenation followed by a convolutional layer. The convolutional layer in the decoder network learns how to fuse activations at different scales instead of simply summing the activations. The idea of fusing activations instead of summing the scores was mentioned in [17], but the latter was preferred for memory efficiency. To achieve memory efficiency, we use a small and fixed number of filters (e.g., 16) per layer. Each scale in the encoder network reuses the information from previous layers, allowing us to reduce the number of filters at each convolutional layer without compromising accuracy. The second type of skip connection, which wraps around the convolutional layers in the encoder network (Fig. 1 dashed lines on the right), makes it possible to reuse features from previous layers. This type of skip connection has been shown to be useful for efficiently training deep convolutional neural networks in the ResNet [25] and DenseNet [35] papers.

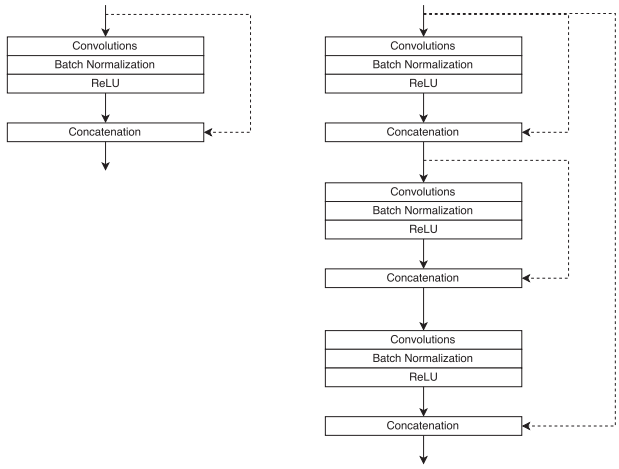


Fig. 2. Convolutional blocks at a single scale. A single convolutional block (left) and a block with three convolutional layers (right).

In our model, we adopt a bottom-up approach, by gradually increasing the network complexity. The simplest version of DeepWaterMap has a single convolutional layer at each scale (see Fig. 1). More complex versions use multiple convolutional layers per scale, where the added layers can learn more complex feature hierarchies and discover more complex patterns. To achieve memory efficiency, the last skip connection in the complex versions skips over the middle convolutional blocks (see Fig. 2). We tested the networks with one, three, and five convolutional layers per scale, and chose the number of convolutional layers in a convolutional block to be 3, since further increasing the number of layers did not improve the overall accuracy at the global scale (see Section IV). All three variants of our model produced visually similar results where the difference in the overall accuracy was observed only at the global scale (see Table I).

We set the number of scales to 10 to maximize the receptive field for the input size so that the model can make use of all

TABLE I
COMPARISON OF MODELS: A TRADITIONAL MLP AND DEEPWATERMAP WITH ONE-, THREE-, AND FIVE-LAYER CONVOLUTIONAL BLOCKS

	Precision	Recall	Com. Err.	Om. Err.	F1
MNDWI	0.55	0.98	0.45	0.02	0.70
MLP	0.61	0.67	0.39	0.33	0.64
DeepWaterMap-1	0.81	0.94	0.19	0.06	0.87
DeepWaterMap-3	0.91	0.88	0.09	0.12	0.90
DeepWaterMap-5	0.92	0.87	0.08	0.13	0.90

TABLE II
CONFUSION MATRIX FOR THE MLP PREDICTED RESULTS

Actual \Predicted	Land	Water	Snow/Ice	Shadow	Cloud
Land	**0.78**	0.06	0.01	0.01	0.13
Water	0.01	**0.67**	0.14	0.17	0.01
Snow/Ice	0.01	0.70	**0.29**	0.00	0.00
Shadow	0.15	0.58	0.06	**0.17**	0.05
Cloud	0.42	0.46	0.04	0.01	**0.07**

TABLE III
CONFUSION MATRIX FOR THE RESULTS PREDICTED BY DEEPWATERMAP WITH THREE-LAYER CONVOLUTIONAL BLOCKS

Actual \Predicted	Land	Water	Snow/Ice	Shadow	Cloud
Land	**0.91**	0.01	0.01	0.04	0.03
Water	0.00	**0.88**	0.02	0.07	0.03
Snow/Ice	0.00	0.02	**0.88**	0.02	0.08
Shadow	0.05	0.33	0.00	**0.59**	0.03
Cloud	0.26	0.03	0.06	0.11	**0.55**

contextual information available in a given sample input. The scales are implemented by pooling and upsampling layers that downsample and upsample the layer activations by factors of two, respectively. The pooling layers perform a max-pooling operation using a window size of 2×2, by forward propagating the maximum value within this window. The upsampling layers use transposed convolutions having parameters initialized to compute bilinear interpolation. This architecture provides a broad description of the context for a given spatial location through a hierarchy of multiscale features. Despite its depth, our model architecture allows the number of trainable parameters to remain small (1.5 M parameters in our largest model as compared to 134 M parameters in the original fully convolutional network architecture), thereby greatly reducing the risk of overfitting, as well as memory and processing power requirements. All variants of our model required less than a minute to fully process a full-size Landsat tile on an NVIDIA Tesla P100 GPU.

All of the convolutional layers except the first and last layers deploy 3×3 filters. The first and last layers consist of 1×1 filters, acting as point operations that compute weighted averages across filter activations, to minimize the loss of detail arising from the spatial convolutions.

The convolutional layers in DeepWaterMap are followed by batch normalization [36] and rectified linear unit (ReLU) activation layers (i.e., $\max(0, x)$). Batch normalization, which normalizes layer activations by the batch mean and variance, serves two main purposes in our model. First, it reduces the internal covariate shift problem [36]. Covariate shift refers to the phenomenon where the distribution of inputs at each layer changes as the previous layer parameters are updated. In deep network architectures, even small changes in the distributions of the outputs of the early layers are amplified through the network, thereby causing changes in the distribution of the internal layer inputs, and ultimately the output classifications. This complicates the training of deep neural networks and slows convergence during training. Second, batch normalization enables multiscale feature concatenation by making the magnitudes of different scale activations comparable. Naively concatenating the layer activations without any type of normalization scheme could cause features having larger magnitudes to dominate features having smaller magnitudes.

The final convolutional layer in our model has one filter for each class label, which acts as a scoring layer on the class probabilities. This layer uses a normalized exponential function (softmax) at the output to obtain pseudoprobabilities on the class labels. Finally, pixels where the water class has the greatest probability are labeled as water.

III. DATA PREPARATION AND TRAINING

We matched the Landsat 7 ETM+ images in the GLS2000 collection [34] with the corresponding per-pixels labels in the GLCF inland water dataset [2] to create the training and test datasets for all variants of the DeepWaterMap model. We included all reflective bands except the panchromatic channel. The panchromatic channel, which has higher resolution than the rest of the channels, could be included to compute higher resolution water maps if ground-truth labels were available that matched the resolution of the band.

Certain classes in the dataset, such as snow/ice, shadow, and clouds, have a relatively smaller number of pixels compared to the others. Using a uniform cost function in such a dataset could cause the classes with a relatively higher occurrence, such as land and water, to dominate the model. This class imbalance problem can be addressed using a class-weighted cost function, where a higher cost is assigned to misclassification of smaller classes. We used a median frequency balanced cross-entropy function [37] as the cost function to be minimized during training. Median frequency balancing assigns a weight to a class as $w_c = f_{\text{median}}/f_c$, where f_c is the frequency of a class c and f_{median} is the median of class frequencies. Using this median frequency balanced cost function encourages the models to separate snow, ice, shadow, and clouds from water, despite their relatively rare occurrence.

Our models can input images of arbitrary size during inference. However, during training, all images in a minibatch need to have the same dimensions, and the layer activations for the entire batch need to fit the available memory. Therefore, we cropped 512×512 pixel nonoverlapping patches from the Landsat images, using a sliding window. We skipped "empty" patches where more than 99% of the pixels were labeled as

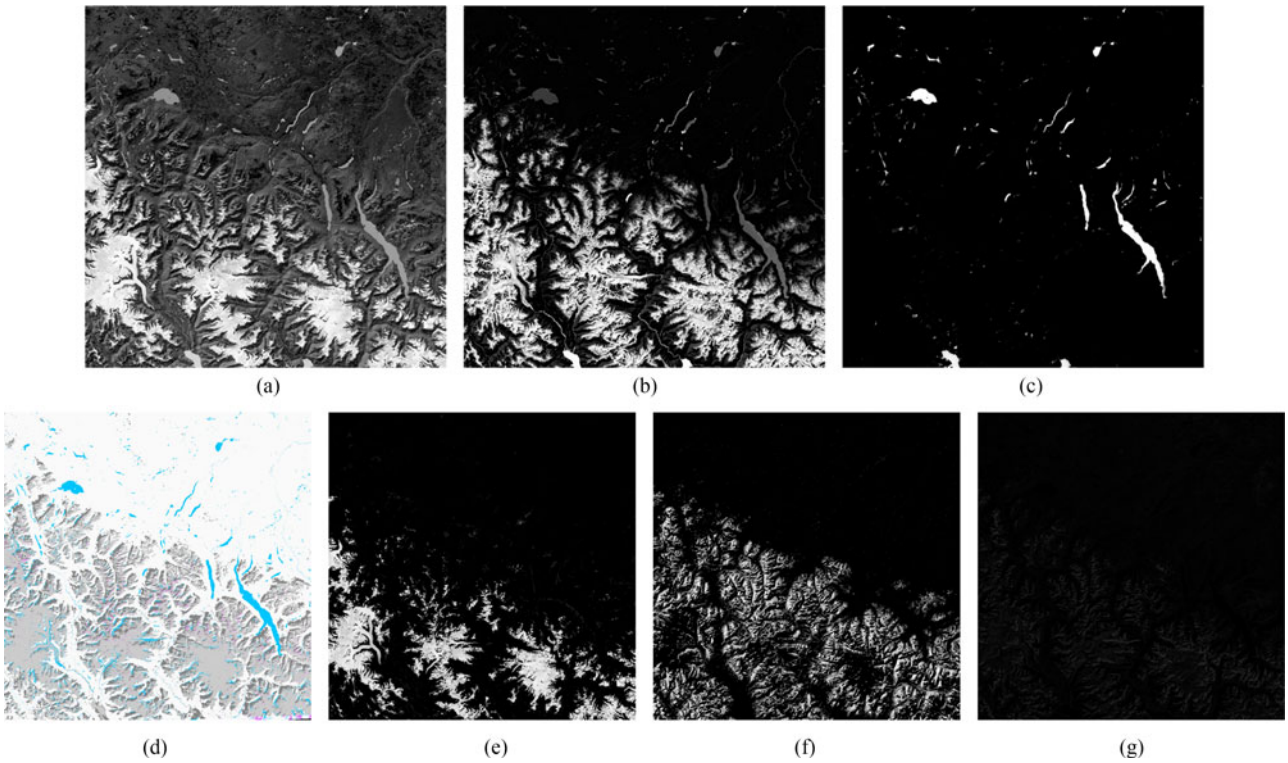


Fig. 3. High latitudes in North America: WRS-2 path/row: 49/24, British Columbia, Canada. (a) MNDWI response, (b) traditional MLP estimate for water probability, (c) DeepWaterMap-3 estimate for water probability, (d) corresponding labels in the GLCF GIW dataset (blue: water, pink: snow/ice, and dark and light gray: cloud and shadows), (e)–(g) DeepWaterMap-3 estimates for snow/ice, shadow, and cloud probabilities, respectively. DeepWaterMap successfully distinguishes between water, snow/ice, and shadow, while MNDWI and MLP fail to separate these classes from water. (a) MNDWI, (b) MLP water, (c) DeepWaterMap water, (d) GLCF GIW labels, (e) DeepWaterMap snow/ice, (f) DeepWaterMap shadow, (g) DeepWaterMap cloud.

land. The resulting dataset contained more than 1.4 million labeled multispectral image patches. We randomly selected 80% of these patches for training and the remaining 20% for testing.

When the training data are scarce, transferring pretrained parameters from existing models as in [31], and [38] or preprocessing the input data as in [32] and [33] can be useful. Given the large number of samples in our training set, we did not need to transfer features or preprocess the input images. Training our models from scratch allowed for a greater flexibility in our model architecture. Using the input images as is without any preprocessing let our models learn to extract useful features directly from data.

We designed our model to utilize all context information available in a given training sample by maximizing the receptive field. We chose the number of scales to be $\lfloor \log_2 \min(N, M) + 1 \rfloor$ that evaluates to 10 for a training input size $N = M = 512$. Thus, the coarsest scale had a receptive field of 512 $\times$ 512 pixels. In other words, the coarsest scale had access to all pixels in the input.

Very deep convolutional neural networks, like our DeepWaterMap model, have difficulty converging if the parameters are randomly initialized. The weight initialization method described in [39] provides a robust scheme for initializing very deep models. We initialized the weights in all convolutional layers, except the upsampling layers (which are initialized to compute bilinear interpolation) using this scheme.

We optimized the weights using the adaptive moment estimation (Adam) algorithm [40] using the recommended default hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a base learning rate $\lambda = 10^{-4}$. The Adam algorithm computes adaptive learning rates for different parameters and reduces the impact of tuning the hyperparameters on convergence. We trained all models at once, without training in stages or fine tuning, until the training loss converged.

We trained three different versions of the DeepWaterMap model, having one, three, and five convolutional layers per scale, respectively. We shuffled the training set once before training and trained the models with minibatches of eight samples. Training and testing all three models took less than 3 days on a server equipped with three NVIDIA Tesla P100 GPUs. As a benchmark, we also trained a traditional multilayer perceptron (MLP) on the same training set. The benchmark neural network had 30 hidden nodes, similar to the model in [12], which was also trained to classify cloud, shadow, water, snow/ice, and clear sky pixels.

IV. Results

We tested each model, namely the MLP and DeepWaterMap with one-, three-, and five-layer convolutional blocks, with regards to water pixel classification performance on the test set. We also ran a simple water classifier on the test set by thresholding

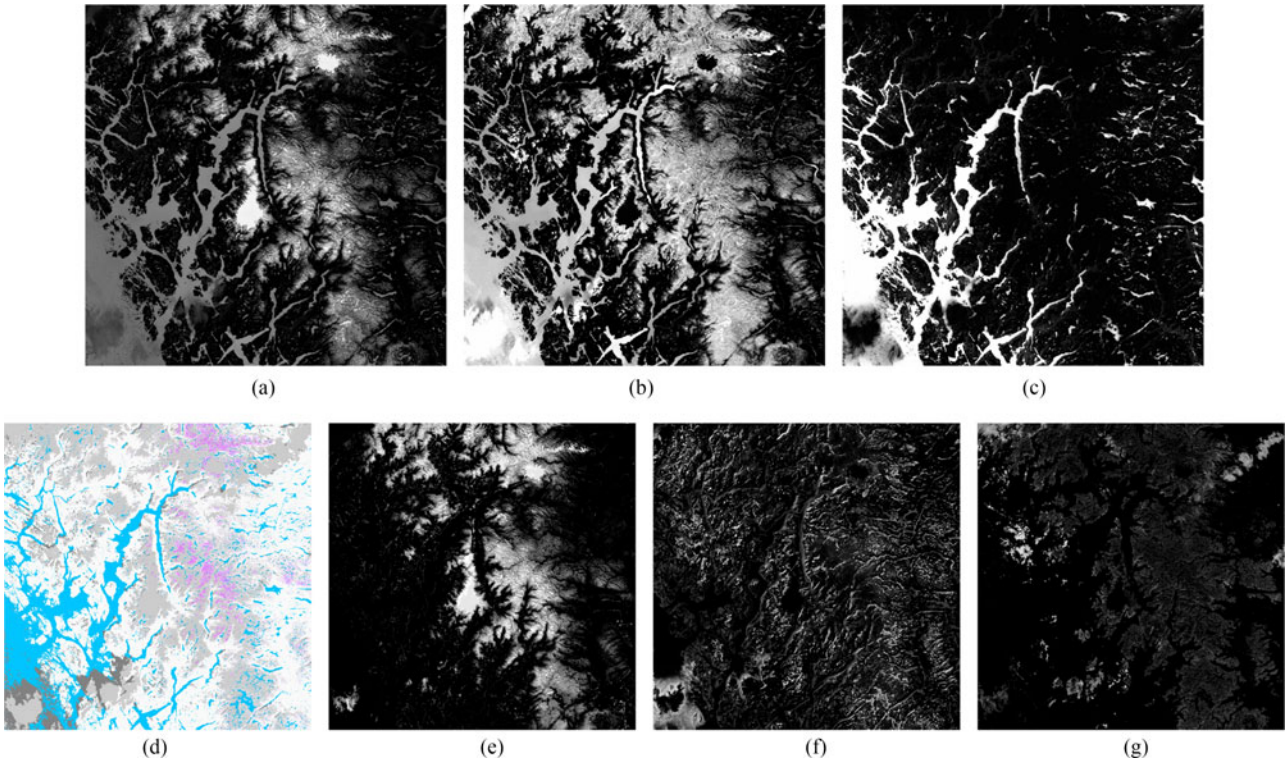


Fig. 4. High latitudes in Europe: WRS-2 path/row: 200/18, Bergen, Norway. (a) MNDWI response, (b) traditional MLP estimate for water probability, (c) DeepWaterMap-3 estimate for water probability, (d) corresponding labels in the GLCF GIW dataset (blue: water, pink: snow/ice, dark and light gray: cloud and shadows), (e)–(g) DeepWaterMap-3 estimates for snow/ice, shadow, and cloud probabilities, respectively. DeepWaterMap successfully distinguishes between water from snow/ice, shadow, and clouds, while MLP classifies them as water. (a) MNDWI, (b) MLP water, (c) DeepWaterMap water, (d) GLCF GIW labels, (e) DeepWaterMap snow/ice, (f) DeepWaterMap shadow, (g) DeepWaterMap cloud.

the MNDWI response at zero, as suggested in the original MNDWI paper [9]. We compared the models using precision (user accuracy), recall (producer accuracy), and corresponding commission and omission errors. Precision denotes the ratio of pixels that are correctly classified as water to all pixels classified as water, while recall is the ratio of detected water pixels to all ground-truth water pixels. As an overall performance measure, we used the F1-score, which is the harmonic mean of precision and recall (see Table I). The simple MNDWI classifier yielded many false positives, which led to a high commission error. The MLP model had lower commission error and higher omission error rates as compared to MNDWI. All three versions of the DeepWaterMap models delivered better overall performance than the MNDWI and MLP classifiers. Increasing the number of layers in the convolutional blocks in the DeepWaterMap models improved the F1-score, saturating at three layers per block. Given that the ground truth had commission errors $<5\%$ and omission errors $<15\%$ relative to established national datasets [2], the water classification performance of our three and five-block models was close to the limits defined by the training data.

The confusion matrices (see Table II and III) show that our model significantly outperformed the traditional MLP approach at discriminating water from other classes. The traditional MLP model learns the spectral response (pixel intensity values) for different class labels. Our fully convolutional models, on the other hand, are capable of learning multiscale shape and texture features in addition to spectral response. These features help discriminate between classes where the spectral responses may be similar, such as water and shadows.

We show qualitative results on some images obtained from across the globe that are part of the GLS 2000 collection of Landsat images (see Figs. 3–7). The images include areas having varying characteristics: high latitudes (see Figs. 3 and 4), river channels in a tropical rainforest (see Fig. 5), urban areas (see Fig. 6), and river deltas with vegetation (see Figs. 7 and 8) in different continents. We visualize the model outputs by mapping the class probabilities p_c to a grayscale ramp, where $p_c = 0$ is black and $p_c = 1$ is white. We also show the GLCF GIW dataset labels for the corresponding regions for reference. The qualitative results were aligned with the quantitative results. The visualizations show that a simple MLP network trained on a global dataset poorly separates water from snow/ice, shadow, cloud, and urban areas as compared to the DeepWaterMap model. In urban areas (see Fig. 6), even the simple MNDWI index provides better separation between land and water, since it was designed to suppress built-up noise.

DeepWaterMap was able to successfully detect underrepresented classes, including snow/ice, shadow, and clouds, despite their relatively lower accuracy in the quantitative results. One reason that the quantitative results show lower accuracy on these classes may be that clouds and shadows are not discrete objects.

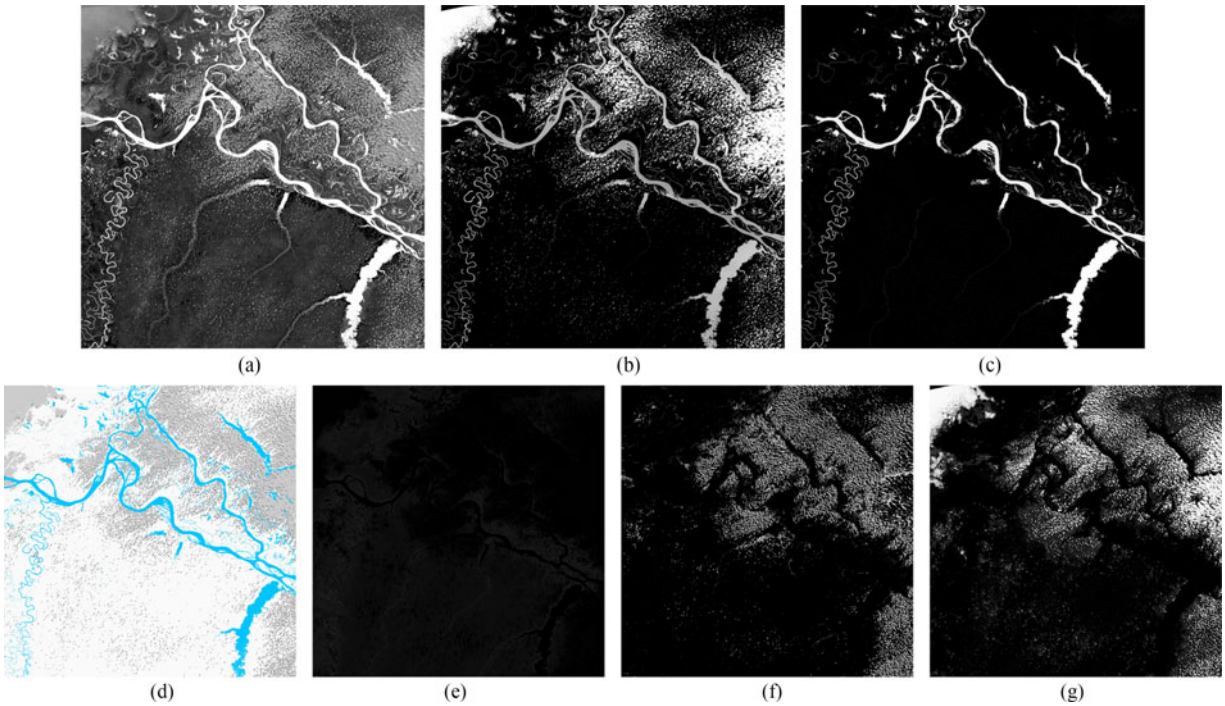


Fig. 5. River channels in a tropical rainforest: WRS-2 path/row: 1/62, Amazonas, Brazil. (a) MNDWI response, (b) traditional MLP estimate for water probability, (c) DeepWaterMap-3 estimate for water probability, (d) corresponding labels in the GLCF GIW dataset (blue: water, pink: snow/ice, dark and light gray: cloud and shadows), (e)–(g) DeepWaterMap-3 estimates for snow/ice, shadow, and cloud probabilities, respectively. DeepWaterMap successfully detects clouds and their shadows, while MLP assigns them high probabilities of being water. (a) MNDWI, (b) MLP water, (c) DeepWaterMap water, (d) GLCF GIW labels, (e) DeepWaterMap snow/ice, (f) DeepWaterMap shadow (g) DeepWaterMap cloud.

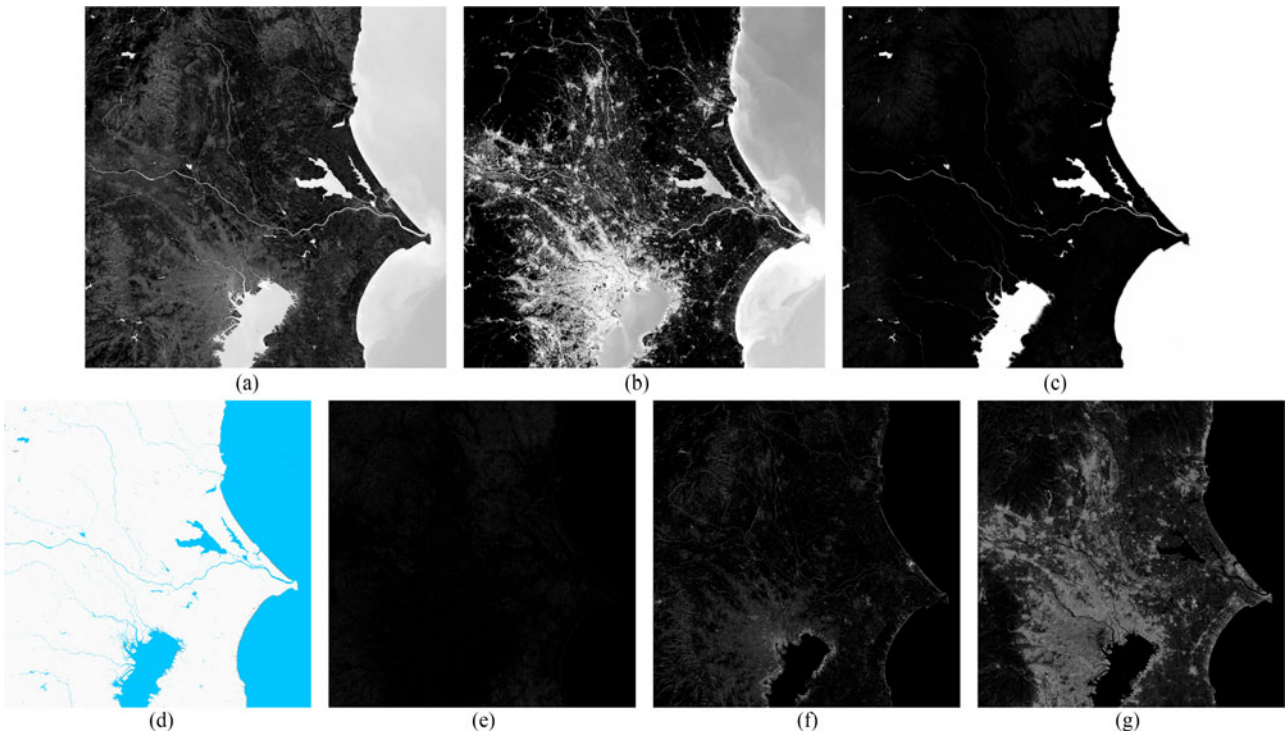


Fig. 6. Coastal urban area: WRS-2 path/row: 107/305, Tokyo, Japan. (a) MNDWI response, (b) traditional MLP estimate for water probability, (c) DeepWaterMap-3 estimate for water probability, (d) corresponding labels in the GLCF GIW dataset (blue: water, pink: snow/ice, dark and light gray: cloud and shadows), (e)–(g) DeepWaterMap-3 estimates for snow/ice, shadow, and cloud probabilities, respectively. DeepWaterMap correctly classifies water, while MLP fails to suppress built-up noise. (a) MNDWI, (b) MLP water, (c) DeepWaterMap water, (d) GLCF GIW labels, (e) DeepWaterMap snow/ice, (f) DeepWaterMap shadow, (g) DeepWaterMap cloud.

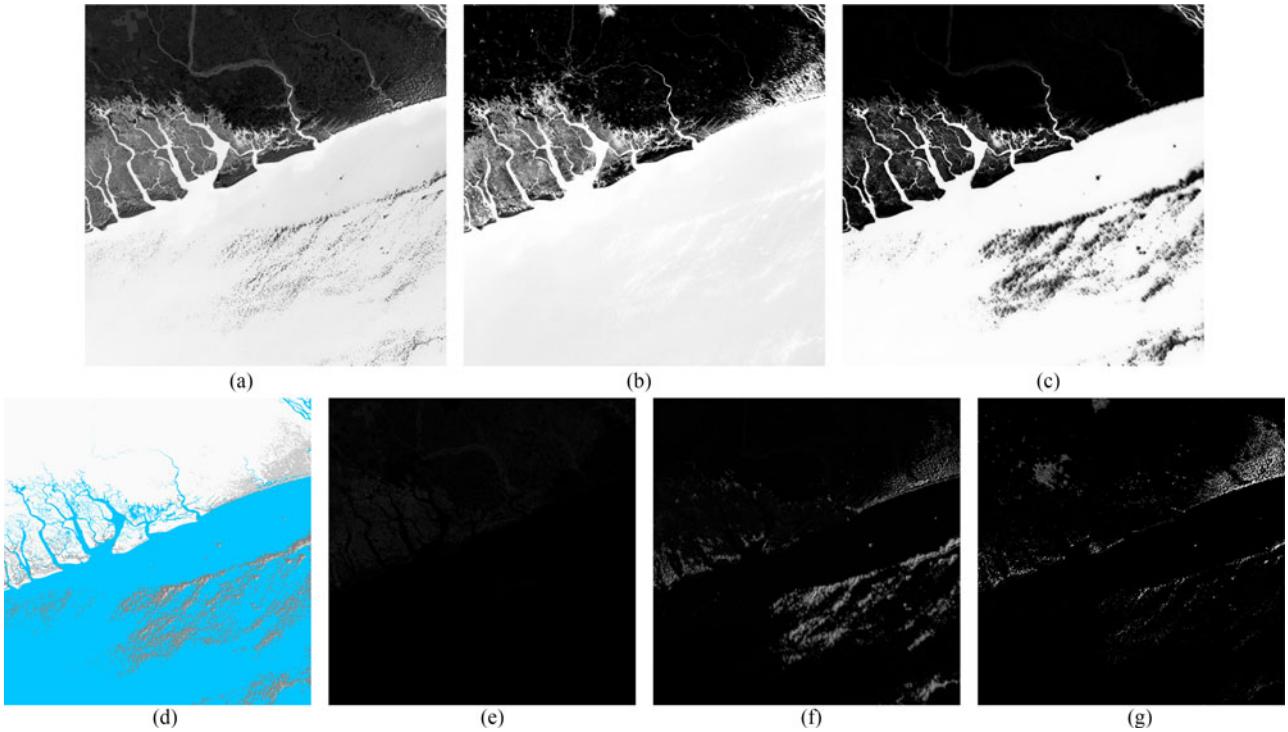


Fig. 7. River delta: WRS-2 path/row: 188/57, Niger delta, Nigeria. (a) MNDWI response, (b) traditional MLP estimate for water probability, (c) DeepWaterMap-3 estimate for water probability, (d) corresponding labels in the GLCF GIW dataset (blue: water, pink: snow/ice, dark and light gray: cloud and shadows), (e)–(g) DeepWaterMap-3 estimates for snow/ice, shadow, and cloud probabilities, respectively. DeepWaterMap separates vegetation, clouds, and cloud shadows from water.

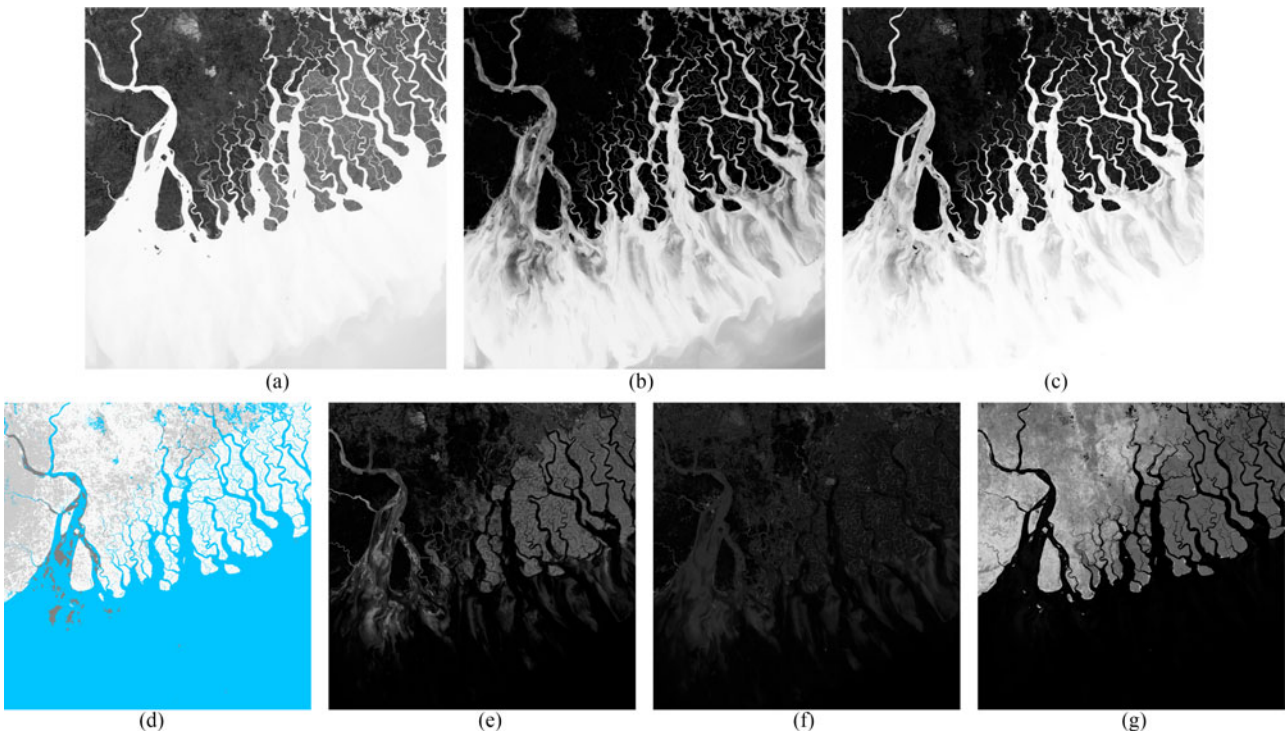


Fig. 8. River delta with mangrove forest: WRS-2 path/row: 138/45, a portion of the Brahmaputra–Jamuna delta, India and Bangladesh. (a) MNDWI response, (b) traditional MLP estimate for water probability, (c) DeepWaterMap-3 estimate for water probability, (d) corresponding labels in the GLCF GIW dataset (blue: water, pink: snow/ice, dark and light gray: cloud and shadows), (e)–(g) DeepWaterMap-3 estimates for snow/ice, shadow, and cloud probabilities, respectively. DeepWaterMap separates vegetation from water. However, false positives are observed in the cloud and snow/ice classes.

Thus, it is difficult to define binary labels on the cloud and shadow classes. Furthermore, the errors on these classes were not reported in the GLCF dataset, which we used as the ground truth. As may be seen from the qualitative results, the labels for these classes were not always precise in the ground-truth dataset. Overall, our model learned to generalize well from noisy data. However, on some input images (e.g., Fig. 8), the model delivered false positives on these underrepresented classes, leading to confusion between the nonwater classes. Some of the underrepresented classes had higher weights in the cost function during training due to class balancing, which likely led to these false positives.

As shown in these experiments, DeepWaterMap was able to generalize the characteristics of water globally, resulting in a high classification accuracy, particularly for the water class. No noticeable loss of detail was observed in the outputs of DeepWaterMap, showing that the model was able to efficiently learn to fuse multiscale features.

We tested our model globally and focused on its ability to learn features at the global scale; independent of the type of terrain and the atmospheric conditions. Our model works well across terrain types and atmospheric conditions. Finding a way to segment the entire earth into different types of terrains and test our model for different earth regions would require a rather major effort. Yet, we recognize that such a study would be of great interest.

V. Conclusion

We presented a deep fully convolutional neural network model, called DeepWaterMap, to map surface water on Landsat imagery. In our model, we adopted a data-driven approach, thereby removing the need for manually selected threshold values and other hand-crafted rules on different regions and conditions. The model learns the global characteristics of land, water, snow/ice, shadow, and clouds, including their shape, texture, and spectral response. The model separates between these classes using Landsat bands, without requiring ancillary data. The multiscale feature fusion in the model helps preserve the amount of detail while taking the context into account during per-pixel classification of the input. Our results show that our model performs significantly better than the simple modified normalized difference water index and the traditional multilayer perceptron approach at discriminating water from other surface land cover.

DeepWaterMap can be applied on a variety of different problems involving diverse terrains, seasonal states, and particular water networks (e.g., deltas). With minimal modification, DeepWaterMap can be trained for other tasks involving remotely sensed images, such as classifying other types of land cover (e.g., vegetation, forests, and urban areas). The maps generated by our model would help us better understand environmental change and predict our planet's future.

References

[1] J.-F. Pekel, A. Cottam, N. Gorelick, and A. S. Belward, "High-resolution mapping of global surface water and its long-term changes," *Nature*, vol. 540, pp. 418–422, 2016.

[2] M. Feng, J. O. Sexton, S. Channan, and J. R. Townshend, "A global, high-resolution (30-m) inland water body dataset for 2000: First results of a topographic–spectral classification algorithm," *Int. J. Digit. Earth*, vol. 9, no. 2, pp. 113–133, 2016.

[3] N. Mueller *et al.*, "Water observations from space: Mapping surface water from 25 years of Landsat imagery across Australia," *Remote Sens. Environ.*, vol. 174, pp. 341–352, 2016.

[4] D. Yamazaki, M. A. Trigg, and D. Ikeshima, "Development of a global˜ 90 m water body map using multi-temporal Landsat images," *Remote Sens. Environ.*, vol. 171, pp. 337–351, 2015.

[5] C. Verpoorter *et al.*, "Automated mapping of water bodies using Landsat multispectral data," *Limnol. Oceanogr. Methods*, vol. 10, pp. 1037–1050, 2012.

[6] M. Carroll, J. R. Townshend, C. M. DiMiceli, P. Noojipady, and R. Sohlberg, "A new global raster water mask at 250 m resolution," *Int. J. Digit. Earth*, vol. 2, no. 4, pp. 291–308, 2009.

[7] A. Karpatne, A. Khandelwal, X. Chen, V. Mithal, J. Faghmous, and V. Kumar, "Global monitoring of inland water dynamics: State-of-the-art, challenges, and opportunities," in *Computational Sustainability*. Cham, Switzerland: Springer, 2016, pp. 121–147.

[8] S. K. McFeeters, "The use of the normalized difference water index (NDWI) in the delineation of open water features," *Int. J. Remote Sens.*, vol. 17, no. 7, pp. 1425–1432, 1996.

[9] H. Xu, "Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery," *Int. J. Remote Sens.*, vol. 27, no. 14, pp. 3025–3033, 2006.

[10] G. L. Feyisa, H. Meilby, R. Fensholt, and S. R. Proud, "Automated water extraction index: A new technique for surface water mapping using Landsat imagery," *Remote Sens. Environ.*, vol. 140, pp. 23–35, 2014.

[11] H. Bischof, W. Schneider, and A. J. Pinz, "Multispectral classification of landsat-images using neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 30, no. 3, pp. 482–490, 1992.

[12] M. J. Hughes and D. J. Hayes, "Automated detection of cloud and cloud shadow in single-date Landsat imagery using neural networks and spatial post-processing," *Remote Sens.*, vol. 6, no. 6, pp. 4907–4926, 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[14] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations*, 2015.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.

[17] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[18] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.

[19] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," in *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, doi: 10.1109/TPAMI.2016.2644615.

[20] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," arXiv:1511.02680, 2015.

[21] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," in *Proc. Int. Conf. Learning Representations*, 2016.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[23] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*. Cham, Switzerland: Springer, 2014, pp. 740–755.

[24] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3128–3137.

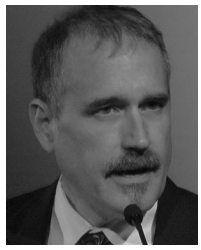
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[26] 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, 2016, Art. no. 329.

[27] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.
[28] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," arXiv:1508.00092, 2015.
[29] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: A learning framework for satellite imagery," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst*, 2015, Art. no. 37.
[30] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.
[31] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14 680–14 707, 2015.
[32] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. 2015 IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 4959–4962.
[33] B. Pan, Z. Shi, and X. Xu, "R-vcanet: A new deep-learning-based hyperspectral image classification method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1975–1986, May 2017.
[34] G. Gutman, R. Byrnes, M. Covington, C. Justice, S. Franks, and R. Headley, "Towards monitoring land-cover and land-use changes at global scale: The global land use survey," *Photogrammetric Eng. Remote Sens.*, vol. 64, pp. 6–10, 2005.
[35] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learning*, 2015.
[37] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2650–2658.
[38] O. A. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2015, pp. 44–51.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
[40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, 2015.

Furkan Isikdogan received the B.S. degree in computer engineering from Yildiz Technical University, Istanbul, Turkey, in 2011, the M.S. degree in computer engineering from Bogazici University, Istanbul, Turkey, in 2013, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2017.

He is currently an Imaging Algorithm Staff Engineer with Motorola Mobility/Lenovo, Chicago, IL, USA. His research interests include image and video processing, computer vision, machine learning, and remote sensing.



Alan C. Bovik (F'95) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1980, 1982, and 1984, respectively.

He is currently the Cockrell Family Regents Endowed Chair Professor at the University of Texas at Austin, Austin, TX, USA. His books include *The Handbook of Image and Video Processing* (Academic, 2000), *Modern Image Quality Assessment* (Morgan & Claypool Publishers, 2006), and *The Essential Guides to Image and Video Processing* (Academic, 2009).

Dr. Bovik received the 2017 Edwin H. Land Medal from the Optical Society of America, a 2015 Primetime Emmy Award for Outstanding Achievement in Engineering Development, and the 2013 IEEE Signal Processing Society 'Society' Award, along with about 10 journal 'best paper' awards. He cofounded and was longest-serving Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and created the IEEE International Conference on Image Processing in Austin, Texas, 1994.

Paola Passalacqua received the B.S. degree in environmental engineering from the University of Genoa, Genoa, Italy, in 2002, and the M.S. and Ph.D. degrees in civil engineering from the University of Minnesota, Minneapolis, MN, USA, in 2005 and 2009, respectively.

She is currently an Assistant Professor of environmental and water resources engineering in the Civil, Architectural and Environmental Engineering Department, University of Texas at Austin, Austin, TX, USA. Her research interests include network analysis and dynamics of hydrologic and environmental transport on river networks and deltaic systems, lidar and satellite imagery analysis, multiscale analysis of hydrological processes, and quantitative analysis and modeling of landscape forming processes.

Dr. Passalacqua received the National Science Foundation CAREER award (2014) and several teaching awards including the 2016 Association of Environmental Engineering and Science Professors (AEESP) Award for Outstanding Teaching in Environmental Engineering and Science.