Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images

Claudio Persello, *Senior Member, IEEE*, and Alfred Stein

***Abstract*— This letter investigates fully convolutional networks (FCNs) for the detection of informal settlements in very high resolution (VHR) satellite images. Informal settlements or slums are proliferating in developing countries and their detection and classification provides vital information for decision making and planning urban upgrading processes. Distinguishing different urban structures in VHR images is challenging because of the abstract semantic definition of the classes as opposed to the separation of standard land-cover classes. This task requires extraction of texture and spatial features. To this aim, we introduce deep FCNs to perform pixel-wise image labeling by automatically learning a higher level representation of the data. Deep FCNs can learn a hierarchy of features associated to increasing levels of abstraction, from raw pixel values to edges and corners up to complex spatial patterns. We present a deep FCN using dilated convolutions of increasing spatial support. It is capable of learning informative features capturing long-range pixel dependencies while keeping a limited number of network parameters. Experiments carried out on a Quickbird image acquired over the city of Dar es Salaam, Tanzania, show that the proposed FCN outperforms state-of-the-art convolutional networks. Moreover, the computational cost of the proposed technique is significantly lower than standard patch-based architectures.**

***Index Terms*— Convolutional neural networks (CNNs), deep learning, image classification, informal settlements, very high resolution (VHR) satellite imagery.**

I. Introduction

INFORMAL settlements are a global urban phenomenon. According to UN-Habitat, the United Nations (UN) program for human settlements, informal settlements (or slums) are residential areas that: 1) lack legal tenure; 2) may not comply with building regulations; and 3) are usually characterized by dense substandard housing, poor living conditions, and deficiency in basic services and city infrastructure [1]. According to UN statistics, around one-quarter of the world urban population lives in slums and this proportion rises to 56% in sub-Saharan Africa [2].

Since the availability of very high resolution (VHR) satellite images, remote sensing (RS) has been used for mapping location and extent of informal settlements, providing essential information for managing and planning urban upgrading projects [3]. Formal and informal settlements can be distinguished in VHR images based upon physical and morphological characteristics of the urban structure. Slums are usually densely built-up areas characterized by irregular layout of buildings and almost no presence of vegetation. Fig. 1 shows an example of formal and informal settlements in Dar es Salaam. The detection of slums in VHR images is, however, a difficult task. The spectral information alone is insufficient for discriminating different urban typologies. In order to accurately distinguish informal settlements from other built-up areas, it is necessary to extract spatial features capable of capturing long-range pixel dependency in the image. State-of-the-art methods are based on the extraction of features specifically designed to address the problem at hand. Popular choices are texture statistics, local binary patterns, morphological profiles, oriented gradients, wavelet transforms, and segment-based features. These methods depend on several free parameters, which are usually set according to user experience or by trial and error. An exhaustive optimization of those parameter values is computationally costly, especially when large spatial neighborhoods are considered.

Fig. 1. Subset of the Quickbird data set considered in this letter illustrating different urban structures in Dar es Salaam: 1) formal urban area in the top and 2) an informal settlement in the bottom of the image.

Deep learning methods such as convolutional neural networks (CNNs) can overcome the above-mentioned issue by automatically learning spatial features from the input image [4]. CNNs are composed of a sequence of processing layers that perform three main operations: 1) 2-D convolutions; 2) unit-wise nonlinear activations; and 3) spatial pooling with subsampling. Weights and biases of the convolution operations are parameters learned through a supervised

Manuscript received June 21, 2017; revised September 6, 2017; accepted October 11, 2017. Date of publication November 3, 2017; date of current version December 4, 2017. *(Corresponding author: Claudio Persello.)*

The authors are with the Faculty of Geo-Information Science and Earth Observation, University of Twente, 7522 NB Enschede, The Netherlands (e-mail: c.persello@utwente.nl; a.stein@utwente.nl).

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

Digital Object Identifier 10.1109/LGRS.2017.2763738

learning process in order to minimize the classification error. Standard architectures use a series of convolutional layers to extract feature maps, which are then flattened into a 1-D vector and fed to a fully connected network. The convolutional layers are responsible for learning the spatial features, whereas the fully connected layers learn the classification rule to be applied to the extracted feature vector. The network is trained in an *end-to-end* fashion; hence, feature extraction and classification occur simultaneously in a single framework. This approach has shown to be effective in various computer vision tasks, including multimedia image classification where one label is assigned to the entire input scene. Deep CNNs have been successfully applied to image categorization benchmarks, considerably outperforming techniques based on hand-crafted features.

CNNs have also been adapted to perform pixel-wise image classification, also known as semantic segmentation. The standard patch-based approach consists in training the CNN to label the central pixel of patches extracted from the input image [5]. This, however, results in redundant processing at inference time and therefore in high computational cost when applied to large RS images. In this letter, we adopt fully convolutional networks (FCNs), which are trained to infer pixel-wise labels of the entire input image or patch directly. In these networks, the fully connected layers are usually substituted by one or multiple layers that upsample the feature maps extracted by the convolutional layers to the resolution of the input image [6]–[8]. Shelhamer *et al.* [6] adapted contemporary CNNs into FCNs and fine-tuned them to address semantic segmentation. In [7] and [8], architectures with multiple deconvolution and unpooling layers have been adopted. In the context of RS, similar FCN architectures based on a downsample-then-upsample scheme have been applied to the classification of aerial [9] and satellite images [10].

In this letter, we introduce deep FCNs for the detection of informal settlements in VHR images. We design a novel architecture using six layers of dilated convolutions interleaved by max pooling with no downsampling and nonlinear activation functions. Such a network does not require deconvolution or upsampling modules, because every layer is designed to output feature maps of the same spatial resolution of the input image. In order to capture large patterns in the image, instead of downsampling the image, we use dilated convolutions to enlarge the receptive field of the network while restraining the number of learnable parameters and therefore limiting the risk of overfitting [11].

II. Deep FCN With Dilated Convolutions

The proposed FCN consists of a sequence of spatial filter banks with parameters that are learned to extract informative spatial features and gradually transform the input image into the classification map. Unlike standard convolutional filters, with kernel values that are fixed *a priori* to extract specific image features (e.g., edges), the kernel values of the FCN are learned in a supervised manner by minimizing a loss function.

A. Convolution With Dilated Kernel

The main building blocks of our network are convolutional layers. They compute the convolution of the input image with a bank of filter kernels and add a bias term. The learnable weights of the filter bank are represented by a 4-D array $\mathbf{w} \in \mathbb{R}^{H\times W\times D\times K}$, where H and W are the height and width of the kernel, D is the number of input feature channels, and K is the number of filters. Additional parameters that control the size of the output are the *stride* and *zero padding*. The stride s is the spatial interval between convolution centers and defines the downsampling factor ($s = 1$ means no downsampling). Padding consists in adding zeros around the border of the input image before applying the filter. The parameter p defines the number of pixels used for zero padding. Given an input image $\mathbf{x}$ of size $M \times N \times D$, the tth channel of the output feature map $\mathbf{y}$ is given by

$$\mathbf{y}_{qrt} = b_t + \sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{d=1}^{D} \mathbf{w}_{ijdt} \cdot \mathbf{x}_{s(q-1)+i-p,s(r-1)+j-p,d} \quad (1)$$

assuming equal strides s in both spatial dimensions. The size of the output feature map equals $M' \times N' \times K = \lfloor((M - H + 2p)/s) + 1\rfloor \times \lfloor((N - W + 2p)/s) + 1\rfloor \times K$ (Fig. 2).

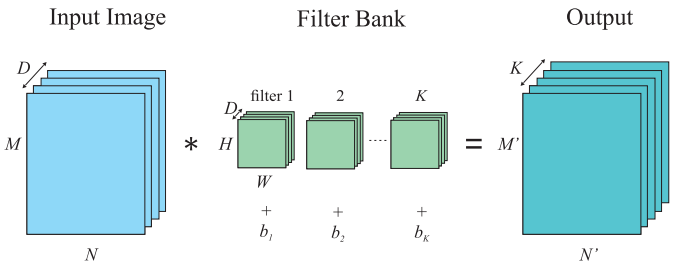

Fig. 2. Convolution of an input image with a filter bank.

In order to capture long-distance dependency, one should adopt kernels of large spatial support $H \times W$. Using large filters, however, increases the number of parameters, making the training more difficult and decreasing the generalization capability of the network. To alleviate this problem, several techniques adopt downsampling in the convolutional layers, which then requires an upsampling strategy [6]–[9]. In this letter, we adopt convolutions with dilated kernels (DKs), or dilated convolutions, instead of downsampling. This allows us an exponential expansion of the receptive field without increasing the number of learnable parameters per layer [11]. DKs are obtained by inserting zeros between filter elements, effectively enlarging the spatial support of the filter without increasing the number of elements. Fig. 3 shows the dilation of a 3×3 filter with increasing dilation factors from one to four. The spatial support of a DK becomes $H' \times W' = d(H - 1) + 1 \times d(W - 1) + 1$, where d is the dilation factor.

B. Proposed Architecture

Our FCN architecture consists of six convolutional layers using DKs and one final classification layer with a 1×1 convolution and a softmax loss function. The structure of this architecture, named FCN-DK6, is reported in Table I. The convolutional layers use 5×5 kernels with increasing dilation factors d from one to six that capture larger and

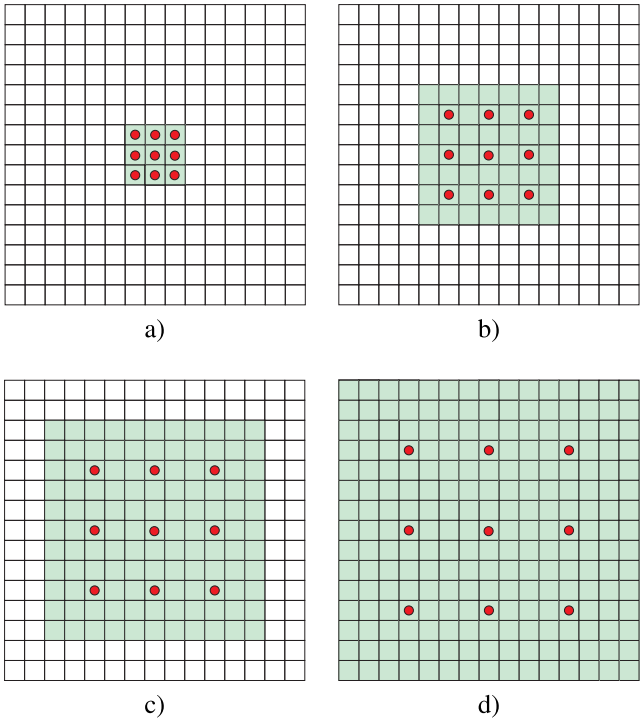

Fig. 3. Kernels with increasing dilation factors. Red circles represent learnable filter weights. (a) 3×3 kernel with no dilation (or dilation factor one). Same filter dilated with a factor (b) two, (c) three, and (d) four. The area marked in green shows the receptive field obtained after consecutive application of filters (a)–(c). The number of parameters is the same in each layer, while the receptive field grows exponentially.

larger contextual information in deeper layers resulting in an exponentially growing receptive field. Convolutions are interleaved by max pooling and nonlinear activations. Max pooling returns the maximum value within a square window and assigns it to the central pixel of the window. The window size is set to be equal to the corresponding DK support. As activations, we adopt leaky rectified linear units (lReLUs) with a leak factor of 0.1 [12]. The stride s is set equal to one for every layer of the network to avoid downsampling. We use zero padding to keep the same spatial dimension of the feature maps in each convolutional layer. Unlike patch-based architectures, the proposed FCN can be directly applied to the classification of images with different sizes, not necessarily matching the size of the input patches used for training it. The classification of large RS images can be performed per blocks of a size that can be tuned according to hardware specifications.

We also consider shallower variants of the proposed architecture of different depths. We refer to the different networks as FCN-DKx, where x indicates the number of convolutional layers included in the network.

III. Experiments and Results

We experimentally evaluated the performance of the proposed FCN for the detection of informal settlements in VHR images and compared the results against state-of-the-art techniques: 1) pixel-based support vector machine (SVM); 2) SVM with texture features extracted from the gray-level co-occurrence matrix (GLCM) [13]; 3) patch-based CNN (PB-CNN); and 4) deconvolutional FCN (FCN-DEC) utilizing a downsample-then-upsample strategy. For SVM, we adopted a radial basis function kernel; its spread and the regularization parameter are set according to hold-out validation. GLCM features are extracted using a 125×125 window and a lag of one pixel in both spatial directions. The patch-based CNN consists of three convolutional layers with 5×5 filters interleaved by stride two max-pooling and ReLU activations and one fully connected layer with softmax classification. The FCN-DEC network consists of a downsampling (encoder) and an upsampling block (decoder) as in [9]. The encoder uses three convolutional layers with 5×5 filters, stride two max-pooling, and lReLU activations with a leak factor of 0.1. The decoder has three deconvolution layers interleaved by lReLU activations. The patch size is 125×125 pixels for all our networks.

TABLE I
Proposed Deep FCN With Six Layers of Dilated Convolutions (FCN-DK6)

Layer	module type	dimension	dilation	stride	pad
DK1	convolution	$5 \times 5 \times 4 \times 16$	1	1	2
	lReLU				
	max-pool	5×5		1	2
DK2	convolution	$5 \times 5 \times 16 \times 32$	2	1	4
	lReLU				
	max-pool	9×9		1	4
DK3	convolution	$5 \times 5 \times 32 \times 32$	3	1	6
	lReLU				
	max-pool	13×13		1	6
DK4	convolution	$5 \times 5 \times 32 \times 32$	4	1	8
	lReLU				
	max-pool	17×17		1	8
DK5	convolution	$5 \times 5 \times 32 \times 32$	5	1	10
	lReLU				
	max-pool	21×21		1	10
DK6	convolution	$5 \times 5 \times 32 \times 32$	6	1	12
	lReLU				
	max-pool	25×25		1	12
class.	convolution	$1 \times 1 \times 32 \times 2$	1	1	0
	softmax				

A. Data Set

As case study, we considered the city of Dar es Salaam, where about 70% of the population is living in informal settlements scattered in different parts of the large urban area. We used a pan-sharpened Quickbird satellite image acquired in 2007. The four multispectral bands have a spatial resolution of 60 cm. Labeled reference information is obtained from the land use reference map [14] and updated by visual interpretation. We addressed the binary classification task of distinguishing "informal settlements" from the class "rest," which mainly includes formal urban settlements and limited areas of vacant and agricultural land. We used five tiles of 2000×2000 pixels taken from different areas of the city, which capture different types of formal and informal settlements in relatively balanced proportions. Three tiles were used for training (named TR1, TR2, and TR3) and two for testing the considered classifiers (TS1 and TS2). The adopted image tiles are shown in Fig. 4.

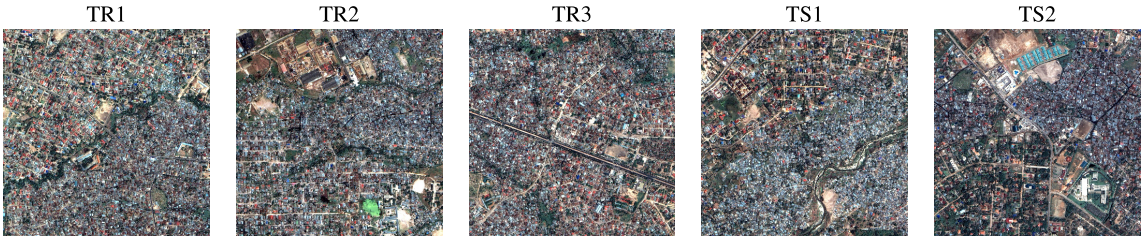

Fig. 4. Tiles obtained from the Quickbird image of Dar es Salaam, Tanzania. TRs 1–3 are used for training and TS1-2 for testing.

B. Training the Networks

The networks are trained using stochastic gradient descend with momentum. We used a training set of 3000 fully labeled patches randomly extracted from the three training tiles (1000 samples from each tile). We used minibatches of 32 samples, a momentum of 0.9, and a weight decay factor of $5 \cdot 10^{-4}$. Batch normalization was applied after every convolution. Dropout with 50% rate was used in the last classification layer. We first trained FCN-DK3 by randomly initializing the weights. Then we trained FCN-DK4 by initializing the weights of the first three layers according to FCN-DK3 and the fourth layer randomly. Then we progressively trained deeper networks including the additional convolutional layers one by one up to FCN-DK6. The networks were trained for 150 epochs with a learning rate of 10^{-4} and then for another 20 epochs with a learning rate of 10^{-5}. This multistage training proved effective: in the first stage, the training error is substantially reduced, but the error on the validation set shows an unstable behavior, whereas in the second stage, the network is tuned with a lower learning rate stabilizing both training and validation errors. We also observed the importance of batch normalization, which considerably speeds up the training. The trained networks were then applied to the classification of the two test tiles. The classification of test tiles is performed using blocks of 200×2000 pixels loaded into GPU memory.

The networks considered in this letter are implemented using the MatConvNet library version 1.0-beta-23 compiled with CUDA-8 toolkit and cuDNN support.[1]

C. Classification Results

The accuracies are computed with the reference maps of the two test tiles. The obtained results are shown in Tables II and III for TS1 and TS2, respectively. Accuracies are reported in terms of overall accuracy (OA), average class accuracy (AA), i.e., mean producer's accuracy (PA), and PA of the two classes [15]. Reference and classification maps are shown in Fig. 5.

We observe that the accuracy of the pixel-based SVM classification is poor on both test tiles, as expected. This confirms that the spectral information alone is not sufficient for discriminating the two abstract land-use classes. SVM classification of TS2 results in 17% higher OA than TS1. This suggests that the two information classes are spectrally more different in TS2 than in TS1. GLCM features improve the separability of the classes, resulting in higher classification accuracies (especially on TS1). The PB-CNN classification reaches significantly higher accuracy by automatically learning contextual features from the data. The maps produced by PC-CNN are nonetheless noisy with many isolated pixels falsely classified as "informal settlements." FCN-DEC produces more regular maps, but the PA of "informal settlements" is low. This is also evident from the classification maps showing many missed detection errors.

The highest accuracies are obtained by the proposed FCN-DK architecture. We observe the benefit of deeper architectures using convolutional kernels with larger dilation factors. They can learn more complex and abstract features from a larger spatial neighborhood leading to higher accuracy. The maps obtained by FCN-DK6 are also more regular than those obtained by other techniques.

The proposed FCN-DK architecture also offers an advantage in terms of computational cost with respect to PB-CNN. One tile is classified in 2.67 s with FCN-DK6 whereas PB-CNN takes 110 min (average times over 10 repeated experiments).

TABLE II
CLASSIFICATION ACCURACIES OBTAINED BY THE CONSIDERED METHODS ON IMAGE TILE TS1

Classifier	OA (%)	AA (%)	PA informal (%)	PA rest (%)
SVM	59.46	59.25	30.41	88.90
SVM+GLCM	67.14	67.31	40.42	94.21
PB-CNN	76.77	76.87	60.33	93.42
FCN-DEC	67.50	67.69	38.69	96.68
FCN-DK3	67.75	67.92	40.81	95.04
FCN-DK4	72.14	72.28	50.84	93.71
FCN-DK5	79.27	79.38	62.41	96.34
FCN-DK6	81.31	81.40	67.89	94.90

TABLE III
CLASSIFICATION ACCURACIES OBTAINED BY THE CONSIDERED METHODS ON IMAGE TILE TS2

Classifier	OA (%)	AA (%)	PA informal (%)	PA rest (%)
SVM	77.01	72.58	56.16	89.01
SVM+GLCM	77.89	72.71	53.47	91.96
PB-CNN	84.39	80.71	67.02	94.40
FCN-DEC	80.88	75.09	53.60	96.58
FCN-DK3	82.74	77.55	58.29	96.82
FCN-DK4	82.62	77.44	58.16	96.71
FCN-DK5	84.70	79.90	62.09	97.71
FCN-DK6	86.09	81.74	65.58	97.91

[1] http://www.vlfeat.org/matconvnet/

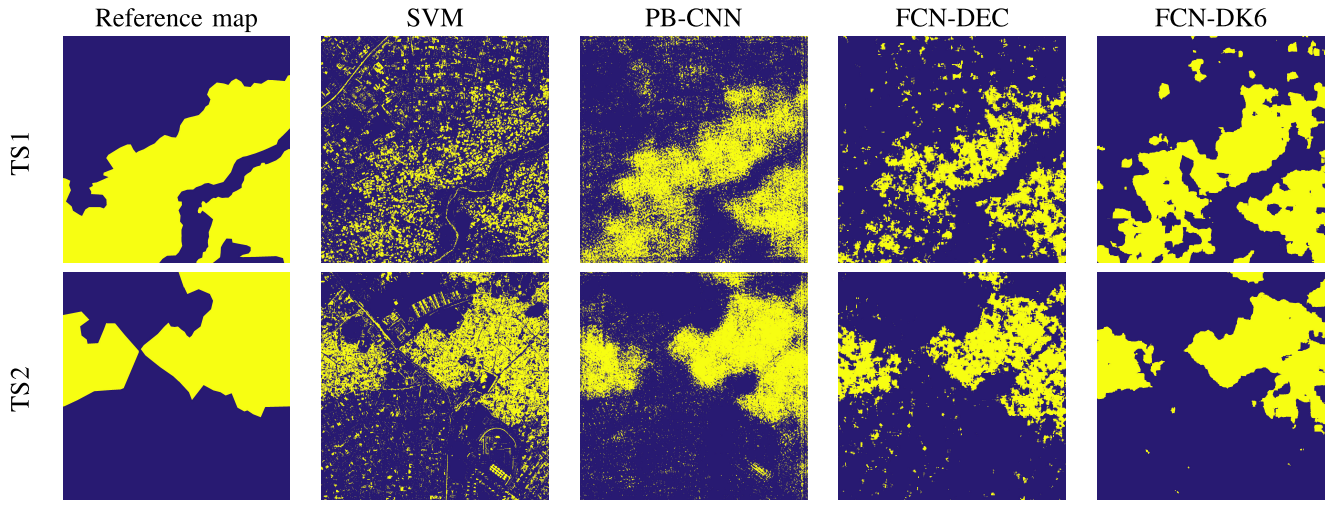

Fig. 5. Reference and classification maps obtained by the investigated techniques. The first row reports the maps of tile TS1 and the second row of TS2. "Informal settlements" are given in yellow and the "rest" in blue.

The training time is about 2500 s for both architectures. All experiments were performed on a desktop workstation with an Intel Xeon CPU E5-1650 v4 at 3.6 GHz, 40 GB of RAM, and an Nvidia Titan X (Pascal) GPU.

IV. Conclusion

The novel deep network proposed in this letter is capable of learning a hierarchy of abstract appearance features capturing long-range pixel dependencies while keeping a limited number of network parameters. Experimental results carried out on a Quickbird image acquired over Dar es Salaam show the effectiveness of the proposed technique in accurately distinguishing informal settlements from other land-use classes. The proposed architecture using six convolutional layers performs better than state-of-the-art architectures such as PB-CNNs and deconvolutional networks. Moreover, the fully convolutional architecture allows us to significantly reduce the processing time of patch-based networks at inference time.

Compared with deconvolutional networks using downsampling and upsampling layers [6]–[9], we would argue that the proposed architecture offers a more simple and flexible design approach. There is no need to match feature map sizes in the downsampling and upsampling parts to produce maps of the required dimension. With the proposed structure, new layers can be added to increase the receptive field without the need for redesigning the rest of the network. The depth can be adjusted depending on the spatial resolution of the RS image and the type of information classes to be discriminated. Therefore, variants of our network can be designed to address different land-cover or land-use classification applications.

This letter shows that informal settlements can be effectively detected in VHR images by the proposed automatic technique. This can facilitate a systematic monitoring of slums and provide useful information for international pro-poor policy development and for planning urban upgrading projects. Future developments will aim at studying the transferability of the model to different cities.

Acknowledgment

The authors would like to thank Dr. M. Kuffer and Dr. R. Sliuzas for kindly providing us the Quickbird data and for the useful discussions.

References

[1] UN-Habitat, "Informal settlements," UN-Habitat, New York, NY, USA, May 2015.
[2] UN-Habitat, "Urbanization and development: Emerging futures," UN-Habitat, New York, NY, USA, 2016.
[3] M. Kuffer, K. Pfeffer, and R. Sliuzas, "Slums from space—15 years of slum mapping using remote sensing," *Remote Sens.*, vol. 8, no. 6, p. 455, 2016.
[4] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[5] J. R. Bergado, C. Persello, and C. Gevaert, "A deep learning approach to the classification of sub-decimetre resolution aerial images," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 1516–1519.
[6] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[7] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. ICCV*, 2016, pp. 1520–1528.
[8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[9] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2017.
[10] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.
[11] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–13.
[12] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–6.
[13] M. Kuffer, K. Pfeffer, R. Sliuzas, and I. Baud, "Extraction of slum areas from VHR imagery using GLCM variance," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 5, pp. 1830–1840, May 2016.
[14] R. Sliuzas, A. Hill, C. Lindner, and S. Greiving, "Dar es Salaam Land use and informal settlement data set," NASA Socioecon. Data Appl. Center, Columbia Univ., New York, NY, USA, Tech. Rep., 2016.
[15] R. G. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data*. Boca Raton, FL, USA: CRC Press, 1999.