ISPRS Journal of Photogrammetry and Remote Sensing 142 (2018) 106–123

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

A deep learning approach to DTM extraction from imagery using rule-based training labels

C.M. Gevaert*, C. Persello, F. Nex, G. Vosselman

Dept. of Earth Observation Science, ITC, University of Twente, Enschede, The Netherlands

ARTICLE INFO

Keywords:
Digital Terrain Models (DTM)
Unmanned Aerial Vehicles (UAV)
Aerial photogrammetry
Deep learning
Fully Convolutional Networks (FCN)

ABSTRACT

Existing algorithms for Digital Terrain Model (DTM) extraction still face difficulties due to data outliers and geometric ambiguities in the scene such as contiguous off-ground areas or sloped environments. We postulate that in such challenging cases, the radiometric information contained in aerial imagery may be leveraged to distinguish between ground and off-ground objects. We propose a method for DTM extraction from imagery which first applies morphological filters to the Digital Surface Model to obtain candidate ground and off-ground training samples. These samples are used to train a Fully Convolutional Network (FCN) in the second step, which can then be used to identify ground samples for the entire dataset. The proposed method harnesses the power of state-of-the-art deep learning methods, while showing how they can be adapted to the application of DTM extraction by (i) automatically selecting and labelling dataset-specific samples which can be used to train the network, and (ii) adapting the network architecture to consider a larger surface area without unnecessarily increasing the computational burden. The method is successfully tested on four datasets, indicating that the automatic labelling strategy can achieve an accuracy which is comparable to the use of manually labelled training samples. Furthermore, we demonstrate that the proposed method outperforms two reference DTM extraction algorithms in challenging areas.

1. Introduction

Airborne Laser Scanning (ALS), satellite imagery, and aerial or UAV imagery can provide a *Digital Surface Model* (DSM) which describes the elevation of the Earth's surface. This model describes the elevation of the top of objects, i.e. the elevation of the ground plus the height of objects such as buildings and vegetation which is on top of the surface. However, many applications actually require a model where these elevated objects are removed, i.e. a *Digital Terrain Model* (DTM), as depicted in Fig. 1. The difference between the DSM and DTM is referred to as a normalized Digital Surface Model (nDSM), and gives the height of the elevated objects. The conversion of a DSM to a DTM is known in literature as DTM extraction, bare-ground extraction, or point cloud filtering. This process generally consists of two phases: first selecting pixels or points which represent the ground and then using these points to interpolate a surface model of the terrain.

Most DTM extraction algorithms have been tested on relatively easy datasets (Tomljenovic et al., 2015). However, we can identify a number of specific scenarios which present difficulties for DTM extraction from point clouds of urban areas (Fig. 2). A number of difficulties arise due to errors inherent in the data itself. For example shadows cause difficulties for dense matching algorithms, resulting in noise in the point cloud (Fig. 2a). Also, lack of texture or unsatisfactory camera calibration may cause noise or outliers in the point cloud (Fig. 2b). The DSM interpolation step may also cause errors, such as increasing the extent of elevated objects when using Inverse Distance Weighting (Fig. 2c) or artefacts along overhanging objects when using Delaunay triangulation (Fig. 2d). Other sources of difficulties for DTM extraction algorithms are due to the characteristics of the scene itself. For example, sloped surfaces may cause a step-like pattern where ground and off-ground cannot be distinguished (Fig. 2e) or off-ground objects to be co-planar with the ground (Fig. 2f). Finally, elevated objects which are significantly larger than the other objects in the scene (Fig. 2g) or agglomerations of neighbouring objects (Fig. 2h) form contiguous off-ground areas which affects the size of the local neighbourhood which must be considered to identify ground points as most algorithms somehow assume that ground points will locally be the lowest point. In our approach, we demonstrate how complementary information from the imagery can be included to successfully extract a DTM in these challenging areas.

Existing algorithms for DTM extraction from DSMs or point clouds can be roughly divided into five groups: (i) morphological filtering, (ii) progressive densification, (iii) surface-based, (iv) segment-based, and

* Corresponding author.
E-mail addresses: c.m.gevaert@utwente.nl (C.M. Gevaert), c.persello@utwente.nl (C. Persello), f.nex@utwente.nl (F. Nex), george.vosselman@utwente.nl (G. Vosselman).

https://doi.org/10.1016/j.isprsjprs.2018.06.001
Received 31 August 2017; Received in revised form 30 May 2018; Accepted 1 June 2018
Available online 15 June 2018

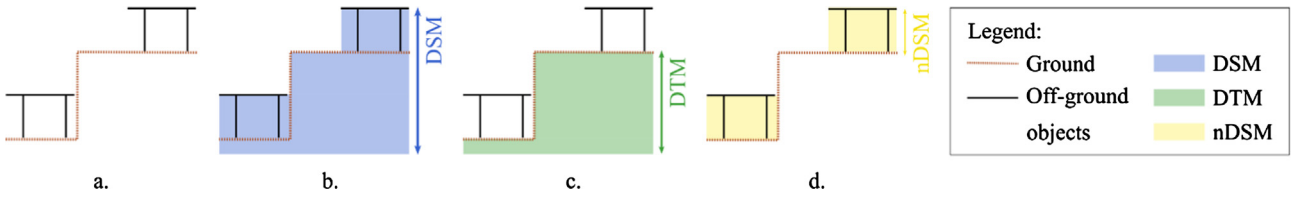

Fig. 1. Given a scene with the ground and objects such as buildings (a), the Digital Surface Model (DSM) provides the height of the ground plus any objects on top of it (b), the Digital Terrain Model (DTM) filters off-ground objects and therefore provides the elevation of only the ground surface (c), and the normalized Digital Surface Model (nDSM) represents the difference between the DSM and DTM, essentially giving the height of the objects on top of the terrain (d).

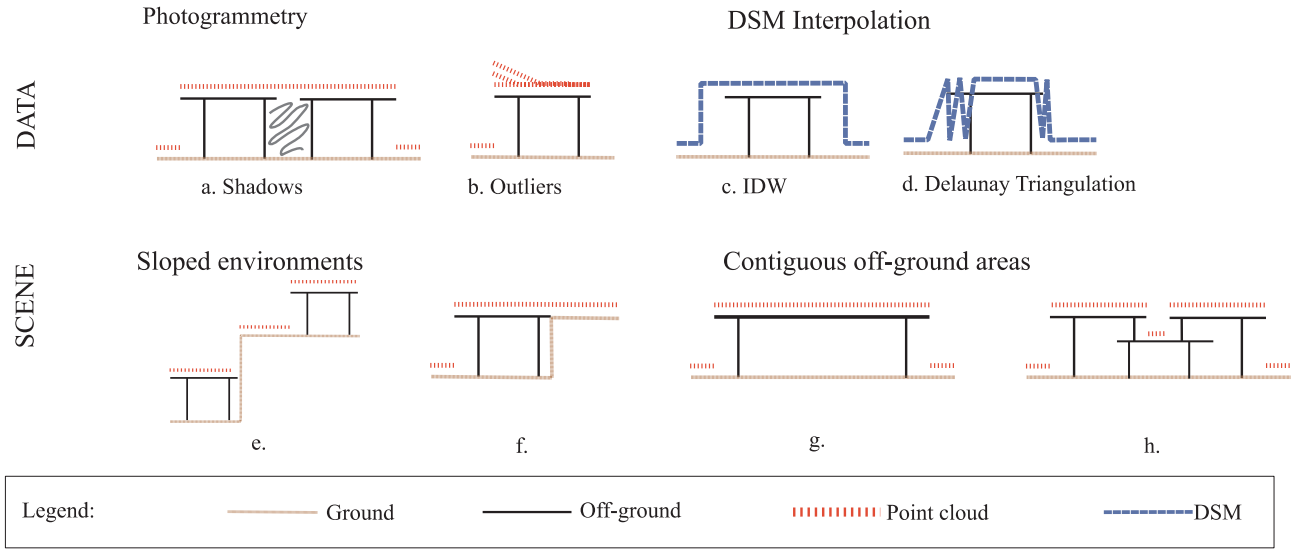

Fig. 2. An overview of sources of errors in DTM extraction algorithms. The data itself has errors, such as shadows (a) and outliers (b) which are by-products of the photogrammetric workflow. Also, DSM interpolation methods such as Inverse Distance Weighting (IDW) (c) and Delaunay Triangulation (d) create artefacts in the DSM. Scene characteristics such as sloped environments (e and f) and contiguous off-ground areas due to exceptionally large buildings (g) or connected buildings (h) also cause difficulties.

(v) deep learning methods. In *morphological filtering*, the ground is defined as the lowest point within a specified neighbourhood. Variations of this method include: making the threshold dependent on the distance to the centre point (Sithole and Vosselman, 2005) or adapting the filter to the slope calculated from an existing DTM (Debella-Gilo, 2016; Sithole and Vosselman, 2005). Morphological methods are very sensitive to the size of the search neighbourhood. For example, if the element is too small, it may cause elevated objects slightly lower than the surrounding objects to be mistakenly labelled as ground (e.g. Fig. 2h). To avoid this, some approaches use neighbourhoods of various sizes. For example, Kilian et al. (1996) use structuring elements of various sizes and then link the likelihood of a point being considered ground with the size of the structuring element for which the point is labelled as ground. Similarly, Mongus et al. (2014) use morphological profiles of various sizes, and record: the largest response, the size of the structuring element at the first response, and the cumulative sum of responses up to the largest response as three features. These are said to reflect the height of features compared to the direct surrounding, planimetric size of the elevated object, and estimation of the height of the object.

After the ground points are obtained through the filtering, an interpolation can be performed to obtain the DTM surface. For example, a Triangulated Irregular Network (TIN) represents the surface through a series of triangles where the vertices are the ground points. This can be done through a Delaunay triangulation, which constructs a TIN in such a way that no points are within the circumcircle of one of the surface triangles, and the vertices of all triangles are maximized (Lawson, 1972). This is a common method, and a wide range of adaptations have been developed to optimize it (Tsai, 1993). Another interpolation method is Inverse Distance Weighting (IDW), where all points within a given neighbourhood are utilized as input for the surface, but nearer points are given more weight on the surface estimation than further points (Hohn, 1991). The performance of the interpolation algorithms depends on e.g. surface characteristics and dataset density (Chaplot et al., 2006). However, in the current study we focus on the correct identification of ground points, and a further comparison of interpolation methods is not considered.

Progressive densification methods select a number of 'seed' points which are likely to represent the ground, and then successively add points to those classified as ground. For example, Axelsson (2000) used a grid to select the lowest points which are then used to construct an initial TIN model. This TIN is progressively densified by adding points which are less than a user-defined distance from an existing TIN face, and form an angle less than a user-defined threshold with the three vertices of this face. With a total error of 11.2% algorithm had comparatively good results on the ISPRS benchmark set (Sithole and Vosselman, 2004); though it is said to have difficulties in identifying cliffs and sharp ridges (e.g. Mongus et al., 2014; Zhang et al., 2016b).

Surface-based or *interpolation* methods estimate a surface from all the input points and suppress the influence of off-ground points on the interpolation. For example, at the first iteration, a surface can be interpolated using all available points. One can then assume that points on the ground are likely to be below the interpolated surface. These lower points are then assigned a higher weight in the interpolation for the next iteration(Kraus and Pfeifer, 1998). Alternatively, an active shape method can be applied, which describes the surface as a rubber cloth and forms it to the laser points in a bottom up fashion (Elmqvist et al., 2001). The surface is adjusted iteratively using an energy

function which weighs the 'stiffness' of the interpolated surface (internal force) against the individual point observations (external force). Zhang et al. (2016b) propose a similar approach based on 'cloth simulation filtering' to identify potential ground points. Surface-based methods experience difficulties in areas with steep slopes (Liu, 2008), which may require explicit post-processing (e.g. Zhang et al., 2016b).

Segment-based methods generally consist of 3 steps: (1) the segmentation of a point cloud or DSM, (2) the classification of the segments as ground or non-ground, and (3) the interpolation of the DTM from ground segments (Beumier and Idrissa, 2016). Point cloud segmentation may use profiles (Sithole and Vosselman, 2005) or region-growing techniques. For the latter, local minima are often used to obtain seed points, which are densified through e.g. planar segmentation (Pérez-Garcia et al., 2012), or similarity of normal vectors (Tóvári and Pfeifer, 2005). For 2D raster methods, slope is often used to define segmentation boundaries. For example, Hingee et al. (2016) calculate the slope of the DSM, which is used to segment the raster. Segments where majority of pixels are flowing 'in' are candidates for ground, then surface fitting is applied. Tomljenovic et al. (2016) also uses slope to delimit segments, the largest segment is considered to be the ground. Note that Mongus et al. (2014) mention that slope-based filtering doesn't work well in sloped study areas. Yan et al. (2012) use a locally lowest points to initiate the region growing segmentation, where slope is used to determine whether pixels are included in the segment or not. They define a segment as terrain or non-terrain based on the signed height differences between neighbouring segments. Beumier and Idrissa, (2016) use a maximum height difference and two-step connected component algorithm to define the segments, and additionally define a minimum region size parameter. Segment size and its relative elevation to the neighbouring segments are used to identify ground segments. Segment-based methods may speed up processing compared to pixel- or point-based methods and reduce sensitivity to noisy data, though the quality of the results is heavily dependent on the quality of the segmentation to begin with.

Finally, *deep learning* algorithms have recently been improving accuracies on a wide range of supervised classification tasks (e.g. He et al., 2016; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014). In computer vision applications, Convolutional Neural Networks (CNNs) were used to give a single semantic label to an entire image patch. CNNs consist of a combination of: convolutional layers which apply a series of filters to the input image, nonlinear activation layers which allow complex representations to be learned, and pooling components to help prevent overfitting. CNNs have been very successful in classification tasks which require assigning a label to an image patch or scene (He et al., 2016; Krizhevsky et al., 2012). More recently, Fully Convolutional Networks (FCNs) have been developed for tasks which require assigning a label to each pixel within an image, i.e. semantic segmentation (Shelhamer et al., 2017). They are more efficient for semantic segmentation than conventional CNNs as they avoid redundant calculations, improve memory efficiency and incorporate more training data into the optimization of the weights. Furthermore, a significant benefit of FCNs is that, unlike patch-based CNNs, they can be easily applied to images with different dimensions.

Due to the convincing results achieved on computer vision benchmarks, CNNs (Hu et al., 2015; Mboga et al., 2017; Romero et al., 2016; Zhang et al., 2016a) and FCNs (Persello and Stein, 2017; Sherrah, 2016) are also increasingly being applied to classify satellite and aerial imagery in remote sensing applications. For example, some studies utilize networks trained on computer vision datasets (Audebert et al., 2017) or synthetic multispectral imagery (Kemker and Kanan, 2017), and fine-tune the weights using real aerial imagery. Deep learning has also been applied for DTM extraction from point clouds, where Hu and Yuan (2016) recently achieved state-of-the-art results on the ISPRS benchmark dataset using a CNN. The authors first convert the point cloud into a 2D grid consisting of three attributes: the minimum, maximum and mean height per grid cell. More than 17 million pre-labelled training samples were then used to train a CNN capable of distinguishing ground vs. non-ground points. The method obtained accurate results, but required a large amount of labelled data.

We foresee that there are three main concerns which must be overcome in order to efficiently exploit the power of deep learning for DTM extraction. Firstly, the collection of a sufficient amount labelled training data for training the networks is costly and time-consuming. In some cases, such labelled data may be available due to extensive manual labour, but here we consider cases where such labelled data is not available. Secondly, previous DTM extraction algorithms indicate that it is important to consider elevation differences over a local neighbourhood which exceeds the size of the largest off-ground object in the scene. However in the case of DTM extraction from extremely high resolution UAV data products, covering such an extensive area would require very large image patches as input for a FCN. The challenge is therefore how to consider the information over a large area while limiting the number of network parameters which must be tuned as well as the size of the input patch used to train the network. Thirdly, even if a network is correctly tuned, there are still cases in which using only the elevation information is not enough to distinguish ground from off-ground samples (e.g. Fig. 2e,f).

In the field of large-scale urban scene reconstruction, researchers have shown how incorporating both 3D and 2D information is beneficial. For example, to jointly perform image segmentation and dense stereo reconstruction. This can be done by jointly optimizing the random field formulations of both problems (Ladický et al., 2012). At an object level, learning the mean shape of an object from 3D scans can be combined with image-based cues of anchor points to improve the accuracy of multiview stereo workflows (Bao et al., 2013). Probabilistic models (Ulusoy et al., 2017) or 3D deep learning strategies (Riegler et al., 2017) can learn object shapes to support 3D reconstruction in occluded or texture-less areas. Classification problems on a larger scale also benefit from the integration of 2D and 3D information. For example, a voxel's preference for a certain semantic class (image-based cues) can be supplemented by the likelihood of certain surface orientations (3D geometric cues) (Hane et al., 2013). Smart strategies using hierarchical voxel schemes can be used to maintain a high classification accuracy while reducing the memory and computational times enormously (Blaha et al., 2016). Other workflows combine an even wider range of data sources: from OpenStreetMap, LiDAR, aerial photography and semantic data for large-scale scene reconstruction (Cabezas et al., 2015). Although scientific research displays great potential for this field of large-scale 3D scene reconstruction and semantic interpretation, we acknowledge that a number of applications simply require an input DTM. The purpose of this manuscript is therefore to exploit these observations of synergies between information contained in the visual and geometric information of a scene, but applied to a more simple task of DTM extraction with more conservative computational and data requirements.

More specifically, this paper proposes the use of deep learning in the form of a FCN for DTM extraction. DSMs derived from photogrammetric point clouds and the corresponding true-orthophotos are used as input. The utilization of both sources of information is one of the main points of our approach, and is key to DTM extraction in challenging areas. Our method uses a simple rule-based mechanism to automatically identify ground and off-ground samples which are then used to train the network, thus eliminating the need to collect large sets of manually labelled training data. Secondly, the network takes a large surface area into account by considering topographic features derived from DSM (which summarize the height of a pixel to the local topographical tendencies) and by applying dilated filters in the network architecture. Finally, difficult scenarios which may confuse existing DTM methods are solved by exploiting the RGB information obtained from the UAV imagery in conjunction with the DSM. In the following manuscript we describe the proposed method for DTM extraction and demonstrate its accuracy using three challenging datasets. The use of the three different

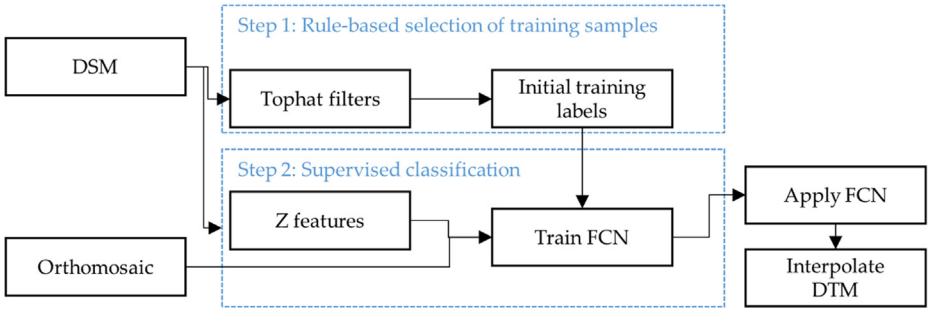

Fig. 3. Workflow of the proposed methodology. The first step consists of applying top-hat filters to the DSM to select and label initial training samples. The second step combines the RGB channels of the orthomosaic with features derived from the DSM together with the labelled samples from the first step to train a FCN. This FCN is then applied to the entire dataset to identify the ground samples, which can then be used to create a DTM through interpolation.

datasets attests to the versatility of the method for VHR aerial imagery due to the dataset characteristics (i.e. two were acquired with a UAV and one through aerial imagery and all three have different spatial resolutions) and scene characteristics.

The proposed methodology is assessed by casting it as a binary classification problem (i.e. ground vs. off-ground). We illustrate the importance of combining image-based and DSM-based features by performing sets of FCN experiments using differing input features. Furthermore, we perform experiments using the ground-truth labels vs. the rule-based training labels to support the claim that simple morphological rules are a viable alternative to manually labelling the large number of training samples required to train deep networks for DTM extraction. Further experiments compare the proposed network architecture to deeper networks, apply the algorithm to the ISPRS benchmark data, and consider the possibility of direct regression-based DTM prediction.

2. Proposed method

The proposed methodology consists of two steps (Fig. 3). The first step is a rule-based selection of a set of candidate ground (S_g) and off-ground (S_o) training samples. The idea is that if a supervised classification will follow the initial rule-based method, it is not necessary to label *all* of the pixels as ground or off-ground. Rather, it suffices to have a large number of confident samples. In this case, simple morphological filters are applied to the DSM to select the training samples, as the filters can be executed quickly and the algorithm parameters are intuitive to the user (i.e. neighbourhood search window clearly corresponds to the expected size of the object in the scene).

The second step consists of a supervised classification combining radiometric features from the imagery with geometric features from the DSM to refine the initial labelling. Image-based classification in urban settings can be challenging due to high within-class variability (e.g. different roof materials and colours as well as the presence of clutter on roofs) and low between-class variability (e.g. especially when ground and roof pixels appear similar in true colour). Contextual features such as texture can improve the separability of these two classes (Gevaert et al., 2017), but hand-crafting informative texture features can be challenging. Therefore, we use a FCN as a classifier. In addition to the recent success of deep learning methods for various image classification tasks, their ability to learn powerful contextual features from the data itself supports the development of automatic workflows. Once the pixels corresponding to ground samples have correctly been identified, the final DTM can be interpolated. In the following sections, we describe both steps in more detail.

2.1. Rule-based training sample selection using morphological filters

Let us define $\gamma_{w_s}(DSM)$ as a morphological top-hat filter on the DSM. This filter returns the height of the central point above the lowest point in a disk-shaped neighborhood w_s with a radius of s. However, rather than utilizing multiple scales (Mongus et al., 2014), we utilize only two neighbourhoods: w_{small} and w_{big}. The first filter, w_{small}, is used to identify off-ground objects. The idea is that pixels which are higher than their direct neighbors provides a set of confident off-ground samples (i.e. the filter indicates that the pixel is higher than neighbours within a small neighbourhood), and that these selected samples will be representative of the image-based characteristics of off-ground objects within the dataset. Problems with contiguous elevated objects (e.g. Fig. 2g and h) are addressed as we assume that pixels along the edges of elevated objects (such as roofs in the figure) will have a similar appearance in the image as pixels in the interior of these roofs.

The set of off-ground training samples is defined as:

$$S_o = \{\gamma_{w_{small}}(DSM) > \tau_o\}, \quad (1)$$

where τ_o represents the threshold in meters which defines the minimum height difference between a DSM pixel and its neighbours to be considered as off-ground. Later experiments on the ISPRS benchmark indicated that datasets containing large buildings with flat roofs, unique roofing material and located on flat terrain (e.g. industrial areas) are not always captured by the rule in (1). These areas can therefore benefit from an additional criterion to select off-ground samples: $(DSM - \zeta_{w_{small}}(DSM)) > \tau_o$, where ζ is a morphological erosion filter.

Similarly, we can consider that ground pixels have a minimal response to $\gamma_{w_{big}}(DSM)$. That is to say, a ground point is likely to be lower than pixels within a larger neighbourhood. As with other morphological methods, this search range should be large enough to extend over large objects in the scene, yet not too large as this will be problematic in sloped areas. The set of ground training samples S_g is then:

$$S_g = \{\gamma_{w_{big}}(DSM) < \tau_g\}. \quad (2)$$

In practice, we set $\tau_g = 0.5 \cdot \tau_o$ which reduces the number of parameters to be tuned by the user. Thus, ground and off-ground samples are selected and labelled automatically for each dataset through two simple rules which require the user to tune only three intuitive parameters: w_{small}, w_{big}, and τ_o.

2.2. Fully convolutional neural networks

The selected training samples are used to train a FCN. Detailed descriptions are available regarding the applications of CNN (Castelluccio et al., 2015; Hu et al., 2015), and FCNs (Persello and Stein, 2017; Sherrah, 2016) for image classification tasks in remote sensing. When applying a FCN to DTM extraction applications, especially when utilizing data with an extremely high spatial resolution such as those acquired with UAVs, one of the main concerns is how to consider a large spatial extent without increasing the computational costs of the network. Considering contextual information over a large spatial extent is important for DTM extraction algorithms. For example, the search neighbourhood in morphological filtering methods should be larger than the largest off-ground object. Similarly, when using a FCN for DTM extraction, the receptive field of final layer should be large enough to capture relative elevation differences between off-ground pixels and the surrounding ground pixels. We do this in two ways: by adapting the network architecture and through the use of specialized feature inputs.

Both CNNs and FCNs can be defined as a sequence of layers which generally consist of convolutional, nonlinear activation, and pooling components. Using the same notation as (Volpi and Tuia, 2017), the convolutional layers consist of a set of K' filters with a size of $M \times M \times K$, where M is the width and height of the square filter and K corresponds to the number of input channels of the previous layer. For example, for an RGB image this K would have a value of three. Each filter is convolved over the input layer $\boldsymbol{x}$, producing a response $\boldsymbol{x}'_{ijk'}$ for the k'^{th} filter at row i and column j of the output layer $\boldsymbol{x}'$ as follows:

$$\boldsymbol{x}'_{ijk'} = \sum_{k=1}^{K} \sum_{q=1}^{M} \sum_{p=1}^{M} \boldsymbol{w}_{pqk} \bullet \boldsymbol{x}_{pqk} + b, \tag{3}$$

where $\boldsymbol{w}_{pqk}$ is the filter value of row p, column q, and channel k of the input layer and b are the bias parameters which are learned by training the network. One of the main advantages of the convolutional layers is that, once optimized in the training stage, the weights of the filter are fixed as it passes over the image in the testing stage. This not only decreases the number of parameters to be learned, but also introduces translation invariance. The dimensions of $\boldsymbol{x}'$ depend on the stride (s) and padding (z). The stride is the interval for which each convolution is calculated. A stride equal to one indicates that the convolution is calculated for each pixel of $\boldsymbol{x}$ whereas values higher than one indicate that pixels are skipped and $\boldsymbol{x}'$ will therefore be downsampled. The padding indicates the number of zeros added to the border of the input image to enable pixels along the edges of $\boldsymbol{x}$ to be processed. The receptive field of a filter refers to the area of the original input image which affects the filter response. This can be increased by applying multiple convolutional layers, increasing the size of the filters, or increasing the stride. Another way to increase the receptive field without increasing the number of variables to be tuned is by inserting a defined number (d) of 0 s between weights of $\boldsymbol{w}$. This technique is known as the atrous method (Chen et al., 2015) or dilation (Yu and Koltun, 2016). For an input layer $\boldsymbol{x}$ with dimensions $W \times H \times K$, the dimensions of $\boldsymbol{x}'$ will then be: $\left(\frac{W+2z-M(d-1)-1}{s}\right) \times \left(\frac{H+2z-M(d-1)-1)}{s}\right) \times K'$.

Convolutions are generally followed by a nonlinear activation, which introduces non-linearity into the system thus allowing more complex representations to be learned. One of the most common methods currently used is the Rectified Linear Unit (ReLU), defined as $\boldsymbol{x}'_{k'} = \max(0, \boldsymbol{x}_{k'})$ (Nair and Hinton, 2010). This function is capable of efficient network training and it avoids the vanishing gradient problem (He et al., 2015).

The third main component of FCNs are the pooling layers. The purpose of pooling layers is to summarize the filter responses and improve translation invariance (Krizhevsky et al., 2012). A common strategy is max-pooling, which returns the highest response over a small window (generally 2×2 or 3×3). Pooling layers commonly utilize a stride set equal to the pooling size window. This returns a single value for each (e.g. 2×2) window and thus downsamples the image. In semantic segmentation applications, the final output layer should have the same dimensions as the input layer. Therefore, the network may make use of deconvolutional layers which again upsample the features at a later stage in the network (Shelhamer et al., 2017; Volpi and Tuia, 2017). Alternatively, it is possible avoid downsampling in the pooling layer by avoiding pooling layers altogether or by using pooling layers with a stride equal to one (Sherrah, 2016). Results of the ISPRS 2D semantic labelling contest[1] suggest that the latter strategy is competitive with more complex deconvolutional strategies (Volpi and Tuia, 2017).

2.3. Proposed network

In our proposed network, we therefore utilize a FCN with no downsampling to ensure that the output ground prediction map will automatically have identical dimensions as the input dataset. The network consists of three convolutional layers (Table 1). The first two are followed by ReLUs, and a max-pooling with no downsampling. As there is no downsampling, no deconvolutional layers are needed to ensure the output map has the same dimensions as the input map. This strategy has been previously used by FCN architectures for the classification of satellite imagery (Persello and Stein, 2017; Sherrah, 2016). The receptive field is increased greatly in the second convolutional layer through the use of dilated filters. The use of dilated filters also introduces a multi-scale effect, which is typically achieved through downsampling. However, the use of dilated filters as opposed to downsampling has the additional benefit that fewer parameters are required, thus speeding up training and reducing potential overfitting (Yu and Koltun, 2016). The architecture used here also reduces overfitting by introducing a batch normalization (Ioffe and Szegedy, 2015) after each convolution and dropout (Srivastava et al., 2014) after the final convolution.

Table 1
An overview of the FCN network architecture utilized for the DTM extraction.

Layer	Filter size M pixels)	Filter dilation d (pixels)	Number of filters K'	Padding z (pixels)	Receptive field size (pixels)
Convolutional1	5×5	1	16	2	5×5
Batch normalization					
ReLU					
Pooling1	3×3	–	–	1	7×7
Convolutional2	9×9	6	16	24	55×55
Batch normalization					
ReLU					
Pooling2	3×3	–	–	1	57×57
Convolutional3	1×1	1	2	0	57×57
Batch normalization					
Dropout					

The second manner to increase the extent under consideration by the FCN is by incorporating DSM feature which describe the topography over a large area as input channels for the network. We choose to include these DSM features as they allow the network to consider topographical variations over an extended area without increasing the computational costs of training the network. I.e., while the utilization of dilated kernels may reduce the number of parameters to be tuned, it still requires a larger input patch for training the network and thereby increases the memory requirements. Remember that existing DTM extraction methods indicate that the relative height must be considered over an area larger than the largest elevated object (i.e. Fig. 2g,h). If a dataset has a spatial resolution of 3 cm, then it requires a receptive field of for example 667×667 pixels in order to take a 20×20 m area into account, which is considerably larger than the patch size of the current network. Therefore, we include features which describe the surface topography over a larger area as input for the FCN. These features are inspired by DTM extraction methods which take an existing DTM into account (Debella-Gilo, 2016; Sithole and Vosselman, 2005) and the surface-based or interpolation methods (Kraus and Pfeifer, 1998). Namely, we define a grid over the DSM and select the point with the elevation which corresponds to the lowest 10% of the pixels in the cell. We avoid selecting the lowest point per cell as photogrammetric point clouds may contain many outliers which could negatively affect the interpolation (Nex and Gerke, 2014). A bicubic interpolation is then applied to these lowest points. In addition to the original DSM, we utilize two grids: one of 1×1 m to preserve local topographical details and the other of 20×20 m to describe the general surface topography. The combination of these three features representing the absolute

[1] http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html.

Table 2
Description of the different feature sets used to train the FCN.

Feature set name	Data source(s)	Number of channels	Description
RGB	Image	3	Red, green, blue colour channels
Z	DSM	3	DSM Local topography: interpolation of lowest elevation decile every 1 m General topography: interpolation of lowest elevation decile every 20 m
nZ	DSM	2	DSM – Local topography DSM – General topography
DTM	DSM	1	An approximated DTM formed by interpolating a surface from all pixels labelled as ground
nDSM	DSM	1	An approximated normalized DSM formed by subtracting the DTM above from the input DSM

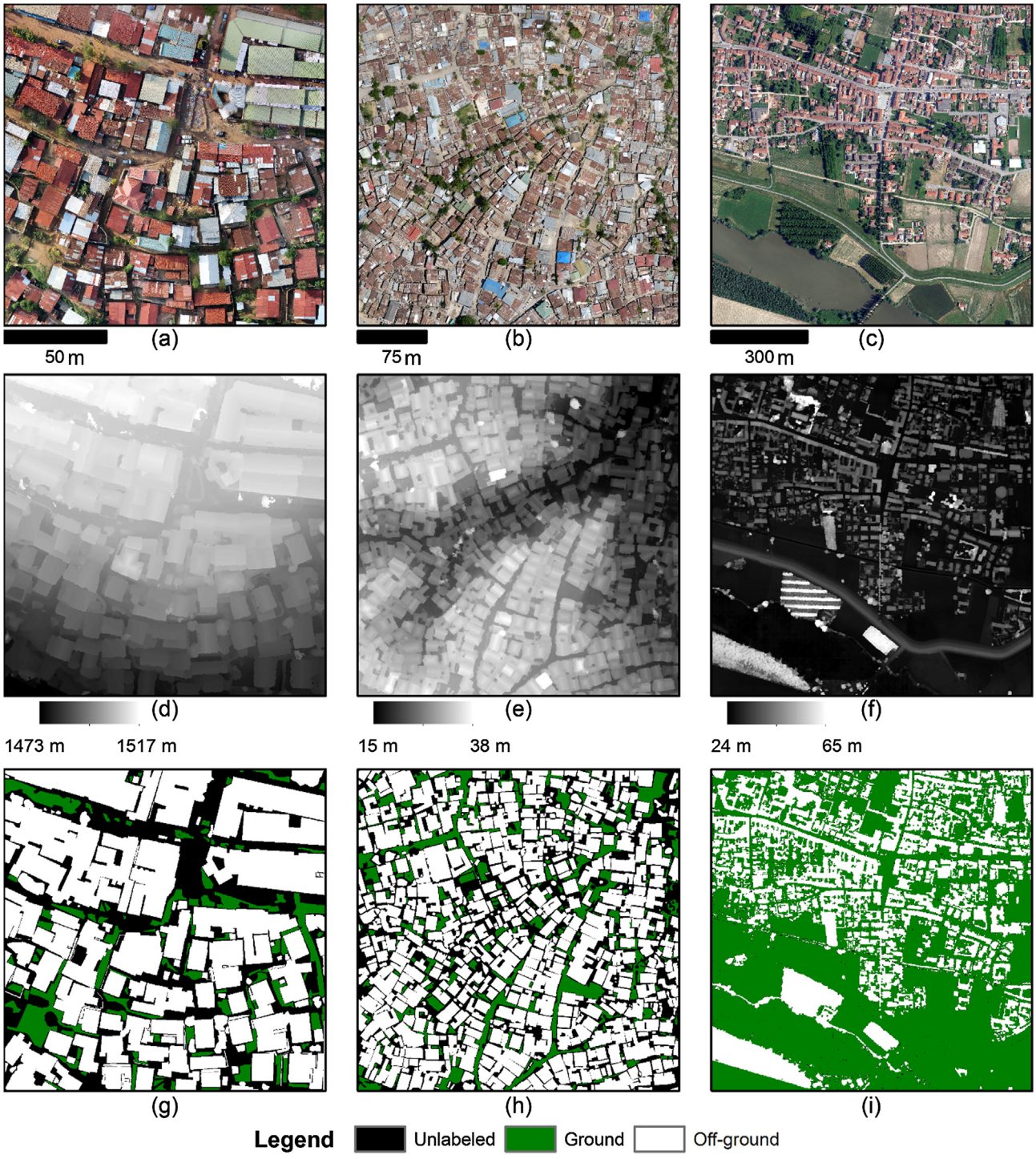

Fig. 4. Images of the Kigali (a), Dar es Salaam (b), and Lombardia (c) datasets, and their respective DSMs (d–f) and manual reference data (g–i).

height forms one feature set (Z). Another feature set simulates the height of objects above these surfaces and consists of the difference between the DSM and the two interpolated height features (nZ). An overview of these feature sets is given in Table 2.

3. Experimental analysis

3.1. Data sets

3.1.1. Kigali, Rwanda

The first dataset consists of UAV imagery collected over an informal settlement in Kigali, Rwanda (Fig. 4a,d). Images were collected with a DJI Phantom 2 Vision+ quadcopter and processed with Pix4Dmapper to obtain a DSM and true-colour orthomosaic with a spatial resolution of 3 cm. A subset of 5000 × 5000 pixels (150 × 150 m) was selected which contains densely grouped buildings separated by narrow footpaths which are often shadowed. The terrain of the lower part of the image contains steep slopes, making it a challenging scene for DTM extraction algorithms. More information regarding the UAV data collection and processing can be found in Gevaert et al., (2017). The reference data (Fig. 4g) was manually created by visual interpretation.

3.1.2. Dar es Salaam, Tanzania

The second dataset consists of UAV imagery over Dar es Salaam, Tanzania (Fig. 4b,e). The images were collected in 2015 with a SenseFly eBee mounted with a 14 MP Canon Powershot RGB camera in the context of a World Bank project (Dar Ramani Huria[2]). These images were processed with Pix4Dmapper to obtain a DSM and true-colour orthomosaic with a spatial resolution of 5 cm. A subset of 6000 × 6000 pixels (300 × 300 m) was selected for the current analysis. The area again covers an informal settlement. Although the area is not as steeply sloped as in Kigali, the area also challenging due to the presence of contiguous off-ground areas and spectral similarity between the ground and off-ground objects. Reference data for the ground and off-ground object classes was again manually digitized over the orthomosaic (Fig. 4h).

3.1.3. Lombardia, Italy

The third dataset was obtained over Lombardia, Italy with a Vexcel UltraCam Xp on May 29, 2015. The aerial images were processed to obtain an orthomosaic and DSM with a Ground Sampling Distance (GSD) of 20 cm. A subset of 5000 × 5000 pixels (1000 × 1000 m) was selected for the experimental analyses. The area consists of a residential area, river, dense forests, agricultural fields and a dike (Fig. 4c,f). A DTM of this area was obtained by the Compagnia Generale Ripreseaeree (CGR S.p.A.) by manually editing the DSM. Therefore, the reference data for the classification part of the experimental analyses was determined by classifying all pixels where the difference between the DSM and DTM was greater than 50 cm as off-ground, and pixels where they were equal as ground. Pixels where the difference was between 0 and 0.5 m were left unlabelled (Fig. 4i).

3.1.4. ISPRS benchmark dataset

The proposed method was also tested on the ISPRS 2D Semantic Labelling dataset of Vaihingen[3]. The dataset consists of 33 tiles, for which orthophotos and DSMs with a spatial resolution of 9 cm are provided. Sixteen tiles have reference labels corresponding to six semantic classes: impervious surfaces, buildings, low vegetation, trees, cars and clutter/background. In accordance with the benchmark, a 3 × 3 erosion filter was used on these reference data to remove border pixels from the quality analyses. It should be noted that while it is useful to test the algorithm using an existing benchmark, it is not the optimal dataset to demonstrate the utility of DTM-extraction techniques such as ours which targets two classes: ground and non-ground. We therefore consider the ISPRS class “impervious surfaces” to equate ground, whereas non-ground consists of the ISPRS classes: buildings, trees, and cars. The ISPRS classes low vegetation and clutter are not considered in our accuracy analysis due to inconsistencies. For example, the “low vegetation” class contains both shrubs (off-ground) and grass (ground) pixels.

3.2. Experimental set-up

3.2.1. Setting up the proposed network – feature sets, reference labels and dilation

For each of the three datasets, experiments were conducted which trained a FCN with randomly selected patches and utilizing either the true reference labels or the labels assigned through the proposed morphological rule-based method. Ideally, the classification accuracy of the training samples labelled through the proposed rule-based method should approximate the accuracies obtained when using the manually labelled reference data. Furthermore, we motivate the use of dilated filters in the network architecture by providing the results of a FCN in which no dilation is applied in the second convolutional layer.

The parameter values for the rule-based method were tuned on the Kigali dataset, and the same parameters ($w_{small} = 6$m, $w_{big} = 20$m, $\tau_o = 1.0$m, and $\tau_g = 0.5$m) are applied to the Dar es Salaam and Lombardia datasets. Experimental analyses indicated that slight variations in w_{small} (0.2–1 m), w_{big} (10–20 m), and τ_o (0.4–1 m) did not significantly change the results for these three datasets. Given the feature sets described in Table 2, experiments were performed using FCNs exploiting only the imagery (RGB), only the DSM (Z and nZ), or both imagery and DSM (RGBZ, RGBnZ, RGBDTM, RGBnDSM). Note that the DTM feature sets, obtained by interpolating the elevation values of pixels labelled as ‘ground’, were calculated separately for both the true reference labels and the labels assigned through the rule-based method. All features were normalized according to the maximum and minimum values of the respective dataset.

For each of these combinations, three folds of 2000 randomly selected patches of 167 × 167 pixels were used to train a FCN using stochastic gradient descent (SGD) with momentum (Krizhevsky et al., 2012) and a batch-size of 32. The networks were trained with a learning rate of 0.0001 for 30 epochs followed by another 10 epochs with a training rate of 0.00001. Weights for all convolutional layers were initialized using the improved Xavier initialization to $\sqrt{\frac{2}{M^2 \cdot K}}\mathcal{N}(0, 1)$ (He et al., 2015). The dropout rate of the final convolution as 0.5, and a batch size of 32 was used. The network was implemented in MatConvNet[4].

The accuracy assessment is conducted using the mean Producer’s Accuracy (mPA) and mean User’s Accuracy (mUA), providing the average and standard deviation across the three folds of randomly selected samples. The mPA (Eq. (4).) and mUA (Eq. (5) are calculated using the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) of the ground class.

$$mPA = \frac{\left(\left(\frac{TP}{TP+FN}\right) + \left(\frac{TN}{TN+FP}\right)\right)}{2} \quad (4)$$

$$mUA = \frac{\left(\left(\frac{TP}{TP+FP}\right) + \left(\frac{TN}{TN+FN}\right)\right)}{2} \quad (5)$$

3.2.2. Comparison with deeper network architectures

The proposed method utilizes a much smaller network than those which are proposed for other deep learning tasks. This was a conscious

[2] http://ramanihuria.org/.

[3] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html.

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

Table 3
An overview of the three FCN network architectures FCN-DK4, FCN-DK5, FCN-DK6 tested to address the effects of added depth to the DTM extraction problem. FCN-DK4 is made up of layers DK1 up to DK4.

Name	Included in FCN-			Layer	Filter size M (pixels)	Filter dilation d (pixels)	Number of filters K'	Padding z (pixels)	Receptive field size (pixels)
	DK4	DK5	DK6						
DK1	x	x	x	Convolution1 Batch normalization lReLU	5 × 5	1	16	2	5 × 5
	x	x	x	Pooling1	5 × 5	–	–	2	9 × 9
DK2	x	x	x	Convolution2 Batch normalization lReLU	5 × 5	2	32	4	17 × 17
	x	x	x	Pooling2	9 × 9	–	–	4	25 × 25
DK3	x	x	x	Convolution3 Batch normalization lReLU	5 × 5	3	32	6	37 × 37
	x	x	x	Pooling3	5 × 5	–	–	6	49 × 49
DK4	x	x	x	Convolution4 Batch normalization lReLU	5 × 5	4	32	8	65 × 65
	x	x	x	Pooling4	5 × 5	–	–	8	81 × 81
DK5		x	x	Convolution5 Batch normalization lReLU	5 × 5	5	32	10	101 × 101
		x	x	Pooling5	5 × 5	–	–	10	121 × 121
DK6			x	Convolution6 Batch normalization lReLU	5 × 5	6	32	12	145 × 145
			x	Pooling6	5 × 5	–	–	12	169 × 169
Classification	x	x	x	Convolution7 Dropout Loss	1 × 1	1	2	0	169 × 169

choice, as deeper networks also have more parameters and therefore require more sophisticated hardware and longer training times than smaller networks. In this study, a preference was given to smaller networks, as the network is trained and tested separately for each dataset. However, we provide comparisons with deeper network architectures in order to justify this decision. For this purpose we select three FCNs with Dilated Kernels (DK) which were specifically designed for remote sensing applications (Persello and Stein, 2017). Table 3 displays the network architecture, where three networks with varying depths were tested: FCN-DK4, FCN-DK5, and FCN-DK6. These networks have 4, 5, and 6 convolutional layers respectively (e.g. FCN-DK5 contains all of the layers DK1 – DK5 in Table 3 but not DK6). Similar to the proposed network, the FCN-DK networks consist of modules of a convolutional layer followed by batch normalization, a non-linear activation function (in this case leaky Rectified Linear Units or lReLU) and a max-pooling layer with no downsampling. Dilated convolutions are used to increase the receptive field size while limiting the number of parameters. Experiments were run using the rule-based reference labels and the RGBnZ feature set and the same three folds of training samples. All networks were trained with 150 epochs at a learning rate of 10^{-6} followed by another 20 epochs with a learning rate of 10^{-7}. Shallower network architectures may require less epochs, but using the same hyper-parameters for all four network architectures enables a fairer comparison.

3.2.3. Comparison with existing DTM extraction methods

The proposed method is compared to two existing DTM extraction algorithms, namely LAStools[5] which implements a variation of progressive densification DTM extraction (Axelsson, 2000) and gLidar[6] which is based on differential morphological profiles (Mongus et al., 2014). There are three parameters for the LAStools implementation: the step size, bulge, and standard deviation. The step size indicates the dimensions of the grid used to select the initial ground samples. This parameter was optimized for each dataset by trying a step size of 5 m to 40 m at 5 m intervals. The bulge parameter refers to the height in meters that the TIN is allowed to go up during the refinement stage. Values from 0.3 to 1.8 m in steps of 0.3 were tested for each dataset. The final parameter refers to the maximal standard deviation for planar patches, values from 0 to 40 cm were tested in steps of 10 cm. All parameter combinations were tested for the three datasets, and the combination which maximized the mPA regarding the true reference data are reported. For the gLidar implementation, parameter settings described in the work by Mongus et al. (2014) were set to: $S = 50$m, $k = 0.01$, $n = 0.10$, and $b = 0.5$. A detailed description of the meaning of these parameters can be found in the original presentation of the algorithm (Mongus et al., 2014). Finally, we also compare the proposed method to the manually generated DTM for the Lombardia dataset. The pixels which were labelled as ground by the proposed method were selected and a bilinear interpolation was performed to construct a DTM. The cumulative error between the predicted and reference DTMs for the pixels labelled as ground are provided.

3.2.4. Performance on the ISPRS benchmark

The sixteen labelled tiles of the ISPRS benchmark were used to test the performance of our proposed algorithm. Although good results were obtained with the parameter settings used for the previous datasets ($w_{small} = 6$m, $w_{big} = 20$m, $\tau_o = 1.0$m, and $\tau_g = 0.5$m), the presence of larger buildings and a relatively flat terrain in the ISPRS benchmark dataset caused slightly better results to be obtained with $w_{small} = 3$m and $w_{big} = 30$m. As the roofing material of these larger buildings was also different from the surrounding buildings, the additional criterion for off-ground samples mentioned in Section 2.1 was implemented.

[5] https://rapidlasso.com/lastools/.
[6] https://gemma.feri.um.si/gLiDAR/index.html.

Table 4
The accuracy of the proposed FCN strategies for classifying ground vs. off-ground pixels in the Kigali, Dar es Salaam, and Lombardia datasets. The labels of the training samples are either obtained from the reference data (ref) or the rule-based morphological method (mph) whereas the input feature channels are either derived from the image (RGB), DSM (Z, nZ) or both RGB and DSM (RGBZ, RGBnZ, RGBDTM, RGBnDSM). The average and standard deviation of the mPA and mUA for three folds of randomly selected training data is presented.

Labels	Features	Mean Producer's Accuracy (%)						Mean User's Accuracy (%)					
		Kigali		Dar es Salaam		Lombardia		Kigali		Dar es Salaam		Lombardia	
ref	RGB	94.8	± 0.2	94.7	± 2.0	89.5	± 0.5	95.2	± 0.9	93.6	± 2.5	90.0	± 0.9
	Z	65.4	± 3.0	77.5	± 0.5	97.1	± 0.3	87.6	± 1.5	87.7	± 0.9	97.0	± 0.2
	nZ	62.2	± 4.6	77.9	± 3.8	95.3	± 1.0	83.8	± 3.6	84.9	± 0.7	94.4	± 2.6
	RGBZ	96.1	± 0.7	95.8	± 1.2	97.7	± 0.0	**97.4**	± 0.0	97.2	± 0.3	97.9	± 0.1
	RGBnZ	94.6	± 0.7	96.1	± 1.3	97.2	± 0.2	96.3	± 1.0	98.1	± 0.8	97.4	± 0.3
	RGBDTM	94.3	± 0.8	93.6	± 1.7	91.5	± 0.6	96.0	± 0.5	95.2	± 0.7	91.9	± 0.2
	RGBnDSM	**97.9**	± 0.4	**99.0**	± 0.3	**99.1**	± 0.4	96.9	± 0.6	**98.6**	± 0.4	**99.3**	± 0.2
ref	RGBnZ (no dilation)	91.9	± 1.4	93.5	± 1.0	95.5	± 0.1	93.9	± 0.5	95.6	± 0.6	96.1	± 0.3
mph	RGB	93.9	± 0.3	92.6	± 1.2	88.3	± 0.2	88.0	± 1.0	94.3	± 0.2	88.7	± 0.2
	Z	81.3	± 1.2	75.8	± 4.2	**95.2**	± 0.2	77.0	± 1.5	87.5	± 1.5	94.1	± 0.3
	nZ	74.3	± 0.4	80.6	± 3.5	94.1	± 0.4	69.9	± 1.9	85.0	± 1.0	92.3	± 0.9
	RGBZ	91.4	± 1.3	**94.3**	± 1.0	**94.7**	± 0.0	88.5	± 0.4	95.8	± 0.4	93.5	± 0.3
	RGBnZ	92.8	± 0.3	**95.0**	± 0.4	**94.7**	± 0.2	83.9	± 0.8	95.7	± 0.5	**93.7**	± 0.2
	RGBDTM	**94.0**	± 0.2	91.6	± 1.8	89.7	± 0.5	**89.3**	± 1.7	94.8	± 0.2	88.9	± 0.4
	RGBnDSM	92.7	± 0.3	92.9	± 1.5	93.9	± 0.0	87.0	± 0.2	**95.9**	± 0.2	92.3	± 0.2

Conform to the benchmark, the User's Accuracy (precision), Producer's Accuracy (recall), and F1-scores are provided as quality metrics for the ground (impervious surfaces) and non-ground (buildings, trees, and cars) classes.

3.2.5. Regression-based DTM experiments

An interesting question is whether the proposed method can be altered to directly predict the DTM using regression-based deep learning rather than using first classifying the ground pixels and then interpolating the DTM (as proposed above). Such a regression-based method would consist of five steps. The first step is the rule-based identification of ground vs. off-ground samples using the same methodology as defined in Section 2.1. Secondly, a nDSM can be approximated by calculating the difference between the input DSM and an initial DTM obtained by interpolating the pixels labelled as ground in the previous step. The third step then consists of training a regression FCN rather than the classification FCN proposed in Section 2.2. Changing the classification FCN to a regression FCN can be done by replacing the softmax loss function with a ℓ_2 loss function to minimize the squared Euclidean distance between the height predicted by the network and the nDSM created from the rule-based labels in the previous step. The fourth step then consists of applying this trained (regression) FCN to the entire dataset to obtain a complete nDSM. Finally, the fifth step then consists of subtracting the FCN-nDSM from the input DSM to obtain the DTM of the entire area. This method was tested for the Kigali and Dar es Salaam datasets. The Mean Error (ME) and Root-Mean-Square Error (RMSE) for the entire scene as well as only the ground pixels are presented as quality metrics.

4. Results

4.1. Setting up the proposed network – feature sets, reference labels and dilation

The results obtained by the proposed FCN method according to various combinations of training labels and input channels is presented in Table 4 and Figs. 5–7. The first observation is that networks which utilize both image-based and DSM-based input channels (i.e. RGBZ, RGBnZ, RGBDTM and RGBnDSM) outperform networks which utilize only DSM-based (Z, nZ) channels for the Kigali and Dar es Salaam datasets. Using only image-based (RGB) channels as input obtains good results for the Kigali dataset, though the inclusion of elevation information clearly improves the results in Dar es Salaam and Lombardia. When true reference labels are available, the RGBnDSM method has the highest performance. This is logical as the nDSM input channel constructed using the true reference labels essentially defines the height of objects above the ground. However, the nDSM feature constructed using the rule-based training labels is an imperfect representation as these rule-based training labels may be erroneous or incomplete thereby causing the nDSM feature to be inaccurate.

Rather, the RGBDTM input channels achieve the highest mPA when using the rule-based training labels for the Kigali dataset. In this case, using only image-based features (RGB) works quite well for the Kigali dataset which may be due to the fact that the ground and elevated objects are more easily distinguished using spectral features in this dataset and that the topographic information is less informative due to the steep slopes in the area. Some of the errors in the top left corner (Fig. 5b) are due to inconsistencies in the UAV flight operations, resulting in a blurring of the orthomosaic and a loss of texture. Previous research indicated that texture was an important cue for distinguishing building roofs from ground (Gevaert et al., 2016). One of the assumptions of our method is that the pixels along the edges of contiguous elevated objects will have a similar appearance as the central parts of those objects. This example in the top-left part of the Kigali dataset is a case where this assumption does not hold, as some pixels in the central parts of the contiguous buildings have a blurred texture (unlike the pixels along roof edges). This may cause errors in the classification results and interpolated DTM. The RGBnZ works best for the Dar es Salaam and Lombardia datasets. For the Lombardia dataset, using the Z channels as input for the FCN slightly outperforms the sets using both image-based and DSM-based combinations. Most of the errors in the Lombardia dataset are due to the assignation of incorrect labels to an elevated road during the rule-based label assignation which are used to train the FCNs (Fig. 7a), causing systematic mislabeling of this road as off-ground (Fig. 7b). Furthermore, there are some errors in the vegetation in the lower left corner, where errors in the rule-based labels caused by systematic tree height differences are propagated in the classification and interpolated DTM. The proposed FCN-RGBnZ method performs better than gLidar in these areas, although Lastools appears to perform best in this particular situation. The large extent of contiguous off-ground objects (forests) and relatively flat terrain in the Lombardia dataset suggests that increasing w_{big} could achieve better results.

These results indicate that although there are slight differences according to the scene characteristics of the various datasets, the RGBnZ

Fig. 5. Classification maps of the Kigali dataset for the rule-based training labels (a), FCN-RGBnZ (b), gLidar (c) and Lastools (d).

input channels generally achieve a high and reliable classification accuracy when using the rule-based initialization of training labels. Indeed, some errors in the initial labelling of the Kigali dataset (Fig. 5a) are corrected in the FCN-RGBnZ output (Fig. 5b). This indicates that the proposed FCN does more than 'fill in the gaps' by relearning the top-hat heuristic used to generate the training labels. We furthermore see that for all three datasets, the network which does not include the dilation in the convolutional layers performs worse than the proposed network when using RGBnZ features. Using this proposed strategy with rule-based training labels RGBnZ features and dilated convolutional layers, we can accurately classify ground vs. off-ground objects with an mPA of 92.8% to 95.0% and an mUA of 83.9% to 93.7% for the three datasets. These results, which exploit simple rules to label the training samples, have an mPA of only 1.8% (Kigali), 1.1% (Dar es Salaam), and 2.5% (Lombardia) lower than FCNs trained using manually-labelled training samples.

4.2. Comparison with deeper network architectures

Results indicate that adding additional convolutional layers in this

Fig. 6. Classification maps of the Dar es Salaam dataset for the rule-based training labels (a), FCN-RGBnZ (b), gLidar (c) and Lastools (d).

application does not lead to an increased accuracy. Table 5 displays the average accuracies obtained for each of the three folds of the Kigali, Dar es Salaam, and Lombardia datasets. The mean producer's accuracy remains around 93.4% for Kigali, 95.8% for Dar es Salaam, and 94.8% for Lombardia. The differences in the accuracies reported in Tables 5 and 4 are due to changes in the training rate and number of epochs. Table 6 presents the number of false positives and negatives. The networks generally show similar tendancies for the three datasets – Kigali has a larger number of false positives than false negatives, whereas Lombardia has relatively more false negatives. The additional depth of FCN-DK4, FCN-DK5 and FCN-DK6 comes with higher computing requirements, as illustrated in Table 7. The FCN-DK6 network has 120 000 parameters which require 462 KB of memory which takes around 5.42 h to train. However, the FCN-RGBnZ network requires only 23,000 parameters which require 90 KB of memory and 1.92 h of traning time. The smaller network can achieve a slightly higher accuracy than the deeper architectures in only 35% of the time.

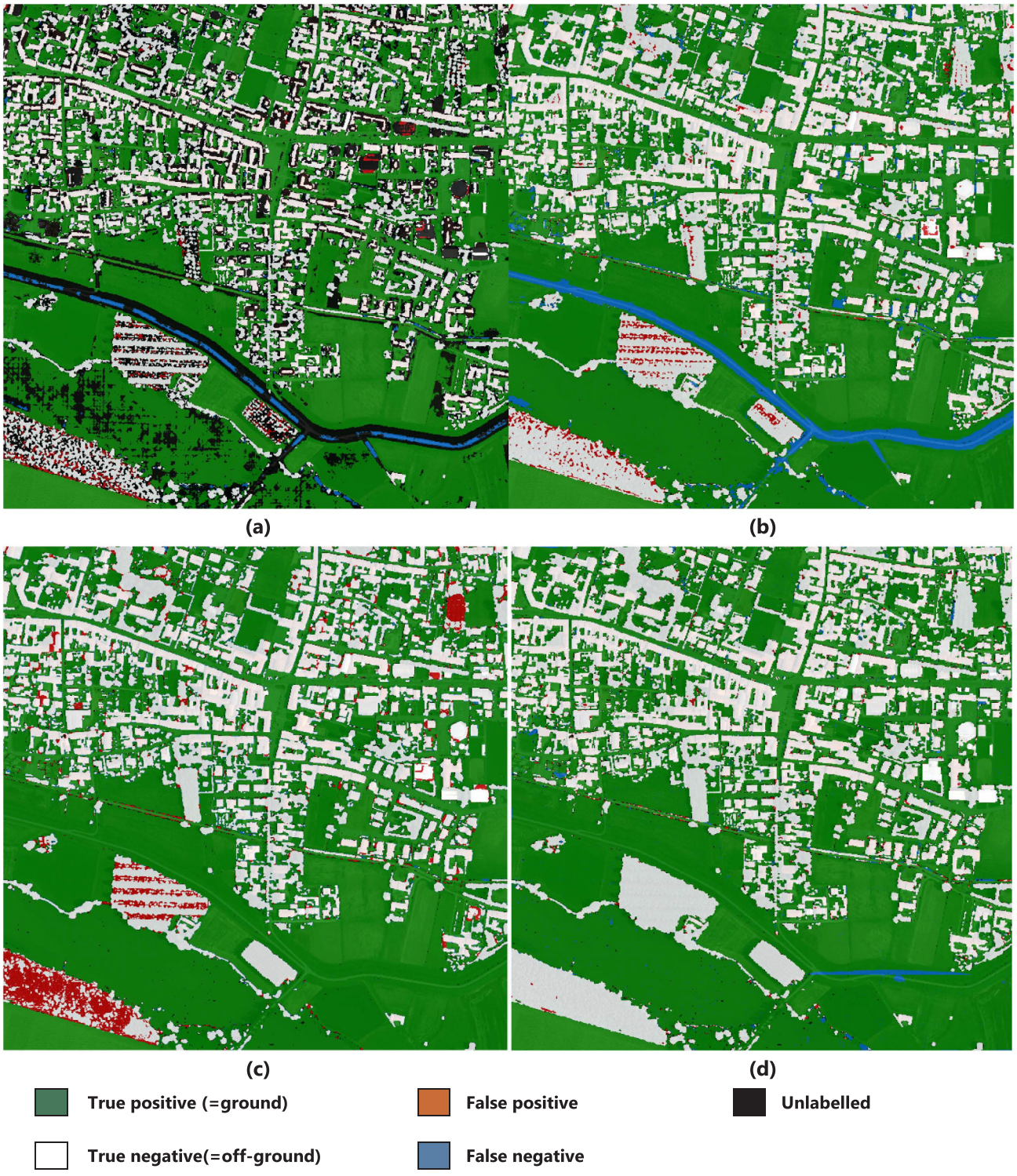

Fig. 7. Classification maps of the Lombardia dataset for the rule-based training labels (a), FCN-RGBnZ (b), gLidar (c) and Lastools (d).

Table 5
The mPA and mUA of FCN-RGBnZ (the proposed network), FCN-DK4, FCN-DK5, and FCN-DK6 for Kigali, Dar es Salaam (Dar), and Lombardia.

FCN Network	Overall Accuracy (%)			Mean Producer's Accuracy (%)			Mean User's Accuracy (%)		
	Kigali	Dar	Lombardia	Kigali	Dar	Lombardia	Kigali	Dar	Lombardia
FCN-RGBnZ (proposed)	93.5	97.6	95.1	93.5	95.9	94.9	83.7	95.3	94.2
FCN-DK4	93.4	97.8	94.7	93.4	96.0	94.7	83.5	95.9	93.7
FCN-DK5	93.6	97.8	94.8	93.6	95.6	94.7	83.9	96.3	93.9
FCN-DK6	93.2	97.9	94.7	93.2	95.6	94.7	83.0	96.4	93.8

Table 6
The number of false negatives and false positives of FCN-RGBnZ (the proposed network), FCN-DK4, FCN-DK5, and FCN-DK6 for the three datasets.

FCN Network	False negatives			False positives		
	Kigali	Dar es Salaam	Lombardia	Kigali	Dar es Salaam	Lombardia
FCN-RGBnZ (proposed)	152,887	269,509	752,471	979,921	345,090	456,848
FCN-DK4	142,741	278,738	845,153	1,005,187	284,432	455,564
FCN-DK5	152,936	312,670	800,301	961,631	241,046	468,322
FCN-DK6	155,716	320,196	832,636	1,040,235	226,908	456,014

Table 7
Characteristics of the four FCN network architectures.

FCN Network	FCN-RGBnZ	FCN-DK4	FCN-DK5	FCN-DK6
Number of parameters	23,000	67,000	92,000	120,000
Memory requirement for parameters (KB)	90	260	361	462
Average training time (hours)	1.92	3.35	4.33	5.42
Final receptive field size (pixels)	57 × 57	81 × 81	121 × 121	169 × 169

Table 8
The mPA and mUA of LAStools, gLidar, the rule-based labels (Step 1), and FCN-RGBnZ (Step 2) for the three datasets. For the rule-based labels, we provide the mPA of the training samples which were labeled, and the mPA penalizing unlabelled pixels as classification errors in parentheses.

DTM extraction algorithm	Mean Producer's Accuracy (%)			Mean User's Accuracy (%)		
	Kigali	Dar es Salaam	Lombardia	Kigali	Dar es Salaam	Lombardia
LAStools	84.4[a]	83.8[b]	98.1[c]	67.3[a]	74.1[2]	97.7[c]
gLidar	85.3	85.7	93.1	66.2	75.5	94.4
Rule-based labels (Step 1)	95.1 (71.3)	96.2 (68.6)	97.7 (75.5)	90.8 (56.6)	97.4 (65.3)	97.4 (96.6)
FCN-RGBnZ (Step 2)	92.8	95.0	94.7	83.9	95.7	93.7

[a] The best LAStools results for the Kigali dataset were obtained using a step of 20 m, bulge of 0.3 m, and standard deviation of 30 cm.
[b] Using a step of 20 m, bulge of 0.3 m, and standard deviation of 40 cm.
[c] Using a step of 40 m, bulge of 1.8 m, and standard deviation of 0 cm.

4.3. Comparison with existing DTM extraction methods

The proposed method also clearly outperforms the reference methods both visually (Figs. 5–7) and quantitatively (Table 8). Note that two accuracy measures are provided for the rule-based labels in Table 8. As the morphological selection method does not label the entire image, we provide the accuracy of the labelled samples, and the accuracy where unlabelled samples are considered as errors in parentheses. The proposed method outperforms the reference methods for all three datasets with a single exception. LAStools slightly outperforms the automated method for the Lombardia dataset. However, it should be noted that the LAStools parameters were optimized separately for each dataset to maximize the accuracy on the testing data, whereas the proposed method utilized the same parameters for all datasets and is therefore more easily implemented in automatic workflows. The proposed method outperforms LAStools in the Kigali (increasing the mPA by 8.4% and mUA by 16.6%) and Dar es Salaam (increasing the mPA by 11.2% and mUA by 21.6%) datasets. In the Kigali dataset, both LAStools and gLidar clearly suffer from the steep slopes in the lower half of the image, where parts of the roofs are misclassified as terrain (Fig. 5c,d) in a clear example of the problem illustrated in Fig. 2e. This effect is clearly lower using the proposed FCN-RGBnZ method,

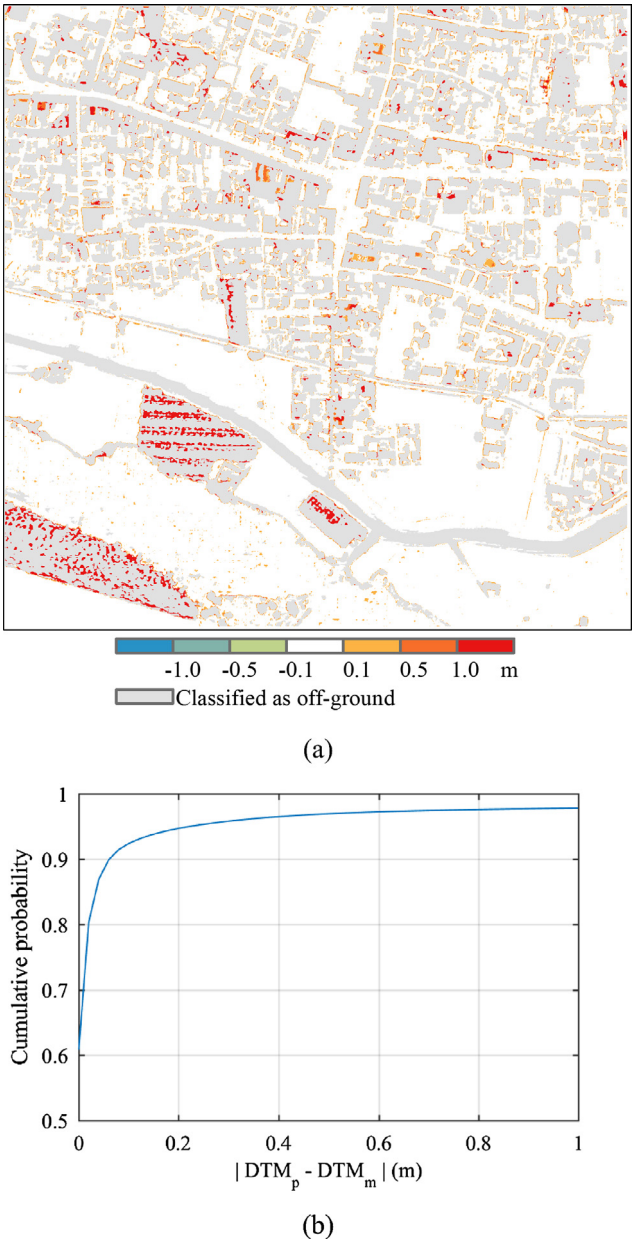

Fig. 8. A visualization of the predicted DTM (DTM_p) minus the manual DTM (DTM_m) for the Lombardia dataset (a), and the cumulative probability of this difference for pixels classified as ground by the proposed algorithm (b).

illustrating the importance of including RGB information in areas where the surface topography is complicated (Fig. 5b). Indeed, when using only the height information (i.e. Z and nZ feature sets in Table 4), LAStools and gLidar outperform the FCN in Lombardia and have a higher mPA in Kigali and Dar es Salaam. In the Dar es Salaam dataset, contiguous roof-tops (Fig. 2g,h) appear to cause many problems errors

Table 9
User's Accuracy (= precision), Producer's Accuracy (= recall), and F1-scores for the FCN-RGBnZ algorithm applied to the ISPRS benchmark dataset. The top row presents the average percentage for all sixteen tiles, the rows below indicate the results of a tile with a high accuracy and lower accuracy.

	Ground (impervious surfaces)			Off-ground (buildings, trees, and cars)		
	User's Accuracy (%)	Producer's Accuracy (%)	F1 score	User's Accuracy (%)	Producer's Accuracy (%)	F1 score
All tiles with reference labels	92.2	74.5	82.0	87.4	96.9	91.8
Tile with high accuracy (N° 34)	92.9	88.1	90.4	95.7	97.1	96.1
Tile with lower accuracy (N° 21)	91.0	71.5	80.1	87.1	96.5	91.5

for gLidar and LAStools (Fig. 6c,d). In the Lombardia dataset, the proposed method outperforms the two reference methods in the correct classification of forested areas as off-ground. These areas in the bottom left and top right corners of the image are clearly visible as false positives in the gLidar results (Fig. 7c). However, the Lombardia dataset also clearly illustrates how samples on the elevated road crossing the centre of the dataset were mislabelled in the first rule-based step (Fig. 7a), causing systematic errors in the prediction map obtained by the proposed method (Fig. 7b).

A comparison between the DTM obtained through the proposed method and a manual editing is provided in Fig. 8. Considering only areas labelled as ground by the proposed method, there was a mean error of 0.16 m and a mean absolute error of 0.18 m compared to the manually edited DTM. This indicates that there is a small bias of less than one GSD in results of the proposed method, which is slightly higher than the reference DTM provided. 93.1% of the pixels have an absolute difference of less than 10 cm in the two DTMs – which is less than half the GSD – and 96.9% have an absolute difference of less than 0.5 m (Fig. 8b).

4.4. Results on the ISPRS benchmark dataset

Table 9 displays the quantitative accuracies of FCN-RGBnZ applied to the ISPRS benchmark. Impervious surfaces are classified as ground with a User's Accuracy of 92.2% and a Producer's Accuracy of 74.5%. The three ISPRS classes of buildings, trees, and cars are classified as off-ground objects with a User's Accuracy of 87.4% and Producer's Accuracy of 96.9%. These results indicate that there are more false negatives than false positives in the results, which can also be observed visually (see Fig. 9). Some of these errors can be attributed to inconsistencies in the benchmark labels. For example, the central area of Fig. 9c indicates false positives in the central area, where the ISPRS reference label is tree (Fig. 9a). However, a visual analysis of the image (Fig. 9b) suggests that these pixels could indeed be ground in between the trees. The results in Table 9 indicate a relatively large error due to pixels labelled as impervious surfaces to be classified as off-ground (i.e. false negatives). A visual analysis of the results indicates that such false negatives (Fig. 9g) often occur in shadowed streets, where the reference label indicates impervious surface (Fig. 9f), but the DSM actually shows relatively high elevation values (Fig. 9e) and there are few visual cues in the image due to the shadows (Fig. 9d). The different semantic labels and inconsistencies between the reference labels and input data make it difficult to compare the results of the FCN-RGBnZ method proposed for DTM extraction with the other contributions to the ISPRS benchmark.

4.5. Regression-based DTM experiments

The error metrics in Table 10 indicate that the nDSM returned by the regression-based FCN are an average of 23 cm higher in the Kigali dataset than the reference nDSM values. This is 46 cm in the Dar es Salaam dataset. One difficulty in DTM prediction is that it isn't clear which 'terrain' height to assign to the terrain under building located on a slope. I.e. would it be correct to interpolate the height of the surrounding terrain, or should we assume the floor is flat and assign the elevation of the lowest floor to the entire building footprint? Due to such confusions, we also include error metrics of the nDSM predictions for pixels *labelled as ground* in the reference data. Table 10 indicates that the ME of the ground pixels is actually much higher than the global average, overestimating the reference elevation data by 1.74 m in Kigali and 2.18 m in Dar es Salaam. The RMSE of the ground pixels is also higher than that of the entire dataset in both cases. Further investigations indicated that although the average nDSM values of the predicted and reference datasets were similar, the variance of the predicted nDSM was much lower than that of the reference data. In essence, all height values are therefore closer to the mean nDSM value of the dataset. This in turn causes the overestimation of the height of ground pixels.

5. Discussion

In UAV applications, variations in flight heights and camera parameters are likely to cause a wider diversity in the spatial resolution of datasets. The representation of off-ground objects in datasets with a spatial resolution of 3 cm, 5 cm, or 20 cm for example, will be quite different. It is unclear how this wide variation of spatial resolution in UAV datasets will influence the parameters learned by a FCN. Hu and Yuan (2016) address this problem by summarizing the elevation information contained by point clouds in a grid of a fixed spatial resolution. Although this allows the utilization of a single FCN trained for various study areas and datasets, this strategy will not exploit the full information contained in a dataset which has a higher point density (or spatial resolution) than the trained network. Therefore, an important characteristic of the method proposed here is that it demonstrates that it is feasible to train and apply a FCN on each dataset independently. The present manuscript demonstrates this using datasets from UAV or aerial imagery, but the method is not limited to these types of images. It would also be possible to apply the proposed method to satellite imagery with a lower spatial resolution and a larger extent. Applications which intend to cover a larger extent may benefit from larger sample sizes to train the network – this stresses the advantage of using the rule-based strategy to provide labels for training. The labelling and selection of training samples is completely automated in the proposed methodology. It is therefore in principle extendible to very large datasets. Furthermore, although training is time-consuming, FCNs are very fast in the testing phase and would therefore be a viable option in the classification of ground over large extents.

Although the point clouds obtained from dense matching tend to contain more random noise errors than LiDAR point clouds (Nex and Gerke, 2014), the simultaneous acquisition of both elevation and radiometric information can be seen as an advantage of UAV datasets and aerial photogrammetry in general. Making use of the complementary information in imagery may help distinguish ground from off-ground areas when the elevation information itself is not sufficient. However, it is important to note that the rule-based selection of training samples is only an estimation, and that mislabelled samples may cause systematic errors in the output of the FCN. For example, when considering scenes with steep slopes where ground and off-ground objects present a step-like pattern (i.e. Fig. 2e), the rule-based selection of training samples based on morphological filters will not be able to distinguish between ground and off-ground objects. However, if such geometrically ambiguous areas form a minority in the dataset, then a

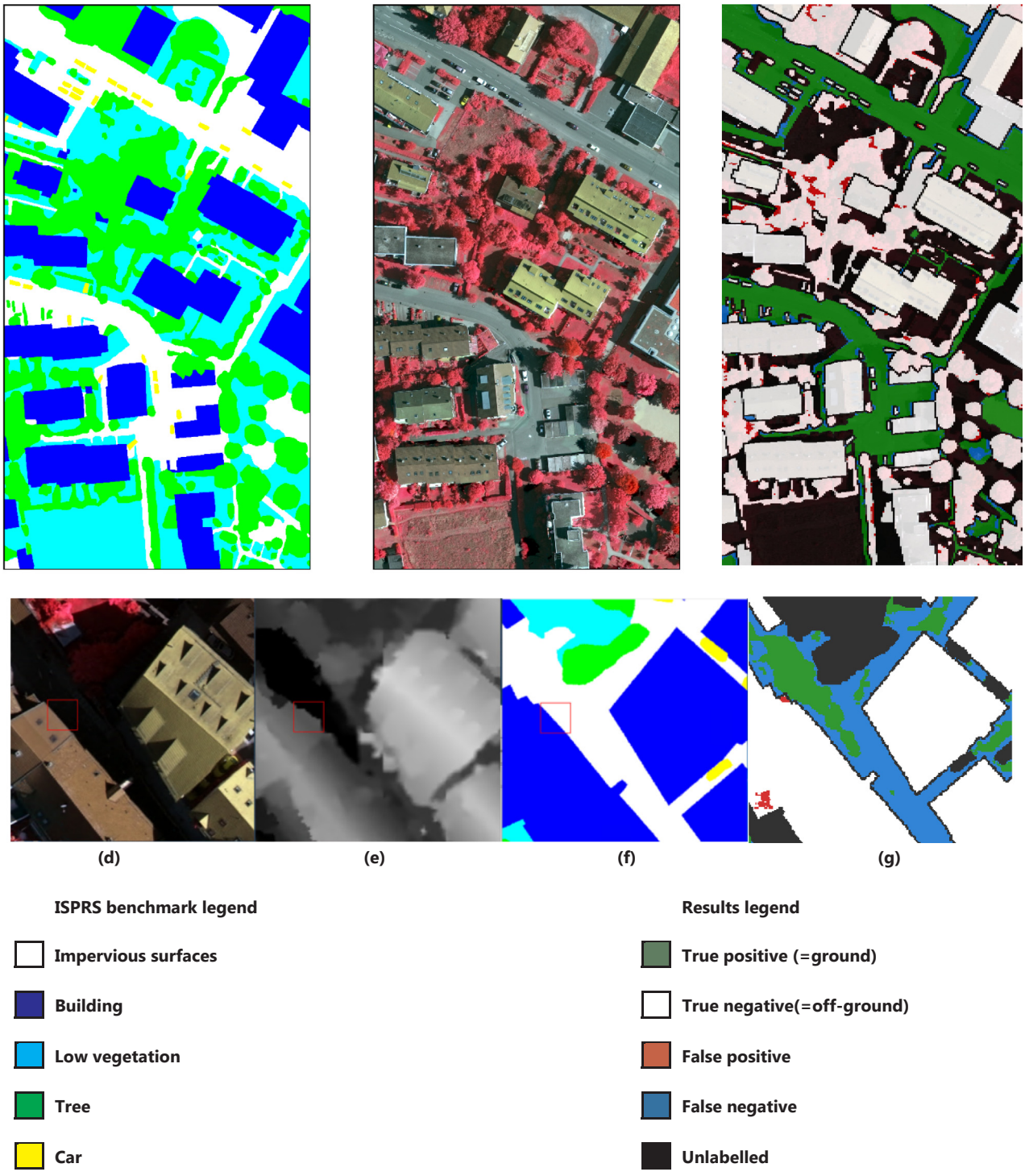

Fig. 9. Input ISPRS reference labels (a) and false-color images (b), and the FCN-RGBnZ results (c) of tile 34. The bottom row presents an example of causes of false negatives in tile 05. Note the narrow streets which are labelled as impervious surfaces in the reference data (f), but are classified as off-ground by our algorithm (g) due to the combination of shadows in the imagery (d) and elevated values in the DSM (e).

sufficient number of correct ground vs. off-ground training samples can be collected. If a sufficient number of correct samples are captured and utilized to train the supervised classifier, and presuming the radiometric information from the imagery is capable of distinguishing between the ground and off-ground objects, then these initial errors may be corrected in the second step.

This second step, the exploitation of deep learning methods refers to a field of research which is currently developing rapidly. It is likely that emerging network architectures developed in the near future may further increase the accuracy of the proposed method. However, the observations of the present paper may serve to guide users towards the selection of a suitable network architecture. Firstly, one important issue is the redundancy of calculations when performing a pixel-based classification or semantic segmentation as it is known in the computer vision community. This motivated the selection of a FCN architecture rather than a CNN architecture in the current paper. Other emerging options such as PixelNet (Bansal et al., 2017) could be considered in the future. Similarly, due to the high spatial resolution compared to the size

Table 10
ME and RMSE of the nDSM predictions obtained with the regression-based FCN calculated over the entire dataset (i.e. both ground and off-ground objects) or only the pixels labelled as ground. All values are in meters.

Dataset	ME[a] (m) Entire dataset	RMSE (m) Entire dataset	ME (m) Only ground	RMSE (m) Only ground
Kigali	0.23	1.62	1.74	1.85
Dar es Salaam	0.46	1.53	2.18	2.21

[a] The ME is calculated as the predicted nDSM minus the reference nDSM. Positive values therefore indicate that the predicted elevation overestimates the reference values.

of off-ground objects, it is important to increase the receptive field of the network. In the present case, this was done through the use of dilated filters and adjusted DSM features. Alternative strategies could include multi-scale approaches (Farabet et al., 2013) or skip-architectures (Song et al., 2017). Thirdly, the depth of the network architecture, or number of layers should also be considered. In general, the success of deep architectures may be attributed to their ability to learn complex patterns in very difficult classification tasks.

Network architecture may also be one of the underlying reasons behind the high errors obtained in the regression-based experiments. The main problem was the lower variance of the output nDSM predictions, causing the elevation of ground pixels to be overestimated. One hypothesis is that this has to do with the ℓ_2 loss function which penalizes outliers. In the case of DTM extraction, small errors overestimating the height of bare ground may be more concerning than larger errors underestimating the height of buildings. Further experiments could try using other loss functions such as the Huber loss (Huber, 1964) or Tukey's biweight function (Belagiannis et al., 2015) which others have found to be less sensitive to outliers when tuning deep regression networks. Another strategy could be the introduction of skip connections, which proved to be key to obtaining realistic height estimations from monocular imagery (Mou and Zhu, 2018). Further experimental analysis could focus on such direct height estimations to complement classification-based DTM extraction techniques such as the methodology proposed here.

In this application of DTM extraction, we are not interested in separating numerous abstract classes associated with complex appearance features like in other computer vision problems. In the considered application, the network should be able to capture features from both ground and non-ground, integrating radiometric and geometric variables. Results show that shallow networks with large receptive fields perform as good or better than deeper networks. On the other hand, shallower networks have less parameters, are easier and faster to train, less prone to overfitting, and generally more robust to different radiometric/geometric characteristics of the data set. The applicability of developments regarding the further reduction of parameters in deep learning networks could be analyzed in future works.

Furthermore, the DTM extraction algorithm proposed here has been designed to be trained and tested on a single dataset – focusing on UAV datasets which may have a limited extent and therefore limited number of training samples. If one were to instead combine UAV data from a large number of sources, which may become feasible in the near future due to the wider availability of UAV imagery, e.g. OpenAerialMap, using a deeper network architecture trained on all of these images could be an alternative strategy. In this case it would be important that the selected datasets represent challenging situations (such as those depicted in Fig. 2) in order to ensure that the network is able to handle them. Again, it depends on whether the user would like to have a quick DTM extraction tailored specifically to a single (UAV) dataset (i.e. the purpose of the current manuscript), or a general deep network trained applicable to a larger spatial extent.

Finally, another important consideration is how to assess the quality of a DTM extraction algorithm. In this case, we define the DTM as a classification problem, similar to Sithole and Vosselman (2004). Other studies use the vertical accuracy of a DTM compared to Ground Control Points (GCPs) collected in the field with GPS (Höhle and Höhle, 2009; Hugenholtz et al., 2013). However, we should remember that the final product is an interpolated DTM surface. As such, false positive rates which introduce errors into the interpolation could be more malign than false negatives which lower the detail of the reconstructed surface. Further research could consider how to assess the quality of DTM extraction methods without the presence of alternative DTMs or the costly collection of GCPs in the field.

6. Conclusions

Existing algorithms for DTM extraction still face difficulties due to data outliers and geometric ambiguities of the scene due to contiguous off-ground areas or sloped environments. This work postulates that in such cases, the radiometric information contained in aerial imagery may be leveraged to distinguish between ground and off-ground objects. This is particularly relevant for, but not limited to, UAV datasets which simultaneously acquire both elevation and radiometric information.

The proposed method uses two simple rules based on morphological filters to select examples of ground and off-ground objects using the DSM. The underlying idea is not to use these rules to label the entire dataset, but rather to select reliable samples which together describe the variability in the geometric and radiometric attributes of both classes. These samples are then used to train a supervised classifier, which labels each pixel in the entire scene and may correct errors in the initial labelling. We propose using a FCN, as deep learning methods are currently state-of-the-art in supervised classification problems. Improvements to deep learning methods are rapidly evolving, therefore it is plausible that the network architecture presented here could be improved according to the continued developments in this field.

In this research we address a number of issues which are important when adapting deep learning methods to DTM extraction. Firstly, we bypass the costly requirement of large amounts of training data by employing simple rules to automatically select and label representative samples from the dataset itself. By training the FCN for each dataset, we can account for both differences in the spatial resolution of different datasets as well as the natural variability of objects in different parts of the world. Secondly, we illustrate how FCNs can be adapted to consider the topographical variations over a larger area without increasing the computational complexity of the algorithm. This is done both by considering dilated filters in the network architecture and through the inclusion of feature channels which summarize variations in the elevation over larger areas.

The proposed method is successfully tested using three photogrammetric datasets with different spatial resolutions and covering scenes containing areas which challenge DTM extraction methods, as well as the ISPRS benchmark dataset. The datasets used for testing are relatively small but the results can easily be applied to larger study areas, or imagery and DTMs with a lower spatial resolution. We demonstrate the improvements of the proposed method with respect to two reference DTM extraction algorithms.

Acknowlegement

The authors would like to express their sincere gratitude to the

Compagnia Generale Ripreseaeree (http://www.cgrspa.com/) and Dar Ramani Huria/the World Bank (http:// http://ramanihuria.org/) for providing access to the Lombardia and Dar es Salaam datasets which have provided us the opportunity to evaluate the various features across different study areas. They would also like to express their gratitude to the University of Twente/Faculty ITC, City of Kigali, and other officials who facilitated the collection of the Kigali dataset.

References

Audebert, N., Le Saux, B., Lefèvre, S., 2017. Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (Eds.), Computer Vision – ACCV 2016. Springer International Publishing, Cham, pp. 180–196.

Axelsson, P., 2000. DEM generation from laser scanner data using adaptive TIN models. Int. Arch. Photogramm. Remote Sens. 33, 111–118.

Bansal, A., Chen, X., Russell, B., Gupta, A., Ramanan, D., 2017. PixelNet: Representation of the pixels, by the pixels, and for the pixels.

Bao, S.Y., Chandraker, M., Lin, Y., Savarese, S., 2013. Dense object reconstruction with semantic priors. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1264–1271. https://doi.org/10.1109/CVPR.2013.167.

Belagiannis, V., Rupprecht, C., Carneiro, G., Navab, N., 2015. Robust optimization for deep regression. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2830–2838. https://doi.org/10.1109/ICCV.2015.324.

Beumier, C., Idrissa, M., 2016. Digital terrain models derived from digital surface model uniform regions in urban areas. Int. J. Remote Sens. 1–17. http://dx.doi.org/10.1080/01431161.2016.1182666.

Blaha, M., Vogel, C., Richard, A., Wegner, J.D., Pock, T., Schindler, K., 2016. Large-scale semantic 3D reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3176–3184. https://doi.org/10.1109/CVPR.2016.346.

Cabezas, R., Straub, J., Fisher, J.W., 2015. Semantically-aware aerial reconstruction from multi-modal data. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2156–2164. https://doi.org/10.1109/ICCV.2015.249.

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land Use Classification in Remote Sensing Images by. Convolut. Neural Networks. CoRR abs/1508.0.

Chaplot, V., Darboux, F., Bourennane, H., Leguédois, S., Silvera, N., Phachomphon, K., 2006. Accuracy of interpolation techniques for the derivation of digital elevation models in relation to landform types and data density. Geomorphology 77, 126–141. http://dx.doi.org/10.1016/j.geomorph.2005.12.010.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2015. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: International Conference on Learning Representations (ICLR 2015). San Diego.

Debella-Gilo, M., 2016. Bare-earth extraction and DTM generation from photogrammetric point clouds including the use of an existing lower-resolution DTM. Int. J. Remote Sens. 37, 3104–3124. http://dx.doi.org/10.1080/01431161.2016.1194543.

Elmqvist, M., Jungert, E., Lantz, F., Persson, A., Soderman, U., 2001. Terrain modelling and analysis using laser scanner data. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 34 (3/W4), 219–226.

Farabet, C., Couprie, C., Najman, L., Lecun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. 35, 1915–1929. http://dx.doi.org/10.1109/TPAMI.2012.231.

Gevaert, C., Persello, C., Sliuzas, R., Vosselman, G., 2016. Integration of 2D and 3D features from UAV imagery for informal settlement classification using Multiple Kernel Learning. In: International Geoscience and Remote Sensing Symposium (IGARSS). https://doi.org/10.1109/IGARSS.2016.7729385.

Gevaert, C.M., Persello, C., Sliuzas, R., Vosselman, G., 2017. Informal settlement classification using point-cloud and image-based features from UAV data. ISPRS J. Photogramm. Remote Sens. 125, 225–236. http://dx.doi.org/10.1016/j.isprsjprs.2017.01.017.

Hane, C., Zach, C., Cohen, A., Angst, R., Pollefeys, M., 2013. Joint 3D scene reconstruction and class segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 97–104. https://doi.org/10.1109/CVPR.2013.20.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034.

Hingee, K., Caccetta, P., Caccetta, L., Wu, X., Devereaux, D., 2016. Digital Terrain from a two-step segmentation and outlier-based algorithm. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat Inf. Sci. 233–239.

Höhle, J., Höhle, M., 2009. Accuracy assessment of digital elevation models by means of robust statistical methods. ISPRS J. Photogramm. Remote Sens. 64, 398–406. http://dx.doi.org/10.1016/j.isprsjprs.2009.02.003.

Hohn, M.E., 1991. An Introduction to Applied Geostatistics, Computers & Geosciences. Oxford University Press, New York https://doi.org/10.1016/0098-3004(91)90055-I.

Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. http://dx.doi.org/10.3390/rs71114680.

Hu, X., Yuan, Y., 2016. Deep-learning-based classification for DTM extraction from ALS point cloud. Remote Sens. 8, 730. http://dx.doi.org/10.3390/rs8090730.

Huber, P.J., 1964. Robust estimation of a location parameter. Ann. Math. Stat. 35, 73–101. http://dx.doi.org/10.1214/aoms/1177703732.

Hugenholtz, C.H., Whitehead, K., Brown, O.W., Barchyn, T.E., Moorman, B.J., LeClair, A., Riddell, K., Hamilton, T., 2013. Geomorphological mapping with a small unmanned aircraft system (sUAS): Feature detection and accuracy assessment of a photogrammetrically-derived digital terrain model. Geomorphology 194, 16–24.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML.

Kemker, R., Kanan, C., 2017. Deep Neural Networks for Semantic Segmentation of Multispectral Remote Sensing Imagery. arXiv Prepr. arXiv1703.06452.

Kilian, J., Haala, N., Englich, M., 1996. Capture and evaluation of airborne laser scanner data. Int. Arch. Photogramm. Remote Sens. 31, 383–388.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS J. Photogramm. Remote Sens. 53, 193–203. http://dx.doi.org/10.1016/S0924-2716(98)00009-4.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105.

Ladický, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.H.S., 2012. Joint optimization for object class segmentation and dense stereo reconstruction. Int. J. Comput. Vis. 100, 122–133. http://dx.doi.org/10.1007/s11263-011-0489-0.

Lawson, C.L., 1972. Transforming triangulations. Discrete Math. 3, 365–372. http://dx.doi.org/10.1016/0012-365X(72)90093-3.

Liu, X., 2008. Airborne LiDAR for DEM generation: some critical issues. Prog. Phys. Geogr. 32, 31–49. http://dx.doi.org/10.1177/0309133308089496.

Mboga, N., Persello, C., Bergado, J., Stein, A., 2017. Detection of informal settlements from VHR Images using convolutional neural networks. Remote Sens. 9, 1106. http://dx.doi.org/10.3390/rs9111106.

Mongus, D., Lukač, N., Žalik, B., 2014. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. ISPRS J. Photogramm. Remote Sens. 93, 145–156. http://dx.doi.org/10.1016/j.isprsjprs.2013.12.002.

Mou, L., Zhu, X.X., 2018. IM2HEIGHT: Height Estimation from Single Monocular Imagery via Fully Residual Convolutional-Deconvolutional Network.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814.

Nex, F., Gerke, M., 2014. Photogrammetric DSM denoising. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 40, 231.

Pérez-Garcia, J.L., Delgado, J., Cardenal, J., Colomo, C., Ureña, M.A., 2012. Progressive densification and region growing methods for LIDAR data classification. Int. Arch. Photogramm. Remote Sensing, Spat. Inf. Sci. 39 (B3), 155–160.

Persello, C., Stein, A., 2017. Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. IEEE Geosci. Remote Sens. Lett. 14, 2325–2329. http://dx.doi.org/10.1109/LGRS.2017.2763738.

Riegler, G., Ulusoy, A.O., Bischof, H., Geiger, A., 2017. OctNetFusion: Learning Depth Fusion from Data.

Romero, A., Gatta, C., Camps-Valls, G., 2016. Unsupervised deep feature extraction for remote sensing image classification. IEEE Trans. Geosci. Remote Sens. http://dx.doi.org/10.1109/TGRS.2015.2478379.

Shelhamer, E., Long, J., Darrell, T., 2017. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 640–651.

Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv Prepr. arXiv1606.02585.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv Prepr. arXiv1409.1556.

Sithole, G., Vosselman, G., 2005. Filtering of airborne laser scanner data based on segmented point clouds. Int. Arch. Photogramm. Remote Sensing, Spat. Inf. Sci. 36 (Part 3), W19.

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 59, 85–101. http://dx.doi.org/10.1016/j.isprsjprs.2004.05.004.

Song, X., Herranz, L., Jiang, S., 2017. Depth CNNs for RGB-D scene recognition: learning from scratch better than transferring from RGB-CNNs. Aaai 4271–4277.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Tomljenovic, I., Höfle, B., Tiede, D., Blaschke, T., 2015. Building extraction from airborne laser scanning data: an analysis of the state of the art. Remote Sens. 7, 3826–3862. http://dx.doi.org/10.3390/rs70403826.

Tomljenovic, I., Tiede, D., Blaschke, T., 2016. A building extraction approach for Airborne Laser Scanner data utilizing the Object Based Image Analysis paradigm. Int. J. Appl. Earth Obs. Geoinformation 52, 137–148.

Tóvári, D., Pfeifer, N., 2005. Segmentation based robust interpolation-a new approach to laser data filtering. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 36, 79–84.

Tsai, V.J.D., 1993. Delaunay triangulations in TIN creation: an overview and a linear-time algorithm. Int. J. Geogr. Inf. Syst. 7, 501–524. http://dx.doi.org/10.1080/02693799308901979.

Ulusoy, A.O., Black, M.J., Geiger, A., 2017. Semantic multi-view stereo: jointly estimating objects and voxels. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4531–4540. https://doi.org/10.1109/CVPR.2017.482.

Volpi, M., Tuia, D., 2017. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 55, 881–893.

Yan, M., Blaschke, T., Liu, Y., Wu, L., 2012. An object-based analysis filtering algorithm for airborne laser scanning. Int. J. Remote Sens. 33, 7099–7116. http://dx.doi.org/10.1080/01431161.2012.699694.

Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions. In: ICLR, p. 13.

Zhang, L., Zhang, L., Du, B., 2016a. Deep learning for remote sensing data: a technical tutorial on the state of the art. IEEE Geosci. Remote Sens. Mag. http://dx.doi.org/10.1109/MGRS.2016.2540798.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016b. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. Remote Sens. 8, 501. http://dx.doi.org/10.3390/rs8060501.