remote sensing

MDPI

Article

A Novel Object-Based Deep Learning Framework for Semantic Segmentation of Very High-Resolution Remote Sensing Data: Comparison with Convolutional and Fully Convolutional Networks

Maria Papadomanolaki [1,2,*], Maria Vakalopoulou [2] and Konstantinos Karantzalos [1]

[1] Remote Sensing Laboratory, National Technical University of Athens, 15780 Zographos, Greece; karank@central.ntua.gr

[2] Centre de Vision Numérique, CentraleSupélec, INRIA, Université Paris-Saclay, 91190 Gif sur Yvette, France; mariavakalopoulou@gmail.com

* Correspondence: mar.papadomanolaki@gmail.com

Received: 24 December 2018; Accepted: 15 March 2019; Published: 21 March 2019

check for updates

Abstract: Deep learning architectures have received much attention in recent years demonstrating state-of-the-art performance in several segmentation, classification and other computer vision tasks. Most of these deep networks are based on either convolutional or fully convolutional architectures. In this paper, we propose a novel object-based deep-learning framework for semantic segmentation in very high-resolution satellite data. In particular, we exploit object-based priors integrated into a fully convolutional neural network by incorporating an anisotropic diffusion data preprocessing step and an additional loss term during the training process. Under this constrained framework, the goal is to enforce pixels that belong to the same object to be classified at the same semantic category. We compared thoroughly the novel object-based framework with the currently dominating convolutional and fully convolutional deep networks. In particular, numerous experiments were conducted on the publicly available ISPRS WGII/4 benchmark datasets, namely Vaihingen and Potsdam, for validation and inter-comparison based on a variety of metrics. Quantitatively, experimental results indicate that, overall, the proposed object-based framework slightly outperformed the current state-of-the-art fully convolutional networks by more than 1% in terms of overall accuracy, while intersection over union results are improved for all semantic categories. Qualitatively, man-made classes with more strict geometry such as buildings were the ones that benefit most from our method, especially along object boundaries, highlighting the great potential of the developed approach.

Keywords: earth observation; satellite data; machine learning; convolutional neural networks; GEOBIA; object-based image analysis; superpixels; anisotropic diffusion

1. Introduction

Semantic segmentation has received much research and development effort since it plays an important role in many critical computer vision tasks such as scene understanding, pattern recognition, object detection and tracking, etc. Several approaches have been adopted in order to improve classification results and create powerful, generic models independent of the training dataset. Currently, deep learning methods deliver state-of-the-art results in numerous image classification benchmark datasets based on mainly two different architectures. In particular, the patch-based methods [1,2] are based on convolutional networks (CNNs) [3–5] that receive as input fixed-size patches centered on each image pixel and thus, every single pixel is represented by the corresponding image region of this specific patch. Although such models are able to perform quite well, especially for sparse annotated datasets [6], they require much computational power which sometimes exceeds the capacity of available resources [7]. The second architecture is based on fully convolutional networks (F-CNNs) [8], which consist only of convolutional layers. Unlike patch-based architectures, they can deliver dense predictions as they do not contain fully connected layers with fixed size dimensions. Several F-CNNs have been proposed in the literature [9,10] and currently outperform other approaches in several benchmarks. However, there are still significant challenges towards the effective detection of objects with specific geometry, a detection that retains and follows object edges and boundaries.

In the remote sensing community, object-based image analysis (also known as geographic object-based image analysis) employs a classification procedure based on image objects (i.e., image segments), ameliorating results for numerous mapping tasks [11]. The terms object, segment or superpixel refer to a particular image region which can be enclosed in a polygon while all included pixels have common attributes (e.g., spectral values) and ideally belong to the same semantic category. Object-based approaches have delivered quite promising results in many applications especially in cases with very/ultra high-resolution data and when combined with knowledge-based and/or other machine learning frameworks [12,13]. However, object-based frameworks are still employing mainly shallow, kernel-based classifiers or integrating superpixel information during a pre-/post-processing step.

To this end, in this study, we designed, developed and validated an object-based, deep-learning framework that can integrate object representations in the neural network independent of the employed deep architecture for the given semantic segmentation task. Under such a framework, an additional loss quantity penalizes the pixels included in each superpixel to have the same semantic label. More specifically, the contributions of this work are twofold. Firstly, we propose a novel, object-based semantic segmentation approach based on deep neural networks, which can exploit information from object representations and constrain accordingly the predictions. According to our knowledge it is the first time that a generic object-specific loss for F-CNNs is presented. Our goal is to constructively combine the networks' rich feature representations with object-based information derived from groups of pixels and investigate the potentials by comparing the results with plain deep architectures. Secondly, we present a thorough experimental design, and the comparison and validation on several state-of-the-art deep learning models from the two dominating architectures, i.e., CNNs and F-CNNs, and the proposed object-based one on publicly available remote sensing datasets.

2. Related Work

2.1. Convolutional Networks for Semantic Segmentation (Patch-Based Learning)

Patch-based models can extract complicated features by combining spectral and spatial information at the same time. Historically, these were the first architectures adapted from the remote sensing community for the semantic segmentation of very high resolution datasets [14,15]. In [16], the authors examined three different ways of exploiting multiple convolutional architectures (PatreoNet [17], AlexNet [4], CaffeNet [18], GoogLeNet [19], VGG ConvNets [5], and OverFeat ConvNets [20]): (i) training the architectures from scratch using only the dataset of interest; (ii) employing pre-trained convolutional networks and fine tuning them according to the dataset of interest; and (iii) using a pre-trained convolutional network as a feature extractor and replacing the last softmax layer with an SVM classifier. Various patch-based convolutional architectures (DenseNet121 [21], InceptionV3 [19], VGG19 [5], Xception [22], ResNet50 [23], and InceptionResNetV2) are also explored in [24] for the successful mapping of wetlands. In addition, different urban environment categories are detected in [25] by exploiting ResNet [23] and VGG [5]. Similar approaches have also been explored for crop identification from high resolution imagery [26]. Moreover, the proposed patch-based method in [27] employs an AlexNet-based pre-trained architecture integrated with Spatial Pyramid Pooling (SPP) and Side Supervision (SS) techniques. The former term implies the concatenation of all the convolved feature maps that are produced from the intermediate pooling layers. SS on the other hand indicates the enrichment of the classification decision mechanism with a convex strategy that performs intermediate supervision. This strategy forces the classification output to depend not only on the final layer outcome, but also on the intermediate parts of the network since objective functions are added to each hidden layer. Lastly, the authors of [28] improved the performance of convolutional architectures by enriching the standard RGB patch information with Local Binary Patterns (LBP) features [29]. This extra feature information is incorporated into the training process using late and early fusion techniques experimenting with two different schemes: VGG-M [30] and ResNet [23]. In the late fusion case, the RGB and LBP features are fed separately into two different network streams and later fused in the fully-connected layers. Conversely, in the early fusion case, the raw RGB and LBP features are concatenated in a single vector and then passed to the network for training.

2.2. Fully-Convolutional Networks for Semantic Segmentation (Pixel-Based Learning)

Fully convolutional networks currently do deliver state-of-the-art results in several semantic segmentation benchmark challenges. In particular, several successful segmentation approaches based on fully-convolutional encoder–decoder architectures [31–33] are presented in the DeepGlobe CVPR-2018 challenge, e.g., Iglovikov et al. [34] extended a semantic segmentation model to perform instance segmentation for building surfaces. In particular, the authors employed a U-Net-like [9] architecture where the encoder is replaced with the first five convolutional blocks of the WideResNet-38 network [35]. The model output is a two-channel volume, one channel related to the binary building–non-building mask, and the other related to touching borders of building instances. Similarly, Seferbekov et al. [36] dealt with the automatic multi-class land segmentation problem using a Feature Pyramid Network [37] whose encoder is based on the ResNet50 network [23].

In general, it seems that encoder–decoder architectures have been widely applied on very high resolution remote sensing datasets owing to their promising results. For example, in [38], variations of the fully convolutional SegNet [10] architecture are employed for semantic segmentation on the ISPRS (WGII/4) benchmark dataset. Similarly, Audebert et al. [39] exploited a multi-scale SegNet, which outputs classification maps at different resolutions, while early and late fusion techniques [40] related to the dataset's DSM (digital surface model) are also investigated. In addition, the authors of [41] based their experiments on an encoder–decoder U-Net-like [9] deep architecture to effectively distinct sea from land areas. Marmanis et al. [42] also explored the use of F-CNNs [8] by providing spectral and elevation information to an ensemble of CNNs.

Recently, the authors of [43] tried to tackle the segmentation problem and preserve object boundaries [44–46] by formulating a bidirectional network called RiFCN (Recurrent Network in Fully Convolutional Network). The forward stream of this network is based on the VGG-16 [5] architecture while the backward stream deconvolves the pooled feature maps, with each deconvolution taking as input not only the features produced by the forward pass, but also features from the previous deconvolution levels. Similarly, Marmanis et al. [44] proposed a fully-convolutional architecture where boundary information is integrated to the learning process. Specifically, color and elevation information is passed through a boundary-detection encoder–decoder architecture, which outputs a scalar image of boundary-likelihoods. Then, this scalar volume is concatenated with the original raw image and is given as input to different segmentation encoder–decoder models, the results of which are averaged to produce the final classification scores. Moreover, the authors of [47] proposed a two-step process involving the training of a F-CNN for building detection combining the predictions with a CRF to integrate information from the boundaries of the objects.

Continuing with analogous approaches, the authors of [48] utilized a fully-convolutional encoder-decoder framework where the encoder is based on the ResNet architecture [23] followed by atrous spatial pyramid pooling [49]. An additional multi-scale loss function applied on different scales of the produced features is also exploited and finally a superpixel-based dense conditional random fields framework is used as a post-processing step to smooth the results.

2.3. Object-Based Learning for Semantic Segmentation

Instead of exploiting raw image features extracted by a deep learning network, several recent studies try to use the object/super-pixel information during semantic segmentation tasks. Objects/super-pixels also serve the need of lower computational complexity, which is usually required since for deep convolutional networks millions of parameters must be tuned consuming significant GPU and CPU core hours. The authors of [50] enriched the objects' representations by including information from proximal, distant and global regions. The local region describes the pixels included in the segment while the proximal region incorporates the local region and is usually twice the object's size. The distant region represents an even larger surrounding part of the object and finally the global region describes the entire image scene. This enhanced feature representation, which describes better the different scale dependencies, is then given as input to a feedforward multilayer network. An asymmetric loss function is also considered in this network which balances the weights among frequent and non frequent classes. Recently, in [51], gridized superpixels were employed for the extraction of salient objects. Gridized superpixels resemble pixels, since they are similar in shape producing in this way a grid-like oversegmentation. Each segment is represented by its mean color value, thus resulting in much lower image dimensions. The corresponding binary ground truth images (salient–non-salient) are similarly processed by assigning the dominant label value in each segmented region. The encoded images are then given as input to a fully convolutional network with residual blocks and, eventually, the predicted output is reconstructed back to its original dimensions.

Object information has been also integrated during pre-/post-processing steps with promising results. Recently, the authors of [52] examined different segmentation algorithms for training data formulated as follows: patches of different sizes (32 × 32, 64 × 64 and 128 × 128) centered to each object are extracted and then reshaped to 228 × 228, while labels are given according to the dominating value inside each superpixel. All reshaped patches go through a pretrained AlexNet network to produce a final feature vector, which is then incorporated to a linear SVM classifier. During the testing phase, each segmented region is labeled with the class that the model predicted for the corresponding centered patch. Moreover, further experiments are conducted using also context features of neighboring segments included inside a certain radius for each superpixel. In a similar manner, the authors of [53] also adopted the preprocessing segmentation step, however they employed a different technique for extracting image patches. Superpixels replace pixels based on the assumption that the over-segmentation of an image produces superpixels that are very similar in shape and size (usually also rounded without following object geometry and boundaries, such as in [54]). This approach reduced drastically the amount of testing time by making the sliding-window process much more rapid. Reported performance accuracy is similar to the standard pixel-based sliding-window method.

3. Materials and Methods

3.1. The Developed Object-Based Learning Framework

In the aforementioned related studies, the object information is mainly integrated under a preprocessing or postprocessing manner. Therefore, the main motivation here was to design a deep learning framework that could efficiently integrate object representations. To do so, we included object-based constraints by incorporating elegantly simplified image representations as priors to an additional loss term in the deep neural network. In Figure 1, a representative graphical illustration of the developed approach is presented.

More specifically, let us define an image I_s represented by a set of patches $\{P_i\}$ for $i = 1, 2, \ldots, Z$ with corresponding dense ground truth annotation S_s including a set of labels $\{l_i\}$ for $i = 1, 2, \ldots, K$. Both the image and the corresponding ground truth include $s = 1, 2, \ldots, D$ pixels.

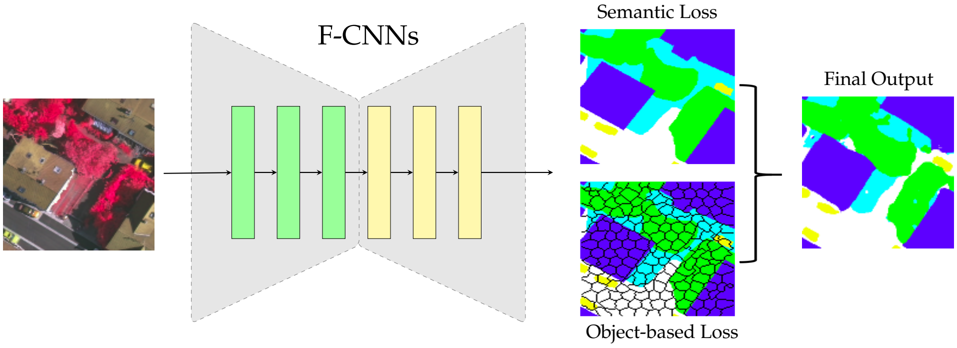

Figure 1. Illustration of the proposed object-based deep learning framework. The input image is fed to a F-CNN architecture with encoding (green) and decoding (yellow) layers. During the pixel-based optimization procedure, the semantic loss is calculated by comparing the network output with the reference data, while the object-based loss constrains the semantic labels to be the same with the dominant label inside each superpixel. The two losses are then combined together to produce the final segmentation map.

In our experiments, we dealt with a classification problem with more than two classes and thus we employed the multiclass cross entropy for the optimization of all the architectures

$$L_1 = -\sum_{l=1}^{K} y_{s,l} log(p_{s,l}), \tag{1}$$

where $y_{s,l}$ is a binary indicator that shows if class l is the correct answer for observation s and $p_{s,l}$ holds the probability that observation s belongs to class l.

Image objects can be generated by any segmentation algorithm to create regions with similar spectral or any other characteristics. To this end, let us consider, without loss of generality, a set of objects or segmented regions $\{C_i\}$ for $i = 1, 2, \ldots, M$. Assuming that all the pixels inside an object should have the same label, we formulate our loss as

$$L_2 = \sum_{i=1}^{M} \sum_{s \in C_i} \psi(\arg\max_{l}(p_{s,l}), l_d(C_i)), \tag{2}$$

where $\arg\max_l(p_{s,l})$ indicates the label with the maximum probability of s pixel of the object C_i and $l_d(C_i)$ the dominant category of the object C_i. As $\psi(\cdot)$, one can consider any distance function. In our case, and for all experiments, we used a Potts model such as

$$\psi(\arg\max_{l}(p_{s,l}), l_d(C_i)) = \begin{cases} 0 & \arg\max_l(p_{s,l}) = l_d(C_i) \\ c_1 & \arg\max_l(p_{s,l}) \neq l_d(C_i) \end{cases}, \tag{3}$$

where c_1 is a constant value that defines the penalty that will be given to each pixel of the object that does not belong to the dominating class. In all our experiments, we set $c_1 = 1$.

At this point, we should mention that the L_2 penalizes different predictions than the dominant label l_d and does not take into account the type of class that is different. The loss works similar to a smoothness term, enforcing homogeneity inside the regions of the objects C_i. Hence, the optimal label for each pixel is obtained by the weighted (w_1) sum of the classification and object-based losses as follows:

$$L_t = L_1 + w_1 \cdot L_2 \tag{4}$$

3.2. Implementation Details

In this section, we provide a brief overview of the plain patch-based and pixel-based methods that were employed to compare the results with the proposed framework and conduct a comprehensive evaluation. Moreover, implementation details, selected hyperparameters as well as the required computational training duration are presented for each model. All experiments were conducted using the PyTorch deep learning framework [55].

3.2.1. Patch-Based Learning

Three commonly used architectures were implemented, namely ConvNet, AlexNet and VGG-16, which provide one single l_i per P_i. In particular, ConvNet has a relatively simple architecture. There are 4 blocks of layers: 2 convolutional and 2 fully-connected [56]. The first convolutional layer includes 3 × 3 kernels, a stride of 1 and padding equal to 0. It is then followed by a ReLU layer and a maxpooling operation of 3 × 3 kernels and a stride of 2. The second convolutional block follows the same pattern and 2 fully-connected layers follow to produce the final classification product.

The AlexNet architecture is comprised of 8 blocks of layers following the same sequence: 5 convolutional and 3 fully-connected [56]. Giving some more details, the first convolutional layer applies 3 × 3 filters with a stride of 1, followed by a ReLU activation function and a max-pooling operation of kernels and stride equal to 3 × 3 and 2, respectively. The second convolutional layer follows the same pattern while the third and fourth lack the max-pooling operation. The fifth convolutional layer is again the same only this time the max-pooling filters are of size 2 × 2. After that, some fully-connected layers produce linear transformations while at the same time dropout layers mask part of the input rejecting samples that do not meet certain probability expectations, reducing at the same time the overfitting chances. More precisely, the probability threshold is equal to 0.5 instructing in this way the layers to reject 50% of input elements using Bernoulli distribution binary samples.

The VGG-16 architecture, which is much deeper [56] than the previous ones, consists of a block where the training data are subject to convolutional filters, batch normalization filters and a ReLU activation function. Data dimensions change only in terms of depth after being passed through this block since the convolution filters are of size 3 × 3 using a stride and padding of 1. Such a convolutional block appears 13 times throughout the whole architecture while dropout and max-pooling layers are also evenly distributed among the blocks.

Regarding the implementation details of the patch-based architectures the ConvNet, AlexNet and VGG-16 were optimized by the standard Stochastic Gradient Descent with a learning rate of 0.04, a momentum of 0.9, a weight decay of 0.0005 and a batchsize equal to 100.

3.2.2. Pixel-Based Learning

Different fully convolutional architectures were tested in this work and we present them in this section. Starting with the SegNet architecture [10], it includes an encoder and a decoder part. The input image is passed firstly through the encoder to be downsampled to a very low resolution, while at the same time a variety of features are calculated. In this specific case, the encoder consists of 5 convolution blocks. Each block computes consecutive convolutional, activation function and batch normalization operations. The convolutional operation involves filters of size 3 × 3, a stride of 1 and padding equal to 1. One max-pooling operation, which applies 2 × 2 filters with a stride of 2, is also included in each block. In this way, the original input volume is downsampled in half five times. Next, the input image is passed through the decoder where upsampling procedures take place in a symmetric manner. To be more specific, the decoder also consists of 5 convolution blocks, while the maxpooling operations are replaced by unpooling operations which bring the dataset back to its original size. At the end the model generates a heatmap where each pixel consists of n probability values, where n is the total number of semantic classes. It should be also noted that all the activation operations apply the rectified linear unit (ReLU) function to the input, apart from the last layer which produces the final segmentation output using a softmax activation.

The U-Net architecture [9] is also based on a downsampling-upsampling procedure but involves skip connections that concatenate feature maps between the encoder and the decoder. The employed

U-Net's encoder consists of five convolutional blocks. Each one applies 2 convolutions to the input in the form of Conv-Batch-ReLU. The convolution operations always involve 3 × 3 filters with both stride and padding being equal to 1. In the first convolutional block the depth is increased to 64, while the height and width dimensions remain unchanged. The following convolutional blocks reduce the dimensions in half using a 2 × 2 maxpooling operation and double the input depth, apart from the last block where the depth is unaltered. After the encoder, the decoder receives the low resolution volume to upsample it back to its original dimensions. For this purpose, four convolutional blocks of the same form are used, this time applying 2 × 2 upsampling operations. In addition, the resulted feature map of each upsampling operation is concatenated with the feature map of the symmetrical block existing in the encoder part. In this way, higher resolution information is combined with lower resolution information producing more sophisticated features and maintaining spatial knowledge. Finally, at the end of the model, a 1 × 1 convolution operation is applied to produce the final probability heat map for the existing classes.

Apart from the SegNet and U-Net models, the Fully Convolutional Network (FCN), which was proposed by Long et al. [8], was also employed. The authors proposed to replace the last fully-connected layers of various architectures with convolutional ones allowing in this way the model to produce heat maps instead of simple 1-D predictions. This is accomplished by exploiting deconvolutional layers, which are considered as a backward convolution that restores the downsampled volume. The actual upsampling is achieved by using a 32-stride deconvolution at the end of the model. Such models are independent of the input data dimensions and thus very flexible and easy to use. To boost the performance, a further skip layer was added, which collects information from intermediate parts of the model. In particular, apart from the 32-stride deconvolution, additional information from higher resolution layers is added by employing 16-stride and 8-stride layers. Then, the predictions of all upsampled skip layers are combined to produce the final classification map. The convolutionalized architectures were AlexNet, VGG and GoogLeNet. In all our experiments, we used the convolutionalized VGG-16 architecture with 32-stride and 16-stride deconvolutions. Firstly, the 32-stride fully convolutional model was trained using the VGG-16 weights as initialization. Then, the 16-stride network was in turn initialized with the parameters that the 32-stride scheme produced.

The term "pixel-based" implies that the network learns to provide some dense predictions for each pixel s of the patch P_i. The patch information is passed through the model which downsamples it to a low resolution and then upsamples it back to the original dimensions following an autoencoder scheme. Pixel-based architectures are fully convolutional and in this work we employed three commonly used ones for very high-resolution satellite data i.e., SegNet, U-Net and FCN-16. These ones were also employed and integrated in the developed object-based framework (as described in Section 3.1). Regarding the implementation details, in the case of SegNet, the weights were initialized using the pretrained VGG-16 ImageNet model. All pixel-based models were optimized by the Stochastic Gradient Descent with a batchsize, learning rate, momentum and weight decay equal to 10, 0.01, 0.9 and 0.0005 respectively.

3.2.3. Generation of Objects/Superpixels

One important component of the employed framework is the strategy for generating the objects. Regarding the image segmentation process, the choices are plenty; however, they were narrowed down significantly since the goal here was to select methods that respect the following criteria:

(i) Perform a nonlinear, anisotropic diffusion by taking also into account the fact that signal continuity in spectrum is, usually, more plausible than continuity in space

(ii) Take into account the fact that objects/segments/superpixels in the spatial directions should be enhanced, smoothed and elegantly simplified while their contours/edges/boundaries must remain perfectly spatially localized: no edge displacements, intensity shifts or spurious extrema should occur

(iii) Tackle only the kind of noise that never forms a coherent structure in both spatial and spectral directions

Among these methods, segments that have been produced through an effective, anisotropic data simplification process, able to integrate spatial and spectral information while respecting the aforementioned criteria are the ones based on scale space morphological filtering and anisotropic diffusion markers [57,58]. Such simplification techniques have been applied successfully for edge detection, image segmentation tasks [59] as well as smoothing, simplifying and reducing the dimensionality of hyperspectral data [60,61].

In particular, comparing with standard superpixel methods such as SLIC [54], the selected anisotropic morphological levelings (AMLs) can simplify the optical, multispectral data while preserve successfully image contours and object boundaries (Figure 2). More specifically, during the superpixel creation, although the parameterized color proximity and spatial proximity distances are combined, the resulting objects tend to present rather rounded shapes that do not approximate adequately and correctly object boundaries. This is mainly due to the fact that, for larger objects or objects with irregular shapes, the spatial distances outweigh color proximity, giving more relative importance to spatial proximity than spectral coherence.

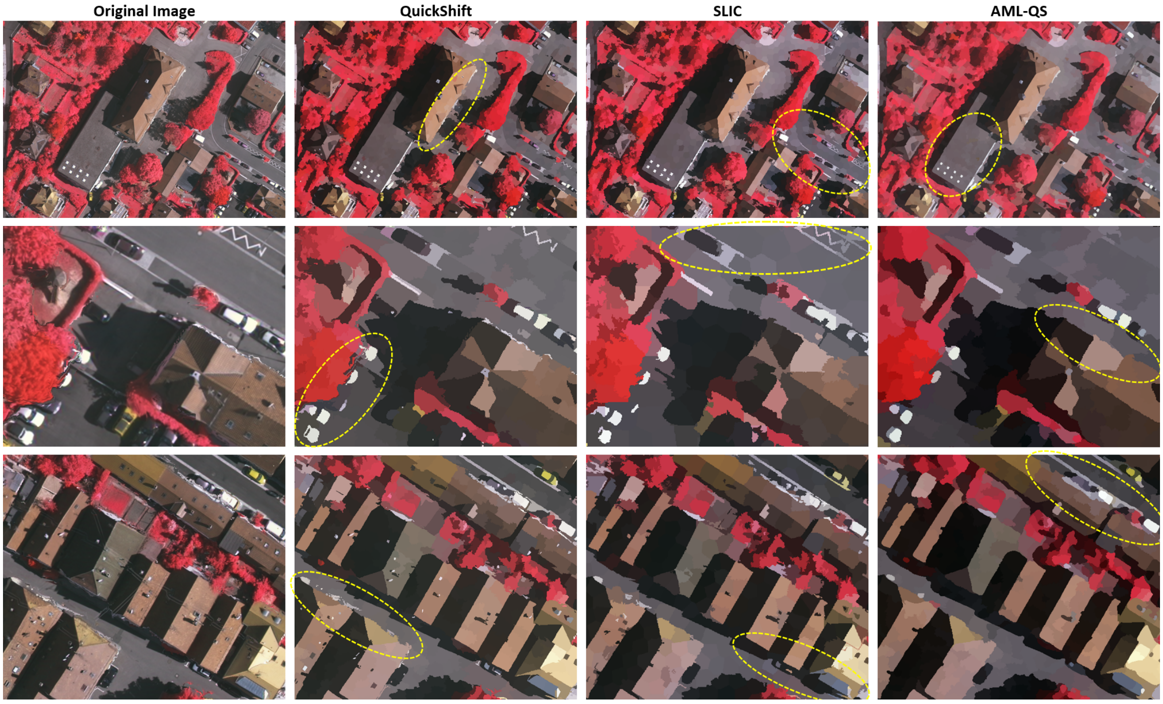

Figure 2. Comparing image objects derived from the Quickshift and SLIC algorithms with the proposed AML object-based approach (AML-QS). Image crops from the Vaihingen dataset are presented. Yellow dashed line: indicative areas where the proposed approach results into more semantically merged objects and more clear, accurate edges and boundaries.

This produces compact superpixels that do not adhere well to image boundaries. In the proposed processing pipeline, the dataset is simplified with AMLs in order to efficiently preserve the boundaries/edges of a variety of objects representing classes such as roads, roofs, trees, pavements, cars, other man-made objects, soil, vegetation, etc. Regarding the selected scale of filtering, although objects appear in different scale in the images and a standard object-based image analysis pipeline proposes a multiscale procedure [11,12], here we simplify the dataset at a relatively small scale creating a number of segments inside a single semantic object and letting the convolutional layers address the scale variance. From the simplified data, objects are then derived from the Quickshift segmentation algorithm [62] where each pixel is represented by a feature vector of its spectral information based on which the visually similar image regions are formed. The algorithm generates a forest of pixels based on which the objects/segments are created. In all our experiments, we assigned the values of 0.5, 2 and 12 to the ratio, kernelsize and maxdist parameters, respectively.

In Figure 2, the original raw image crops are presented in the left column along with the corresponding results from the Quickshift, SLIC and the proposed here AML-QS procedure. After a close look, one can observe that the objects derived from the AML-QS approach have been more semantically merged. In numerous cases, small objects (e.g., roof materials, roof objects, chimneys, asphalt lines, etc.) appear in a more spectral compact representation without spurious extrema. Moreover, the object boundaries and edges are represented more accurately, several small linear features that made object edges more noisy are not present while small objects with a specific shape (e.g., squared white rooftop skylights in the top row) have retained accurately boundaries and geometry.

For simplicity reasons, from now on we refer to the object-based approaches with the following abbreviations: OB_Snet for object-based SegNet, OB_Unet for object-based U-Net and OB_FCN for object-based FCN-16. The training processes for the object-based architectures were implemented using the exact same parameters as described in the plain pixel-based architectures (Section 3.2.2) in order to conduct an accurate comparison between them, i.e SegNet was trained with the same hyperparameters as OB_Snet; the same applies to OB_Unet and OB_FCN. Regarding the additional object-based loss function (Section 3.1), the value of w_1 was defined using grid search for each architecture. In particular, regarding the Vaihingen dataset, w_1 was equal to 2 for OB_Snet and OB_Unet, and 0.2 for OB_FCN. For the Potsdam dataset, w_1 was equal to 2 and 1 for OB_Snet and OB_Unet, respectively.

3.3. Dataset and Training Procedure

All our experiments were performed on the two publicly available ISPRS (WGII/4) benchmark datasets provided by Commission III of the ISPRS [63], depicting two different cities of Germany: Vaihingen and Potsdam. Both regions have been annotated with six different classes: *Impervious Surfaces*, *Buildings*, *Low Vegetation*, *Trees*, *Cars* and *Clutter*, which represents everything else that is not included in the other five classes. Regarding Vaihingen, it consists of 33 very high resolution images of average size 2494 × 2064 that have 3 available channels (InfraRed, Red, and Green) and a ground sample distance of 9 cm (i.e., the real ground value that corresponds to the distance of two adjacent pixel centers). In the case of Potsdam, 38 ortho-rectified images are available, with a size of 6000 × 6000 and a ground sample distance of 5 cm. Here, 4 different spectral channels are available: Red, Green, Blue and InfraRed. It should be noted that the different categories of this dataset are not proportionally balanced, i.e., some categories (e.g., *Buildings*) are much more common comparing to others (e.g., *Cars*). In Table 1, the proportion of each class in relation to the training images is presented.

Table 1. Proportion of each semantic category in the training datasets. The proportion corresponds to the number of pixels belonging to the specific class divided by the total number of pixels in the training datasets.

Category	Vaihingen	Potsdam
Impervious_Surfaces	0.293	0.299
Buildings	0.269	0.282
Low_Vegetation	0.194	0.209
Trees	0.224	0.144
Cars	0.013	0.017
Clutter	0.007	0.048

The Vaihingen dataset of the ISPRS benchmark consists of 16 tiles for training and 17 for testing purposes. For our experiments, we further divided the 16 training tiles into 14 for training (i.e., Areas 11, 13, 1, 21, 23, 26, 28, 30, 32, 34, 37, 3, 5 and 7) and 2 for validation (i.e., Areas 15 and 17). Regarding Potsdam, there are 24 training and 14 testing tiles. In a similar way, from the 24 training tiles, we used

17 for training (i.e., Areas 2_10, 3_10, 3_11, 3_12, 4_11, 4_10, 5_10, 5_12, 6_8, 6_9, 6_10, 6_11, 6_12, 7_7, 7_9, 7_11 and 7_12) and 7 for validation (i.e., Areas 2_11, 2_12, 4_10, 5_11, 6_7, 7_8 and 7_10).

For all the patch-based architectures, 29 × 29 patches were extracted randomly taking 1% of each class from every training image, resulting approximately in 1.1 million training and 38 thousand validation patches. All data were normalized by subtracting the mean and dividing by the standard deviation of the three available channels.

For the pixel-based architectures, patches of size 256 × 256 were extracted from the Vaihingen images using a step of 64 along both rows and columns forming in this way overlapping small regions. Approximately 13,800 training and 120 validation patches were created. All data were normalized before being processed by the networks via mean and standard deviation. In the case of Potsdam, patches were again of size 256 × 256 but this time they were formed with a step of 128 creating approximately 34,400 patches for training and 3700 for validation.

Training Time and Optimal Stop Points

All implemented models included in this work were trained using early stopping criteria. More precisely, the learning procedure was ceased when two requirements were satisfied. Firstly, the validation accuracy should not be increased during a specific number of epochs, which is called patience. Here, we used a patience of 10 epochs. Secondly, the difference between the training and validation accuracy should be minimum. If these criteria were met, then the training process was finished. All training tasks were assigned to the same GeForce GTX 1080 GPU. In Table 2, we provide all the relevant information, including computational costs and number of epochs for each of the architectures for both datasets. In general, the object based approaches were the more time demanding for training.

Table 2. The required training time in minutes as well as the optimal epoch that was picked for each architecture are presented for Vaihingen (**a**) and Potsdam (**b**). In bold are the proposed in this paper frameworks, i.e., OB_Snet, OB_Unet and OB_FCN.

(**a**)

	Mins/Epoch	**Optimal Epoch**	**Total Mins**
ConvNet	1	30	30
AlexNet	1.5	32	48
VGG-16	12	22	264
SegNet	20	51	1020
U-Net	20	31	620
FCN-16	20	50	1000
OB_Snet	30	42	1260
OB_Unet	30	38	1140
OB_FCN	30	40	1200

(**b**)

	Mins/Epoch	**Optimal Epoch**	**Total Mins**
SegNet	30	34	1020
U-Net	30	43	1290
OB_Snet	45	32	1440
OB_Unet	80	39	3120

3.4. Quantitative Evaluation Metrics

To assess the quality of the results, we employed four different evaluation metrics: Overall Accuracy, Precision, Recall and F1 score. They are all expressed through the calculated TP (True Positives), FP (False Positives) and FN (False Negatives). If we have a class l, then TP is the number of pixels that have been correctly classified as l. FP is the number of pixels that have been wrongly classified as l. Finally, FN represents the pixels that belong to l but the model has associated them to some other class.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad Overall Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Moreover, to further assess the performance of the developed approach, we employed the Intersection-over-Union (IoU) and the Hausdorff distance (HD), which indicate how close to the ground truth are the predicted objects. In Equation (5), A and B are two different data samples. Since in our case we have a multi-class segmentation problem, the IoU is calculated on each semantic category separately. Regarding the Hausdorff distance, it measures the maximum distance that exists between the predicted object and the ground truth. For two different data samples (i.e., A and B), the Hausdorff distance can be expressed as Equation (6), where a and b are the points of A and B, while $d(a, b)$ is the L2 norm.

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \tag{5}$$

$$HD(A, B) = \max_{a \in A}(\min_{b \in B}(d(a, b))) \tag{6}$$

4. Experimental Results and Discussion

4.1. Quantitative Evaluation

The developed object-based learning frameworks (i.e., OB_Snet, OB_Unet and OB_FCN) were applied to the Vaihingen dataset and compared with the performance of the patch-based (i.e., ConvNet, AlexNet and VGG-16) and pixel-based fully-convolutional (i.e., SegNet, U-Net and FCN-16) networks. The quantitative results regarding the calculated Overall Accuracy rates are presented in Figure 3 (left). The two object-based frameworks that resulted in higher accuracy rates (i.e., OB_Snet and OB_Unet) in Vaihingen were also applied to the Postdam dataset (Figure 3, right) as well. More specifically:

Patch-based Learning Frameworks: Figure 4 presents the quantitative results for ConvNet, AlexNet and VGG-16 for the Vaihingen dataset. The highest Overall Accuracy (OA) rate resulted from AlexNet (i.e., 83.10%), with ConvNet resulting into the second highest outcome (i.e., 82.50%) and VGG-16 giving the lowest OA rate (i.e., 79.79%). We can observe that, even though VGG-16 is the deepest architecture, the F1 rates were lower comparing with the other two models. Between ConvNet and AlexNet, the latter delivered higher accuracy rates, while the ConvNet achieved higher F1 rate only for the *Cars* class.

Pixel-based Learning Frameworks: Quantitative results from the fully-convolutional frameworks (SegNet, U-Net and FCN-16) are presented in Figure 5 (left). Generally, comparing with the patch-based frameworks, the F1 rates were higher apart from the case of the FCN-16 where certain classes (e.g., *Low_Vegetation* and *Trees*) were outperformed by the patch-based frameworks in terms of F1 score. The FCN-16 F1 rates were lower than SegNet for all class categories. The U-Net network achieved very similar results to SegNet with *Cars* reaching the highest F1 rate among the pixel-based methods. The overall accuracy rates were equal to 89.40, 86.38 and 88.32 for SegNet, FCN-16 and U-Net, respectively.

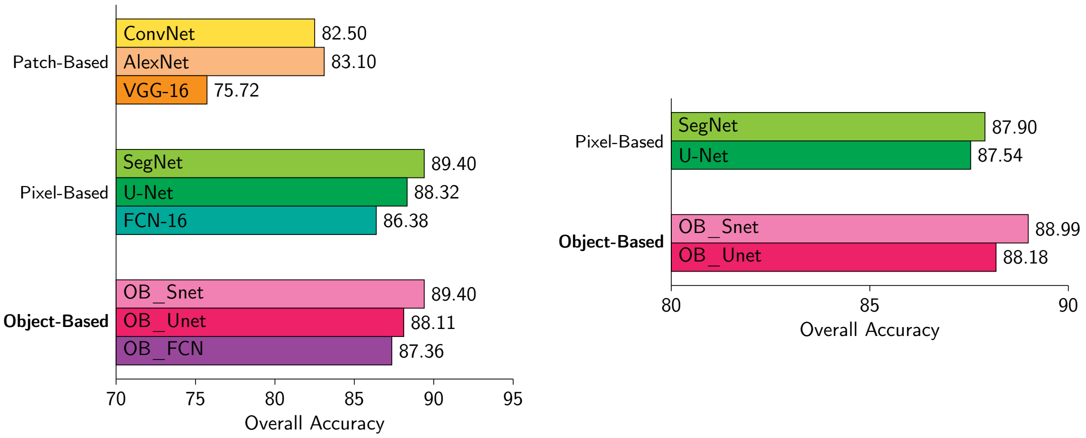

Figure 3. The resulting Overall Accuracy (OA) rates for the Vaihingen (**left**) and Postdam (**right**) datasets after the application of the developed object-based learning frameworks as well as the current state-of-the-art (either patch-based or pixel-based) learning networks.

ConvNet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.875**	0.042	0.060	0.007	0.012	0.003
building	0.120	**0.815**	0.046	0.004	0.011	0.003
low_veg	0.048	0.017	**0.753**	0.178	0.002	0.001
tree	0.009	0.001	0.122	**0.876**	0.001	0.000
car	0.096	0.069	0.031	0.003	**0.795**	0.006
clutter	0.269	0.357	0.065	0.005	0.093	**0.210**
Precision/Correctness	0.836	0.919	0.736	0.827	0.506	0.485
Recall/Completeness	0.875	0.815	0.753	0.867	0.795	0.210
F1	**0.855**	**0.864**	**0.744**	**0.846**	**0.618**	**0.293**

(a)

AlexNet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.855**	0.059	0.048	0.010	0.023	0.003
building	0.095	**0.849**	0.034	0.004	0.015	0.002
low_veg	0.045	0.026	**0.753**	0.169	0.006	0.001
tree	0.007	0.002	0.119	**0.870**	0.002	0.000
car	0.043	0.077	0.011	0.003	**0.860**	0.005
clutter	0.198	0.364	0.041	0.007	0.100	**0.290**
Precision/Correctness	0.859	0.897	0.762	0.832	0.390	0.580
Recall/Completeness	0.855	0.849	0.753	0.870	0.860	0.290
F1	**0.857**	**0.872**	**0.758**	**0.850**	**0.537**	**0.387**

(b)

VGG-16

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.852**	0.034	0.092	0.004	0.017	0.000
building	0.194	**0.736**	0.055	0.001	0.014	0.000
low_veg	0.027	0.018	**0.834**	0.117	0.004	0.000
tree	0.006	0.002	0.199	**0.792**	0.001	0.000
car	0.056	0.036	0.014	0.001	**0.892**	0.001
clutter	0.254	0.400	0.100	0.001	0.117	**0.129**
Precision/Correctness	0.793	0.919	0.672	0.873	0.450	0.828
Recall/Completeness	0.852	0.736	0.834	0.792	0.892	0.129
F1	**0.821**	**0.817**	**0.744**	**0.830**	**0.599**	**0.223**

(c)

Figure 4. Resulting Confusion Matrices for ConvNet, AlexNet and VGG-16 frameworks on the Vaihingen dataset.

Developed Object-based Framework: In Figure 5 (right) the quantitative results of the developed OB_Snet, OB_FCN, and OB_Unet are presented for the Vaihingen dataset. In particular, the OAs were 89.40, 87.36 and 88.11, respectively. Comparing with the aforementioned pixel-based frameworks, one can observe that the additional object-based loss managed to even slightly increase the accuracy rates for the FCN-16 case, produced equal OA for the SegNet case and resulted into a lower score by 0.2% for the U-Net case. This can be also viewed in Figure 3 where all the resulting OA rates for both datasets are presented. Indeed, one can observe that the object-based framework in all cases managed to outperform the pixel-based frameworks in the Potsdam dataset by more than 1%. The more effective performance of the object-based approach was also observed for all classes in the Potsdam dataset. In particular, the resulting confusion matrices (Figure 6) as well as the resulting F1 scores for the six different classes (Figure 7) demonstrate the improvement.

SegNet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.924**	0.028	0.037	0.007	0.003	0.000
building	0.036	**0.944**	0.017	0.002	0.001	0.000
low_veg	0.044	0.015	**0.822**	0.119	0.000	0.000
tree	0.012	0.002	0.096	**0.890**	0.000	0.000
car	0.137	0.043	0.005	0.002	**0.810**	0.002
clutter	0.352	0.274	0.011	0.003	0.048	**0.312**
Precision/Correctness	0.908	0.947	0.826	0.879	0.850	0.928
Recall/Completeness	0.924	0.944	0.822	0.890	0.810	0.312
F1	**0.916**	**0.946**	**0.824**	**0.885**	**0.830**	**0.467**

(a)

FCN-16

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.883**	0.047	0.041	0.022	0.005	0.001
building	0.083	**0.874**	0.035	0.006	0.002	0.000
low_veg	0.060	0.036	**0.721**	0.182	0.001	0.000
tree	0.023	0.006	0.096	**0.875**	0.000	0.000
car	0.339	0.070	0.011	0.010	**0.567**	0.002
clutter	0.293	0.387	0.017	0.011	0.053	**0.239**
Precision/Correctness	0.840	0.895	0.786	0.807	0.736	0.876
Recall/Completeness	0.883	0.874	0.721	0.875	0.567	0.239
F1	**0.861**	**0.884**	**0.752**	**0.840**	**0.641**	**0.376**

(b)

U-Net

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.930**	0.022	0.035	0.009	0.003	0.001
building	0.057	**0.914**	0.022	0.003	0.001	0.003
low_veg	0.048	0.014	**0.779**	0.158	0.000	0.001
tree	0.011	0.002	0.077	**0.910**	0.000	0.000
car	0.099	0.058	0.003	0.002	**0.835**	0.002
clutter	0.320	0.311	0.016	0.005	0.029	**0.319**
Precision/Correctness	0.891	0.951	0.833	0.848	0.867	0.661
Recall/Completeness	0.930	0.914	0.779	0.910	0.835	0.319
F1	**0.910**	**0.932**	**0.805**	**0.878**	**0.850**	**0.430**

(c)

OB_Snet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.933**	0.022	0.035	0.007	0.003	0.000
building	0.043	**0.933**	0.017	0.002	0.001	0.004
low_veg	0.048	0.016	**0.812**	0.123	0.000	0.000
tree	0.011	0.002	0.087	**0.900**	0.000	0.000
car	0.125	0.043	0.003	0.002	**0.827**	0.000
clutter	0.336	0.252	0.010	0.003	0.043	**0.356**
Precision/Correctness	0.902	0.953	0.835	0.876	0.858	0.728
Recall/Completeness	0.933	0.943	0.823	0.888	0.842	0.478
F1	**0.917**	**0.948**	**0.829**	**0.882**	**0.850**	**0.577**

(d)

OB_FCN

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.907**	0.043	0.026	0.019	0.005	0.000
building	0.070	**0.900**	0.023	0.006	0.001	0.000
low_veg	0.075	0.047	**0.685**	0.192	0.000	0.000
tree	0.025	0.009	0.070	**0.896**	0.000	0.000
car	0.344	0.075	0.006	0.007	**0.568**	0.000
clutter	0.360	0.361	0.007	0.010	0.095	**0.167**
Precision/Correctness	0.840	0.892	0.836	0.807	0.734	0.977
Recall/Completeness	0.907	0.900	0.685	0.896	0.568	0.167
F1	**0.872**	**0.896**	**0.753**	**0.849**	**0.641**	**0.286**

(e)

OB_Unet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.935**	0.022	0.029	0.009	0.003	0.002
building	0.061	**0.915**	0.018	0.003	0.001	0.002
low_veg	0.060	0.015	**0.753**	0.172	0.001	0.001
tree	0.012	0.002	0.068	**0.918**	0.000	0.000
car	0.081	0.057	0.003	0.003	**0.856**	0.000
clutter	0.273	0.324	0.012	0.008	0.048	**0.335**
Precision/Correctness	0.882	0.949	0.849	0.839	0.820	0.699
Recall/Completeness	0.935	0.915	0.753	0.918	0.856	0.335
F1	**0.907**	**0.932**	**0.798**	**0.877**	**0.838**	**0.452**

(f)

Figure 5. Resulting Confusion Matrices for the developed object-based learning frameworks (OB_Snet, OB_Unet and OB_FCN) on the Vaihingen dataset (**right**) and the corresponding ones for the state-of-the-art fully convolutional (SegNet, U-Net and FCN-16) networks (**left**).

SegNet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.921**	0.019	0.036	0.014	0.000	0.010
building	0.033	**0.938**	0.016	0.003	0.000	0.010
low_veg	0.032	0.006	**0.899**	0.054	0.000	0.009
tree	0.032	0.004	0.143	**0.816**	0.002	0.003
car	0.013	0.014	0.002	0.009	**0.948**	0.014
clutter	0.317	0.128	0.162	0.028	0.007	**0.358**
Precision/Correctness	0.895	0.945	0.799	0.888	0.944	0.649
Recall/Completeness	0.921	0.938	0.899	0.816	0.948	0.358
F1	**0.908**	**0.941**	**0.846**	**0.851**	**0.946**	**0.461**

(a)

U-Net

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.923**	0.018	0.033	0.013	0.000	0.013
building	0.041	**0.927**	0.014	0.004	0.000	0.013
low_veg	0.036	0.006	**0.895**	0.052	0.000	0.010
tree	0.034	0.004	0.145	**0.811**	0.002	0.004
car	0.013	0.019	0.001	0.010	**0.941**	0.016
clutter	0.315	0.130	0.162	0.025	0.008	**0.359**
Precision/Correctness	0.887	0.944	0.801	0.891	0.950	0.598
Recall/Completeness	0.923	0.927	0.895	0.811	0.941	0.359
F1	**0.904**	**0.936**	**0.846**	**0.849**	**0.945**	**0.449**

(b)

OB_Snet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.932**	0.015	0.030	0.013	0.000	0.010
building	0.033	**0.943**	0.012	0.003	0.000	0.009
low_veg	0.034	0.005	**0.895**	0.055	0.000	0.010
tree	0.027	0.004	0.115	**0.850**	0.002	0.002
car	0.015	0.007	0.000	0.009	**0.952**	0.015
clutter	0.319	0.108	0.164	0.022	0.012	**0.375**
Precision/Correctness	0.897	0.955	0.825	0.893	0.940	0.667
Recall/Completeness	0.932	0.943	0.895	0.850	0.952	0.375
F1	**0.914**	**0.949**	**0.859**	**0.871**	**0.946**	**0.480**

(c)

OB_Unet

Reference Predicted	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.920**	0.019	0.040	0.013	0.000	0.008
building	0.030	**0.938**	0.017	0.004	0.000	0.011
low_veg	0.028	0.006	**0.902**	0.054	0.000	0.009
tree	0.026	0.003	0.132	**0.834**	0.002	0.003
car	0.011	0.015	0.001	0.008	**0.954**	0.011
clutter	0.302	0.132	0.192	0.025	0.009	**0.339**
Precision/Correctness	0.904	0.945	0.797	0.889	0.945	0.654
Recall/Completeness	0.920	0.938	0.902	0.834	0.954	0.339
F1	**0.912**	**0.941**	**0.846**	**0.861**	**0.949**	**0.447**

(d)

Figure 6. Resulting Confusion Matrices for the developed object-based learning frameworks (OB_Snet and OB_Unet) in the Potsdam dataset (**right**) and the corresponding ones for the fully convolutional (SegNet and U-Net) networks (**left**).

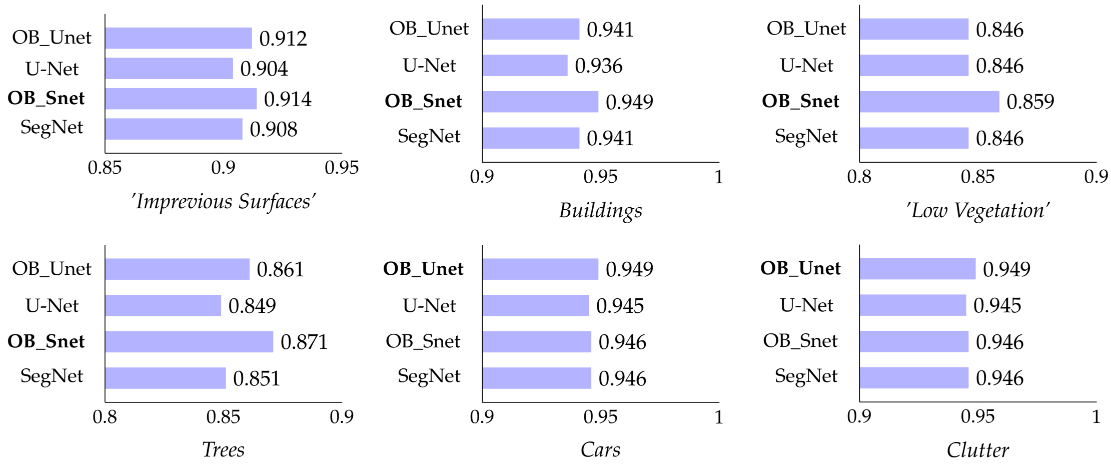

Figure 7. The resulting F1 scores from all considered methods for all semantic categories of Potsdam dataset. In all cases, the developed object-based learning framework managed to outperform the current state-of-the-art fully convolutional networks. In bold are the higher achieved F1 scores.

We can notice that all the resulting F1 rates were higher than the standard pixel-based fully convolutional networks. Especially for the *Buildings* and *Trees*, the object-based framework managed to produce better results. As far as the IoUs of the different semantic categories are concerned, one can observe in Table 3 (left) that object-based approaches have produced slightly better results in the case of *Impervious_Surfaces*, *Trees*, *Cars* and *Clutter* for the Vaihingen dataset. The highest IoU rates have been achieved by SegNet and OB_Snet, while U-Net and OB_Unet seem to have detected much more successfully the *Clutter* category. Regarding FCN-16 and OB_FCN, one can notice from Table 3 (left) that they delivered the lowest IoU values. Comparing with Vaihingen, in the Potsdam dataset, the developed framework performed more effectively, probably because it managed to exploit more efficiently the additional spectral information (i.e., the blue channel) that was available. This is also obvious from the IoU values that were produced from the testing images (Table 3, right) since the highest IoU for each semantic category resulted from the proposed object-based frameworks. Specifically, OB_Snet has achieved the most successful rates for *Impervious_Surfaces*, *Buildings*, *Low_Vegetation*, *Trees* and *Clutter*, while OB_Unet had higher rates in the *Cars* case.

Table 3. Intersection-over-Union (IoU) results for the Vaihingen (**left**) and Potsdam (**right**) datasets. The overall IoU value for each semantic category is calculated by adding the IoUs of all the testing images and dividing by their number.

	Vaihingen						**Potsdam**			
Model / **Category**	**SegNet**	**OB_Snet**	**U-Net**	**OB_Unet**	**FCN-16**	**OB_FCN**	**SegNet**	**OB_Snet**	**U-Net**	**OB_Unet**
Impervious_Surfaces	78.38	**78.64**	77.74	77.53	74.47	76.13	79.50	**80.27**	79.01	80.21
Buildings	**85.85**	85.41	83.74	83.83	78.69	80.45	86.70	**88.07**	85.74	86.79
Low_Vegetation	**63.13**	63.10	60.07	60.50	57.72	58.09	68.73	**70.36**	68.81	68.77
Trees	73.45	**74.09**	72.21	72.15	70.94	72.71	69.58	**72.37**	69.43	70.87
Cars	62.20	**63.43**	64.59	64.07	47.96	49.81	81.05	80.87	81.28	**81.69**
Clutter	0.00	0.00	8.22	**9.66**	7.13	0.00	24.14	**25.92**	24.16	24.04

4.2. *Qualitative Evaluation*

For the qualitative evaluation, three figures demonstrating the results of the patch-based (Figure 8: ConvNet, AlexNet and VGG-16), the fully convolutional pixel-based (Figure 9: SegNet, U-Net and FCN-16) and the developed object-based deep learning frameworks (Figure 10: OB_Snet, OB_Unet and OB_FCN) on the Vaihingen dataset are presented.

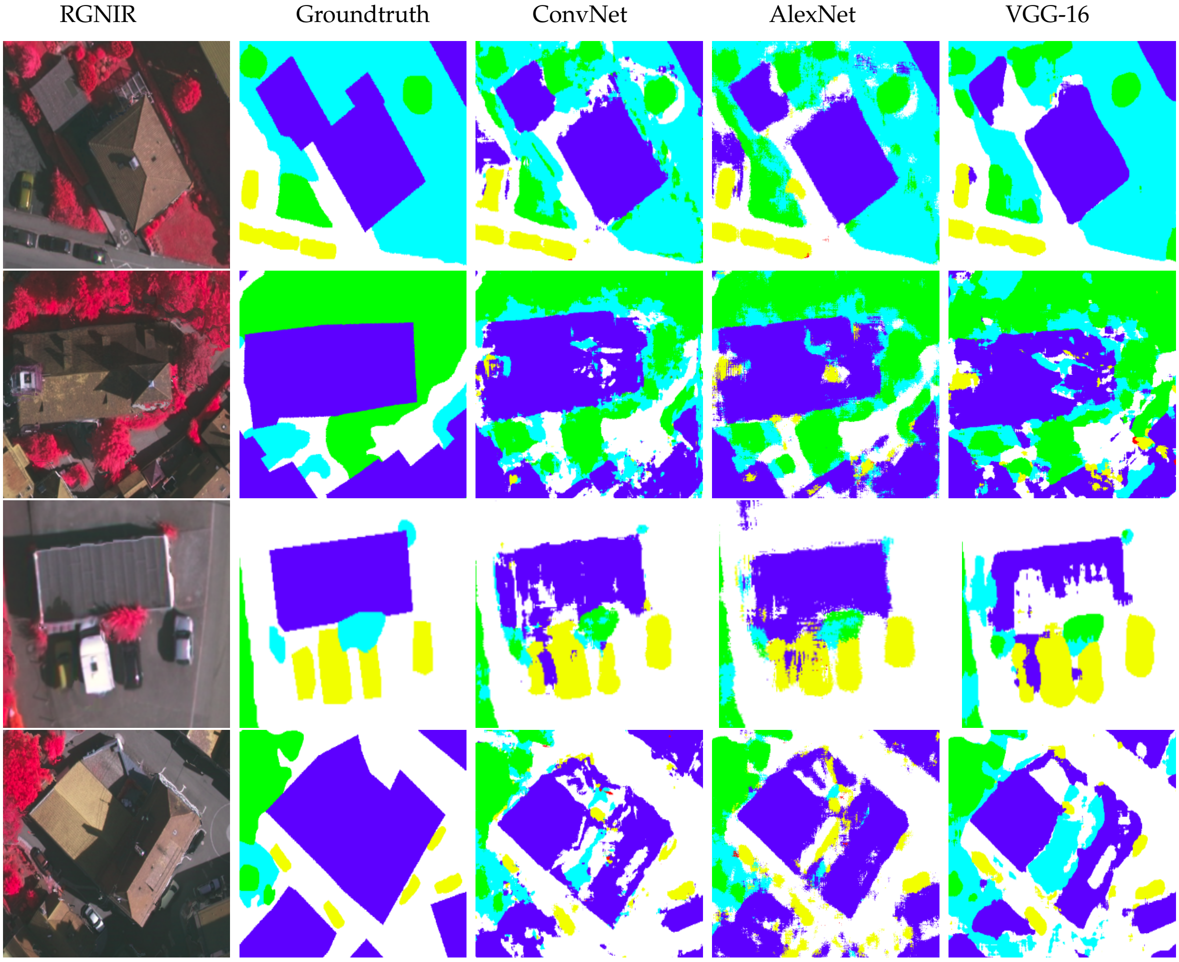

Figure 8. Experimental results from the ConvNet, AlexNet and VGG-16 convolutional networks on indicative regions from the Vaihingen dataset. Along with a false color composite (R-G-NIR), the corresponding ground truth is presented as well (White, *Impervious_Surfaces*; Blue, *Buildings*; Light Blue, *Low_Vegetation*; Green, *Trees*; Yellow, *Cars*; Red, *Clutter*).

In particular, even though patch-based architectures achieved relatively high quantitative results, the predicted map is noisy with significant gaps and fragmented outputs. In Figure 8, one can observe in indicative regions of the testing areas representative examples of the noisy results. It is obvious that the various classes are not well separated and boundaries are scattered and blurry. The corresponding results on the same zoomed regions that were derived from the fully convolutional pixel-based (SegNet, FCN-16 and U-Net) networks are presented in Figure 9. We can notice that the results are not so noisy and not so fragmented. However, certain objects have not been detected accurately in terms of object compactness, overall geometry and accuracy along their boundaries.

In Figure 10, the corresponding results from the same regions of the Vaihingen testing dataset are presented for the developed object-based learning approach. By comparing Figures 9 and 10, one can notice after a close look that in several cases the additional loss made the model more effective, for example in cases such as the depicted building in the second row of Figures 9 and 10 which was more accurately detected by the developed OB_Unet network. Generally speaking, comparing with the plain fully convolutional networks (in Figures 9 and 10 as well as in Figure 11 for the Potsdam testing dataset), one can observe that the resulting shapes and overall geometry were persevered comparing with the ground truth for the developed object-based networks. This was mainly due to the additional object-based priors that were integrated in the process which force and constrain the model to retain object shapes. Moreover, we observed that building boundaries derived from the object-based frameworks were more accurate, which was also justified by the HD metrics computed for the class *Buildings* on both datasets (Table 4).

Table 4. The calculated mean Hausdorff distance (HD) between the predicted and reference *Buildings* pixels for the Vaihingen and Potsdam datasets.

	Vaihingen						**Potsdam**			
Category \ **Model**	**SegNet**	**OB_Snet**	**U-Net**	**OB_Unet**	**FCN-16**	**OB_FCN**	**SegNet**	**OB_Snet**	**U-Net**	**OB_Unet**
Buildings	21.35	**20.89**	21.52	22.08	26.63	25.05	45.36	**41.87**	59.66	44.70

In particular, the HD metric calculates the maximum distance between the predicted *Buildings* pixels and the ones from the reference data. In the Vaihingen case, the object-based approaches resulted generally in better HD metrics with the object-based SegNet approach being closer to the reference geometry attaining a HD score equal to 20.89. Regarding Potsdam, both proposed object-based techniques performed better that the standard ones, especially if one compares the standard U-Net (with a score of almost 60) with the proposed OB_Unet (with a score of less than 45). In particular, the best HD score was achieved by OB_Snet and was equal to 41.87.

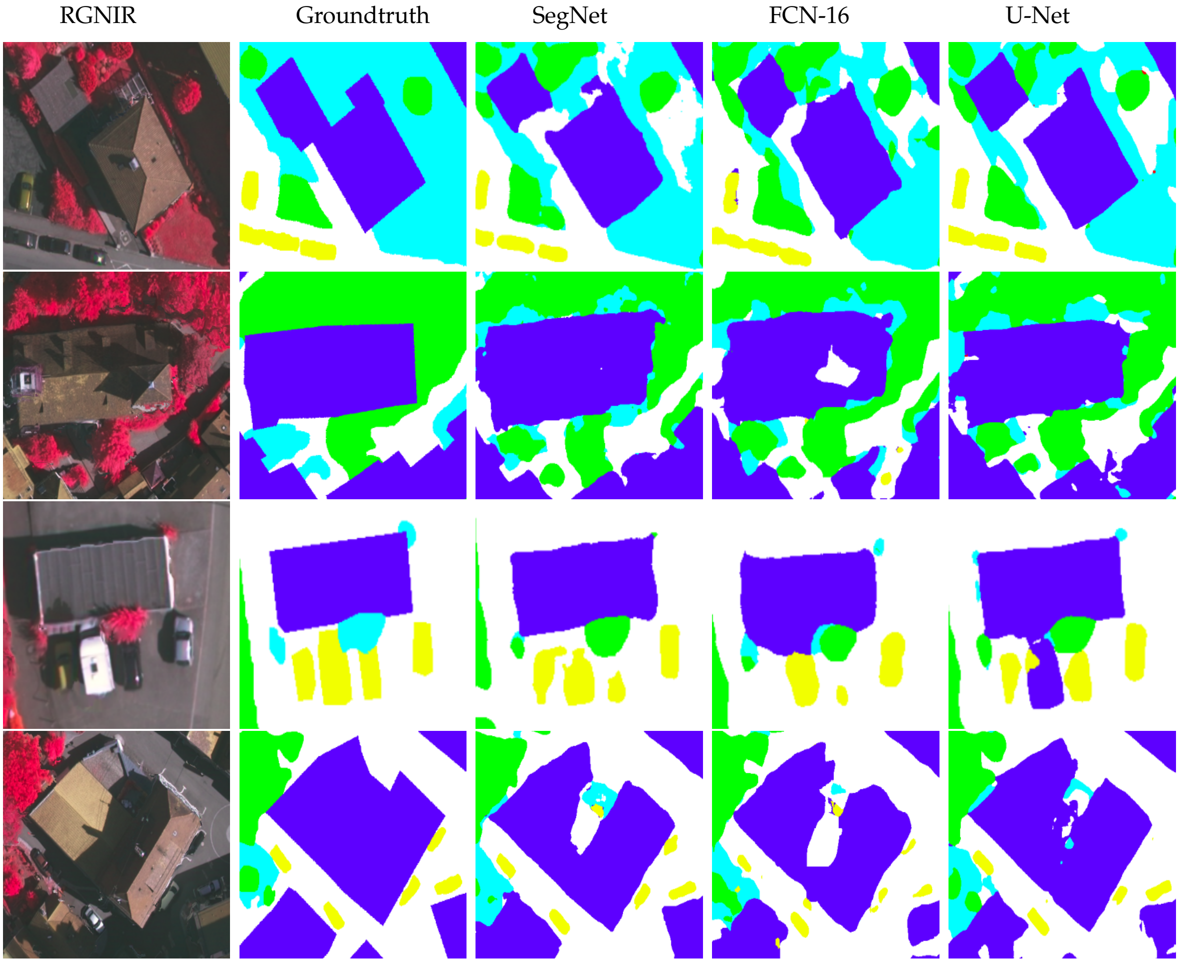

Figure 9. Experimental results from the SegNet, FCN-16 and U-Net fully convolutional networks on indicative regions from the Vaihingen dataset. Along with a false color composite (R-G-NIR), the corresponding ground truth is presented as well (White, *Impervious_Surfaces*; Blue, *Buildings*; Light Blue, *Low_Vegetation*; Green, *Trees*; Yellow, *Cars*; Red, *Clutter*).

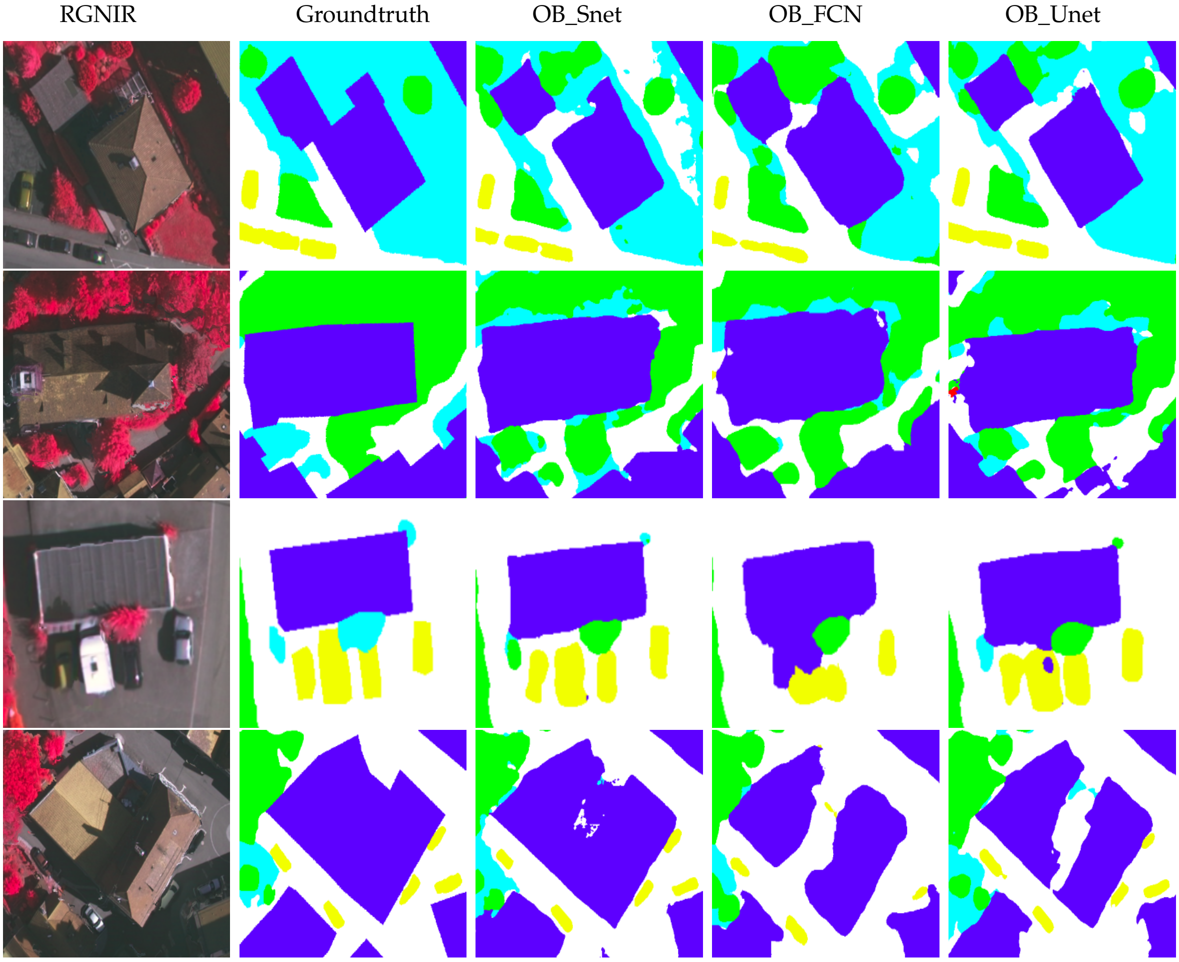

Figure 10. Experimental results from the developed OB_Snet, OB_FCN and OB_Unet networks on indicative regions from the Vaihingen dataset. Along with a false color composite (R-G-NIR), the corresponding ground truth is presented as well (White, *Impervious_Surfaces*; Blue, *Buildings*; Light Blue, *Low_Vegetation*; Green, *Trees*; Yellow, *Cars*; Red, *Clutter*).

4.3. Discussion

From both the aforementioned quantitative and qualitative evaluation results, the following outcomes can be highlighted. Generally speaking, for semantic segmentation tasks in very high resolution images, the fully convolutional frameworks (both pixel-based and object-based) are more robust and effective than the patch-based ones. More specifically, the overall accuracy levels raised by 5–10% depending on the model that was employed in every case. The developed object-based learning approach generally ameliorated the overall accuracy rates and F1 scores or improved the resulting accuracy rates per-class.

Moreover, as far as the Vaihingen dataset is concerned, the recall rates of *Impervious_Surfaces* and *Cars* were ameliorated (see also Figure 5) for all object-based models (OB_Snet, OB_FCN, and OB_Unet). In addition, the object-based OB_FCN network outperformed the standard FCN-16 by 1% in terms of overall accuracy (see also Figure 3) with all classes except *Clutter* achieving higher F1 rates.

Regarding the Potsdam dataset, object-based methods produced higher accuracy rates for both object-based models (i.e., OB_Snet and OB_Unet, Figure 3). Specifically, the resulting F1 scores were improved for the *Impervious_Surfaces*, *Buildings*, *Low_Vegetation* and *Trees* while they stayed almost the same for the remaining classes (Figure 7).

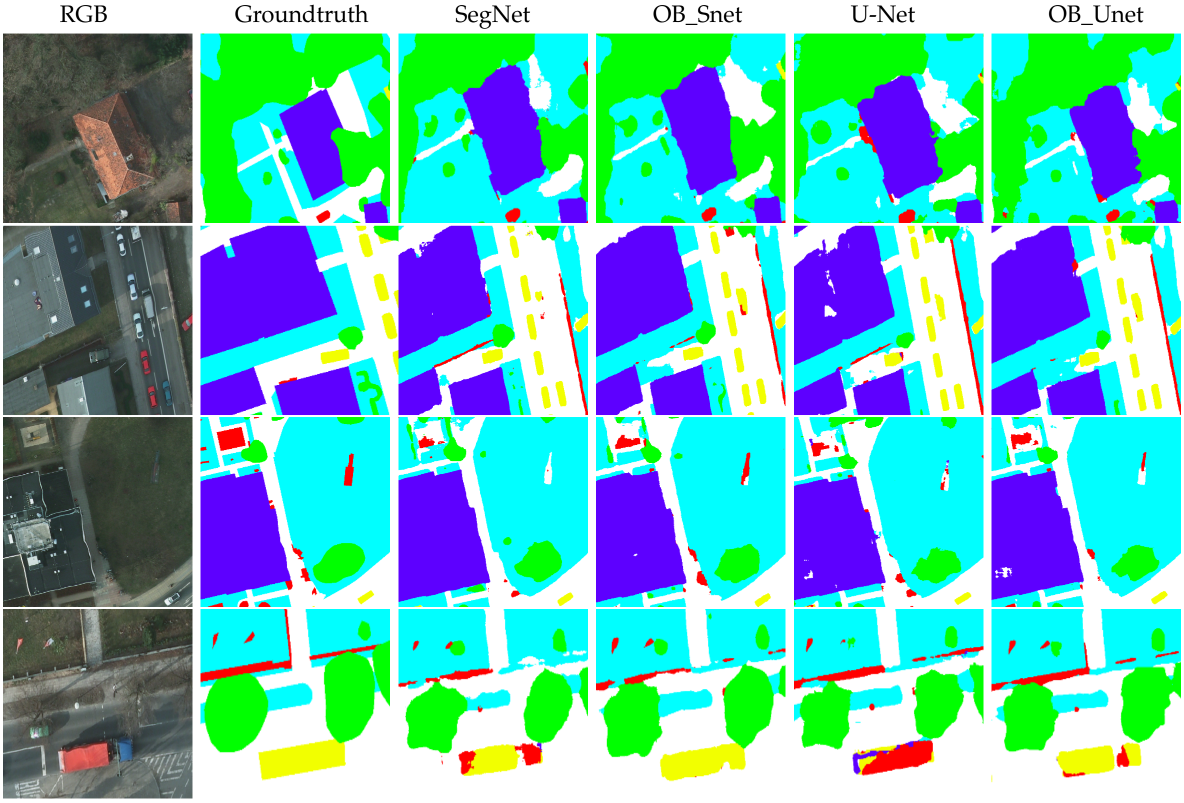

Figure 11. Experimental results from the developed OB_Snet and OB_Unet networks on indicative regions from the Potsdam dataset. The corresponding ground truth along with the results from the state-of-the-art SegNet and U-Net deep networks are presented as well (White, *Impervious_Surfaces*; Blue, *Buildings*; Light Blue, *Low_Vegetation*; Green, *Trees*; Yellow, *Cars*; Red, *Clutter*).

Judging by the achieved performance on the Vaihingen and Potsdam datasets, the more the spectral information is available in the datasets, the higher are the resulting accuracy rates for the object-based approach, which is logical and in accordance with the literature. However, for the object-based procedure this can be further justified by the fact that the AML implementation takes into account the fact that edges can occur also along the spectral dimension and not only the spatial image domain represented through e.g., *lab*/ *cielab* color spaces. Indeed, the proposed approach preserves both spatial as well as spectral edges and object boundaries.

Generally speaking, it should be also noted that regarding the size of the calculated objects and their scale, relatively smaller objects are more preferable than larger ones. In particular, if the average size of a single object is large, then it is highly likely that it actually consists of more than one semantic class. In such cases, the proposed approach cannot address properly the semantic segmentation task and most probably will assign the dominating label.

Comparison with Other State-Of-The-Art Methods on the ISPRS Dataset

Apart from the aforementioned comparison with the state-of-the-art networks, we also compared our results with other methods existing in the literature tested on the publicly available ISPRS dataset, which employ object-based approaches, preserving shapes and boundaries. For example, Marmanis et al. [44] exploited a fully-convolutional architecture that takes advantage of boundary-related information. Regarding the results of this specific method on the Vaihingen dataset, the OA and F1 rates are similar to the ones reported by our proposed object-based method even if they use additional information related to the publicly available Digital Surface Model of the ISPRS dataset. In fact, our *Cars* F1 rate is higher by 0.029 comparing to the method in [44], which indicates that the lack of DSM information for *Cars* degrades the quality of their outcome. Regarding

Potsdam, Marmanis et al. [44] only conducted experiments on the validation set. These results also indicate that even with additional information they are similar to ours. Furthermore, Liu et al. [46] attempted to preserve object boundaries by exploiting features from both VHR images and LiDAR data. Specifically, both data sources are passed through a fully convolutional network which produces probabilistic predictions. Then, a higher-order CRF receives the fused classification outputs and the final segmentation map is formulated through graph cut inference methods [64]. The OA of this method is 0.5% lower than ours for the Potsdam dataset. At the same time, our method attains higher F1 rates for all semantic categories, especially for *Cars* where our results are equal to 0.949 compared to the 0.928 of Liu et al. [46]. Moreover, comparing with the method in [48] (more detailed in the Introduction), which reports OA rates at 87.0% and 88.4% for the Vaihingen and Potsdam, respectively, the method proposed here outperforms the method in [48] in both datasets. In [45], the authors tried to preserve spatial boundary information by performing simultaneously semantic segmentation and edge detection using an encoder–decoder architecture. This is achieved by incorporating additional intermediate supervisions in the form of weighted losses related to the edge ground truth. Even though the achieved results are not based on the ISPRS Vaihingen testing benchmark, they are similar to our proposed method, with OA being 0.5% lower than the one we have presented. In the same way, Liu et al. [65] utilized an HourGlass-like architecture inspired by Newell et al. [66], the result of which is then post-processed by weighted belief propagation [67]. In this case, the Potsdam OA rates are higher than our method (89.42%), whereas, in the Vaihingen case, the best OA rate is equal to 88.82%, which is approximately 0.6% lower than our quantitative results.

5. Conclusions

In this study, an object-based deep learning framework was designed and developed based on the integration of AML simplification and a loss function that constrains the learning process with object priors. The developed approach is generic and can be integrated with different fully convolutional deep networks. The method ultimately can enforce pixels belonging to the same object to be classified on the corresponding dominant class retaining spectral and spatial characteristics. Based on the quantitative evaluation, higher accuracy rates, overall and per-class, were achieved comparing with the state-of-the-art. Qualitatively, the method also demonstrated more compact and less noisy outcomes while it retained more effectively the overall shape, geometry, object edges and boundaries. Among the future perspectives are the automation of the learning weights in the loss function as well as the integration of different simplification scales and image representations towards tackling more efficiently scale space issues.

Author Contributions: M.P. designed the methodology, implemented the software, performed experiments and validation, and wrote, edited, and reviewed the manuscript. M.V. implemented part of the methodology, performed the validation procedure, and edited and reviewed the manuscript. K.K. conceptualized the approach, designed the methodology, and edited and reviewed the manuscript.

Funding: Part of this research was funded by the Research Committee of the National Technical University of Athens (Scholarship Grant).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNNs	Convolutional Neural Networks
F-CNNs	Fully Convolutional Neural Networks
AML	Anisotropic Morphological Levelings
AML-QS	Anisotropic Morphological Levelings-Quickshift segmentation algorithm
OB_Snet	Object-based SegNet
OB_Unet	Object-based U-Net

OB_FCN	Object-based FCN-16
OA	Overall Accuracy
IoU	Intersection over Union
HD	Hausdorff Distance
SLIC	Simple Linear Iterative Clustering
VGG	Visual Geometry Group
DSM	Digital Surface Model
VHR	Very High Resolution
CRF	Conditional Random Field

References

1. Zhu, X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. doi:10.1109/MGRS.2017.2762307. [CrossRef]
2. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional Neural Networks for Large-Scale Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*. doi:10.1109/TGRS.2016.2612821. [CrossRef]
3. Springenberg, J.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. In Proceedings of the International Conference on Learning Representations (ICLR), Workshop Track, San Diego, CA, USA, 7–9 May 2015.
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1*; Curran Associates Inc.: Red Hook, NY, USA, 2012; NIPS'12, pp. 1097–1105.
5. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
6. Papadomanolaki, M.; Vakalopoulou, M.; Karantzalos, K. Patch-based deep learning architectures for sparse annotated very high resolution datasets. In Proceedings of the 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, 6–8 March 2017.
7. Volpi, M.; Tuia, D. Dense Semantic Labeling of Subdecimeter Resolution Images with Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 881–893. [CrossRef]
8. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
9. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
10. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
11. Blaschke, T.; Hay, G.J.; Kelly, M.; Lang, S.; Hofmann, P.; Addink, E.; Queiroz Feitosa, R.; van der Meer, F.; van der Werff, H.; Vancoillie, F.; et al. Geographic object-based image analysis: Towards a new paradigm. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 180–191. [CrossRef]
12. Tzotsos, A.; Karantzalos, K.; Argialas, D. Object-based image analysis through nonlinear scale-space filtering. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 2–16. [CrossRef]
13. Audebert, N.; Le Saux, B.; Lefévre, S. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sens.* **2017**, *9*, 368. doi:10.3390/rs9040368. [CrossRef]
14. Vakalopoulou, M.; Karantzalos, K.; Komodakis, N.; Paragios, N. Building detection in very high resolution multispectral data with deep learning features. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015.

15. Paisitkriangkrai, S.; Sherrah, J.; Janney, P.; Van-Den Hengel, A. Effective Semantic Pixel Labelling with Convolutional Networks and Conditional Random Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Boston, MA, USA, 7–12 June 2015.
16. Nogueira, K.; Penatti, O.A.; dos Santos, J.A. Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. *Pattern Recognit.* **2017**, *61*, 539–556. doi:10.1016/j.patcog.2016.07.001. [CrossRef]
17. Nogueira, K.; Miranda, W.O.; Santos, J.A.D. Improving Spatial Feature Representation from Aerial Scenes by Using Convolutional Networks. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI '15, Salvador, Brazil, 26–29 August 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 289–296. doi:10.1109/SIBGRAPI.2015.39. [CrossRef]
18. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. CAFFE: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678. doi:10.1145/2647868.2654889. [CrossRef]
19. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
20. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; Lecun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR2014), CBLS, Banff, AB, Canada, 14–16 April 2014.
21. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
22. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2016; pp. 770–778.
24. Mahdianpari, M.; Salehi, B.; Rezaee, M.; Mohammadimanesh, F.; Zhang, Y. Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sens.* **2018**, *10*, 1119. doi:10.3390/rs10071119. [CrossRef]
25. Albert, A.; Kaur, J.; Gonzalez, M.C. Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, Halifax, NS, Canada, 13–17 August 2017; ACM: New York, NY, USA, 2017; pp. 1357–1366. doi:10.1145/3097983.3098070. [CrossRef]
26. Karakizi, C.; Karantzalos, K.; Vakalopoulou, M.; Antoniou, G. Detailed Land Cover Mapping from Multitemporal Landsat-8 Data of Different Cloud Cover. *Remote Sens.* **2018**, *10*, 1214. [CrossRef]
27. Han, X.; Zhong, Y.; Cao, L.; Zhang, L. Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification. *Remote Sens.* **2017**, *9*, 848. doi:10.3390/rs9080848. [CrossRef]
28. Anwer, R.; Khan, F.; van de Weijer, J.; Molinier, M.; Laaksonen, J. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *138*, 74–85. Project: 112403, doi:10.1016/j.isprsjprs.2018.01.023. [CrossRef]
29. Ojala, T.; Pietikäinen, M.; Mäenpää, T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. doi:10.1109/TPAMI.2002.1017623. [CrossRef]
30. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. *Return of the Devil in the Details: Delving Deep into Convolutional Nets*; CoRR: Leawood, KS, USA, 2014; abs/1405.3531.
31. Filin, O.; Zapara, A.; Panchenko, S. Road Detection with EOSResUNet and Post Vectorizing Algorithm. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018.

32. Rakhlin, A.; Davydow, A.; Nikolenko, S. Land Cover Classification From Satellite Imagery with U-Net and Lovasz-Softmax Loss. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
33. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
34. Iglovikov, V.; Seferbekov, S.; Buslaev, A.; Shvets, A. TernausNetV2: Fully Convolutional Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
35. Bulò, S.R.; Porzi, L.; Kontschieder, P. In-place Activated BatchNorm for Memory-Optimized Training of DNNs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5639–5647.
36. Seferbekov, S.; Iglovikov, V.; Buslaev, A.; Shvets, A. Feature Pyramid Network for Multi-Class Land Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
37. Lin, T.Y.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
38. Audebert, N.; Le Saux, B.; Lefèvre, S. Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks. In Proceedings of the Asian Conference on Computer Vision (ACCV16), Taipei, Taiwan, 20–24 November 2016.
39. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]
40. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. In Proceedings of the 13th Asian Conference on Computer Vision, ACCV, Taipei, Taiwan, 20–24 November 2016.
41. Li, R.; Liu, W.; Yang, L.; Sun, S.; Hu, W.; Zhang, F.; Li, W. DeepUNet: A Deep Fully Convolutional Network for Pixel-Level Sea-Land Segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**. doi:10.1109/JSTARS.2018.2833382. [CrossRef]
42. Marmanis, D.; D. Wegner, J.; Galliani, S.; Schindler, K.; Datcu, M.; Stilla, U. Semantic Segmentation of Aerial Images with an Ensemble of CNNs. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-3*, 473–480. doi:10.5194/isprs-annals-III-3-473-2016. [CrossRef]
43. Mou, L.; Zhu, X. *RiFCN: Recurrent Network in Fully Convolutional Network for Semantic Segmentation of High Resolution Remote Sensing Images*; CoRR: Leawood, KS, USA, 2018; abs/1805.02091.
44. Marmanis, D.; Schindler, K.; Dirk Wegner, J.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an Edge: Improving Semantic Image Segmentation with Boundary Detection. *ISPRS J. Photogramm. Remote Sens.* **2016**, *135*. doi:10.1016/j.isprsjprs.2017.11.009. [CrossRef]
45. Liu, S.; Ding, W.; Liu, C.; Liu, Y.; Wang, Y.; Li, H. ERN: Edge Loss Reinforced Semantic Segmentation Network for Remote Sensing Images. *Remote Sens.* **2018**, *10*, 1339. doi:10.3390/rs10091339. [CrossRef]
46. Liu, Y.; Piramanayagam, S.; Monteiro, S.T.; Saber, E. Dense Semantic Labeling of Very-High-Resolution Aerial Imagery and LiDAR with Fully-Convolutional Neural Networks and Higher-Order CRFs. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1561–1570.
47. Vakalopoulou, M.; Bus, N.; Karantzalos, K.; Paragios, N. Integrating edge/boundary priors with classification scores for building detection in very high resolution data. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017.
48. Wang, Y.; Liang, B.; Ding, M.; Li, J. Dense Semantic Labeling with Atrous Spatial Pyramid Pooling and Decoder for High-Resolution Remote Sensing Imagery. *Remote Sens.* **2018**, *11*, 20. doi:10.3390/rs11010020. [CrossRef]
49. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]

50. Mostajabi, M.; Yadollahpour, P.; Shakhnarovich, G. Feedforward semantic segmentation with zoom-out features. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3376–3385.
51. Aytekin, Ç.; Ni, X.; Cricri, F.; Fan, L.; Aksu, E. Memory-Efficient Deep Salient Object Segmentation Networks on Gridized Superpixels. In Proceedings of the 20th IEEE International Workshop on Multimedia Signal Processing, MMSP 2018, Vancouver, BC, Canada, 29–31 August 2018; pp. 1–6.
52. Audebert, N.; Boulch, A.; Randrianarivo, H.; Le Saux, B.; Ferecatu, M.; Lefévre, S.; Marlet, R. Deep Learning for Urban Remote Sensing. In Proceedings of the Joint Urban Remote Sensing (JURSE), Dubai, UAE, 6–8 March 2017.
53. Gonzalo-Martin, C.; Garcia-Pedrero, A.; Lillo, M.; Menasalvas, E. Deep learning for superpixel-based classification of remote sensing images. In Proceedings of the GEOgraphic-Object-Based Image Analysis (GEOBIA), Enschede, The Netherlands, 14–16 September 2016.
54. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
55. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. In Proceedings of the NIPS-W, Long Beach, CA, USA, 4–9 December 2017.
56. Papadomanolaki, M.; Vakalopoulou, M.; Zagoruyko, S.; Karantzalos, K. Benchmarking Deep Learning Frameworks for the Classification of Very High Resolution Satellite Multispectral Data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, 83–88. doi:10.5194/isprs-annals-III-7-83-2016. [CrossRef]
57. Meyer, F.; Maragos, P. Nonlinear Scale-Space Representation with Morphological Levelings. *J. Vis. Comun. Image Represent.* **2000**, *11*, 245–265. doi:10.1006/jvci.1999.0447. [CrossRef]
58. Karantzalos, K.; Argialas, D.; Paragios, N. Comparing morphological levelings constrained by different markers. In Proceedings of the 8th International Symposium on Mathematical Morphology, Rio de Janeiro, Brazil, 10–13 October 2007; pp. 113–124.
59. Karantzalos, K.; Argialas, D. Improving edge detection and watershed segmentation with anisotropic diffusion and morphological levellings. *Int. J. Remote Sens.* **2006**, *27*, 5427–5434. [CrossRef]
60. Velasco-Forero, S.; Angulo, J. Morphological scale-space for hyperspectral images and dimensionality exploration using tensor modeling. In Proceedings of the 2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Grenoble, France, 26–28 August 2009; pp. 1–4.
61. Karantzalos, K. Intrinsic dimensionality estimation and dimensionality reduction through scale space filtering. In Proceedings of the 2009 16th International Conference on Digital Signal Processing, Santorini-Hellas, Greece, 5–7 July 2009; pp. 1–6. doi:10.1109/ICDSP.2009.5201196. [CrossRef]
62. Vedaldi, A.; Soatto, S. Quick Shift and Kernel Methods for Mode Seeking. In Proceedings of the European Conference on Computer Vision, ECCV, Marseille, France, 12–18 October 2008.
63. ISPRS. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 20 March 2019).
64. Kohli, P.; Ladický, L.; Torr, P.H. Robust Higher Order Potentials for Enforcing Label Consistency. *Int. J. Comput. Vis.* **2009**, *82*, 302–324. doi:10.1007/s11263-008-0202-0. [CrossRef]
65. Liu, Y.; Minh Nguyen, D.; Deligiannis, N.; Ding, W.; Munteanu, A. Hourglass-ShapeNetwork Based Semantic Segmentation for High Resolution Aerial Imagery. *Remote Sens.* **2017**, *9*, 522. doi:10.3390/rs9060522. [CrossRef]
66. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In Proceedings of the 14th European Conference Computer Vision, ECCV, Amsterdam, The Netherlands, 11–14 October 2016.
67. Murphy, K.P.; Weiss, Y.; Jordan, M.I. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI, Stockholm, Sweden, 30 July–1 August 1999.