Expert Systems With Applications 183 (2021) 115403

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Combining max-pooling and wavelet pooling strategies for semantic image segmentation

André de Souza Brito [a,*], Marcelo Bernardes Vieira [b], Mauren Louise Sguario Coelho de Andrade [c], Raul Queiroz Feitosa [d], Gilson Antonio Giraldi [a]

[a] Coordination of Applied and Computational Mathematics, National Laboratory for Scientific Computing, Petropolis, RJ, Brazil
[b] Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil
[c] Federal Technological University of Paraná, Ponta Grossa Campus, PR, Brazil
[d] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Keywords:
Convolutional neural networks
Semantic segmentation
Max pooling
Wavelet pooling
IRRG images

ABSTRACT

This paper presents a novel multi-pooling architecture generated by combining the advantages of wavelet and max-pooling operations in convolutional neural networks (CNNs), focusing on semantic segmentation tasks. CNNs often use pooling to reduce the number of parameters, improve invariance to certain distortions, and enlarge the receptive field. However, pooling can cause information loss and thus is detrimental to further operations such as feature extraction and analysis. This problem is particularly critical for semantic segmentation, where each pixel of an image is assigned to a specific class to divide the image into disjoint regions of interest. To address this problem, pooling strategies based on wavelets-operations have been proposed with the promise to achieve a better trade-off between receptive field size and computational efficiency. Previous works have confirmed the superiority of wavelet pooling over the traditional one in semantic segmentation tasks. However, we have observed in our computational experiments that the expressive gains reported from the use of wavelet pooling in other segmentation tasks were not observed in the scope of aerial imagery due to imprecision in the segmentation of image details. The combination of wavelet pooling and max-pooling, a solution not yet reported in the literature, can address that issue. Such gap observed in the pooling area motivated the two proposals that are the main contributions of this paper: (a) A new multi-pooling strategy combining wavelet and traditional pooling in a new network structure suitable for aerial image segmentation tasks; (b) Two-stream architectures using the traditional max-pooling and wavelet pooling as streams. These proposals were implemented using the Segnet, a known architecture for semantic segmentation. The computational experiments, based on the IRRG images from the Potsdam and Vaihingen data sets, demonstrated that the proposed architectures surpassed the original Segnet architecture's performance with results comparable to state-of-the-art approaches.

1. Introduction

In the last decade, the evolution of deep learning methods has led to a radical change in computer vision and image analysis tasks, in areas like medicine (Lu et al., 2018; Piccialli et al., 2021), remote sensing imaging (Chai et al., 2020; Duan et al., 2017), surrogate models (Alizadeh, Allen et al., 2020; Alizadeh et al., 2019; Jia et al., 2020), blockchain and applications (Soltanisehat et al., 2020), complex systems in energy generation and distribution (Alizadeh, Beiragh et al., 2020; Alizadeh, Soltanisehat et al., 2020), among others. Indeed, this is due to the generalization capabilities of deep neural networks in conjunction with the availability of large training data sets (Goodfellow et al., 2016). Analysis and processing of visual databases, in particular, were significantly improved by the application of a class of deep architectures called Convolutional Neural Networks (CNNs) (Khan et al., 2020). CNNs are composed of sequential blocks of convolution layers followed by a pooling layer that provides a hierarchical data representation. The image processing and analysis tasks that have incorporated CNN approaches branch out into image super-resolution (Wang et al., 2020), denoising (Tian et al., 2020), segmentation (Minaee et al., 2021), classification (Druzhkov & Kustikova, 2016), to name just a few.

* Corresponding author.
E-mail addresses: andre.brito@ice.ufjf.br (A. de Souza Brito), marcelo.bernardes@ufjf.edu.br (M.B. Vieira), mlsguario@utfpr.edu.br (M.L.S.C. de Andrade), raul@ele.puc-rio.br (R.Q. Feitosa), gilson@lncc.br (G.A. Giraldi).

https://doi.org/10.1016/j.eswa.2021.115403
Received 7 December 2020; Received in revised form 10 May 2021; Accepted 9 June 2021
Available online 12 June 2021

Given an input, a CNN can analyze its features from a low level in the first convolutional layers to a high level in the deeper ones (Boureau et al., 2010). The most used pooling strategies are max-pooling and average pooling that involve downsampling of the feature maps. The geometric downsampling is defined by the pooling hyperparameters, i.e., stride and filter size. Convolutions with stride can also be used for downsampling in deep networks, as used in the ResNet network (He et al., 2016).

Besides the downsampling and complexity reduction, the max-pooling operation improves recognition accuracy by focusing on relevant information. These are essential advantages, but, inherently, the max-pooling is a non-reversible operation (Zeiler & Fergus, 2014). Hence, there is information loss in the downsampling. Consequently, this can cause side effects that propagate distortions throughout image processing tasks based on encoder–decoder architectures, like segmentation, reconstruction, and denoising, for instance, Liu et al. (2019) and Ramanarayanan et al. (2020). Such duality, composed of max-pooling advantages and weaknesses, has motivated several works in the research area of pooling strategies in deep learning (Akhtar & Ragavendran, 2020). The latter reference classifies pooling techniques as value-based, probability-based, rank-based, and transformed domain pooling methods. Max-pooling is a value-based method. Probability-based and rank-based incorporates stochastic elements in the pooling operation. They are efficient strategies for the regularization of CNN models. However, those methods do not focus on the preservation of details during pooling.

Within the context of detail-preserving pooling approaches, Liu et al. (2019) proposes the multi-level wavelet CNN (MWCNN) model that belongs to the last category of pooling methods mentioned above. In this strategy, a discrete wavelet transform (DWT) replaces the pooling operations. As the DWT is invertible, image information and intermediate features are preserved during the downsampling scheme. Besides, both frequency and location information of feature maps are captured by DWT, which helps preserve detailed texture when using multi-frequency feature representation. The inverse discrete wavelet transform (IWT) with expansion convolutional layer is adopted to restore resolutions of feature-maps when the U-Net architecture (Ronneberger et al., 2015) is used as a backbone. Besides, the element-wise summation is chosen to combine feature maps, enriching feature representation. Their experimental results show the effectiveness of MWCNN in tasks such as image denoising, single image super-resolution, image artifacts removal, and object classification (Liu et al., 2019).

In the same line, Ramanarayanan et al. (2020) proposed a modification to the U-Net architecture to preserve and recover fine details. The proposed network is an encoder–decoder CNN that incorporates a wavelet packet transform with residual learning called WCNN. The authors also proposed a deep cascaded framework (DC-WCNN) which consists of cascades of WCNN and k-space data fidelity units to achieve high-quality multi-resolution reconstruction. In Williams and Li (2018) wavelet pooling is presented as an alternative to traditional neighborhood pooling. This method decomposes features up to second-level components and discards the first-level subbands to reduce feature dimensions. This approach avoids the problem of multiplication of subbands observed by Ramanarayanan et al. (2020). However, some information is lost as a consequence of subbands elimination.

In this context, using max-pooling with traditional neighborhood sampling has the advantage of computational efficiency, with the downside of possibly losing information. On the other hand, although appropriate care must be taken with computational complexity, wavelet theory opens several exploratory possibilities to represent signal details in CNNs due to the frequency and localization characteristics of the wavelet transform. However, the expressive gains reported from the use of wavelet pooling in other segmentation tasks were not observed in our computational experiments with aerial imagery. A reason for this problem is the imprecision in the definition of the boundaries of the objects. We claim that the combination of wavelet pooling and max-pooling, a proposal not yet reported in the literature, is a solution to address that issue. Such gap observed in the pooling area composes the underlying rationale for the research presented in this paper, which is steered by two questions: (a) How to combine wavelet pooling and max-pooling to put together their advantages for correctly segmentation of images such as aerial ones? (b) How to perform this task in the way of avoiding incorporating their disadvantages? The multi-pooling architecture proposed in this paper is a novel solution that addresses these two questions and fulfills the mentioned gap. It is significant for deep learning architectures because it allows putting together a reversible process, computed by DWT, with max-pooling ability to focus on relevant information using downsampling.

In this line, the main objectives of this paper are two-fold: Firstly, to propose a new pooling scheme that combines max-pooling and wavelet pooling, and secondly, to design and test a new encoder unity and CNN frameworks using the novel pooling strategy. To implement such architectures, we follow a network in network methodology (Lin et al., 2014) focusing on building micro neural networks incorporating convolutions and pooling strategies. For designing the encoder unity, we go back to the Resnet basic units that adopt convolution filters followed by batch normalization (BN) and rectified linear unit (ReLU) operations, named Conv-BN-ReLU in the following, besides a shortcut connection that is not applied in this work. Moreover, the Segnet (Badrinarayanan et al., 2017) and other architectures, like its variant named WSegnet (Li & Shen, 2020), apply Conv-BN-ReLU in series followed by a pooling layer. Inspired by these works, we propose a new encoder unity composes of two Conv-BN-ReLU in sequence, followed by our multi-pooling block.

In this way, the contributions of this paper are: (a) Proposal of a new multi-pooling scheme that combines max-pooling and wavelet pooling; (b) New encoder unit formed by our multi-pooling network plugged in the output of Conv-BN-ReLU blocks; (c) The Multi-pooling Segnet (MPSegnet) architecture using the new encoder unit; (d) Two-streams Segnet+MPSegnet and Segnet+WSegnet, where each network is trained separately; (e) Computational experiments using the Vaihingen and the Potsdam data sets (Rottensteiner et al., 2014), demonstrating the potential of new architectures for semantic segmentation with competitive results against state-of-the-art methods.

The proposed multi-pooling network has two branches, one with max-pooling and one with wavelet pooling. It applies 1×1 convolutions to efficiently change the output signal dimensions for each branch. The strategy of using 1×1 convolutions has been applied for dimensionality reduction in Liu et al. (2019) and to match dimensions in the implementation of shortcut connections in He et al. (2016). In our CNN framework, the role of 1×1 convolutions needs some clarification. Firstly, DWT is a reversible process, whereas max-pooling is not. Thus, we need to combine them carefully to avoid introducing redundancy in the network flow. On the other hand, the max-pooling operation discards irrelevant information through downsampling. This process hopefully improves feature extraction, which is a property out of the capabilities of DWT. Therefore, the key idea is to define a balanced learning process that can learn filter banks with the ability to reduce the 4-channels of DWT signals without losing DWT representation while intensifying the max-pooling result. The solution encountered was to apply 1×1 convolutions to process the DWT and the max-pooling outputs separately to yield two-channel signals that are concatenated at the end of the multi-pooling network. We remark that our multi-pooling network is different from the one presented in Wang et al. (2017), which concatenates max-pooling results obtained from different pooling neighborhoods.

Our method is applied to the semantic segmentation task, which seeks to accurately label each pixel of an image by assigning it to a specific class (Liu et al., 2017). Fully Convolutional Neural Networks (FCNN) are precursors of the current state-of-the-art methods for semantic segmentation, like U-Net and Segnet, as pointed out in Ulku and Akanguduz (2019). These networks are encoder–decoder architectures that are composed of two parts. The encoder gradually reduces the

spatial dimension with pooling layers, while the decoder gradually recovers the object details and the spatial dimension (Ulku & Akanguduz, 2019).

The U-Net applies 2×2 max-pooling operation with stride 2 for downsampling, while every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution (Ronneberger et al., 2015). The novelty of Segnet lies in how the decoder upsamples its lower resolution input feature maps. More specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This particularity eliminates the need for upsampling learning. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps (Badrinarayanan et al., 2017). The original Segnet architecture is composed of 13 Conv-BN-ReLU layers and five max-pooling layers in the encoder. Each encoder layer has a corresponding decoder layer, but the max-pooling is replaced by the upsampling layer (Badrinarayanan et al., 2017).

In our work, we explore the multi-pooling approach in two ways, using the Segnet architecture as a backbone. Firstly, our multi-pooling units are plugged in the Conv-BN-ReLU block's output to design the new encoder unit. The MPSegnet maintains the depth of the encoder to five multi-pooling layers. Secondly, inspired by Yazdizadeh et al. (2020), we propose a two-stream architecture composed of the original Segnet and the novel MPSegnet that are trained separately. The outputs of Segnet and MPSegnet are combined using an arithmetic average to get the final semantic segmentation. This two-stream network is called Segnet+MPSegnet henceforth. Besides, we test another simpler two-stream network composed of the original Segnet and a variation of it, named WSegnet (Li & Shen, 2020), generated by replacing the max-pooling to DWT and the unpooling scheme to the IWT. This hybrid network is called Segnet+WSegnet throughout this text. The motivation to propose this second two-stream architecture is to compare the first one (Segnet+MPSegnet) against a hybrid CNN that does not apply our multi-pooling strategy.

In the computational experiments, we conduct tests using Vaihingen, and the Potsdam data sets (Rottensteiner et al., 2014), considering the overall accuracy, the average F1 score, and the average intersection over union. In these tests, we aim to: (i) Put the original Segnet against the proposed MPSegnet, Segnet+MPSegnet, and Segnet+WSegnet; (ii) Perform visual analysis to discuss the capabilities and weaknesses of our proposals; (iii) Analyze the computational overhead when adopting the proposed multi-pooling strategy; (iv) Conduct comparisons with state-of-the-art techniques (Section 7.5). Item (i) shows that the MPSegnet is better or close to Segnet for all experiments performed. This fact emphasizes the capabilities of our multi-pooling proposal to preserve details near the objects' frontiers, as also confirmed in item (ii). Moreover, Segnet+MPSegnet and Segnet+WSegnet surpasses MPSegnet in most of these tests.

Besides the IRRG channels, the data sets applied also offer the digital surface model (DSM) and the blue channel. However, since the DSM and four channels images are not usually found in free aerial image data sets, we train our models using only the IRRG information. The state-of-the-art techniques considered in item (iii) use the blue and the DSM besides the IRRG. Also, we do not have access to the counterpart method codes. Despite these facts, we have decided to present our results against the counterpart ones found in the literature. We obtain competitive results of overall accuracy in the Vaihingen data set. In the Potsdam data set, the MPSegnet overcomes the proposed Segnet+MPSegnet and Segnet+WSegnet, concerning overall accuracy. When considering the counterpart methods of the literature, the MPSegnet surpasses only the Segnet and WSegnet networks.

Although Segnet+MPSegnet and Segnet+WSegnet give outstanding results in most tests, their computational cost is almost twice the MPSegnet's one. We emphasize that MPSegnet is the instantiation of the main ideas of this paper: multi-pooling network and its combination with Conv-BN-ReLU. The obtained accuracy shows that it is a promising proposal for semantic segmentation. Moreover, the computational complexity analysis of item (iii) showed that the overhead observed for MPSegnet due to the introduction of our multi-pooling strategy is not restrictive if compared with WSegnet.

The remaining text is organized as follows. In Section 2, we review works related to the foundations of this paper. Next, Section 3 presents concepts behind the proposed multi-pooling framework. Section 4 presents the contributions of this paper. The description of the data sets used and the samples preprocessing is performed in Section 5. The evaluation scores are presented in Section 6. The computational experiments are discussed in Section 7. Finally, Section 8 gives conclusions and future works.

2. Related work

The foundations of the present work belong to pooling strategies (Akhtar & Ragavendran, 2020), semantic segmentation (Ulku & Akanguduz, 2019), and incorporation of wavelets into CNNs (Bae et al., 2017; Han & Ye, 2018). Pooling operations are revised and classified in the Akhtar and Ragavendran (2020) survey as value-based, probability-based, rank-based, and transformed domain pooling methods. The most used techniques, like max-pooling and average pooling, are included in the former class composed of methods that seek the pooling region's values performing downsampling to choose a single activation. The approaches to perform this task encompass simple strategies that slide a window over the image and select for each pixel the maximum (max-pooling) or the average (average pooling) of intensities inside the neighborhood. Despite the low computational cost of both approaches, it is observed that the performance of max and average pooling is data-dependent, and the corresponding feature may not be efficient for classification problems (Boureau et al., 2010).

In the value-based class, the detail preserving pooling (Saeedan et al., 2018) shares with our work the care with information preservation. This method computes a weighted average in a neighborhood $k_H \times k_W$ of each pixel of the input image, aiming to get a more significant contribution for pixels with a more considerable difference in respect to the central one. The output is obtained by downscaling with a sampling rate of k_H in the horizontal dimension and k_W in the vertical one. Despite the pipeline of operations, the computational overhead is not high, and the practice shows that performance is similar to max or average pooling (Akhtar & Ragavendran, 2020).

Series multi-pooling (Wang et al., 2017) shares with our work the idea of concatenating pooling outputs. However, in Wang et al. (2017) only max-pooling is applied over regions R_0, R_1 and R_2, centered in the target pixel. Specifically, each region has a pre-defined size, and the feature map f_i is obtained by applying max-pooling $2 - i$ times with the final feature vector given by the concatenation $[f_0, f_1, f_2]$. Since the whole methodology is based on max-pooling, it is expected that it suffers from the max-pooling drawbacks at a certain level.

Probability-based pooling methods apply stochastic approaches for combining the activations inside a region. For instance, the stochastic pooling described by Zeiler and Fergus (2013) computes the normalized version of a positive activation field and interprets the result as weights used to combine the features maps inside the window to generate the pooling result. Rank-based pooling methods also use weight masks, but they are learned in the training stage. These masks are applied to combine the feature map information inside the pooling region (Akhtar & Ragavendran, 2020). Both rank and probability-based pooling present stochastic characteristics that improve network precision and avoid overfitting being effective regularizer schemes. However, they do not account for information loss during pooling operations.

The methodology proposed in this paper belongs to the transformed domain pooling methods. According to Akhtar and Ragavendran (2020), the techniques in this category work in the time or frequency domains using the concept of dynamic images, Fourier or

wavelet analysis. The dynamic image approach is introduced by Bilen et al. (2016), aiming to represent a video segment as a single image that is calculated through the concept of rank pooling inserted in a quadratic optimization problem. The corresponding solution is an image that encodes the temporal patterns in the video segment. This approach was applied to RGB images by Bilen et al. (2016) and to RGB+D data by Wang et al. (2018) as a pre-processing step for fine-tuning pre-trained CNNs in action recognition tasks. Despite the promising results presented in Bilen et al. (2016) and Wang et al. (2018), the utilization of this method for other application fields is questionable due to the nature of the video representation produced.

Taking into account the duality between spatial convolutions and element-wise product in the Fourier transform domain, Rippel et al. (2015) proposes a spectral representation for CNNs whose pooling is performed by simple truncation of the discrete Fourier transform. The obtained results can be visualized by taking the inverse transform to return to the original image domain. The method takes advantage of the speedup in the computation of convolutions in the Fourier domain, providing a powerful representation for modeling and training CNNs (Rippel et al., 2015). However, the known problem of Fourier analysis to handle localized signals in the frequency domain (no spectral locality) may bring artifacts when truncating the discrete Fourier transform. The application of wavelet concepts could mitigate this problem, as well as open new possibilities for remote sensing classification (Peker, 2021).

Wavelet pooling (Williams & Li, 2018) also applies the strategy of truncating an image transform to implement pooling operations. In this case, the input signal is decomposed into two levels of the DWT hierarchy, with pooling implemented by discarding the first-level subbands for dimensionality reduction. In another way, but also in the context of pooling implementation through wavelet transforms, authors in Liu et al. (2019) present a CNN architecture that interchanges first-level subbands yielded by the DWT decomposition with CNN blocks to generate the encoder path of a U-Net like architecture. The decoder side is built symmetrically, but replacing the DWT with the IWT.

The work (Han & Ye, 2018) tackles the problem of sparse-view tomography reconstruction through deep architectures. The limitations of U-Net for such application are disclosed by embedding deep operations in the framework of convolution framelets, and variants of U-Net emerge with better reconstruction quality than counterparts due to improvements in the pooling and unpooling layers by using high-pass branches to achieve the perfect reconstruction condition.

The incorporation of wavelets into CNNs, out of the context of pooling strategies, was also reported by Bae et al. (2017) for denoising applications, where the training data, formed by input and the clean label images, is decomposed into four subbands through the DWT. A new multi-channel training data is yielded in the wavelet domain by subtracting the input from the clean label images together with the input images. The proposed CNN has five modules, each with one bypass connection and three Conv-BN-ReLU layers. A similar methodology is developed by Guo et al. (2017) for generating high-resolution versions of low-resolution images.

Considering the above literature review, we can group the works to address the information loss in pooling operations as: (i) Strategies in the image domain based on traditional pooling operations (Saeedan et al., 2018; Wang et al., 2017); (ii) Dynamic image approach (Bilen et al., 2016; Wang et al., 2018); (iii) Fourier transform domain; (iv) Discrete wavelet transforms. The latter surpasses the issues of Fourier representations. Moreover, wavelet transforms offer a general framework that is not restricted in terms of application fields, a problem underwent by the dynamic image technique. However, the max-pooling operation discards irrelevant information, a characteristic out of the capabilities of DWT. Hence the combination of both max-pooling and DWT is a promising idea because it allows fusing the signal representation and reconstruction capabilities of DWT with the focus on significant features of max-pooling. Such combination is a gap in the CNN domain that our work addresses, aiming to improve pooling operations, which constitutes the first contribution of this paper. The obtained multi-pooling scheme is one of the building blocks of the encoder unity proposed that is incorporated in our MPSegnet architecture, the latter constructed using Segnet as the backbone. In the following sections, we will demonstrate the capabilities of our proposals for semantic segmentation of aerial images.

Semantic segmentation could be considered a pixel-wise classification problem in which each pixel is labeled with its object or its region class (Long et al., 2015). In this sense, Long et al. (2015) was the pioneer to train an FCNN end-to-end for pixel-wise prediction and supervised pre-training. In this architecture, the fully connected layer is replaced by the convolution layer for dense prediction. Likewise, the Segnet was proposed a year later by Badrinarayanan et al. (2017), aiming to learn in the encoder–decoder stack trained in a modular and fully supervised manner for pixel-wise labeling. A variation of Segnet, named WSegnet, was also proposed in Li and Shen (2020), generated by replacing the max-pooling to DWT and the unpooling scheme to the IWT.

Besides Segnet and WSegnet, other architectures are important for our work because they are state-of-the-art approaches for semantic segmentation applied in the same databases that we chose to perform our tests (Potsdam and Vaihingen). Specifically, the network used by Kampffmeyer et al. (2016) is an FCNN augmented with the cross-entropy loss function weighted with median frequency balancing to improve segmentation accuracy in the presence of imbalanced classes.

In Volpi and Tuia (2017), it is employed a downsampling-then-upsampling CNN architecture for semantic segmentation of ultra-high spatial resolution images. That work applies a convolution network for land-cover representations in the low-resolution spatial map. It then performs the upsample process through a deconvolution network to get back the original input patch size. On the other hand, the CNN architecture proposed by Liu et al. (2017) is also built using encoding and decoding stages but in an hourglass-shaped network (HSN) design. The architecture applies a composed inception module to replace common convolutional layers to achieve multi-scale receptive areas and skip connections with residual units to mitigate information loss due to the upsampling stage.

Dong et al. (2019) propose the concept of the downsampling dense block for obtaining context information and the upsampling dense block for restoring the original resolution. This end-to-end deep CNN, named DenseU-Net, is applied for pixel labeling in urban remote sensing images. A focal loss function weighted through the median frequency balancing is proposed aiming to increase the accuracy of the small classes. The DenseU-Net is explored in Dong et al. (2020) to build a siamese architecture (SiameseDenseU-Net) composed of two DenseU-Net modules to process, in parallel, both the orthophoto image and the corresponding normalized digital surface model. The generated feature vectors are fused to improve the segmentation capabilities of the network. Since a two-stream architecture is applied, we should compare its result with our Segnet+MPSegnet and Segnet+WSegnet proposed in this paper.

Our work differs from the approaches mentioned earlier in three main points: (i) instead of replacing the original pooling (Li & Shen, 2020), we combined it with the wavelet-based pooling strategy (DWT), generating a novel multi-pooling approach; (ii) differently from other multi-pooling schemes (Wang et al., 2017), we implement our multi-pooling system through a balanced learning process, based on a bank of filters learned by a set of 1×1 convolutions. This approach weighs the contribution of each pooling strategy before the combination stage, aiming to reduce the loss of information; and (iii) the proposed two-stream models combine two networks, trained separately, each one using different poolings instead of varying input data like performed by Dong et al. (2020), showing the viability of this strategy as an alternative for integrating multi-pooling structures in networks.

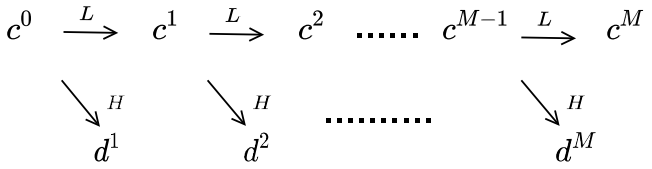

Fig. 1. DWT multiscale decomposition computed through Eqs. (2)–(3).

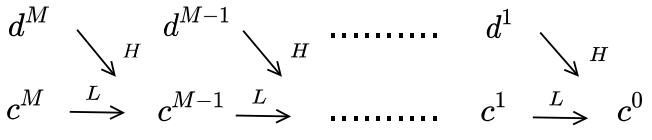

Fig. 2. IWT reconstruction of the original sequence c^0.

3. Theoretical background

In this section, we present the basic concepts behind the proposed multi-pooling framework that includes elements of the DWT for multiscale analysis, convolution theory in the context of CNNs, besides max-pooling, batch normalization, and ReLU activation function. Let us start with the concepts of DWT.

3.1. DWT in signal processing

We consider a signal f in the basic space of the multiscale analysis, named V_0, that is a subset of the space of square-integrable functions $L^2(\mathbb{R})$, as well as an orthogonal scaling function $\varphi \in V_0$. Hence f has the representation:

$$f(x) = \sum_{k\in\mathbb{Z}} c_k^0 \varphi(x-k), \tag{1}$$

with expansion coefficients $c^0 = \{c_k^0 \mid k \in Z\}$ in the space of square-summable sequences indexed in the integer set $\mathbb{Z}$, denoted by $l^2(\mathbb{Z})$. Let ψ denote the orthogonal wavelet corresponding to the scaling function φ, which generates an orthonormal basis:

$$\left\{\psi_{m,k} = 2^{-m/2}\varphi(2^{-m}\cdot - k) \mid m,k \in \mathbb{Z}\right\},$$

of the space $L^2(\mathbb{R})$. Now, we can compute the DWT through the scalar products and corresponding convolutions (Mallat, 1989):

$$c_k^m = \langle f, \varphi_{m,k}\rangle_{L^2} = \sum_{l\in\mathbb{Z}} h_l \langle f, \varphi_{m-1,2k+l}\rangle_{L^2} = \sum_{l\in\mathbb{Z}} h_{l-2k} c_l^{m-1}, \tag{2}$$

$$d_k^m = \langle f, \psi_{m,k}\rangle_{L^2} = \sum_{l\in\mathbb{Z}} g_l \langle f, \varphi_{m-1,2k+l}\rangle_{L^2} = \sum_{l\in\mathbb{Z}} g_{l-2k} c_l^{m-1}, \tag{3}$$

where $g, h \in l^2(\mathbb{Z})$ are convolution kernels and the sequences of coefficients d^m and c^m represent high-pass (H) and low-pass (L) versions, respectively, of the signal c^{m-1} in the $l^2(\mathbb{Z})$ space. In this context, we can perform M levels of decomposition of the initial sequence c^0 through the scheme depicted in Fig. 1. where $L\left(c^{m-1}\right) = c^m$ and $H\left(c^{m-1}\right) = d^m$ are formalized by expressions (2) and (3), respectively.

To compute the reconstruction of the initial sequence c^0, also named inverse discrete wavelet transform (IWT), after M levels of decomposition, it is a matter of taking the sequences of coefficients $\left\{c^M, d^m \mid m = 1,\dots,M\right\}$ and compute the scheme in Fig. 2:
that is implemented through the expression:

$$c_k^{m-1} = \sum_{l\in\mathbb{Z}} c_j^m h_{k-2l} + \sum_{l\in\mathbb{Z}} d_l^m g_{k-2l.}, \quad m = M, M-1, \dots, 1, 0. \tag{4}$$

The above development can be extended to $2D$ discrete signals represented in the space of square-summable sequences indexed in the integer set $\mathbb{Z}\times\mathbb{Z}$, denoted by $l^2(\mathbb{Z}\times\mathbb{Z})$. In this case, the wavelet theory (Meyer, 1992) shows that the DWT decomposition is implemented through four filters: a low-pass LL and three high-pass denoted as LH, HL, and HH. In the case of Haar wavelet, these four convolutional filters are given by the kernels:

$$f_{LL} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad f_{LH} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \tag{5}$$

$$f_{HL} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \quad f_{HH} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{6}$$

So, to compute one level of decomposition, we must perform the operations $\boldsymbol{x}_{LL} = \left(f_{LL} \otimes \boldsymbol{x}\right)\downarrow_2$, $\boldsymbol{x}_{LH} = \left(f_{LH} \otimes \boldsymbol{x}\right)\downarrow_2$, $\boldsymbol{x}_{HL} = \left(f_{HL} \otimes \boldsymbol{x}\right)\downarrow_2$, and $\boldsymbol{x}_{HH} = \left(f_{HH} \otimes \boldsymbol{x}\right)\downarrow_2$, where $\otimes$ represents the convolution operation, and $\downarrow_2$ represents the usual downsampling with factor 2. These combinations of convolution and downsampling implement the operations given by the expressions (2)–(3) in the space $l^2(\mathbb{Z}\times\mathbb{Z})$. For details about the results of these operations, see Liu et al. (2019). We shall observe that the masks in matrices (5)–(6) are not learnable and, consequently, do not introduce extra computation for training.

3.2. CNN elements

Given a volume (tensor) $y \in \mathbb{R}^{M\times N\times D}$ the convolutions in a CNN layer follows:

1. Filter dimension must be such that the number of channels in the input and filter depth is the same; that means, the filter w should satisfies $\{w(m,n,s)\} \in \mathbb{R}^{f\times f\times D}$.
2. We must define the discrete filter $w : W \to \mathbb{R}$, where:

$$W = \{-\xi,\dots,-1,0,1,\dots,\xi\}^2 \times \left\{\left\lfloor -\frac{D}{2}\right\rfloor,\dots,-1,0,1,\dots,\left\lfloor\frac{D}{2}\right\rfloor\right\} \tag{7}$$

   with value $\xi = (f-1)/2$ and matrix $\{w(m,n,s)\}_{(m,n,s)\in W}$ is named the filter mask.
3. Bias given by a scalar value $bias \in \mathbb{R}$.
4. The discrete convolution is computed by:

$$(w \star y)(m,n) = \sum_{(k,l,t)\in W} w(k,l,t)\, y(m+k, n+l, \bar{s}+t), \tag{8}$$

   where $1 \le m \le M$, $1 \le n \le N$, $\bar{s} = median\{1,2,\dots,D\}$ (see Fig. 3).
5. Apply bias to the convolution result:

$$z(m,n) = (w \star y)(m,n) + bias, \tag{9}$$

We can add zeros around the tensor boundary in Fig. 3 to avoid problems with the convolution computation in tensor positions near the volume boundary, a strategy named padding (Goodfellow et al., 2016). In this way, the output of expression (9) is a matrix $z \in \mathbb{R}^{M\times N}$. For data dimensionality reduction, the max-pooling operation can be applied. It performs subsampling of an input image $z \in \mathbb{R}^{M\times N}$. Its computation is based on the expression:

$$\widetilde{z}(i,j) = \max\left\{z(m,n),\ (m,n) \in \mathcal{R}_{ij}\right\}, \tag{10}$$

with $\mathcal{R}_{ij}$ being a neighborhood of pixel (i,j), with size $n_p \times n_p$ and max returns the maximum value ($\widetilde{z}(i,j)$) inside the sub-image in $\mathcal{R}_{ij}$.

The convolution is a fundamental operation in CNN architectures. Another technique that composes the foundations of the Conv-BN-ReLU pipeline used in our proposal is batch normalization. It is a data transform used to accelerate the training of deep neural networks (Ioffe & Szegedy, 2015). In this technique, given a mini-batch $\mathcal{B} = \left\{x_1, x_2, \dots, x_{|\mathcal{B}|}\right\}$ (subset of the training data), it is computed the mini-batch mean (μ_B) and variance (σ_B^2). Then, the samples in $\mathcal{B}$ are normalized as:

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{11}$$

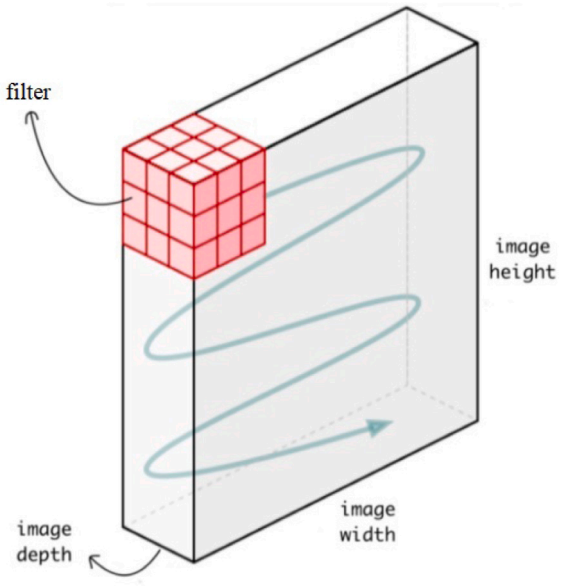

Fig. 3. Representation of the convolution over a volume in the CNN.

with ε used to avoid numerical instability. Next, the algorithm calculates the scale and shift $y_i = \gamma\hat{x}_i + \beta$ where λ and β are parameters that are learned in the training stage.

The ReLU activation function, proposed in Nair and Hinton (2010), is defined as:

$$f(x) = \max\{0, x\}. \tag{12}$$

4. Proposed architectures

MPSegnet: Given these basic wavelet and CNN concepts, we can present our proposal. Considering that our goal is semantic segmentation, the idea is to put together max-pooling and DWT outputs in a learnable way to emphasize the target features. To efficiently implement this idea, we apply 1×1 convolutions.

Hence, if C refers to the number of input channels in the MPSegnet module, then we stretch the max-pooling output through $2C$ convolution filters with the size of $1 \times 1 \times C$ and enhance features in the DWT space by using $2C$ convolution filters with the size of $1 \times 1 \times 4C$. Each filter is implemented following the expression (9) with $\xi = 0$ and $D = C$ generating the equation:

$$z^i(m,n) = \left(w^i \star y\right)(m,n) + bias^i, \quad 1 \le m \le M, \quad 1 \le n \le N, \quad 1 \le i \le 2C,$$

where, like in expression (9), M, N denote the number of rows and columns of the input signal y, w^i denotes weights of output channel i, $bias^i$ is the associated bias. The weights and bias involved in these operations are learned during the training process.

Specifically, as shown in Fig. 4(a), the multi-pooling network has two branches, one of them with max-pooling and the other one with DWT, both followed by 1×1 convolutions that change dimensions of the output signal for each branch to get a balanced learning process. Moreover, inspired by Segnet and other architectures, we apply two Conv-BN-ReLU (Section 3.2) in sequence before our pooling proposal, as represented in the Encoder Unit in Fig. 4(b). The Decoder Unit is the micro neural network also shown in Fig. 4(b). It is applied in the decoder stage, and its pipeline is almost symmetric to the Encoder Unit since we have replaced the multi-pooling layer to the IWT followed by two Conv-BN-ReLU blocks. The MPSegnet architecture is depicted in Fig. 5 where the encoder path is composed by five Encoder Units while the decoder part of the network also contains five Decoder Units, symmetrically arranged with respect to the encoder part.

We shall comment that the application 1×1 convolutions in the multi-pooling network avoids the necessity of using the Segnet scheme of memorizing the position of the maximum feature value in each max-pooling operation. Specifically, consider the simplified example in which the output of our multi-pooling block enters the decoder unit. Hence, a one-to-one inverse operation needs two deconvolutions to be learned during training to get two signals with appropriate dimensions: one to be filtered by the IWT and another one to be processed by the unpooling operator. We have experienced this strategy and the shortcut connections of U-Net without substantially improving the segmentation results but with a considerable increase in the computational complexity. We then discard shortcut connections and adopt only the IWT to perform upsampling without any more elaborated multi-unpooling strategy.

Two-streams: As an alternative to learning specialized fusion modules, we also propose a straightforward scheme based on the concept of two-stream CNNs. This concept initially thought of by Simonyan and Zisserman (2014) in the scope of video classification, quickly became one of the most used techniques to combine knowledge from different sources in problems such as video and image classification (Simonyan & Zisserman, 2014), segmentation (Takikawa et al., 2019), object tracking (Zhang et al., 2020), among others. In our work, we adapted the original structure designed initially by Simonyan and Zisserman (2014) to combine CNNs implemented with different pooling methods.

An overview of the proposed two-stream models is illustrated in Fig. 6. It consists of two Segnet architectures trained separately using different pooling and unpooling strategies followed by a fusion module. This model structure is used to implement the Segnet+WSegnet as well as the Segnet+MPSegnet. As shown in Fig. 6, we opted for a late fusion based on a simple average of the softmax outputs, given its low computational cost, proven efficiency in other domains, and easy implementation (Simonyan & Zisserman, 2014). Moreover, the fusion stage can be detached from the model training, which is desirable in environments with limited computational resources.

During the training and test steps of our two-stream model, all samples are applied to the CNN, and their last layer (softmax layer) feature maps are extracted. Each stream provides a single feature vector per unmanned aerial vehicle image. Then, the softmax scores of these two CNNs are averaged, and the result of this procedure is used to providing the segmentation maps.

5. Data sets

Two well-known semantic segmentation data sets were used to evaluate the effectiveness of our proposed method: the ISPRS Vaihingen and the ISPRS Potsdam (Rottensteiner et al., 2014). Vaihingen is a relatively small village with many detached buildings and story buildings. Potsdam is a typical historic city in Germany, with large building blocks and dense structure arrangement. These data sets were used in the ISPRS 2D semantic labeling contest. They have been also applied for pattern recognition tasks involving semantic segmentation (Chen et al., 2018; Yuan et al., 2021), classification (Guo et al., 2021), feature fusion for segmentation (Dong & Chen, 2021), among others. In the following subsections, we describe these data sets and the data preprocessing techniques applied to use them in conjunction with the developed architectures.

5.1. Data set description

The first data set used, the Vaihingen, contains high-resolution orthophoto maps (titles) and their respective digital surface models (DSMs), extracted from a large orthomosaic taken over the city of Vaihingen in Germany. More precisely, thirty-three title images were sampled, depicting the six most common land cover found in the Vaihingen region: impervious surfaces (imp. surfaces), buildings, low vegetation (low veg.), trees, cars, and clutter/background (clutter).

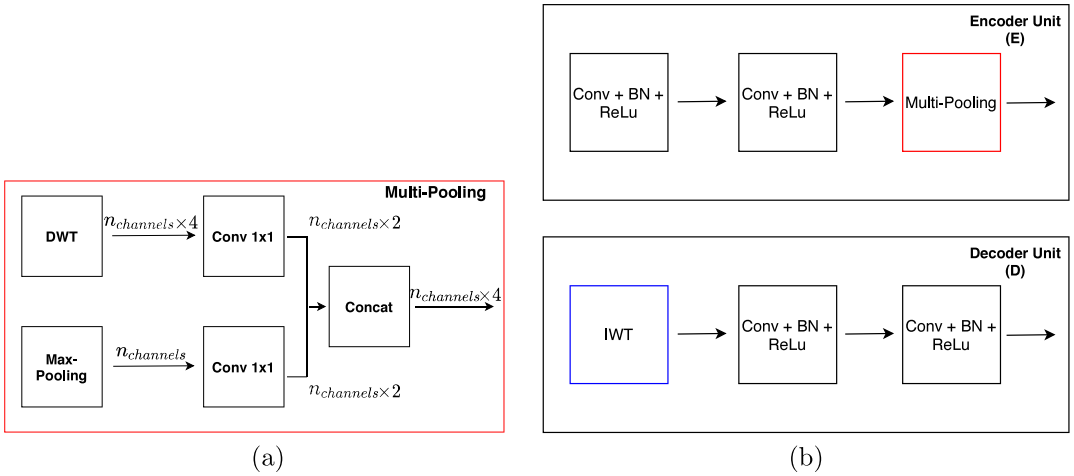

Fig. 4. (a) Multi-pooling architecture composed of two branches implemented by DWT and max-pooling. (b) Encoder and Decoder Unit blocks applied, respectively, in the contraction and expansion stages of the network.

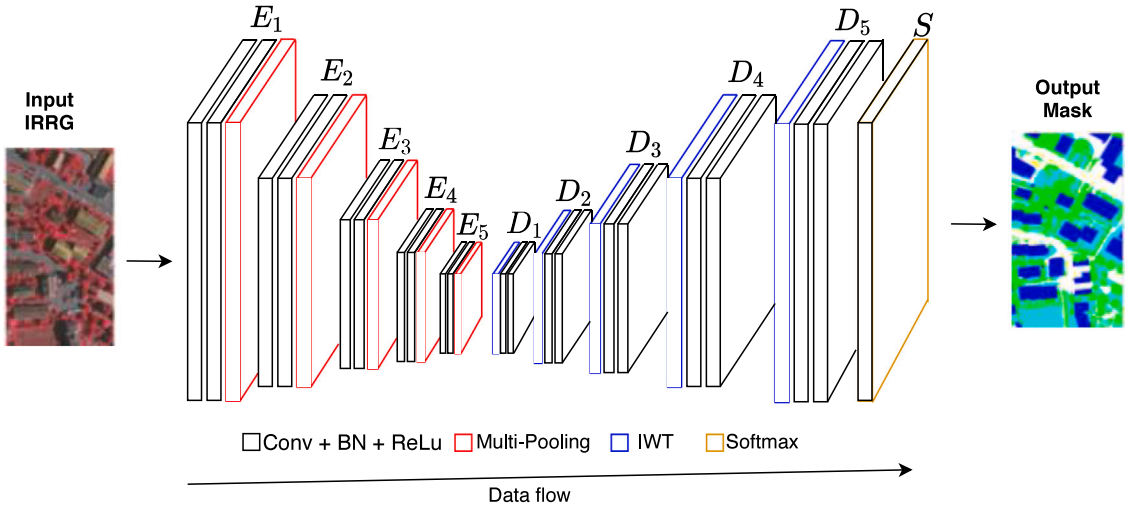

Fig. 5. MPSegnet architecture and data flow where E_i and D_i mean encoder and decoder units, respectively.

Each image has an average size of 2494 × 2064 pixels, a spatial resolution of 9 cm, and three-band channels: near-infrared (IR), red (R), and green (G). It is a data set that covers the major challenges in semantic segmentation, for instance, large variation in object scale, occlusions, cast shadows, reflections, among others. The same evaluation protocol adopted by Liu et al. (2017) was used here. The training and test sets were created using the sixteen annotated sample titles initially provided by the authors' data set. Based on these samples, eleven titles $(1, 3, 5, 7, 13, 17, 21, 23, 26, 32, 37)$ were separated for training, and the other five $(11, 15, 28, 30, 34)$ for testing. The average F1-score, and overall accuracy are reported as the final result.

On the other hand, the Potsdam $2D$ segmentation data set consists of thirty-eight high-resolution orthophoto maps and their normalized DSMs. The image samples in the Potsdam are 6000 × 6000 pixels in size and have a spatial resolution of 5 cm. Although four-band channels are available for the titles (near-infrared, red, green, and blue), we chose to use only the IRRG channels, as performed in the other database used. The ground surface classes presented in the Potsdam are identical to those found in the Vaihingen data set. Following the same training and test protocol used by Liu et al. (2017), the twenty-four titles with ground-truth pixel labels are divided into two groups: the first one with six titles $(02\_12, 03\_12, 04\_12, 05\_12, 06\_12, 07\_12)$ for testing and the second one, composed of eighteen samples for training. As in Vaihingen, the reported results are the average F1-score and the overall accuracy.

Since many real-world remote sensing data sets do not have DSM information, we limit ourselves to using only the information from the IRRG channels of these data sets. In this way, we can provide a more reliable panorama of the model performance in a scenario closer to its actual application.

5.2. Data pre-processing

The number of samples in both data sets is below what is reasonable for training a deep model (Goodfellow et al., 2016). Thus, we decided to implement a data pre-processing protocol similar to that adopted by Liu et al. (2017). In this protocol, random square windows with a fixed size of 256 × 256 pixels (patches) are cut from the sample titles. The obtained image set is further expanded using data augmentation strategies composed by the horizontal flip and vertical flip of the samples. This procedure generates 10,000 new training images per epoch that will later be used to enhance the network's learning process.

At the testing, a sliding window procedure is employed to slice the high-resolution orthophotos into small patches to fed into the network. However, this approach can cause inconsistencies in segmentation, especially at the patch borders, severely impairing the model's accuracy, as demonstrated by Liu et al. (2017). To avoid these border issues,

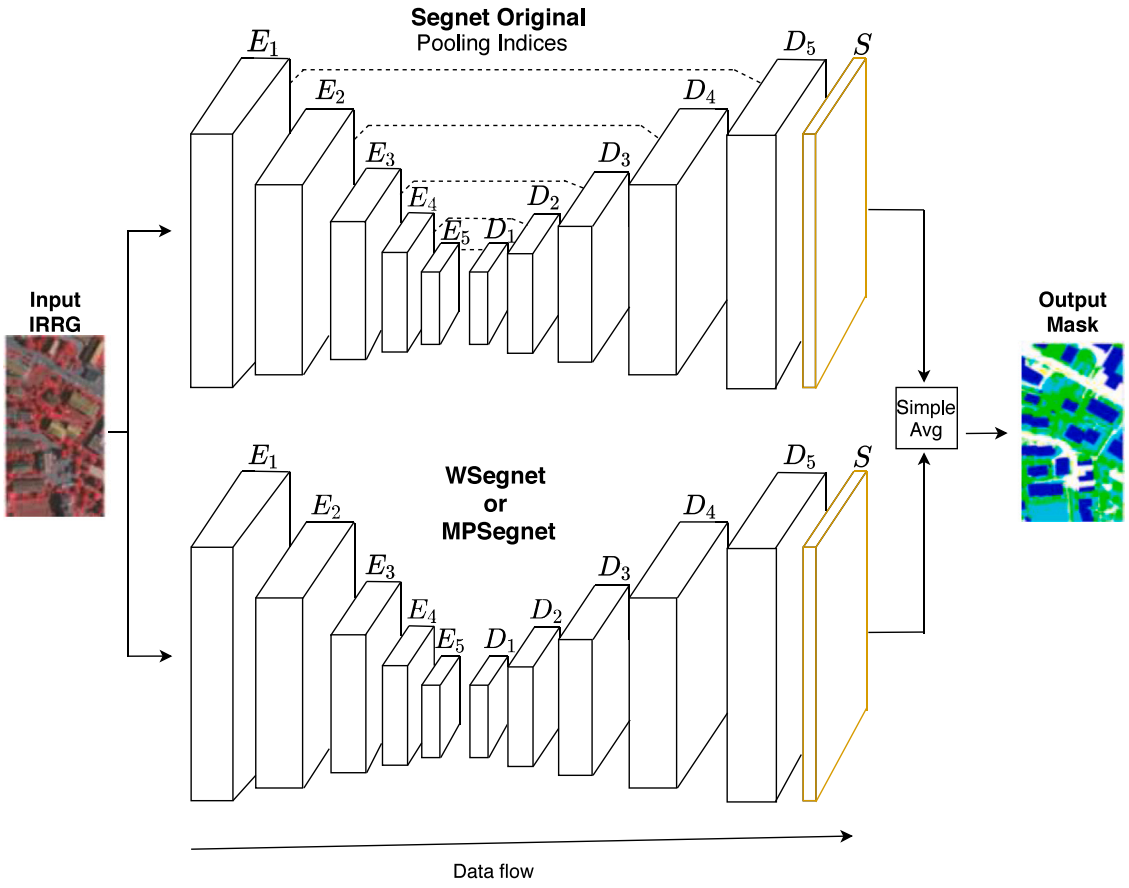

Fig. 6. Two-stream architectures (Segnet + WSegnet and Segnet + MPSegnet) and data flow where E_i and D_i mean encoder and decoder units, respectively.

we follow Farhangfar and Rezaeian (2019) and use overlapped patches with a stride of 16 pixels. The pixel class in these overlapping areas is given by averaging the patches' network outputs that overlap over the pixel.

Contrary to Farhangfar and Rezaeian (2019) and Liu et al. (2017), this overlapping approach was used only in the testing stage because the gains observed with the application of this technique during network training did not justify the computational effort required by it, especially in our environment with limited computational resources.

6. Evaluation metrics

Using the same evaluation protocol established in the 23rd International Society for Photogrammetry and Remote Sensing 2D Semantic Annotation Competition (Rottensteiner et al., 2014), we report two main metrics as the result of the experiments: the overall accuracy and the F1 score. Suppose C is the confusion matrix to multi-class problems, where each entry $C_{i,j}$ refers to the number of samples known to be in class i and predicted to be in class j, with $i, j = 1, 2, \ldots, N$, where N is the number of classes. Hence, the overall accuracy for a specific image is defined as the total number of correctly predicted pixels $\sum_{i=1}^{N} C_{i,i}$ divided by the total number of pixels $\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}$ (with analogous definitions for the overall accuracy (*Overall Acc*) computed over the entire database). The F1-score of the class i ($F1_i$) is the harmonic mean of precision (P_i) and recall (R_i) and is given by:

$$F1_i = 2 \times \frac{P_i \times R_i}{P_i + R_i}, \tag{13}$$

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i}, \tag{14}$$

where TP_i, FP_i and FN_i are, respectively:

- The true positives of class i ($TP_i \equiv C_{i,i}$) that is equal to the number of correctly predicted samples of class i;
- The false negatives of class i ($FN_i \equiv \sum_{j=1, j\neq i}^{N} C_{i,j}$) that is equal to the number of samples known to be in class i but incorrectly classified in other class $j \neq i$;
- The false positives of class i ($FP_i \equiv \sum_{k=1, k\neq i}^{N} C_{k,i}$) that is equal to the number of samples known to be in class $k \neq i$ but incorrectly classified in class i;

We also decided to use the intersection over union (IoU) as an additional evaluation metric. The IoU is widely utilized in the semantic segmentation domain (Minaee et al., 2021) and is defined as follows:

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i}. \tag{15}$$

Contrary to overall accuracy, the F1 score and IoU are metrics computed for each category. However, we also calculated their average values, defined, respectively, as average F1 score (avgF1) and average intersection over union (avgIoU), to complement our experiments section. The avgF1 and the avgIOU are defined as:

$$avgF1 = \frac{2}{N}\left(\sum_{i=1}^{N} \frac{P_i \times .R_i}{P_i + R_i}\right), \tag{16}$$

$$avgIoU = \frac{1}{N}\sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i + FN_i}. \tag{17}$$

Although two versions of ground truth images are available, we opted for the classic pixel-wise version. In the alternative version (border-eroded version), the object's boundaries have been eroded using a 3-pixel radius disk, reducing the impact of dubious border definitions. Nevertheless, this strategy prevents the model from learning to deal with these situations in real-world applications. For this reason, we

Table 1
Comparison of the average F1 score (avgF1), average IoU (avgIoU) and overall accuracy (Overall Acc) for six versions of Segnet in both data sets.

Approach	Vaihingen			Potsdam		
	Overall Acc	avgF1	avgIoU	Overall Acc	avgF1	avgIoU
Segnet (Badrinarayanan et al., 2017)	85.05	82.76	71.19	78.99	70.90	57.42
WSegnet (Li & Shen, 2020)	85.08	82.53	70.91	79.69	71.49	58.12
MPSegnet†	85.32	82.68	71.18	79.75	71.51	58.20
MPSegnet	85.64	82.70	71.24	**80.00**	71.94	58.59
Segnet + MPSegnet	85.87	83.48	72.25	79.91	**72.18**	**58.67**
Segnet + WSegnet	**85.95**	**83.65**	**72.48**	79.78	71.87	58.49

focused on presenting our results using only the conventional pixel-wise ground truth.

7. Experimental results

Extensive experiments were carried out to show the contributions of our multi-pooling strategies. In this section, the experimental setup, the quantitative and qualitative results of the experiments are reported. We ran each method three times using the training and inference procedure presented in Section 5.2. Each run was performed with a different randomization seed, and the metric values reported in the experiments are the average over these executions.

7.1. Implementation details

In the development of the MPSegnet, Segnet+MPSegnet, and the Segnet+WSegnet, we used the original Segnet network presented in Badrinarayanan et al. (2017) as our basis. Network weights were initialized using the ImageNet weights in the encoder part and by the Kaiming He initialization (He et al., 2015) in the remaining layers. The entire architecture was implemented in the Pytorch framework (Paszke et al., 2019). The same training parameters were used in both data sets. The network learning was performed with the following parameters: learning rate of 10^{-2}, batch size of 6, Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9, and categorical cross-entropy loss function (Goodfellow et al., 2016). The learning rate was scaled down by a factor of 10 in the 25th, 35th and 45th epoch and the learning rate reduction was limited to 10^{-5}. The developed networks were trained for 100 epochs. Part of the training set, approximately 33%, was randomly separated as a validation set to estimate these network hyper-parameters. All experiments were performed on a desktop with a Geforce RTX 2070 (8 Gb vRAM).

7.2. Quantitative comparison of pooling methods

Firstly, we analyzed how the key structural changes made to the pooling layers affected the overall performance of the original architecture. We focused on relative accuracy gains verified when comparing the accuracy of networks equipped and not equipped with our multi-pooling proposal. To this end, we put the vanilla Segnet against our three novel versions: MPSegnet, Segnet+MPSegnet, and Segnet+WSegnet. The other tested MPSegnet, named MPSegnet† in the following, was the same network as in Fig. 5 without the 1×1 convolutions in the multi-pooling modules. In MPSegnet†, instead of using a reduction process, as in MPSegnet, we used the components of the two pooling processes (max-pooling and DWT) directly concatenated. To accommodate these extra components in MPSegnet†, we modified the number of input channels of the post-pooling convolutional layers by adding a single 1×1 convolution on the bridge between the encoder and the decoder. This 1×1 convolution is responsible for ensuring that the number of input channels is a multiple of 4, fitting them to the upsampling process carried out by the IWT in the next step. All results are summarized in Table 1. The values of the metrics of the WSegnet are also reported for comparative purposes.

Examining the best results reported in bold in Table 1, the crucial contribution of combining max-pooling and wavelet pooling strategies for semantic segmentation becomes evident for both Vaihingen and Potsdam data sets. It can be seen that in only one of the scenarios, the basic architecture surpassed the Segnet versions introduced by this work. In this situation (second column of Table 1), Segnet's avg F1 score is 82.76% against 82.70% of MPSegnet that corresponds to a slightly higher (0.06%) F1-score in favor of Segnet. The comparison between MPSegnet† and MPSegnet further reinforces the importance of carefully merging pooling features. The simple concatenation applied in MPSegnet† lost in all situations to the multi-pool structure employed by MPSegnet. Based on these results, we decided to discard MPSegnet† from the subsequent analysis.

Furthermore, it is important to highlight the excellent results achieved by both two-stream Segnet architectures (Segnet+WSegnet and Segnet+MPSegnet).

These results demonstrate that this multi-stream scheme widely used to fuse input features in the semantic segmentation domain (Zhang et al., 2019) can also be applied to combine results yielded with CNNs implemented with different pooling methods.

With the next experiment, we seek to understand how the particularities of each land cover class affect the pooling strategies. For this, we particularize the previous analysis to class level. So, following the literature, we report in this experiment the IoU and F1-score for each class. Tables 2 and 3 present the numerical results of the experiment.

The results in Table 2 show that our multi-pool schemes improved the accuracy for all Vaihingen classes. Compared with the original Segnet, the maximum improvement observed in IoU/F1 metrics was 1.43%/0.89%, 1.26%/0.73%, 1.74%/1.33%, 1.10%/0.73% and 0.99%/0.78%, respectively, for imp surfaces, buildings, low veg, trees, and cars classes. Following Liu et al. (2017), we omitted the clutter class when reporting the results of the Vaihingen, given the insignificant number of pixels presented by it.

The avgF1 decrease of MPSegnet observed in the previous experiment can be explained by the F1-score reduction in the car class. We believe that this behavior is related to an early convergence in some of the multi-pooling modules, affecting the extraction of richer car features and, consequently, the performance of MPSegnet to segment this class. We will return to this issue later.

Regarding Potsdam results, we see that our architectures outperformed the original Segnet and the WSegnet in five of the six classes. Using the Segnet as a baseline, the improvement observed in IoU/F1 metrics was 1.34%/0.97%, 0.78%/0.61%, 1.61%/1.25%, 1.23%/0.79% and 2.64%/3.67%, respectively, for imp surfaces, low veg, trees, cars, and clutter classes. As discussed by Li and Shen (2020) and Liu et al. (2019), we believe that the details captured in the high frequency components were the main reason for the better performance of WSegnet in the class building. Especially in densely populated regions, characterized by small objects, high-frequency components can serve, for example, to identify the texture of the buildings. We presume that, in this case, all information in the three original high-frequency components in DWT was essential to capture the fine-scale details needed to segment the building class in the Postdam data set correctly. We believe the lower performance of our approaches for the class

Table 2
Per-class comparison between the F1-score (F1) and IoU for five versions of Segnet in the Vaihingen data set.

Approach	Imp. Surfaces	Buildings	Low Veg.	Trees	Cars
	IoU/F1	IoU/F1	IoU/F1	IoU/F1	IoU/F1
Segnet	77.66/87.43	85.27/92.05	60.80/75.62	73.48/84.71	58.74/74.01
WSegnet	77.78/87.50	85.50/92.19	61.19/75.92	73.13/84.48	56.96/72.58
MPSegnet	78.45/87.92	86.39/92.70	61.73/76.34	74.18/85.18	55.44/71.34
Segnet + MPSegnet	78.87/88.19	86.28/92.64	62.28/76.76	**74.58/85.44**	59.21/74.38
Segnet + WSegnet	**79.09/88.32**	**86.53/92.78**	**62.54/76.95**	74.51/85.39	**59.73/74.79**

Table 3
Per-class comparison between the F1-score (F1) and IoU for five versions of Segnet in the Potsdam data set.

Approach	Imp. Surfaces	Buildings	Low Veg.	Trees	Cars	Clutter
	IoU/F1	IoU/F1	IoU/F1	IoU/F1	IoU/F1	IoU/F1
Segnet	65.22/78.95	74.45/85.35	61.18/75.91	59.49/74.60	65.55/79.19	18.63/31.41
WSegnet	65.99/79.51	**75.63/86.12**	61.63/76.26	60.05/75.04	66.16/79.63	19.30/32.36
MPSegnet	**66.56/79.92**	75.37/85.96	61.83/76.41	**61.10/75.85**	66.45/79.85	20.26/33.69
Segnet + MPSegnet	66.44/79.83	75.23/85.87	**61.96/76.52**	60.89/75.69	**66.78/80.08**	**21.27/35.08**
Segnet + WSegnet	66.16/79.63	75.43/85.99	61.91/76.47	60.45/75.35	66.71/80.03	20.32/33.77

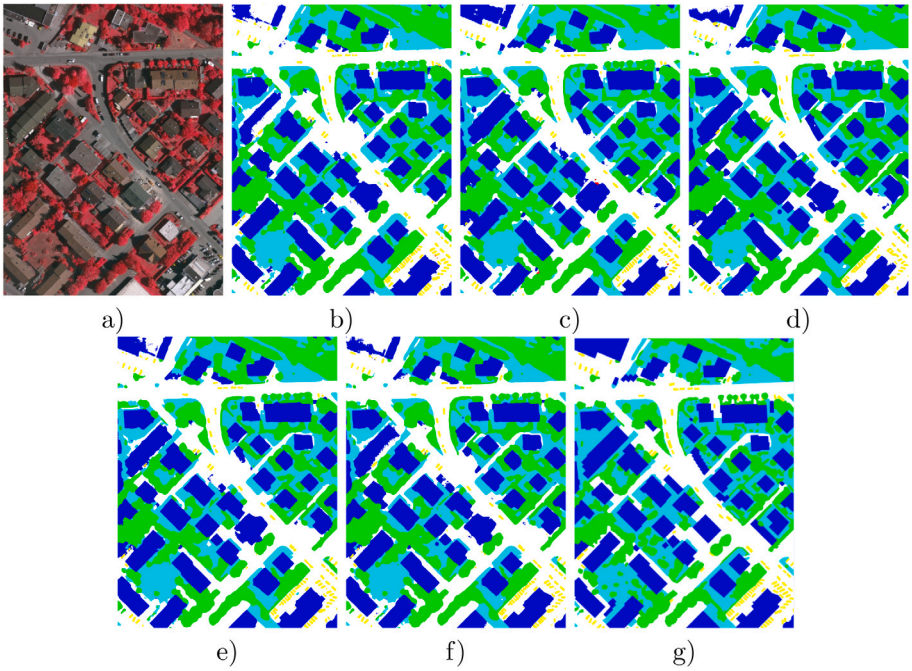

Fig. 7. Segmentation results for Vaihingen Title 30. From (a) to (g), respectively, the original image, the results obtained from Segnet, WSegnet, MPSegnet, Segnet+WSegnet, Segnet+MPSegnet, and the ground truth image. Legend: (white: imp. surface, blue: buildings, cyan: low veg., green: trees, yellow: cars, red: clutter).

buildings can be explained considering that our multi-pooling modules can improve feature extraction by learning filter banks capable of efficiently combining pooling methods. However, in certain situations, this learning process may not converge satisfactorily. As a result, these sub-optimal multi-pooling structures can keep the network below its full potential. Specifically, sub-optimal filter in the MPSegnet pooling layers or losses introduced in the average fusion performed in the two-stream scheme may have caused a partial loss in high-frequency fine-scale details of the buildings class, leading to decreasing in performance in that class.

Analyzing Potsdam results, we also discovered some limitations of adapting the original two-stream architecture (Simonyan & Zisserman, 2014) to combine pooling features. We shall notice that all knowledge learned is usually taken into account during the fusion process. Hence, losses from other operations could impair the performance of this multi-pooling fusion process. The simple average adopted to combine the softmax outputs does not consider that one network can be more accurate than the others as pointed out by Wu et al. (2015). A weighted average can alleviate this problem, but even so, the fusion process is still tied to a global combination of results. As a result, we can see in Table 3 that, in certain situations, the two-stream approaches (Segnet + MPSegnet and Segnet + WSegnet) showed inferior outcomes to those presented by the methods that constitute them. However, this fact does not invalidate the use of multi-stream approaches in this domain. On the contrary, in most situations, the proposed strategy achieved competitive results, as shown in Tables 1–3, with all the advantages already mentioned in Section 4.

7.3. Visual analysis

A visual comparison complements the quantitative analysis and helps to understand the fundamental contributions of each method. We segmented a full title from each data set using all five network architectures to serve as a basis for qualitative analysis. Figs. 7 and 9 show the segmentation outputs obtained from this procedure. Like in Liu et al. (2017), we decided to highlight certain regions of these outputs and depict them in Figs. 8 and 10.

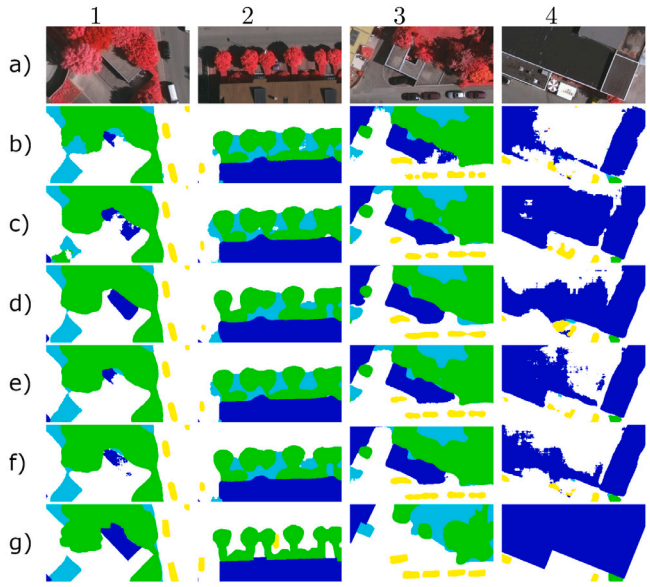

Fig. 8. Some areas zoomed (patches) in the segmentation results of Vaihingen Title 30. From (a) to (g), respectively, the original top image, the results obtained from Segnet, WSegnet, MPSegnet, Segnet+WSegnet, Segnet+MPSegnet, and the ground truth image. Legend: (white: imp. surface, blue: buildings, cyan: low veg., green: trees, yellow: cars, red: clutter). Patches legend: 1: building hidden by trees, 2: dark area caused by tree shadows, 3: miss-labeled building, 4: building roofs with different textures.

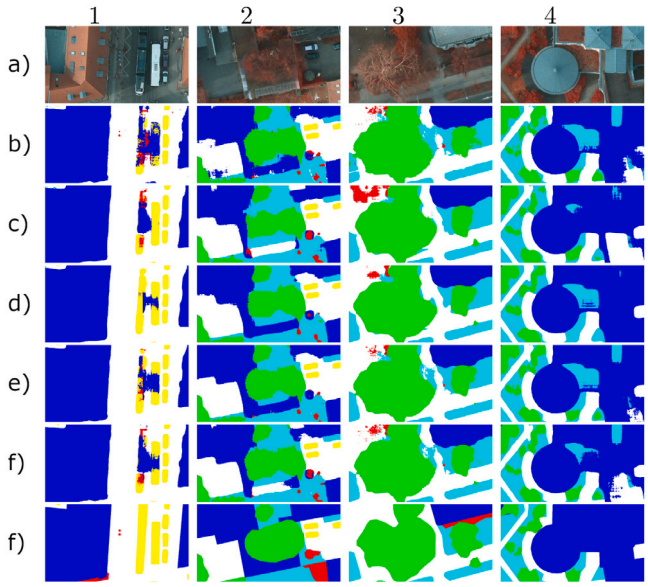

Fig. 10. Some areas zoomed (patches) in the segmentation results of Potsdam Title 5_12. From (a) to (g), respectively, the original top image, the results obtained from Segnet, WSegnet, MPSegnet, Segnet+WSegnet, Segnet+MPSegnet, and the ground truth image. Legend: (white: imp. surface, blue: buildings, cyan: low veg., green: trees, yellow: cars, red: clutter). Patches legend: 1: shadow of buildings covering cars, 2: road hidden by trees, 3: building hidden by low vegetation, 4: building rooftop with similar texture to low vegetation.

As before, we initially focus on Vaihingen results. Looking at Figs. 7 and 8 related to this data set, we can observe the main challenges found in the semantic segmentation domain. Identifying the correct shape of the cars, for example, was a complicated task for all models. This class's segmentation process is challenging due to the high intra-class variability and the small number of pixels compared to other classes. Shadows and occlusions caused by larger objects, such as trees and buildings, on these objects also have a considerable impact on the results.

Compared to the original Segnet, there was an improvement in the correct classification of pixels referring to buildings, trees, and roads with the adoption of the proposed methods. Among the approaches presented in this work, we highlight MPSegnet, which accurately represented the shapes of buildings and the layout of the roads in the Vaihingen Title 30. For instance, in Patch 1 of Fig. 8, we observed that MPSegnet was able to segment the building hidden by trees. In the same Fig. 8, but this time in Patch 2, we highlight the MPSegnet ability to segment part of a region covered by some trees' shade without any height information, which shows its capability to extract contextual information to improve pixel classification. During the experiments, we found inconsistencies in the ground-truth images, such as an unlabeled building, showed in Patch 3 of Fig. 8. However, even in this situation, the methods correctly labeled this object, as seen in Fig. 8.

Concerning the Potsdam, we noticed that shadows, occlusions, distortions and reflections also affected the segmentation results, as can be seen in Figs. 9 e 10. For instance, in Patch 1 of Fig. 10, part of these car-pixels was incorrectly labeled as buildings because of the shadow cast

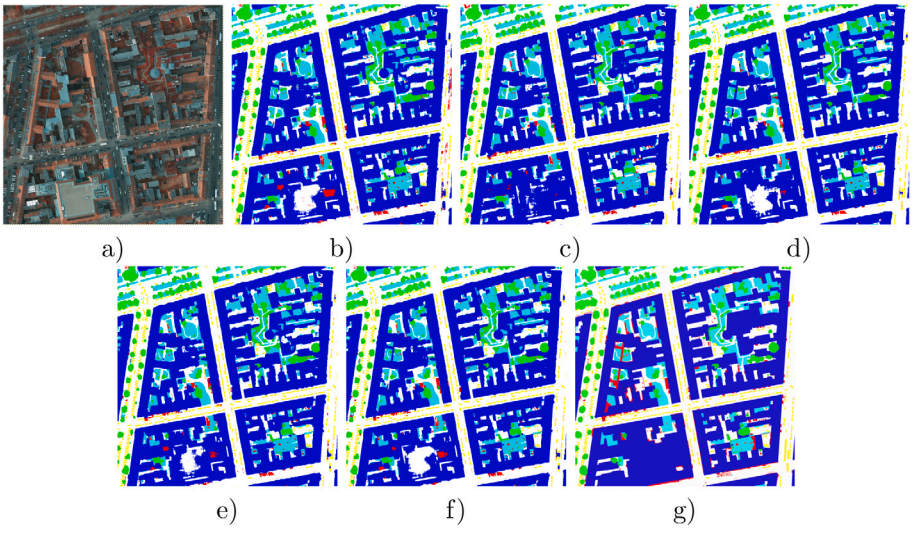

Fig. 9. Potsdam Title 5_12 segmentation results. From (a) to (g), respectively, the original top image, the results obtained from Segnet, WSegnet, MPSegnet, Segnet+WSegnet, Segnet+MPSegnet, and the ground truth image. Legend: (white: imp.surface, blue: buildings, cyan: low veg., green: trees, yellow: cars, red: clutter).

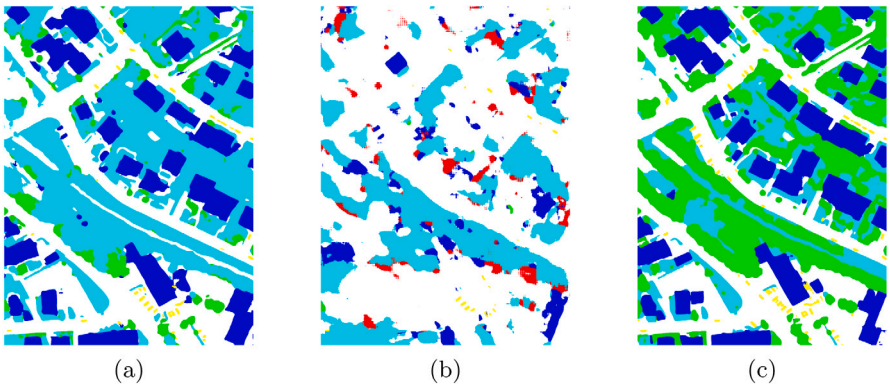

Fig. 11. (a) Segmentation obtained with MPSegnet when making NULL the DWT branch result in the encoder unit E1. (b) Analogous result but now making NULL the max-pooling contribution E1. (c) Ground-truth segmentation.

by buildings on those objects. The presence of more pixels of the clutter class was also a significant challenge for the approaches in Potsdam. The segmentation maps of Fig. 9 show that all methods performed comparatively worse in classifying the Potsdam database, due to the imbalance in the number of pixels between classes, in particular, due to the minority class "clutter".

For the Potsdam data set, the proposed methods produced smoother, visually more realistic contours when compared with Segnet results, which reinforces the effectiveness of the proposed changes in the original architecture. As in the other database, we highlight the good visual output quality of the segmentation outcome of the MPSegnet.

Another relevant issue is the influence of max-pooling and DWT branches in the multi-pooling network of Fig. 4. To visualize each branch's importance in the segmentation result, we take the full title 30 from the Vaihingen data set and segment it using the MPSegnet network with two modifications. Firstly, we set to zero the DWT result in the multi-pooling network of the encoder unit closest to the input image (E1 of Fig. 5). Next, we proceeded analogously but setting to zero the max-pooling pipeline contribution in the same encoder unit. In this way, we focused on the significance of each multi-pooling branch outcome to the final segmentation. We have also applied this protocol to all encoders' units, but the result became severely damaged, making the analysis impossible.

The obtained segmentations are presented in Figs. 11(a)–(b), respectively, together with the ground-truth segmentation in Fig. 11(c). The result 11(a) highlights the importance of DWT for the correct classification of the trees (green). In fact, if we compare Fig. 11(a) and (c), it is noticeable that several regions with objects of the tree class were misclassified as buildings in Fig. 11(a). On the other hand, the comparison between Fig. 11(b) and (c) reinforces the importance of max-pooling in our methodology by focusing on relevant features for pattern recognition since the MPSegnet underwent a considerable decrease in performance when we removed the max-pooling contribution.

The values in Tables 1–3 indicate that the proposed methods outperformed the baselines by a small margin. However, this is not surprising since our proposal aims to mitigate the main disadvantage of pooling operations: the inaccuracies in the segmentation of image details. Such effect can be visualized in Fig. 12 that shows the difference between the segmentation of the full title 30 from the Vaihingen data set obtained through MPSegnet and the ground-through (Fig. 12(a)), the difference between the segmentation of the same full title obtained with the Segnet and the ground-through (Fig. 12(b)) and, finally, the difference between the images in Fig. 12(a) and (b), picture in Fig. 12(c). Despite some differences in the interior of the objects, we notice that the differences in Fig. 12(c) are specially located at their edges. In other words, pooling causes errors in a reduced number of pixels. As the metrics represented in the tables consider all input image pixels, the accuracy gains will necessarily be small numerically. In this way, our proposals' benefits are visible mainly by looking at the semantic segmentation results in the pixels near the objects' borders, as we can see in Fig. 12(c).

Table 4
Comparison of FLOPs, parameters and average times (inference and epoch duration) for Segnet and its variants. Patches of 256 × 256 are used to calculate the FLOPs and to measure the average times. Following the literature, we report the average inference time in milliseconds (ms) and average epoch duration in seconds (s).

Method	Parameters (M)	FLOPs (G)	Average inference time (ms)	Average epoch duration (s)
Segnet	29.46	80.58	45.32	231.64
WSegnet	52.91	131.36	103.68	432.39
MPSegnet†	56.94	139.38	104.48	438.13
MPSegnet	56.38	133.22	104.12	434.27

7.4. Complexity analysis

In practical terms, the complexity of a deep model could be defined in terms of computation overhead and the number of parameters. The number of parameters is obtained directly from the model definition. The computation overhead can be estimated as the maximum number of floating-point operations (FLOPs) needed to run a model instance. Based on these two metrics and in the test-training protocols described in Section 5.2, we evaluated the impact of the proposed multi-pooling strategy on the architecture complexity. To complement the analysis, we also report the single-patch average inference time and the average duration of an epoch, taking as a reference the computational cost of Segnet and WSegnet, since they are baselines for our work. The results are summarized in Table 4.

Observing the lower computational complexity of Segnet, reported in Table 4, compared to other architectures, it is clear that the expansion carried out in the post-pooling convolutional layers to accommodate the additional channels in wavelet-based approaches has considerably increased the complexity of these approaches. The WSegnet results in Table 4 reveal that these five convolutional layers expanded, in the encoder structure, to accommodate the three high-frequency DWT components are responsible for an increase of approximately 44.32%, 38.66%, 56.29%, 46.43%, respectively, in FLOPs, parameters, average inference time and average epoch duration, respectively, if we consider the Segnet results. This type of behavior is already expected since the computational cost of the network is dominated by the convolutional layers (He & Sun, 2015), and our multi-pooling network, for instance, involves two extra convolutions, besides the DWT and IWT computations. Thus, the observed computation overhead is justified.

On the other hand, comparing MPSegnet and WSegnet, we observe an increment of approximately 6.56%, 1.42%, 0.42%, 0.43%,

Fig. 12. (a) Difference between the segmentation of the full title 30 from the Vaihingen data set obtained through MPSegnet and the ground-through. (b) Analogous result but with the segmentation obtained with the Segnet. (c) Difference between the images in Figures (a) and (b).

respectively, for FLOPs, number of parameters, average inference time, and average epoch duration, respectively. These observations validate our choice to adopt 1×1 convolutions in our scheme for combining multiple pooling strategies. The dimensionality reduction provided by this operation made it possible to attach the multi-pooling units to the WSegnet architecture, improving its performance, as reported in the previous sections, without considerably increasing the number of parameters and floating point operations. This fact, elucidated by the comparison of our MPSegnet with its only-concatenated counterpart (MPSegnet†), indicates that adding a pair of 1×1 convolutions to each branch in our multi-pooling structure proved to be computationally cheaper than changing the convolutions in the encoder structure to accommodate another extra channel.

7.5. Comparison to state-of-the-art approaches

Finally, we compare our results with state-of-the-art approaches. Table 6 shows that our methods achieved competitive scores in the Vaihingen data set compared to the literature's approaches.

In this data set, our methods outperformed several approaches that use information beyond the IRRG channels, such as those with an X assigned in the DSM column of Table 6. These approaches use some kind of DSM information along with the IRRG channels to help in the segmentation process. For instance, Liu et al. (2017) employs a normalized DSM version (nDSM), generated by Gerke (2014), as complementary information. As evidenced by Maune (2007), since DSM depicts elevations from the top of reflective surfaces, such as buildings and vegetation, it is beneficial to identify these structures in aerial images. For example, the methods of Liu et al. (2017) were able to distinguish and correctly segment instances of different classes that are visually similar in IRRG images, such as buildings and roads or low vegetation and trees. However, this information is usually not available in aerial images. For this reason, we avoided using this feature in our experiments.

In terms of Overall Acc, only the SiameseDenseU-Net+ MFB_Focal$_{loss}$ surpassed Segnet+MPSegnet and Segnet+WSegnet. The MPSegnet was the third-best method in the corresponding column.

Regarding the avgF1, we notice that our best result, obtained by the Segnet+WSegnet, surpassed those of seven methods among the eleven considered. The second best technique in our suite, regarding the avgF1 score, was the Segnet+MPSegnet that surpassed six of the methods listed in Table 6.

Such difference in the position of our proposals when considering avgF1 and Overall Acc is because the latter depends only on the true positives (given by the C_{ii} in the confusion matrix) while the former is sensitive to FN_i/C_{ii} and FP_i/C_{ii}, as we can see by rewriting expression (16) as:

$$AvgF1 = \frac{2}{N}\left(\sum_{i=1}^{N} \frac{1}{\left(1+\frac{FN_i}{C_{ii}}\right)+\left(1+\frac{FP_i}{C_{ii}}\right)}\right). \tag{18}$$

Table 5
Ratios FN/TP and FP/TP for Segnet+WSegnet in Vaihingen classes.

	Imp. Surfaces	Buildings	Low Veg.	Trees	Cars
FN/TP	0.129	0.088	0.329	0.133	0.575
FP/TP	0.135	0.067	0.270	0.209	0.099

For instance, Table 5 shows the mentioned ratios for all the Vaihingen database classes for Segnet+WSegnet. We notice that except for the class buildings, all values for FN/TP are larger than 0.10, which may produce the difference in the ranking of Segnet+WSegnet approach concerning the considered scores.

Before discussing the Potsdam results, we shall point out a relevant fact about the methodology followed in the computational experiments. Since the Vaihingen data set does not offer the blue channel, we decided to discard this information available in the Potsdam database. Although we know that the blue channel combined with the IRRGs is used by Kampffmeyer et al. (2016), Liu et al. (2017) and Volpi and Tuia (2017) to improve their Potsdam segmentation results, we decided to ignore this extra-channel to perform a fair comparison among the models for both data sets, as done in Section 7.2.

However, such a choice has a price, as we can notice in the Potsdam column of Table 6. The lack of the blue channel may explain the inferior performance of our methods compared to other approaches except Segnet and WSegnet that were overcome by the new techniques.

8. Conclusions and future work

The main objective of this paper is to improve pooling operation in convolutional neural networks (CNNs) for semantic segmentation tasks through the combination of max-pooling and DWT. Such a combination can preserve more details in the frontiers of the objects, and it had not been reported in the context of semantic segmentation yet. Hence, the major motivation is to fulfill that gap. In this way, this paper proposes a novel multi-pooling strategy that combines max-pooling and wavelet pooling. We introduced the MPSegnet that incorporates encoder units built with our multi-pooling proposal. Also, we design and test hybrid Segnet architectures: Segnet+WSegnet and Segnet+MPSegnet. Hence, the main contributions of this paper are: the new multi-pooling scheme, the encoder unit built through it, and the novel architectures MPSegnet, Segnet+WSegnet, and Segnet+MPSegnet.

The computational experiments focused on semantic segmentation of samples of the Vaihingen and the Potsdam data sets (Rottensteiner et al., 2014). The value of our work was attested by the performance of MPSegnet, Segnet+WSegnet, and Segnet+MPSegnet in that application against state-of-the-art methods. Specifically, the MPSegnet surpassed the Segnet in terms of overall accuracy for both the Vaihingen and Potsdam databases highlighting the potential of our multi-pooling module.

Table 6
Comparison of the average F1 score (avgF1) and overall accuracy (Overall Acc) in both data sets. The approaches that use DSM information in conjunction with the IRRG channels are highlighted with an X in the DSM column.

Approach	Vaihingen		Potsdam		DSM
	Overall Acc	avgF1	Overall Acc	avgF1	
FCN (Kampffmeyer et al., 2016)	83.18	79.18	85.04	73.77	X
FPL (Volpi & Tuia, 2017)	83.69	78.56	85.93	72.29	X
Segnet (Badrinarayanan et al., 2017)	85.05	82.76	78.99	70.90	–
WSegnet (Li & Shen, 2020)	85.08	82.53	79.69	71.49	–
HSN (Liu et al., 2017)	84.92	82.88	86.56	72.85	X
HSN+OI (Liu et al., 2017)	85.38	83.51	86.89	73.25	X
HSN+OI+WBP (Liu et al., 2017)	85.39	83.41	87.05	73.44	X
DenseU-Net+CE_{loss} (Dong et al., 2019)	85.28	84.73	–	–	X
DenseU-Net+MFB_$Focal_{loss}$ (Dong et al., 2019)	85.63	84.95	–	–	X
SiameseDenseU-Net+CE_{loss} (Dong et al., 2020)	85.76	84.91	–	–	X
SiameseDenseU-Net+MFB_$Focal_{loss}$ (Dong et al., 2020)	86.20	85.53	–	–	X
MPSegnet	85.64	82.70	80.00	71.94	–
Segnet + MPSegnet	85.87	83.48	79.91	72.18	–
Segnet + WSegnet	85.95	83.65	79.78	71.87	–

The comparison with state-of-the-art techniques, using the mentioned data sets, showed competitive overall accuracy in the Vaihingen data set using only IRRG channels while most of the counterpart approaches applied DSM information (see Table 6). Regarding the Potsdam results, the MPSegnet achieved the best performance among our proposed methods, in terms of overall accuracy, but it surpassed only the Segnet and WSegnet networks in Table 6. We believe that the fact that we did not apply the DSM (or blue) channels is behind this comparatively lower performance. On the other hand, the analysis of computational complexity of Section 7.4 showed that the overhead introduced by our multi-pooling strategy is not restrictive if compared with WSegnet. Such observation validated the adoption of 1×1 convolutions for combining multiple pooling strategies.

Regarding the $avgF1$ score, the ranking of our techniques deteriorated due to the ratios FN_i/C_{ii} and FP_i/C_{ii} in expression (18). Nevertheless, we must emphasize that MPSegnet was close to or even better than schemes that did not use DSM (Segnet and WSegnet) for both the performance measures, as reported in Table 6. Meanwhile, we shall also highlight the following findings:

- The comparison between MPSegnet† and MPSegnet in Table 1 reinforces the importance of 1×1 convolutions for merging pooling features.
- Results of Tables 2–3 show some improvement in per-class recognition when our multi-pooling scheme is adopted. Such fact is highlighted in the visual analysis of Section 7.3.
- The visual analysis of Section 7.3 also shows that identifying the correct shape of the cars is a complicated task for all proposed models.

Despite the observed capabilities of our proposal, we have observed issues in Section 7 for segmentation of cars and in the presence of shadows, occlusions, distortions and reflections pointing out the necessity of further improvements. In this way, a deeper analysis of our multi-pooling networks is needed in order to address those limitations, including comparisons with strategies combining CNN, atrous convolution, and fully connected conditional random fields (Chen et al., 2018). Moreover, we intend to follow Liu et al. (2017) and use DSM information to train our model aiming to improve the recognition of objects strongly described by their elevation measures, such as buildings and trees. We also aim to study ways to address the imbalance frequently found in aerial image data sets. Although our data augmentation protocol worked pretty well for this purpose, more robust strategies can be employed to mitigate this problem, such as new weighted loss functions (Bischke et al., 2018; Bulo et al., 2017; Jadon, 2020), the application of specific data sampling protocols (Liu et al., 2017), and new network structures (Ni et al., 2019). Another possible future research direction refers to applying other wavelet transforms to compose the core of our multi-pooling units. For example, the Stationary Wavelet Transformation (SWT), used by Oliveira et al. (2018) as a pre-processing step, could replace the DWT, opening the possibility of applying our multi-pooling strategy in medical imaging. We also plan to test the multi-pooling scheme to improve the results of other pooling techniques. For instance, by replacing max-pooling with average pooling in Fig. 4. Also, we intend to attach our multi-pooling scheme to other segmentation architectures such as U-Net, DeepLabv3 (Chen et al., 2017), among others.

CRediT authorship contribution statement

André de Souza Brito: Conceptualization, Methodology, Software. **Marcelo Bernardes Vieira:** Formal analysis, Writing - original draft, Writing - review & editing. **Mauren Louise Sguario Coelho de Andrade:** Visualization, Writing - review & editing. **Raul Queiroz Feitosa:** Formal analysis, Validation, Writing - review & editing. **Gilson Antonio Giraldi:** Conceptualization, Writing - original draft, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank the financial support provided by PCI-LNCC, Brazil (grant 444331/2018-2).

References

Akhtar, N., & Ragavendran, U. (2020). Interpretation of intelligence in CNN-pooling processes: a methodological survey. *Neural Computing and Applications*, *32*, 879–898. http://dx.doi.org/10.1007/s00521-019-04296-5.

Alizadeh, R., Allen, J. K., & Mistree, F. (2020). Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design*, *31*, 275–298. http://dx.doi.org/10.1007/s00163-020-00336-7.

Alizadeh, R., Beiragh, R. G., Soltanisehat, L., Soltanzadeh, E., & Lund, P. D. (2020). Performance evaluation of complex electricity generation systems: a dynamic network-based data envelopment analysis approach. *Energy Economics*, *91*, Article 104894. http://dx.doi.org/10.1016/j.eneco.2020.104894.

Alizadeh, R., Jia, L., Nellippallil, A. B., Wang, G., Hao, J., Allen, J. K., & Mistree, F. (2019). Ensemble of surrogates and cross-validation for rapid and accurate predictions using small data sets. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *33*, 484–501. http://dx.doi.org/10.1017/S089006041900026X.

Alizadeh, R., Soltanisehat, L., Lund, P., & Zamanisabzi, H. (2020). Improving renewable energy policy planning and decision-making through a hybrid MCDM method. *Energy Policy*, *137*, Article 111174. http://dx.doi.org/10.1016/j.enpol.2019.111174.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*, 2481–2495. http://dx.doi.org/10.1109/TPAMI.2016.2644615.

Bae, W., Yoo, J., & Chul Ye, J. (2017). Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 1141–1149). http://dx.doi.org/10.1109/CVPRW.2017.152.

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., & Gould, S. (2016). Dynamic image networks for action recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3034–3042). http://dx.doi.org/10.1109/CVPR.2016.331.

Bischke, B., Helber, P., Borth, D., & Dengel, A. (2018). Segmentation of imbalanced classes in satellite imagery using adaptive uncertainty weighted class loss. In *Proceedings of the IGARSS 2018—2018 IEEE international geoscience and remote sensing symposium* (pp. 6191–6194).

Boureau, Y.-L., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 111–118).

Bulo, S. R., Neuhold, G., & Kontschieder, P. (2017). Loss max-pooling for semantic image segmentation. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 7082–7091).

Chai, D. F., Newsam, S., & Huang, J. F. (2020). Aerial image semantic segmentation using DCNN predicted distance maps. *ISPRS Journal of Photogrammetry and Remote Sensing*, *161*, 309–322. http://dx.doi.org/10.1016/j.isprsjprs.2020.01.023.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*, 834–848. http://dx.doi.org/10.1109/TPAMI.2017.2699184.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. CoRR abs/1706.05587. URL: http://arxiv.org/abs/1706.05587. arXiv:1706.05587.

Chen, G., Zhang, X., Wang, Q., Dai, F., Gong, Y., & Zhu, K. (2018). Symmetrical dense-shortcut deep fully convolutional networks for semantic segmentation of very-high-resolution remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*, 1633–1644. http://dx.doi.org/10.1109/JSTARS.2018.2810320.

Dong, R., Bai, L., & Li, F. (2020). SiameseDenseU-Net-based semantic segmentation of urban remote sensing images. *Mathematical Problems in Engineering*, *2020*, 1–14. http://dx.doi.org/10.1155/2020/1515630.

Dong, S., & Chen, Z. (2021). A multi-level feature fusion network for remote sensing image segmentation. *Sensors*, *21*, 1267. http://dx.doi.org/10.3390/s21041267.

Dong, R., Pan, X., & Li, F. (2019). Denseu-net-based semantic segmentation of small objects in urban remote sensing images. *IEEE Access*, *7*, 65347–65356. http://dx.doi.org/10.1109/ACCESS.2019.2917952.

Druzhkov, P. N., & Kustikova, V. D. (2016). A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, *26*, 9–15. http://dx.doi.org/10.1134/S1054661816010065.

Duan, Y., Liu, F., Jiao, L., Zhao, P., & Zhang, L. (2017). SAR image segmentation based on convolutional-wavelet neural network and markov random field. *Pattern Recognition*, *64*, 255–267. http://dx.doi.org/10.1016/j.patcog.2016.11.015.

Farhangfar, S., & Rezaeian, M. (2019). Semantic segmentation of aerial images using FCN-based network. In *2019 27th Iranian conference on electrical engineering (ICEE)* (pp. 1864–1868).

Gerke, M. (2014). *Use of the stair vision library within the ISPRS 2D semantic labeling benchmark (vaihingen): Technical Report*, ITC, University of Twent, http://dx.doi.org/10.13140/2.1.5015.9683.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Guo, Y., Liao, J., & Shen, G. (2021). A deep learning model with capsules embedded for high-resolution image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 214–223. http://dx.doi.org/10.1109/JSTARS.2020.3032672.

Guo, T., Mousavi, H. S., Vu, T. H., & Monga, V. (2017). Deep wavelet prediction for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 1100–1109). http://dx.doi.org/10.1109/CVPRW.2017.148.

Han, Y., & Ye, J. C. (2018). Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE Transactions on Medical Imaging*, *37*, 1418–1429. http://dx.doi.org/10.1109/TMI.2018.2823768.

He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5353–5360). http://dx.doi.org/10.1109/CVPR.2015.7299173.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the international conference on computer vision (ICCV)* (pp. 1026–1034). http://dx.doi.org/10.1109/ICCV.2015.123.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). http://dx.doi.org/10.1109/CVPR.2016.90.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on machine learning (ICML)* (pp. 448–456).

Jadon, S. (2020). A survey of loss functions for semantic segmentation. In *Proceedings of the conference on computational intelligence in bioinformatics and computational biology (CIBCB)* (pp. 1–7).

Jia, L., Alizadeh, R., Hao, J., Wang, G., Allen, J. K., & Mistree, F. (2020). A rule-based method for automated surrogate model selection. *Advanced Engineering Informatics*, *45*, Article 101123. http://dx.doi.org/10.1016/j.aei.2020.101123.

Kampffmeyer, M., Salberg, A.-B., & Jenssen, R. (2016). Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 680–688). http://dx.doi.org/10.1109/CVPRW.2016.90.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*, 5455–5516. http://dx.doi.org/10.1007/s10462-020-09825-6.

Li, Q., & Shen, L. (2020). Wavesnet: Wavelet integrated deep networks for image segmentation. CoRR abs/2005.14461. URL: https://arxiv.org/abs/2005.14461. arXiv:2005.14461.

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. In *Proceedings of the international conference on learning representations*. URL: http://arxiv.org/abs/1312.4400.

Liu, Y., Nguyen, D. M., Deligiannis, N., Ding, W., & Munteanu, A. (2017). Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery. *Remote Sensing*, *9*(522).

Liu, P., Zhang, H., Lian, W., & Zuo, W. (2019). Multi-level wavelet convolutional neural networks. *IEEE Access*, *7*, 74973–74985. http://dx.doi.org/10.1109/ACCESS.2019.2921451.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3431–3440). http://dx.doi.org/10.1109/CVPR.2015.7298965.

Lu, H., Wang, H., Zhang, Q., Won, D., & Yoon, S. W. (2018). A dual-tree complex wavelet transform based convolutional neural network for human thyroid medical image segmentation. In *Proceedings of the IEEE international conference on healthcare informatics (ICHI)* (pp. 191–198). http://dx.doi.org/10.1109/ICHI.2018.00029.

Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*, 674–693. http://dx.doi.org/10.1109/34.192463.

Maune, D. F. (2007). *Digital elevation model technologies and applications: The DEM users manual* (2nd ed.). Bethesda, Md: American Society for Photogrammetry and Remote Sensing.

Meyer, Y. (1992). *Wavelets and operators, Vol. 1*. Cambridge University Press.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–23. http://dx.doi.org/10.1109/TPAMI.2021.3059968.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML)* (pp.807–814).

Ni, Z.-L., Bian, G.-B., Xie, X.-L., Hou, Z.-G., Zhou, X.-H., & Zhou, Y.-J. (2019). RASNet: Segmentation for tracking surgical instruments in surgical videos using refined attention segmentation network. In *2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC)* (pp. 5735–5738).

Oliveira, A., Pereira, S., & Silva, C. A. (2018). Retinal vessel segmentation based on fully convolutional neural networks. *Expert Systems with Applications*, *112*, 229–242. http://dx.doi.org/10.1016/j.eswa.2018.06.034.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035).

Peker, M. (2021). Classification of hyperspectral imagery using a fully complex-valued wavelet neural network with deep convolutional features. *Expert Systems with Applications*, *173*, Article 114708. http://dx.doi.org/10.1016/j.eswa.2021.114708.

Piccialli, F., Somma, V. D., Giampaolo, F., Cuomo, S., & Fortino, G. (2021). A survey on deep learning in medicine: Why, how and when?. *Information Fusion*, *66*, 111–137. http://dx.doi.org/10.1016/j.inffus.2020.09.006.

Ramanarayanan, S., Murugesan, B., Ram, K., & Sivaprakasam, M. (2020). DC-WCNN: A deep cascade of wavelet based convolutional neural networks for MR image reconstruction. In *Proceedings of the 17th IEEE international symposium on biomedical imaging (ISBI)* (pp. 1069–1073). http://dx.doi.org/10.1109/ISBI45749.2020.9098491.

Rippel, O., Snoek, J., & Adams, R. P. (2015). Spectral representations for convolutional neural networks. In *Proceedings of the 28th international conference on neural information processing systems (NIPS), Vol. 2* (pp. 2449–2457).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th international conference of medical image computing and computer-assisted intervention (MICCAI)* (pp. 234–241). http://dx.doi.org/10.1007/978-3-319-24574-4_28.

Rottensteiner, F., Sohn, G., Gerke, M., & Wegner, J. D. (2014). *ISPRS semantic labeling contest*. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html. Accessed on August 20, 2020.

Saeedan, F., Weber, N., Goesele, M., & Roth, S. (2018). Detail-preserving pooling in deep networks. In *2018 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9108–9116). http://dx.doi.org/10.1109/CVPR.2018.00949.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th international conference on neural information processing systems (NIPS)* (pp. 568–576).

Soltanisehat, L., Alizadeh, R., Hao, H., & Choo, K. R. (2020). Technical, temporal, and spatial research challenges and opportunities in blockchain-based healthcare: A systematic literature review. *IEEE Transactions on Engineering Management*, 1–16. http://dx.doi.org/10.1109/TEM.2020.3013507.

Takikawa, T., Acuna, D., Jampani, V., & Fidler, S. (2019). Gated-SCNN: Gated shape CNNs for semantic segmentation. In *2019 IEEE/CVF international conference on computer vision (ICCV)* (pp. 5228–5237). http://dx.doi.org/10.1109/ICCV.2019.00533.

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., & Lin, C.-W. (2020). Deep learning on image denoising: An overview. *Neural Networks*, *131*, 251–275. http://dx.doi.org/10.1016/j.neunet.2020.07.025.

Ulku, I., & Akanguduz, E. (2019). A survey on deep learning-based architectures for semantic segmentation on 2D images. CoRR abs/1912.10230 URL: http://arxiv.org/abs/1912.10230. arXiv:1912.10230.

Volpi, M., & Tuia, D. (2017). Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *55*, 881–893. http://dx.doi.org/10.1109/TGRS.2016.2616585.

Wang, Z., Chen, J., & Hoi, S. C. H. (2020). Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–22. http://dx.doi.org/10.1109/TPAMI.2020.2982166.

Wang, F., Huang, S., Shi, L., & Fan, W. (2017). The application of series multi-pooling convolutional neural networks for medical image segmentation. *International Journal of Distributed Sensor Networks*, *13*, http://dx.doi.org/10.1177/1550147717748899.

Wang, P., Li, W., Gao, Z., Tang, C., & Ogunbona, P. O. (2018). Depth pooling based large-scale 3D action recognition with convolutional neural networks. *IEEE Transactions on Multimedia*, *20*, 1051–1061. http://dx.doi.org/10.1109/TMM.2018.2818329.

Williams, T., & Li, R. (2018). Wavelet pooling for convolutional neural networks. In *Proceedings of the 6th international conference on learning representations (ICLR)*. OpenReview.net, URL: https://openreview.net/forum?id=rkhlb8lCZ.

Wu, Z., Jiang, Y., Wang, X., Ye, H., Xue, X., & Wang, J. (2015). Fusing multi-stream deep networks for video classification. CoRR abs/1509.06086. URL: http://arxiv.org/abs/1509.06086. arXiv:1509.06086.

Yazdizadeh, A., Patterson, Z., & Farooq, B. (2020). Ensemble convolutional neural networks for mode inference in smartphone travel survey. *IEEE Transactions on Intelligent Transportation Systems*, *21*, 2232–2239. http://dx.doi.org/10.1109/TITS.2019.2918923.

Yuan, X., Shi, J., & Gu, L. (2021). A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, *169*, Article 114417. http://dx.doi.org/10.1016/j.eswa.2020.114417.

Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. In *Proceedings of the 1st international conference on learning representations (ICLR)* (pp. 1–9). URL: http://arxiv.org/abs/1301.3557.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the 13th European conference on computer vision (ECCV), Vol. 8689* (pp. 818–833). http://dx.doi.org/10.1007/978-3-319-10590-1_53.

Zhang, B., Kong, Y., Leung, H., & Xing, S. (2019). Urban UAV images semantic segmentation based on fully convolutional networks with digital surface models. In *2019 tenth international conference on intelligent control and information processing (ICICIP)* (pp. 1–6). http://dx.doi.org/10.1109/ICICIP47338.2019.9012207.

Zhang, N., Liu, J., Wang, K., Zeng, D., & Mei, T. (2020). Robust visual object tracking with two-stream residual convolutional networks. CoRR abs/2005.06536. URL: https://arxiv.org/abs/2005.06536. arXiv:2005.06536.