remote sensing

MDPI

Technical Note

MARE: Self-Supervised Multi-Attention REsu-Net for Semantic Segmentation in Remote Sensing

Valerio Marsocci [1,*], Simone Scardapane [2] and Nikos Komodakis [3]

[1] Department of Computer, Control and Management Engineering Antonio Ruberti (DIAG), Sapienza University of Rome, 00185 Rome, Italy
[2] Department of Information Engineering, Electronics and Telecommunication, Sapienza University of Rome, 00184 Rome, Italy; simone.scardapane@uniroma1.it
[3] Computer Science Department, University of Crete, 70013 Heraklion, Greece; komod@csd.uoc.gr
* Correspondence: valerio.marsocci@uniroma1.it

Abstract: Scene understanding of satellite and aerial images is a pivotal task in various remote sensing (RS) practices, such as land cover and urban development monitoring. In recent years, neural networks have become a de-facto standard in many of these applications. However, semantic segmentation still remains a challenging task. With respect to other computer vision (CV) areas, in RS large labeled datasets are not very often available, due to their large cost and to the required manpower. On the other hand, self-supervised learning (SSL) is earning more and more interest in CV, reaching state-of-the-art in several tasks. In spite of this, most SSL models, pretrained on huge datasets like ImageNet, do not perform particularly well on RS data. For this reason, we propose a combination of a SSL algorithm (particularly, Online Bag of Words) and a semantic segmentation algorithm, shaped for aerial images (namely, Multistage Attention ResU-Net), to show new encouraging results (i.e., 81.76% mIoU with ResNet-18 backbone) on the ISPRS Vaihingen dataset.

Keywords: semantic segmentation; self-supervised learning; linear attention; Vaihingen dataset

Citation: Marsocci, V.; Scardapane, S.; Komodakis, N. MARE: Self-Supervised Multi-Attention REsu-Net for Semantic Segmentation in Remote Sensing. *Remote Sens.* **2021**, *13*, 3275. https://doi.org/10.3390/rs13163275

Academic Editors: Amir Hussain, Ahmed Al-Dubai, William (Bill) J Buchanan, Jonathan Wu, Kaizhu Huang, Bin Luo, Jin Tang, Wadii Boulila and Adel M. Alimi

Received: 5 July 2021
Accepted: 14 August 2021
Published: 19 August 2021

1. Introduction

In a wide range of real-world applications, varying from urban planning to precision agriculture, including land cover, infrastructure management and so on, semantic segmentation of aerial and remote sensing (RS) imageries is a pivotal task, which continues to attract great interest [1]. Semantic labeling, that consists in assigning a category to every pixel of the image, is particularly challenging in urban applications [2,3]. In fact, the complicated urban structure leads to interactions among objects, causing occlusions, shadows, and other noisy effects, which worsen the radiometric information of the images [4]. Moreover, artificial manufacts have at least two main issues. On the one hand, objects belonging to different classes could retrieve very similar radiometric information (e.g., low vegetation and trees, or roads and pavements). On the other hand, manufacts of the same semantic class can present very different characteristics, such as color, texture, and shape [5]. Eventually, it could be asserted that semantic segmentation in urban scenes is characterized by a strong intra-class variance and a reduced inter-class one [6,7].

In recent years, deep learning (DL) methods reached state-of-the-art, surpassing traditional methods, in several computer vision (CV) tasks. Many of these find large application also in RS, such as object detection, instance segmentation, and semantic labeling. It is unanimously recognized that DL methods have high generalization capabilities, succeeding in extracting robust and efficient features [8,9]. However, supervised training of these models depends on annotations. More than in other fields, for aerial and RS images, it is difficult to rely on a labeled dataset, in light of the high cost and the amount of effort and time that are required, along with a well founded expertise.

The advent of self-supervised learning (SSL) could handle this problem, reducing the amount of annotated data needed [10–12]. The goal of SSL is to learn an effective visual representation of the input using a massive quantity of data provided without any label [11,13]. To solve this task in CV, we can see the problem as the need to build a well-structured and relevant set of features, able to represent an object in a fruitful way for several downstream tasks. Thus, intelligent systems need not to be structured a priori, but should instead learn about the conformation of the provided data in an unsupervised way [14]. In particular, the majority of unsupervised approaches can be divided into two classes: generative and discriminative. Generative strategies learn to generate pixels in the input space. However, pixel-level generation can be computationally expensive. Discriminative approaches are based on learning the representation of the unlabeled data to solve downstream tasks. Among these techniques, in contrastive learning the goal is to generate a representation of the input, such that similar instances are near each other and far from dissimilar ones [13,15]. Other approaches fulfill the goal of reconstructing a target image, after perturbing it [16,17]. Essentially, SSL is earning more and more interest in CV, reaching state-of-the-art performances in several tasks.

In this technical letter, we propose a new approach, that combines a SSL algorithm (that is, online bag of words [12]) and a semantic segmentation algorithm shaped for aerial images (properly, Multistage Attention ResU-Net [18]), based on a linear attention mechanism (LAM). Particularly, we decided to train the encoder of the Multistage Attention ResU-Net (MAResU-Net), that is a ResNet-18 [19], with the Online Bag of Words (OBoW) method, given its high capability of learning visual representations, that are effective for several downstream task. This characteristic is fundamental, because most of available SSL pretrained models do not perform particularly well on RS data. In fact, due to the problems that we addressed previously, aerial and RS imageries have peculiar characteristics, different from the images taken with close-range cameras. Indeed, unlike, for example, the ImageNet dataset, the RS images present an enormous amount of details to look after, to correctly segment the objects. Moreover, in addition to the aforementioned variance issues, the shapes, the colors and the scale of the objects are crucial in this task [20]. In particular, the edges must have high consideration, as they are changeable and labile much more than in close-range camera images [7].

On the other hand, attention exploits the ability of grasping long-term dependencies of the feature maps, exploring global contextual information of the data. Dot-product attention mechanism, generating response at each pixel by weighting features in the previous layers, expands the receptive field to the whole input feature maps, reaching state-of-the-art performance in many fields [21–23]. However, the memory and computational overhead of the dot-product attention mechanism increases quadratically along with the spatio-temporal size of the input. To alleviate the huge computational costs, ref [24] reduced the complexity from $O(N^2)$ to $O(N\sqrt{N})$; ref [25] to $O(N \log N)$; and [18,26,27] to $O(N)$.

To sum up, the three major contributions offered by the proposed technical letter are the following:

- we show how the SSL approach, designed for generic CV downstream tasks, performs well even on more sectorial areas of image analysis, such as RS;
- we highlight that SSL methods are effective to reduce dependence on well-annotated dataset, currently required to reach high performances in RS tasks, since they require high costs and great need of time, effort, and knowledge to be produced;
- we obtain the best results in literature for the semantic segmentation of the ISPRS Vaihingen benchmark dataset [28] with ResNet-18 as encoder. This confirms an excellent trade-off between the number of parameters and the performance of the model.

The remainder of this technical note is organized as follows: in Section 2 the related works, divided in three subsections, concerning, respectively, semantic segmentation (Section 2.1), semantic segmentation for aerial and RS imageries (Section 2.2) and SSL (Section 2.3), are illustrated; in Section 3 OBoW and MAResU-Net methodologies are briefly explained; in Section 4 the experimental results and the ablation studies are presented.

Finally, in Section 5, a discussion of the results, including the limitations of the proposed strategy, and further developments are dispensed.

2. Related Works

2.1. Semantic Segmentation

Fully convolutional network (FCN) methods have experimented huge progresses in semantic segmentation, following mainly two approaches. On the one hand, dilated convolutions [29,30] have established a strong capability to retain the receptive field-of-view and enhance the performance of the backbone. On the other hand, the encoder–decoder architectures utilize an encoder to obtain multi-level feature maps, which are then incorporated into the final prediction through a decoder [31]. These two strategies can also be combined. An effective example is PSPNet [32], which adopts a pyramid parsing module that exploits global context information by different region-based context aggregations. The local and global clues, concatenated together, make the final prediction more performing.

In addition, several architectures, based on the attention mechanism, often combined with what has been described so far, have been proposed. For example, DANet [23] integrates local features with global dependencies, in an adaptive way. Specifically, the architecture provides two types of attention modules on top of traditional dilated FCN, which model the semantic inter-dependencies, respectively, in spatial and channel dimensions, regardless of their distances. The outputs of these two attention modules are finally summed for the prediction. Other examples of this set of architectures are: PSANet [33], OCNet [34], and CFNet [35].

2.2. Semantic Segmentation for Remote Sensing and Aerial Images

In the last few years, several methodologies specifically shaped for RS images have been proposed. These methods succeed in reaching better performance on this specific task, thanks to their capability of taking in account the variable shapes, scale and edges of the represented objects. In fact, in light of the massive quantity of details in a RS image, the CV semantic segmentation methods often fragment one object into pieces, or confuse adjacent objects, thus failing to segment these objects correctly [20].

Particularly, ResUNet-a [7] uses a U-Net encoder-decoder structure, in combination with residual connections, dilated convolutions, pyramid scene parsing pooling and multi-tasking inference. ResUNet-a infers sequentially the boundaries of the objects, the distance transformation of the segmentation mask, the segmentation mask, and a colored reconstruction of the input. In addition, the authors introduce a novel loss function, modifying the dice loss.

EaNet [20] incorporates a large kernel pyramid pooling module to capture multi-scale context with robust continuous feature relations. An edge-aware loss function (EA loss), based on the dice loss, is presented to guide the EaNet to refine both the pixel-level and context-level information directly from the semantic segmentation prediction.

CE-Net [36] mainly consists of three parts: a ResNet encoder, a context extractor and a decoder module. The context extractor module is formed by a novel dense dilated convolution block and a residual multi-kernel pooling block.

A solution conceived through the use of a transformer has also been recently proposed. Ref. [37] proposes the Swin Transformer [38] as the backbone to extract the context information and designs a decoder of densely connected feature aggregation modules to produce the segmentation map, after restoring the resolution of the input.

All the presented architectures are supervised, that means that are in need of labeled data. The only effort in the direction of SSL applied to RS is presented in [39]. Namely shaped for change detection, the authors propose a self-supervised approach capable to capture better representation for semantic understanding of RS images. Specifically, the network, imitating the discriminator of a generative adversarial network (GAN), is asked to identify different sample patches taken from two temporal images.

2.3. Self-Supervised Learning

Initially, most SSL methods were based on pre-text tasks. The strategies were several, such as patch context [40,41], in-painting [42], colorization [43,44], jigsaw puzzles [45], noise [46], generation [47], rotation [16]), and counting [48].

On the other hand, contrastive-based approaches were proposed. In [11], the authors propose a contrastive framework under which several other algorithms (e.g., Augmented Multiscale Deep InfoMax, i.e., AMDIM [49]; Contrastive Predicting Coding, i.e., CPC [50]; and a simple framework for contrastive learning of visual representations, i.e., SimCLR [13]) can be considered special cases. Yet Another Deep InfoMax (YADIM) [11] is characterized by five parts: data augmentation, needed to generate the anchor, the positive and the negative instances; the encoder, generally a ResNet [19]; the representation extractor, to compare two or more representations [13,50,51]; the similarity measure (e.g., dot product [49,50], cosine similarity [13,51], or bi-linear transformation); and the loss function (e.g., negative contrastive estimation (NCE) [52], triplet loss [53], and InfoNCE [14]). Outside the YADIM framework, there are many other effective approaches. For example, ref [15] proposes a momentum contrast (MoCo) architecture. MoCo uses a moving average network to maintain an effective representation of negative samples taken from a memory bank. Another approach is the one proposed in [54]. Contrastive Multiview Coding (CMC) learns a representation that maximizes the mutual information among various views of the same scene. Barlow Twins [10] proposes an objective function that avoids collapses by measuring the cross-correlation between the outputs of two identical networks fed with distorted versions of a sample.

Nevertheless, to address some of the limitations of contrastive-based approaches (e.g., need for large batch sizes [51] or pairwise comparison), most recently teacher-student (e.g., OBoW [12]), as well as clustering-based (e.g., DeepCluster [55]) approaches have been proposed. For example, Bootstrap Your Own Latent (BYOL) [56] uses a moving average network to produce prediction targets as a mean of stabilizing the bootstrap step. SimSiam [57] maximizes the similarity between two augmentations of one image, using a Siamese network. Swapping Assignments between multiple Views (SwAV) [58] predicts the cluster assignment of a view from the representation of another view of the same image.

3. Methodology

To deal with the challenge of limited annotated training data for RS segmentation, we rely on SSL to learn powerful representations, that can tap on the potential of the large amount of unlabeled data, readily available in RS. Particularly, we decided to use OBoW [12], because it exploits the use of visual words, which are visual concepts localized in the spatial domain (as opposed to global concepts as in most other SSL methods). This could be beneficial for dense prediction tasks, such as semantic segmentation. Furthermore, it exhibits very strong empirical performance. On the other hand, we decided to rely on MAResU-Net, for the semantic segmentation task, because of several reasons. U-Net-based architectures have proven to be an excellent choice for image segmentation tasks. Moreover, the use of a self-attention mechanism has shown to provide high-capacity models that can properly take advantage of large scale datasets. Finally, to deal with the high computational cost of self-attention, we extend the solution proposed by MAResU-Net.

In the following subsections, we provide details about the chosen architectures.

3.1. Self-Supervised Learning for Remote Sensing Using Online Bag of Visual Words

In [17], the authors propose BoWNet, which offers the idea of using Bag of Visual Words (BoW) as targets for SSL. This approach, despite its effectiveness and innovativeness, had some limitations, such as a static visual words vocabulary.

These limits were tackled in [12], where the authors propose an improved solution, that is OBoW.

The BoW reconstruction task involves a student convolutional neural network (CNN) $S(\cdot)$ that learns image representations, and a teacher CNN $T(\cdot)$ that generates BoW targets

used for training the student network. The student $\mathrm{S}(\cdot)$ is parameterized by θ_S and the teacher $\mathrm{T}(\cdot)$ by θ_T.

Inspired by [15], the parameters θ_T of $\mathrm{T}(\cdot)$ are an exponential moving average of the student parameters. As a consequence, the teacher has the same architecture as the student, though maintaining different batch-norm statistics.

To generate a BoW representation $y_T(\mathbf{x})$ out of an image x, the teacher first extracts the feature map $T^l(\mathbf{x}) \in \mathbb{R}^{c_l \times h_l \times w_l}$, of spatial size $h_l \times w_l$ with c_l channels, from its last layer l. It quantizes the c_l-dimensional feature vectors $T^l(\mathbf{x})[u]$ at each location $u \in 1, \ldots, h_l \times w_l$ of the feature map over a vocabulary $V = [\mathbf{v}_1; \ldots; \mathbf{v}_K]$ of K visual words of dimension c_l. The vocabulary V of visual words is a K-sized queue of random features. At each step, after computing the assignment codes over the vocabulary V, V is updated by selecting one feature vector per image from the current mini-batch, removing the oldest item in the queue if its size exceeds K. The feature selection consists of a local average pooling with a 3×3 kernel of the feature map $T^l(\mathbf{x})$ followed by a uniform random sampling of one of the resulting feature vectors. Thus, assuming that the local features in a 3×3 neighborhood belong to one common visual concept, local averaging selects a representative visual-word feature from this neighborhood. This quantization process produces for each location u a K-dimensional code vector $q(\mathbf{x})[u]$ that encodes the assignment of $T(\mathbf{x})[u]$ to its closest visual word. Then, the teacher reduces the quantized feature maps $q(\mathbf{x})$ to a K-dimensional BoW $\widetilde{y}_T(\mathbf{x})$ by channel-wise max-pooling. For this step, a soft-assignment is preferable due to the fact that the vocabulary of visual words is continuously evolving. The soft assignment depends on an adaptive parameter δ. Finally, $\widetilde{y}_T(\mathbf{x})$ is converted into a probability distribution over the visual words by L_1-normalization, i.e.,

$$y_{\mathrm{T}}(\mathbf{x})[k] = \frac{\tilde{y}_{\mathrm{T}}(\mathbf{x})[k]}{\sum_{k'} \tilde{y}_{\mathrm{T}}(\mathbf{x})[k']} \tag{1}$$

To learn effective image representations, the student must predict the BoW distribution over V of an image using as input a perturbed version of that same image. In OBoW, the vocabulary is constantly updated. Therefore, a dynamic BoW-prediction head that can adapt to the evolving nature of the vocabulary is proposed. To that end, the authors employ a generation network $G(\cdot)$ that takes as input the current vocabulary of visual words V and produces prediction weights for them as $G(V) = [G(\mathbf{v}_1); \ldots; G(\mathbf{v}_K)]$, where $G(\cdot)$: $\mathbb{R}^{c_l} \rightarrow \mathbb{R}^c$ consists in a 2-layer multilayer perceptron (MLP) whose input and output vectors are $l2$-normalized and $G(\mathbf{v}_k)$ represents the prediction weight vector for the kth visual word. Therefore, $\widetilde{y}_S(\mathbf{x})$ is computed as follows:

$$y_{\mathrm{S}}(\tilde{\mathbf{x}})[k] = \frac{\exp\left(\kappa \cdot \mathrm{G}(\mathbf{v}_k)^\top \mathrm{S}(\tilde{\mathbf{x}})\right)}{\sum_{k'} \exp\left(\kappa \cdot \mathrm{G}(\mathbf{v}_{k'})^\top \mathrm{S}(\tilde{\mathbf{x}})\right)} \tag{2}$$

where κ is a fixed coefficient that equally scales the magnitudes of all the predicted weights $G(V)$. The K-dimensional vector $y_{\mathrm{S}}(\tilde{\mathbf{x}})[k]$ is the predicted softmax probability of the target $y_T(\mathbf{x})$. Hence, the training loss that is minimized for a single image $\mathbf{x}$ is the cross entropy between the softmax distribution $y_{\mathrm{S}}(\tilde{\mathbf{x}})[k]$ predicted by the student from the perturbed image $\widetilde{\mathbf{x}}$, and the BoW distribution $y_T(\mathbf{x})$ of the unperturbed image $\mathbf{x}$ given by the teacher.

The architecture described so far is represented in Figure 1.

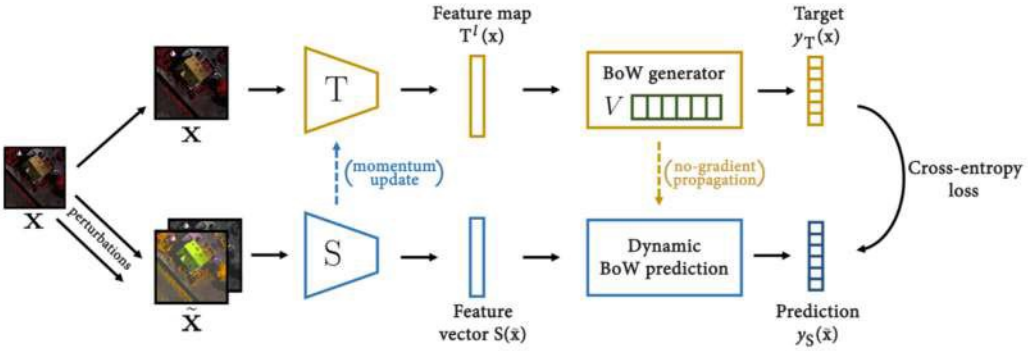

Figure 1. OBoW architecture.

3.2. MAResU-Net

MAResU-Net proposes a modified U-Net architecture. In fact, first of all, the encoder S is made up of a ResNet. Moreover, the blocks of the encoder S and the decoder D are not simple skip connections, but are replaced by attention modules. To reduce the computational times, these modules combine a conventional attention mechanism and a linear one, i.e. LAM.

The loss function follows the formula:

$$L = -\frac{1}{M}\sum_{c=1}^{C}\sum_{m=1}^{M} w_c \cdot y_m^c \cdot \log(h_\theta(x_m, c)) \tag{3}$$

where M is the number of training examples, C the number of classes, y_m^c the target label for training example m of class c, w_c the weight for the class c, x_m the input for the training example m and h_θ the model with the weights θ.

Particularly, if all the w_c are set to 1 and soft assignment is performed, the loss becomes a soft categorical cross entropy (SCE), otherwise it is a weighted categorical cross entropy (WCE).

Eventually, the LAM is presented in the next Section 3.2.1, while the whole architecture is shown in Figure 2.

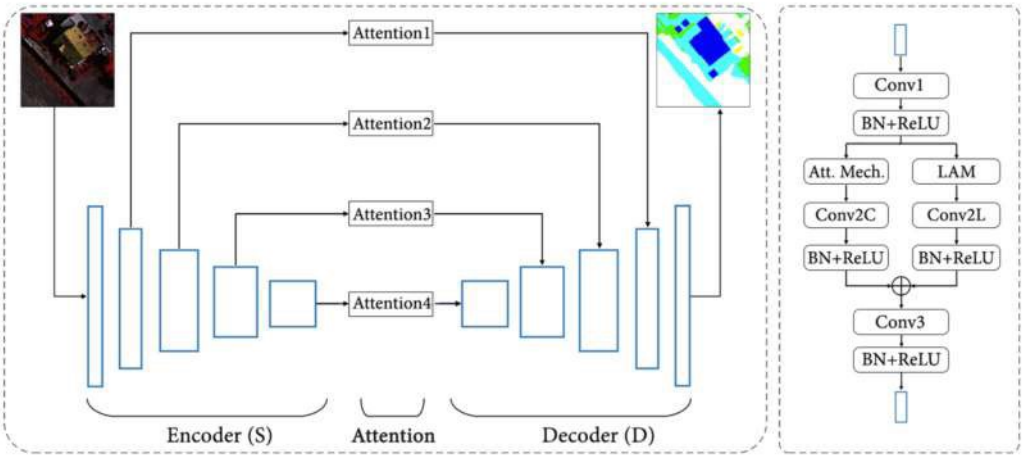

Figure 2. MAResU-Net architecture (**left**) and MAResU-Net attention block (**right**).

3.2.1. Linear Attention Mechanism

Providing N and C as the length of input sequences and the number of input channels, where $N = H \times W$, with H and W the height and width of the input, with the input feature $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N \times C}$, dot-product attention utilizes three projected matrices $\mathbf{W}_q \in \mathbb{R}^{D_x \times D_q}$, $\mathbf{W}_k \in \mathbb{R}^{D_x \times D_k}$, and $\mathbf{W}_v \in \mathbb{R}^{D_x \times D_v}$ to generate the corresponding query

matrix $\mathbf{Q}$, the key matrix $\mathbf{K}$, and the value matrix $\mathbf{V}$. The attention is, according to [21], computed as follows:

$$D(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{4}$$

where d_k is a scale factor. As $\mathbf{Q} \in \mathbb{R}^{N\times D_k}$ and $\mathbf{K}^T \in \mathbb{R}^{D_k\times N}$, the product between $\mathbf{Q}$ and $\mathbf{K}^T$ belongs to $\mathbb{R}^{N\times N}$, which leads to $O(N_2)$ memory and computational complexity. Thus, the i_{th} row of result matrix generated by the dot-product attention module can be written as:

$$D(\mathbf{Q},\mathbf{K},\mathbf{V})_i = \frac{\sum_{j=1}^{N} e^{\mathbf{q}_i^T\mathbf{k}_j}\mathbf{v}_j}{\sum_{j=1}^{N} e^{\mathbf{q}_i^T\mathbf{k}_j}} \tag{5}$$

First generalizing (5), then approximating $e^{\mathbf{q}_i^T\mathbf{k}_j}$ with first-order Taylor expansion, finally $l2$-normalizing the resulting equation, (5) can be written as:

$$D(\mathbf{Q},\mathbf{K},\mathbf{V}) = \frac{\sum_j \mathbf{V}_{i,j} + \left(\frac{\mathbf{Q}}{\|\mathbf{Q}\|_2}\right)^T\left(\left(\frac{\mathbf{K}}{\|\mathbf{K}\|_2}\right)^T\mathbf{V}\right)}{N + \left(\frac{\mathbf{Q}}{\|\mathbf{Q}\|_2}\right)\sum_j\left(\frac{\mathbf{K}}{\|\mathbf{K}\|_2}\right)^T_{i,j}} \tag{6}$$

As $\sum_{j=1}^{N}\left(\left(\mathbf{k}_j/\left(\|\mathbf{k}_j\|_2\right)\right)\mathbf{v}_j^T\right.$ and $\left.\sum_{j=1}^{N}\left((\mathbf{k}_j)/\left(\|\mathbf{k}_j\|_2\right)\right)\right.$ is common to every query, complexity of this linear attention mechanism is $O(N)$. Nevertheless, for the channel dimension dot-product-based attention is computed, considering that the channels of the input C are much less than the pixels. Thus, by taking the ResNet as the backbone, MAResU-Net combines low and high-level feature maps through attention block in multiple stages.

4. Experimental Results

As previously affirmed, we combined OBoW and MAResU-Net, in order to enhance semantic segmentation performances, and validated this intuition on ISPRS Vaihingen benchmark dataset [28]. The general strategy adopted to determine the training process, and summed up in Table 1, is as follows:

- For the OBoW training, we have followed the procedure presented in [12]. We have, thus, selected the hyperparameters that the authors utilized in the semantic segmentation downstream task;
- For the MAResU-Net training, we started from the configuration provided in [18]. First of all, we made some considerations on some issues, concerning the activation function and the loss, then, we explored different sets of hyperparameters, following different intuitions, as we discuss in Section 4.3.

After briefly presenting the data in Section 4.1, we go in the deep on the experimental results in Section 4.2. Finally, the ablation studies are presented in Section 4.3.

4.1. Vaihingen Dataset

To evaluate the performances of the presented strategy, we chose the benchmark dataset of Vaihingen, provided by ISPRS [28]. The dataset contains 33 tiles of variable sizes, each consisting of a true orthophoto (TOP). The ground sampling distance of the images is 9 cm. The images are 8 bit TIFF with three bands, corresponding to near infrared (NIR), red (R), and green (G). The corresponding masks are divided into six classes: background, impervious surface, car, building, low vegetation and trees. Following the approach of [7], we cropped each TOP in a variable number of 256 $\times$ 256 overlapping patches. We ended up with a dataset of more than 7500 images. Then, we divided it in train and validation, respectively, 85% ($\sim$6500 images) and 15% ($\sim$1000 images) of the total.

Table 1. Summary of the training process. In (**a**) the principal training hyperparameters of the two methods are shown. Most of them trace the one presented in [12,18]. In (**b**,**c**) some details about OBoW configuration are offered. In particular, in (**c**) we show the augmentation applied for the training. In (**d**) the training times are shown.

(**a**) Training hyperparameters

Method	Batch Size	LR (final LR)	Optimizer	Scheduler	Augm.	Epochs
OBoW	256	0.3 (0.003)	Adam	Cosine	Radiometric	40
MAResU-Net	64	0.003	Adam	Step	No	150

(**b**) OBoW Configuration

Num. Image Crops	Crop Size	Num. Patches	Patch Size	$1/\delta$	Num. Words	κ
2	160×160	5 of 9	96×96	15	8192	8

(**c**) Data Augmentations application probability (p)

Transformation	p
Color Jittering	0.9
Grayscaling	0.2
Gaussian Blurring	0.7

(**d**) Training time with Tesla V100-SXM2 32 GB GPU

Method	Training Time (it/s)
OBoW	1.39
MAResU-Net	1.09

4.2. Experimental Settings

For the training phase, a single Tesla V100-SXM2 32 GB GPU has been used. Thus, we trained, through OBoW, a ResNet-18 feature extractor, used as encoder in the following MAResU-Net. With specific reference to the latter, we fine tuned the ResNet-18 encoder and trained the decoder. The overall accuracy (OA), accuracy per class, mean intersection over union (mIoU), and F1-score (F1) are the selected evaluation indexes. The background class is excluded from the evaluation, except for the OA, as in [18]. The pretrained model that reaches the best performances is available at the following link: https://github.com/VMarsocci/MARE (accessed on 6 July 2021).

4.2.1. OBoW

Among the several experiments we conducted, the best OBoW configuration, in terms of loss, was trained with a batch size of 256. As optimizer, Adam was preferred, with a cosine scheduler, with a first epoch of warmup. The learning rate started from 0.03 and finished to 0.003. Moreover, κ was set to 8 and the number of words of the dictionary is 8192. The net was trained for 40 epochs. The data have been augmented with strong radiometric transformation, consisting in: color jittering, gray-scaling, and Gaussian blurring. The input images of the student network consist in two sets of crops. The first ones are obtained performing two overlapping crops of size 160×160 on each augmented image. Finally, the second ones are 5 random patches, selected from 9 overlapping crops of size 96×96 of the input image. An example is provided in Figure 3. The training speed time was 1.38 iteration per second.

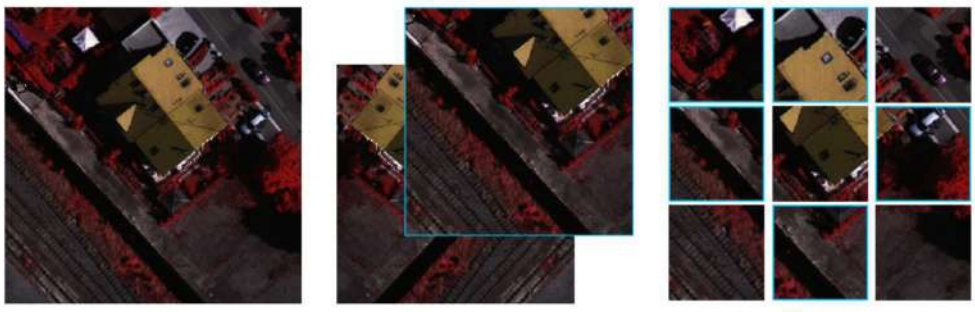

Figure 3. An example of the input images of the OBoW student network.