A Novel Boundary Loss Function in Deep Convolutional Networks to Improve the Buildings Extraction From High-Resolution Remote Sensing Images

Hamidreza Hosseinpour, Farhad Samadzadegan, and Farzaneh Dadrass Javan

Abstract—In recent years, there has been a significant increase in the production of high-resolution aerial and satellite imagery. Analyzing and extracting urban features from these images, especially building components, is a significant challenge in photogrammetry and remote sensing. Meanwhile, deep convolutional neural networks have been used as a powerful model in the semantic segmentation of building features. However, due to the structure of these types of deep convolution networks, accurate retrieval of building boundary information during training will be difficult, and this will lead to ambiguous boundary areas. On the other hand, the use of distributed-based loss functions, such as the binary cross-entropy loss alone cannot improve the segmentation accuracy in the boundary areas of building features. Accordingly, in this article, a derivative boundary loss function is introduced to optimize and enhance the extraction of buildings in border areas. To calculate the proposed boundary loss function, distance transform image (DTI) is obtained from ground-truth images and predicted images, that derived from the segmentation network. In this article, a derivative method for calculating DTI is presented. Another advantage of the proposed loss function is its ability to be used in a wide range of deep convolution segmentation networks. The proposed method was investigated and tested using the ISPRS Potsdam and Vaihingen datasets. The results of this implementation show the superiority in the criteria of evaluation and improvement of buildings extraction if the proposed boundary loss function is used along with the distributed-based loss function.

Index Terms—Boundary loss function, building extraction, deep convolution network, distance transform image (DTI).

I. INTRODUCTION

In the recent decade, the design and manufacturing of remote sensing and photogrammetric sensors, including drone, aerial, and satellite systems, has expanded. The result of this progress and advancement is the increased spatial resolution of images up to a centimeters level, accessible significantly in huge volumes and on a daily basis [1]. High spatial resolution images contain exact spatial and spectral information of the earth so that they are widely applied to a wide range of applications such as urban planning [2], accurate agriculture [3], management of environmental resources [4], monitoring of events and natural disasters [5], and three-dimensional (3-D) modeling reconstruction of buildings [6]. Among these diverse applications, the automatic extraction of buildings from high-resolution images as the most critical urban objects has always attracted the attention of researchers in this field. Regarding the various features of a building (e.g., size and shape) and the intervention of complex areas in high-resolution satellite and aerial images (like roads and parking lots), achieving distinctive inter-class features associated with building features and simultaneously discriminating intraclass features between different features is a very challenging in automatic extraction of building objects [7]. As a result, in photogrammetry and remote sensing, automatic building extraction from high spatial resolution images is a difficult problem.

Thanks to their capabilities in learning features, deep convolutional neural networks (DCNNs) improve the accuracy of semantic segmentation [9]. Convolutional and pooling operations are iterated in the DCNN to increase inputs fields of the original images and obtain deep semantic features. Nonetheless, the downsampling trend reduces the dimensional of the original image. This process leads to the loss of critical spatial details of the images, resulting in inappropriate segmentation with incorrect boundaries. Computer vision scholars have developed different encoder–decoder architectures to effectively merge high-level semantic information with low-level spatial information aiming to reduce spatial details loss and generate more reliable segmentation results [10]. The idea of merging characteristics maps produced from the encoder part is generally adopted in DCNN-based building extraction approaches and helps maintain spatial details [9], [11].

In some research, multilevels features extracted from encoder–decoder architectures have significantly contributed to improving of segmentation problems. But, low-levels features, including boundary information extracted straight from
independent segmentation operations, are erroneous and may adversely impact dense pixel-based segmentation results [12]. According to this point, researchers have recently addressed and studied the ability of joint learning of semantic segmentation and auxiliary applications associated with the shape of interesting objects [13], [14]. Some researchers enhance the segmentation accuracy through a multitask network, including object segmentation with boundary estimation [12], [15]. Yet, using binary values of the extracted boundaries does not consider the connection between neighboring pixels, thus reducing the strongness of the previous methods. Meanwhile, a distance-transform image (DTI) measures the distance between each of the pixels from the closest edge of the objects [16]. The values of pixels change in a fuzzy mode at boundary areas, and its advantage is that it establishes a complementary relationship among the neighboring pixels.

In this article, the idea of applying DTI is used to determine the boundary loss function in DCNNs so that the building extraction accuracy in the vicinity of its boundary areas to the background is improved. The adoption of DTI can be effective as an index feature in detecting boundary areas of the classifier network. In other words, DTI offers an alternative to classical ground truth. A binary mask can be converted to a gray level image, in which the intensities of pixels in the foreground are adjusted based on their distance from the closest boundary [14]. Based on this idea, one of the goals of this study is to present a loss function based on DTI, named the boundary loss function. The purpose of giving this loss function is the more accurate optimization of objects boundaries in the building extraction process. Moreover, considering previous methods are not derivative in calculating the values of DTI and cannot be directly used for network training. The present study introduces a novel derivative approach to calculate the values of DTI in an iterative process based on the morphological dilation operator. By doing this, the network will be able to determine the DTI, and it can be merged with the DCNN to calculate the boundary loss function.

The rest of this article is set out as follows. Section II discusses previous relevant research on building extraction methods based on deep learning. Section III contains a detailed description of the proposed network and boundary loss functions. Then, in Sections IV and V, the details of the implementation experiments and discussion are addressed. Finally, Section VI contains the article’s conclusion.

II. RELATED WORK

A. Semantic Segmentation Based on DCNN

The process of assigning a class label to each pixel in an image is known as semantic segmentation. In recent years, DCNNs have been widely used in semantic segmentation. One such example is the fully convolutional networks (FCNs) [17], [18]. In the DCNN-based semantic segmentation, downsampling processes are used to obtain semantic features in deeper layers. However, essential spatial features are lost as a result of this operation. Deep and initial features are connected via a summation or concatenation operator to combine information from different spatial scales. Popular UNet networks [10], Hourglass-shape networks [19], [20], and Refine-Net networks [21] enjoy such a structure in their architectures. On the other hand, to extract objects with different dimensions, various features with diverse scales are merged through dilated convolutional methods and multiscale pooling layers. In this regard, we can mention popular PSPNet [22] and DeepLabv3+ [23] networks. Generally, feature merging in multilevels enhances the performance of segmentation networks significantly.

B. Building Extraction Based on DCNN

In the past decades, researchers in the fields of photogrammetry and remote sensing have intensively studied the extraction of buildings from aerial and satellite images with a high spatial resolution [7]. Color, spectrum, edges, shapes, and shadows are among the features derived from the images in these studies. Then, in an unsupervised process, one of, or a mixture of these features, is utilized to classify the buildings [24]–[28] or in supervised classification [29]–[32]. Due to the complexity of building architecture in satellite and aerial imagery with high spatial resolution and its similarity to other urban terrains such as roads, the manually created features in traditional building extraction methods need to be more accurate. This sometimes leads to a weakness in the generalization capability of such approaches [33].

Many novel semantic segmentation approaches in remote sensing have recently been developed, and they are based on DCNNs, which can learn highly distinguishable characteristics [34]. Patch-based and FCN-based DCNN methods are the two types of DCNN-based approaches. In the patch-based process, the labeling of a pixel is done by classifying around that pixel [35]–[37]. For classifying all of the image pixels, the patch-based segmentation process must be repeated, resulting in a high computational cost. To address this issue, approaches based on the FCN method have been designed for controlling the dimensions of the input image to the network [12], [38]–[40]. A set of networks based on Hourglass-shaped like UNet networks are developed to fuse multiscale features and generate accurate building extraction outcomes to reduce spatial details loss and generate accurate building extraction outcomes [41]–[46]. In recent years, the multimodal fusion network has been introduced to increase features representation to classify semantic-rich dependencies in the channel and spatial dimensions. Thus, some scholars adopted this idea in segmentation networks to boost the features learned for building representation [47]–[50].

Recent research works have employed tools in the segmentation process, such as using the shape structure of objects [51], [52]. They have used morphological data to enhance the capability of the segmentation networks, such as boundary learning and DTI. To this end, the use of DCNN to detect boundary areas of objects has been proposed as a solution. The trained boundaries can be used to improve the segmentation process. In [46], for instance, an additional network is used to estimate the boundaries. Then, the estimated boundaries with features generated from the image are used as inputs in a segmentation network. In [53], to boost the performance of building extraction, a DTI and the estimated image resulting from classification...
In different encoder–decoders are mixed in parallel. In [54], a signed DTI is presented for building extraction. The idea of this research is to provide a boundary among the buildings at high dimensions, which improves the detection of different building areas with complicated spatial relations.

In some research, boundary information is provided in the network optimization process based on the boundary loss function. For instance, Bokhovkin and Burnaei [55] provide a boundary loss function to improve the segmentation of a high-resolution image using boundary information. On the other hand, the DTI can be used to complete morphological information and improve semantic segmentation. In the field of computer vision, the researchers [56], [57] adopt this idea and introduce loss functions based on the shape of the foreground objects. Karimi and Salcudean [56] calculate the loss function using Hausdorff distance minimization. The value of the loss function is derived in [57] by calculating the joint surface between both the predicted image and the ground-truth. The methods mentioned above are based on developing various network structures to obtain robust features and only solve the problem with asymmetric class distribution. As a result of this issue, the building extraction is more biased toward the background, resulting in cases with missing building boundaries in the results. Therefore, a novel boundary loss function is provided in this study based on the power of the DTI to represent the information about the object’s boundaries. The segmentation network is optimized using the boundary loss function in conjunction with a distributed-based loss function. During training, the combined learning in the optimization task is explicitly enforced, which successfully helped to provide correct building extraction results.

III. PROPOSED METHOD

Fig. 1 illustrates the architecture of the proposed method for extracting the buildings object from high spatial resolution images. In the building extraction phase, a convolutional network similar to the UNet network is incorporated. This architecture is known as one of the successful deep learning architectures in medical images classification. The structure of UNet architecture [10] is an inspiration by FCNNs, and its original idea is merging initial layers with deep layers through concatenation operations. This process provides a more suitable distribution of the gradient value in the network training process and improves the classification accuracy. Moreover, in contrast to the SegNet architecture [20], upsampling operations structure is used instead of a pooling structure. The related details will be discussed in Section III-A.

Nonetheless, regarding the downsampling operations and losing the details of building edge boundaries, the accuracy of the segmentation network is low in these areas. To solve this issue, in addition to the binary cross-entropy loss, a loss function based on boundary areas and using the DTIs is introduced in the suggested architecture in this article. As is observed in Fig. 1, this boundary loss function is calculated independently and does not change the main structure of the segmentation network. Additionally, this loss function can be used in the design of other segmentation neural networks. Section III-B provides the details of the boundary loss function and shows how it is calculated.

A. Segmentation Network

As previously mentioned, the structure of the proposed segmentation network in this article is similar to that of the UNet model. However, some differences have also been applied to the encoder and decoder parts of the original structure of the UNet model. Deep features were extracted in the encoder part of the network using the ResNet-50 network’s architecture [58]. The fully connected layers of this architecture are removed, and the decoder part of the network is added for dense pixel estimation. The use of deep residual networks helps overcome the gradient reduction problem in the network training part. In the training part, the trained parameters of ResNet-50 that applied to ImageNet dataset [59] are used to initialize learning parameters and increase the convergence speed of the model. As is seen in Fig. 1, ResNet-50 network consists of five blocks, named Conv (t, s) in this article, where t and s denote the block number and the number of residual convolutional layers in it, respectively. The first block (Conv (10)) in ResNet-50 lacks residual convolutional layers and uses only one 7x7 convolutional layer. Assuming that the input images to the network have a dimension of WxH, the output dimensions of the feature space of each convolutional block \( E_i, (i \in \{1, 2, 3, 4, 5\}) \) will be equal to values (w/2 x h/2), (w/4 x h/4), (w/8 x h/4), (w/16 x h/16), and (w/32 x h/32). The feature space \( E_6 \) is associated with the end pooling layer and has a dimension of (w/64 x h/64). Similar to the structure of UNet networks, each part of the side features of encoder flow blocks are merged via concatenation operations before upscaling operations with features corresponding to the decoder flow. The flow of a decoder part corresponding to each flow of the encoder block is represented by \( D_m \), and its value will be calculated as follows:

\[
D_m = \begin{cases} 
\text{RUM} + m (\text{conv}_{1 \times 1} (E_m \odot D_{m+1})), & m = 1, \ldots, 5 \\
\text{RUM} + m (E_m) & m = 6 
\end{cases}
\]

where \( \text{conv}_{1 \times 1} \) represents the 1x1 convolution operation with the size of the kernel equals to one, \( E \) represents the encoder features, \( \odot \) denotes the concatenation operation, and \( \text{RUM}+ \) shows the residual upsampling module. In this article, in contrast to conventional upsampling methods [see Fig. 2(a)], the idea is to use residual connection layers in the upsampling convolution block structure [60]. Fig. 2(b) shows the simple residual unit module (RUM) architecture [28]. Based on the design of this module, the \( \text{RUM}+ \) structure has been developed in this research. The input features to this module enter two different flows. In one flow, once the dimensions of the feature are doubled through bilinear upsampling, a residual block is used. In other flow, opposite to the previous case, a residual block is first deployed, and finally, the dimensions of feature space are doubled by the bilinear upsampling layer. Eventually, the results of these two flows are averaged. In the RUM, convolutional operations with kernel dimensions of 3 x 3 are performed in the residual block after batch normalization (BN) [61] and the ReLU activation function [62]. In the last layer of the proposed model,
Fig. 1. Segmentation network architecture with the structure of the proposed boundary loss function.
the predicted building maps is calculated as follows:

\[ P_{rgb} = \text{sig}(\text{conv}^2_{3 \times 1}(D_1)) \]  

(2)

In the above equation, \( P_{rgb} \) is equivalent to the predicted map from building objects, and \( \text{conv}^2_{3 \times 1}(\cdot) \) reduces the dimensions of features channels. Also, the size of kernel is one, and the number of classes is 2, showing building classes and background terrain. The sigmoid function is adopted to produce the map generator function for building estimation and is represented by \( \text{sig}(\cdot) \). Once the estimation map is known, the network is optimally trained in a supervised mode via ground-truth images using binary cross-entropy (BCE) loss function \( L_{BCE} \) and the proposed boundary loss function \( L_{\text{boundary}} \):

\[
L = \begin{cases} 
L_{BCE} & N < \text{iter}/2 \\
\alpha L_{BCE} + \beta L_{\text{boundary}} & N \geq \text{iter}/2
\end{cases}
\]

(3)

In the above equation, \( \alpha \) and \( \beta \) are the balancing coefficients of loss functions \( L_{BCE} \) and \( L_{\text{boundary}} \). \( \text{iter} \) denotes the number of training iterations. The value of \( \beta \) in this article is assumed zero until the number of iterations (i.e., \( N \)) reaches half. The rest of the iterations of the network training step continues with a balancing coefficient between \( \alpha \) and \( \beta \). Details of implementation and quantification of these parameters are mentioned with detail in Section IV-D.

B. Boundary Loss Function

The proposed boundary loss function in this study is calculated based on ground-truth images and predicted maps from the segmentation network [16]. After selecting the predicted maps by the sigmoid function, the binary image of the estimated image is calculated using a simulated binary function. Next, boundary areas of the binary images are extracted for both ground-truth and predicted maps. Then, DTIs for both images of boundary areas are obtained. In this research, inspired by the method presented in [16], DTIs are calculated using the morphological dilation method. Finally, the calculations required to determine the boundary loss function \( L_{\text{boundary}} \) are discussed.

1) Calculation of the Boundary Map: Assume that \( P_{rgb} \) is the predicted map by the segmentation network. This section introduces a function that is responsible for the binarization of \( P_{rgb} \) during the network training interval. One of the essential features of this function is that it is differentiable. Otherwise, backpropagation error operations will not be possible in the network optimization step. The plot of this function in the range 0 to 1 is shown in Fig. 3(a). This function is calculated as follows:

\[
F(x) = \frac{1}{1 + e^{-\gamma(x-T)}}
\]

(4)

where \( \gamma \) and \( T \) are constant values of the binary function, which are considered 15 and 0.5 in the present study. The input value of this function is equal to the predicted maps from the segmentation network, \( P_{rgb} \), and its output is the value of the predicted binary image, \( B_{pred} \) [see Fig. 3(b)].

The Sobel operator [63] is the inspiration to extract boundary areas of the ground-truth and predicted binary image. To this end, convolutional kernels with constant parameters are used for the length and width of the image

\[
S_x = \begin{bmatrix} -2 & 0 & 2 \\ -4 & 0 & 4 \\ -2 & 0 & 2 \end{bmatrix}, \quad S_y = \begin{bmatrix} 2 & 4 & 2 \\ 0 & 0 & 0 \\ -2 & -4 & -2 \end{bmatrix}
\]

(5)

If the ground-truth images and images estimated by the classifier network are assumed as \( B_{GT} \) and \( B_{pred} \), boundary maps for them will be found as follows:

\[
\Psi_{pred} = \frac{1}{2} \| B_{GT} * S_x \| + | B_{pred} * S_y | 
\]

(6)

\[
\Psi_{gt} = \frac{1}{2} \| B_{GT} * S_x \| + | B_{GT} * S_y | 
\]

(7)

In (6) and (7), * is the standard convolution operations, and |. | shows the absolute value of the functions.

2) Calculation of Distance Transform Image: Having the binary image B, its DTI (i.e., \( B_{DTI} \)) is given by the following:

\[
B_{DTI} = \begin{cases} 
\inf_{y \in G_{tn}} x-y_2 & x \in G_{in} \\
0, & \text{others}
\end{cases}
\]

(8)
In the above equation, $x - y_2$ is the Euclidian distance between pixels $x$ and $y$. In the binary image, $G_{in}$ and $\partial G$ represent the foreground space (building areas) with pixel values of 1 and boundary areas with pixel values of 0. Based on this definition, pixel values $B_{DTI}(i, j)$ in building areas are always positive and are zero in background areas. By moving away from boundary areas of the building, $B_{DTI}(i, j)$ values increase. To calculate the proposed boundary loss function in this research, first, the DTI in boundary areas extracted from the ground-truth and the predicted map need to be generated. Despite this, in DTIs calculated for binary images, the value of pixels reduces by moving away from the boundary areas, thus leading to challenges to the optimization process of the deep learning network. To overcome this problem, those boundary areas with pixel values of 1 will be considered the background and zero value for the calculation of DTI. Conversely, background areas with nonzero values are considered. Based on DTI, the boundary loss function ($L_{\text{boundary}}$) with differentiability capability is introduced during network training. In this method, DTI is calculated in $k$ steps and based on the morphological dilation operator. For brevity, the proposed distance transform imagery is named $k$-DTI. Fig. 4 shows a flowchart of the $k$-DTI method. In the following, the details of the calculation of the $k$-DTI are provided. Parameter $k$ will be analyzed as metadata during network training. Considering the boundary images ($I_{0}^{\text{Mask}}$), calculated in previous section, as the input images, by $k \geq 1$, the transformed image is obtained as follows:

\[
I_{k}^{\text{Mask}} = \text{Dilate}(I_{k-1}^{\text{Mask}}) \tag{9}
\]

\[
I_{k}^{T, \text{Mask}} = I_{k}^{\text{Mask}} \oplus I_{k-1}^{T, \text{Mask}} \tag{10}
\]

\[
I_{k}^{DT} = \{ (k+1) - I_{k}^{T, \text{Mask}} \}. \tag{11}
\]

In the above equations, $\text{Dilate(.)}$ shows the morphological dilation operations, $\oplus$ represents the addition operator in tensor mode, and $I_{k}^{DT}$ is the distance transformation image obtained in step $k$. Except for $\text{Dilate(.)}$, all functions are differentiable. To calculate the dilation with differentiability during the network training process, a constant average kernel with dimensions $3 \times 3$ is taken into account as follows:

\[
K_{\text{Smooth}} = \frac{1}{9} \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}. \tag{12}
\]

After applying this kernel to the input binary image through the convolution operator, (4) is used to binarize the image. Equations (9) and (10) are repeated $k$ times considering that parameter $k$ has been selected already. Finally, by applying (11), the DTI is calculated.

3) Calculation of Boundary Loss Function: The boundary loss function is obtained by measuring the distance between boundary areas of the predicted map and the ground-truth image. Assume that $\Psi_{\text{pred}}$ and $\Psi_{\text{gt}}$ represent boundary areas, respectively. Consider point $M$ on boundary areas $\Psi_{\text{pred}}$. The distance between point $M$ and $\Psi_{\text{gt}}$ is usually equal to the shortest distance between $M$ and the points of boundary areas in the ground-truth image $\Psi_{\text{gt}}$, which is calculated as follows:

\[
d(M, \Psi_{\text{gt}}) = \min_{N \in \Psi_{\text{gt}}} \{ d(M, N) \}. \tag{13}
\]

Assume that $\Omega_{\text{gt}}$ and $\Omega_{\text{pred}}$ are the DTIs of $\Psi_{\text{gt}}$ and $\Psi_{\text{pred}}$, respectively. According to the calculation of DTIs, the distance between point $M$ and $\Omega_{\text{gt}}$ in (13) is equal to the mapping distance of point $M$:

\[
d(M, \Psi_{\text{gt}}) = \Omega_{\text{gt}}(M). \tag{14}
\]

Therefore, the distance between boundary areas of the estimated map and ground-truth image is found based on DTIs

\[
\begin{align*}
\text{d} \left( \Psi_{\text{pred}}, \Psi_{\text{gt}} \right) &= \frac{1}{2} \left[ \frac{1}{\Omega_{\text{pred}}} \sum_{M \in \Psi_{\text{pred}}} \Omega_{\text{gt}}(M) \\
&+ \frac{1}{\Omega_{\text{gt}}} \sum_{N \in \Psi_{\text{gt}}} \Omega_{\text{pred}}(N) \right]. \tag{15}
\end{align*}
\]
Fig. 4. Flowchart of calculating the distance transform image (DTI) in the k step.

Values $|\Omega_{\text{pred}}|$ and $|\Omega_{\text{gt}}|$ in the above equation denote the number of points in boundary areas of $\Psi_{\text{pred}}$ and $\Psi_{\text{gt}}$. Equation (15) shows the discrete version in calculating the distance between the two boundary areas. This method is not possible to use in the optimization process because the function is not differentiable. Based on the options available in deep learning software packages, the continuous version of (15) has been developed as follows:

$$d(\Psi_{\text{pred}}, \Psi_{\text{gt}}) = \frac{1}{2} \left[ \frac{\text{GAP}(\Psi_{\text{pred}} \otimes \Omega_{\text{gt}}^{k}) + \epsilon}{\text{GAP}(\Psi_{\text{pred}}) + \epsilon} \right] + \frac{\text{GAP}(\Psi_{\text{gt}} \otimes \Omega_{\text{pred}}^{k}) + \epsilon}{\text{GAP}(\Psi_{\text{gt}}) + \epsilon} + \epsilon.$$  (16)

The above equation, $\otimes$ represents the vector multiplication of two tensors, GAP(.) is the average global pooling, and $\epsilon$ is the normalizing value to avoid division by zero. The boundary loss function is computed as follows, assuming that the total number of training samples is $Q$ in the training process:

$$L_{\text{boundary}} = \frac{1}{Q} \sum_{i=1}^{Q} d(\Psi_{\text{pred}}^{i}, \Psi_{\text{gt}}^{i}).$$  (17)

IV. EXPERIMENTAL RESULT

A. Dataset

Two aerial image datasets, namely the ISPRS Potsdam and Vaihingen datasets, were used to assess the proposed segmentation network and loss function in this study. Both datasets were specially designed to develop algorithms related to segmentation and fusion methods. The first data is related to the part of the Potsdam city, and the second data is corresponding to the Vaihingen city in Germany.

The Potsdam dataset includes 38 orthophoto images with a spatial resolution of 5 cm and dimensions of $6000 \times 6000$ pixels,
containing buildings of different sizes and appearances. Each image can be obtained in one of two modes: red (R), green (G), and blue (B) or near-infrared (NIR), red (R), and green (G). Ground-truth images include six classes of impenetrable land cover, buildings, low vegetation, trees, machines, and other features. According to the information provided by the dataset provider, the training datasets will consist of 24 images, while the evaluation data will include the remaining images. The Vaihingen dataset is cropped from a large orthophoto image. The dimensions of the images in this case, unlike the images in the Potsdam dataset, are not the same, and on average, they have dimensions of 2500 × 2000 pixels. In this dataset, the spatial resolution of the images is 9 cm. Each image is presented in only one mode with three bands: red (R), green (G), and near-infrared (NIR). The number of the classes in this dataset is similar to the Potsdam data class. Sixteen images were used as training data in the proposed network training procedure. The rest of the images were used for network testing based on the information provided by the dataset provider. In both datasets, the class containing the building area is assigned to the value 1, and the other classes are labeled 0. The examples of images associated with these datasets are shown in Fig. 5.

**B. Evaluation Criteria**

The effectiveness of the proposed method is investigated in this study by analyzing the results of the evaluation criteria on the test dataset. Similar to previous studies, three typical criteria are used to analyze the accuracy of the extracted building: overall accuracy (OA), F-score, and intersection over union (IoU). The OA is calculated as the ratio of all successfully predicted pixels to total pixels in the image

\[
    OA = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)
\]

where TP stands for true positive predictions, TN for true negative predictions, FP for false positive predictions, and FN for false negative predictions. In this article, Precision is a measure of the accuracy of minority class (building class) based on the number of accurate predictions of TP in all positive probabilities. At the same time, Recall gives a proposal for the missing buildings which were correctly predicted. The harmonic means of the two Precision and Recall criteria, which can be represented as the following equations, is identical to the F-Score criterion:

\[
    Precision = \frac{TP}{TP + FP} \quad (19)
\]

\[
    Recall = \frac{TP}{TP + FN} \quad (20)
\]

\[
    F\text{-score} = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (21)
\]
The IoU is a criterion for determining how closely the prediction map and ground truth overlap. It is defined as follows:

$$\text{IoU}(P_{\text{pred}}, P_{\text{gt}}) = \frac{|P_{\text{pred}} \cap P_{\text{gt}}|}{|P_{\text{pred}} \cup P_{\text{gt}}|}$$

(22)

The group of predicted pixels is $P_{\text{pred}}$, the group of ground-truth pixels is $P_{\text{gt}}$, and the number of members in the collection is “$|$”. The union and intersection of the two groups are represented by $\cap$ and $\cup$, respectively.

C. Implementation of the Proposed Method

To start the network training process, the value of the initial weight parameters of the ResNet-50 network, which is trained on the ImageNet image set, is used in the encoder part of the segmentation network. The weight parameters of the other network parts were initialized using the method presented in [58].

To implement the proposed strategy, the Pytorch programming package [64] is used in all experiments on a single-core operating system (Tesla K80) based on the graphics processing unit (GPU).

The data augmentation method was utilized to expand the input data and overcome the problem of overfitting during the training process [65]. This method involves randomly rotating images at 90° increments from 0° to 270° and randomly inverting images in horizontal or vertical directions. Due to the limitations of the GPU, the training set’s input images were cropped into 640 x 640 subimages. In each training dataset, 80% of the images were used to train the network, and the rest were used to evaluate the network. The AdaMax function, developed from the well-known Adam optimizer algorithm [66], was used to optimize network weights, with weight decay values of 0.0009 and an initial learning rate of 0.0002. During the learning process, the value of the learning rate decreases by multiplying the learning rate value by the value $(1 - \frac{\text{iter}}{\text{max-iter}})^{\text{power}}$, as a function with “Poly” behavior [67]. “Power” equals 0.35, and the total number of epochs is multiplied by the number of training data in each dataset to get “max-iter.” In this research, the number of training epochs was considered 60. Up to 30 iterations, the coefficient value of the proposed boundary loss function is set to zero, and this function is not evaluated in the training phase of the segmentation network. From 30 to 60 iterations, the boundary loss function was taken into account in network training calculations.

Next section describes how to weigh this loss function. At each repetition stage, the network weight parameters associated with the best evaluation values are stored. In the test process, similar to the training phase, the input images are cropped first. There is no need to augment datasets at this stage, and the final results are obtained after fusing and averaging with each other.

D. Evaluation of the Proposed Boundary Loss Function

In the first loops of model training, the values obtained from the end layer of the segmentation network (i.e., Softmax layer) become close to zero and lead to foreground pixels with zero values. So, it is not possible to train the network by relying merely on the boundary loss function in the training phase of the segmentation network. In this case, the error values do not converge. In other words, the role of the boundary loss function as an additional loss function is to help improve segmentation results in boundary areas via giving larger weights to them. According to (3), only distribution-based loss function like binary cross-entropy loss ($L_{\text{BCE}}$) was adopted up to the 30th iteration to avoid nonconvergence of network training during 60 iterations. From the 30th to the 60th iteration, network training continues by considering the proposed boundary loss function merged with the binary cross entropy.

To weigh the parameters $\alpha$ and $\beta$ of (3), three strategies are considered. For this purpose, first, the value of the parameter $\alpha$ in the calculation of DTI was considered equal to one. In the first strategy, the value of the $\alpha$ is deemed to be similar to one, and $\beta$ will be regarded as constant values during network training. In the second strategy, the $\alpha$ value is assumed to be the same as in the first strategy, and the $\beta$ increases at a constant rate during each network training repetition. In the first step, the $\beta$ value is set to 0.05, and the rate of increase is adjusted so that its value in the last loop is equal to the constant weight of the $\alpha$ value, which is equivalent to one. In the third strategy, the $\alpha$ value is considered equivalent to (1- $\beta$). Similar to the second case, the $\beta$ value in the first step is considered equal to 0.05 and increases during network training. In the first iterations, more weight is assigned to the $L_{\text{BCE}}$ function, and in later iterations, this weight is reduced.

The increase in the $\beta$ is done so that the $\beta$ value does not exceed the $\alpha$ value during training.

Table I shows the results obtained from network training based on using the proposed boundary loss function in different strategies. The results show an improvement in evaluation criteria compared to training only based on the $L_{\text{BCE}}$ function. In Strategy 1, the constant $\beta$ value is increased, and in the case of using $\beta = 1$, the best result is obtained. By comparison, every one of the two proposed scheduling strategies (strategies 2 and 3) produces better outcomes than any constant $\beta$ (strategy 1) without extensive testing. However, the best result is obtained when strategy three is utilized, which is why this strategy has been used in subsequent tests.

As mentioned in Section III-B2, the value of the boundary loss function depends on parameter $k$. Hence, when analyzing the effect of this loss function on the building extraction results, parameter $k$ is considered with a changing domain between 1 and 6. Fig. 6 illustrates the values of IoU and OA during the network training phase. These values were calculated between 30th to 60th iterations. As is shown, the best performance on the network evaluation data during training is obtained for $k = 3$.

Table II shows the quantitative results of the test images for choosing the best parameter of $k$. The results of merging the two $L_{\text{BCE}}$ and $L_{\text{boundary}}$ functions are shown using different values for parameter $k$ in network training. By comparing these results, one can conclude that using the $L_{\text{boundary}}$ function improves building extraction results. Moreover, the effect of different values of $k$ can be analyzed. In this test, selecting $k = 3$ results in obtaining the best accuracy in the evaluation metrics results of the two datasets. In this case, the IoU evaluation metric for building class has increased by 0.57% and 0.39%, respectively, for ISPRS Postdam and Vaihingen datasets. Also,
TABLE I
RESULT OF DIFFERENT STRATEGIES FOR WEIGHTING THE BOUNDARY LOSS FUNCTION

<table>
<thead>
<tr>
<th>Methods</th>
<th>α</th>
<th>β</th>
<th>ISPRS Vaihingen dataset (epoch: 60)</th>
<th>ISPRS Potsdam dataset (epoch: 60)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>α</td>
<td>β</td>
<td>OA (%)</td>
<td>F-score (%)</td>
</tr>
<tr>
<td>BCE only</td>
<td>1</td>
<td>0</td>
<td>95.84</td>
<td>94.14</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td></td>
<td>95.88</td>
<td>94.17</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td></td>
<td>95.90</td>
<td>94.21</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td></td>
<td>96.00</td>
<td>94.35</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td></td>
<td>96.14</td>
<td>94.43</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>96.18</td>
<td>94.48</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td></td>
<td>95.82</td>
<td>94.15</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>95.80</td>
<td>94.10</td>
</tr>
<tr>
<td>Strategy 1 (Constant β)</td>
<td>1</td>
<td>0.05 (initial value)</td>
<td>96.21</td>
<td>94.51</td>
</tr>
<tr>
<td>Strategy 2 (Increase β)</td>
<td>1</td>
<td>0.05 (initial value)</td>
<td>96.25</td>
<td>94.57</td>
</tr>
</tbody>
</table>

Fig. 6. (a) and (b). OA and IoU criteria for the ISPRS Potsdam data set during the network training phase. (c) and (d). OA and IoU criteria for the ISPRS Vaihingen dataset during the network training phase.
HOSSEINPOUR et al.: NOVEL BOUNDARY LOSS FUNCTION IN DEEP CONVOLUTIONAL NETWORKS

TABLE II
IMPACT OF PARAMETER k ON THE VALUES OF EVALUATION CRITERIA CALCULATED FOR TEST DATA

<table>
<thead>
<tr>
<th>Method</th>
<th>ISPRS Vaihingen dataset (epoch: 60)</th>
<th>ISPRS Potsdam dataset (epoch: 60)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OA (%)</td>
<td>F-score (%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>96.25</td>
<td>94.57</td>
</tr>
<tr>
<td>2</td>
<td>96.28</td>
<td>94.60</td>
</tr>
<tr>
<td>a (L_{BCE} + \beta L_{boundary}) (Balancing a and (\beta)) 3</td>
<td>96.32</td>
<td>94.68</td>
</tr>
<tr>
<td>4</td>
<td>96.30</td>
<td>94.67</td>
</tr>
<tr>
<td>5</td>
<td>96.27</td>
<td>94.63</td>
</tr>
<tr>
<td>6</td>
<td>96.24</td>
<td>94.60</td>
</tr>
</tbody>
</table>

Note: All evaluation criteria are expressed in (%). \(L_{BCE}\) and \(L_{boundary}\) represent the binary cross-entropy and proposed boundary loss, respectively.

TABLE III
COMPARING THE WELL-KNOWN INTEGRATED BOUNDARY LOSS FUNCTIONS WITH THE LBCE

<table>
<thead>
<tr>
<th>Loss Function</th>
<th>ISPRS Vaihingen dataset (epoch: 60)</th>
<th>ISPRS Potsdam dataset (epoch: 60)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OA (%)</td>
<td>F-score (%)</td>
</tr>
<tr>
<td>BCE Loss</td>
<td>95.84</td>
<td>94.14</td>
</tr>
<tr>
<td>BCE + Hausdorff Loss</td>
<td>96.16</td>
<td>94.30</td>
</tr>
<tr>
<td>BCE + Surface Loss</td>
<td>96.19</td>
<td>94.32</td>
</tr>
<tr>
<td>BCE + RS-Boundary Loss</td>
<td>95.21</td>
<td>94.48</td>
</tr>
<tr>
<td>BCE + Proposed Loss (k=3)</td>
<td>96.32</td>
<td>94.68</td>
</tr>
</tbody>
</table>

Note: All evaluation criteria are expressed in %.

with increasing the value of \(k\) (\(k>3\)), the effectiveness of the \(L_{boundary}\) function value decreases. The experimental results in Fig. 7 demonstrate that our proposed boundary loss function could extract a more accurate building boundary than the method based on using only the BCE loss function. One of the main reasons for this is that the number of pixels in the background of high spatial resolution remote sensing images is typically much larger than the number of pixels in the building’s objects, causing the training to fall into the local optimum problem when using the distribution loss function in the building extraction process. As a result of this issue, the extraction outputs are more biased toward the background, resulting in missing building components. Therefore, we attempted to add learnable boundary information by introducing DTI to joint-optimize the building outline. Our proposed method more accurately differentiates the borders and shapes between building and background.

V. DISCUSSION

A. Comparison With the State-of-the-Art Loss Functions

The proposed boundary loss function is compared with its counterparts provided in the literature. Accordingly, boundary loss functions based on different approaches, such as Hausdorff loss [56], Surface loss [57], and RS-Boundary loss [55], have been used. In Hausdorff loss and Surface loss, the DTI calculation of ground truth is incorporated. To weigh the softmax prediction outputs, Hausdorff loss used not only the ground truth DTI but also the predicted segmentation DTI. Surface loss, on the other hand, based on the ground truth DTI, assigned weights to the softmax probability outputs. RS-Boundary loss is computed based on a differentiable metric of boundary detection. The implementation of the loss functions mentioned in this section has been done according to the authors’ suggestion, and the same training and test data have been used in all experiments for fairness. Table III shows the results of comparing different boundary loss functions in both ISPRS Potsdam and Vaihingen datasets. The proposed boundary loss function in this article indicates superior performance in identifying and extracting building areas compared to other loss functions. Our proposed loss function improved the building extraction results by 0.71% and 0.47% in terms of IoU metric for both ISPRS Potsdam and Vaihingen datasets, which indicates the strength of the proposed boundary loss function provided for improving the results. On the other hand, Hausdorff loss, Surface loss, and RS-Boundary loss improved the building extraction results by 0.37%, 0.42%, and 0.48% in terms of IoU metric for the ISPRS Vaihingen dataset. This is while the improvement in IoU results for the ISPRS Potsdam dataset is reported to be 0.23%, 0.28%, and 0.34%. Hausdorff loss and Surface loss achieved minor improvements in
Fig. 7. Building extraction results in two modes of use and non-use of the proposed boundary loss function. (a) Input image. (b) and (c) Extracted buildings result without and with using the proposed boundary loss function. FN, FP, and TP pixels are pink, cyan, and white, respectively.
The potential reason may be that building extraction from high spatial resolution aerial or satellite images is much more challenging than segmentation problems in the computer vision community. The building has various locations, shapes, and sizes, while these characteristics are relatively fixed for most computer vision semantic segmentation.

To investigate the effect of boundary loss functions, the prediction heat map of some sample test images from both the ISPRS Vaihingen and Potsdam dataset is shown in Fig. 8. The employment of previously developed boundary loss functions in conjunction with the BCE loss function improves qualitative results. In addition, we see a reduction in noise in...
the extracted buildings’ boundary areas. The areas indicated by yellow square boxes further illustrate the effects of using boundary loss functions. The comparison of the output prediction maps of the extracted building suggests that our proposed loss function contains more semantic information and less noise in the boundary area of the buildings.

### B. Evaluation of the Segmentation Network

As previously mentioned in Section III-A, there is no limitation on using the $L_{\text{boundary}}$ function in other segmentation networks. The proposed boundary loss function can be adopted in various segmentation networks for extracting the building area. In this section, the proposed segmentation network is compared to previously presented popular deep segmentation models. Based on this, SegNet [20], PsPNet [68], Res-U-Net [45], and DeepLabV3+ [69] networks are used. The criterion to select these specific networks is their popularity and availability of their implementation code. To make a fair comparison, all metadata are considered the same in all tests. Moreover, according to Table II, $k = 3$ is chosen to determine the boundary loss function. Table IV lists the results of the comparison between different segmentation networks.

As shown in Table IV, the proposed segmentation model achieves the best evaluation metrics for the Potsdam and Vaihingen datasets. Columns 1 to 3 of Table IV evaluate the computational speed. The multiply-and-accumulates (MACs) volume criterion in neural networks, as well as training and testing duration in image per second, were employed in this section. The difference in computing efficiency between these methods is primarily due to the training and testing process’s time cost, which is generally related to the complexity of the segmentation models. The proposed model has reached an 83.42 value for MACs and a process of 1.73 and 0.163 images per second for network training and testing phases, which shows a suitable speed for calculations compared to other models except PsPNet model. It is worth noting that the PsPNet model enjoys only one convolution block in the decoder block, resulting in reduced calculations volume and reducing the accuracy of evaluation results. So, the weakest segmentation results were obtained by this model. The IoU of PsPNet model equaled 84.6% and 89.65% for Vaihingen and Potsdam datasets, respectively. The evaluation criteria in the popular deeplabV3+ and the model proposed in this research are slightly different. However, the deeplabV3+ has high MACs (342.73) than the other models due to its complex encoder and decoder structure. In its encoder architecture, the Res-U-Net model, like our segmentation model, uses the residual block. Our model, on the other hand, is faster and more efficient than the Res-U-Net model. This is owing to the proposed RUM+ block that uses two residual units in its structure for accurate upsampling in this research. The role of decoder structure in increasing the accuracy of building extraction results was investigated. The literature has paid little attention to the decoder parts of the networks, and superficial convolution
Fig. 10. Comparing of building extraction results in both ISPRS Potsdam and Vaihingen dataset using different segmentation networks. (a) Input image. (b) SegNet. (c) PSPNet. (d) Res-U-Net. (e) DeeplabV3+. (f) Proposed segmentation network. FN, FP, and TP pixels are pink, cyan, and white, respectively.
layers are generally used to restore the lost feature. The design of the proposed RMU+ in this article somehow enhances the restoration of the deep feature in the encoder part of the network. According to Fig. 2, the segmentation model is tested with three decoders to evaluate the ability of the proposed RMU+. The evaluation criteria for the ISPRS Vaihingen and Potsdam datasets are compared in Fig. 9. Experiments demonstrate that the RUM+ (Model-RUM+) model outperforms models that merely employ conventional convolution (Model-CD) or simple RUM (Model-RUM) in the decoder portions.

The building extraction results of some test images from the ISPRS Potsdam and Vaihingen datasets are shown in Fig. 10. Our proposed segmentation network outperforms existing advanced methods and can successfully handle the problems of inadequate building extraction for buildings with complicated shapes and sizes. According to qualitative results, the building extraction results at the boundary areas showed reduced noise in all models. This means that network training based on coded boundary information (i.e., DTI), generated using the last layer of semantic features, could improve the robustness of building extraction in edge parts. On the other hand, the buildings extraction results in Fig. 10 still have misclassification problems in some buildings. The causes of these errors can be divided into two categories. The occurrence of errors in training datasets is the first category, including factors like incorrect building labeling, noise and distortion in satellite or aerial images, and a mismatch between ground-truth and color images. In this case, the possibility of manipulating and solving the mentioned problems is not recommended because it reduces the uncertainty in solving the building extraction problem. The second factor is related to the structure of hourglass deep learning models. In these models, which are mainly composed of two parts, encoder and decoder, the dimensions of the input images are reduced to extract the feature space to overcome the limitations of the computational volume, which can cause the loss of important information of buildings parts. On the other hand, traditional methods of semantic feature recovery in the decoder part do not have the ability to optimally retrieve the buildings, which leads to the extraction of incomplete buildings. In this research, our segmentation method, compared to other methods, produced a small number of FN (pink) and FP (cyan) in the building extraction results. It can be concluded that the performance improvement of the proposed segmentation model is considerably associated with the proposed RUM+.

VI. CONCLUSION

In this article, a new boundary loss function was introduced to enhance the result of the extracted building area in high-resolution aerial and remote sensing images. The boundary loss function is calculated using binary images from the predicted image of the segmentation network and ground-truth image, as well as DTI conversion images with derivability throughout the network training phase. The DTI is calculated based on the proposed derivative morphological dilation in the k step. Since the challenge of extracting buildings from VHR images is characterized by an unbalanced class distribution, integrating the proposed boundary loss with a distributed-based loss function can improve the final result. The proposed approach was tested using the ISPRS Potsdam and Vaihingen datasets. The implementation results show that the proposed approach is beneficial in enhancing building extraction results, particularly in the boundary area. When the parameter value of the k was set to 3 to calculate the boundary loss function, the best result was obtained. In this case, the IoU evaluation metric in both the Potsdam and Vaihingen datasets improves by 0.57% and 0.39%, respectively. In addition, another advantage of the proposed loss function is that it does not depend on the segmentation network and can be used in a wide range of DCNN networks.

As suggested in future research, the proposed approach might be enhanced to handle multiclass semantic segmentation of urban areas. In this study, only VHR orthophoto images were used to classify buildings. However, height data may be useful in future study to ensure accurate building extraction. In other words, obtaining high-resolution 3-D data such as aerial 3-D point clouds or digital surface model data can improve the connection between the classes identified in this study.

REFERENCES


Hamidreza Hosseinpour received the B.Sc. degree in photogrammetry from the University of Tehran, Tehran, Iran, in 2002. He is a Full Professor with the School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran. Since 2012, he is teaching and researching the same field and during the last decade, he is mainly focused on the application of UAVs for Photogrammetry. His publication has been cited more than 2000 times till now (h-index 27) and has supervised more than 50 MSC and Ph.D. students till now.

Farhad Samadzadegan received the Ph.D. degree in photogrammetry from the University of Tehran, Tehran, Iran, in 2002. He is a Full Professor with the School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran. Since 2012, he is teaching and researching the same field and during the last decade, he is mainly focused on the application of UAVs for Photogrammetry. His publication has been cited more than 2000 times till now (h-index 27) and has supervised more than 50 MSC and Ph.D. students till now.

Farzaneh Dadras Javan received the M.Sc. and Ph.D. degrees from the University of Tehran, Tehran, Iran, in 2009 and 2012, respectively, both in photogrammetry. She is working with the Earth Observation Science Department, University of Twente, Enschede, The Netherlands. As her passion for teaching and being involved in research, she started her job as an Assistant Professor with the Photogrammetry and Remote Sensing Department, University of Tehran, Tehran, Iran, in 2016–2020. Since September 2020, she has been involved in a variety of teaching and research activities and supervised more than 20 M.Sc. and the Ph.D. thesis which resulted in the publication of more than 60 publications in international journals and conferences with the Citation score of (h-index) 9.