Improving Indoor Semantic Segmentation with Boundary-level Objectives

Roberto Amoroso$^{[0000-0002-1033-2485]}$, Lorenzo Baraldi$^{[0000-0001-5125-4957]}$, and Rita Cucchiara$^{[0000-0002-2239-283X]}$

University of Modena and Reggio Emilia, Modena, Italy
{name.surname}@unimore.it

Abstract. While most of the recent literature on semantic segmentation has focused on outdoor scenarios, the generation of accurate indoor segmentation maps has been partially under-investigated, although being a relevant task with applications in augmented reality, image retrieval, and personalized robotics. With the goal of increasing the accuracy of semantic segmentation in indoor scenarios, we develop and propose two novel boundary-level training objectives, which foster the generation of accurate boundaries between different semantic classes. In particular, we take inspiration from the Boundary and Active Boundary losses, two recent proposals which deal with the prediction of semantic boundaries, and propose modified geometric distance functions that improve predictions at the boundary level. Through experiments on the NYUDv2 dataset, we assess the appropriateness of our proposal in terms of accuracy and quality of boundary prediction and demonstrate its accuracy gain.

Keywords: Indoor scene understanding · Segmentation · Boundary losses.

1 Introduction

Automatically parsing and understanding pictures of indoor scenes is a core problem in Computer Vision, with a variety of applications ranging from augmented reality interfaces to image retrieval and the navigation of mobile robots in indoor spaces. The goal of the task is that of providing detailed information about the objects in a scene, the layout of the space, and how objects interact with each other [28]. One of the core subtasks which need to be solved in this context is that of performing a semantic segmentation over the input image. While most of the indoor understanding literature has focused on the usage of RGBD data [7, 9, 15], and while most of the semantic segmentation literature has adopted outdoor scenarios [22, 18, 27, 10], some applications require to employ RGB data in indoor contexts. Examples include the understanding of indoor photos taken from mobile phones for augmented reality applications, the processing of pictures taken from social networks and search engines, and every application in which employing a depth camera is not practical.

In such contexts, providing accurate and fine-grained pixel-wise classification without relying on depth data is of great importance. Recently, the research on semantic segmentation models has focused on the introduction of fully convolutional networks [16, 4] which leverage convolutional layers and downsampling operations to achieve a large receptive field, while upsampling operations are employed to increase the output resolution. Although this architectural choice is necessary to encode contextual information and deal with objects at large scales, it also leads to feature smoothing across object boundaries, and thus to a degraded quality in the final result. The segmentation results might look blurry and lack fine object boundary details, thus leading to defects in the results of augmented reality applications.

With the aim of improving the quality of semantic segmentation in indoor scenarios, especially in boundary regions, in this paper, we investigate the design of boundary-aware losses for the optimization of semantic segmentation architectures. We start from two recently proposed loss functions, namely the Boundary loss [12] and the Active Boundary loss [21], and design two improved versions that can significantly increase the overall quality of the segmentation at boundary level. In particular, we improve their formulation in the geometric distance between objects and prove that this results in better segmentation accuracy and better predictions in boundary areas. From an experimental point of view, we assess the effectiveness of the proposed losses on the NYUDv2 dataset for indoor semantic segmentation. We quantify and show, through quantitative and qualitative experiments, the role of both losses in the case of indoor scene segmentation and the appropriateness of the proposed variants.

2 Related Work

Localizing semantic boundaries or exploiting boundary information to improve the semantic segmentation has been the focus of several previous studies [24, 1, 6]. Gated-SCNN [20], for instance, designs a two-stream network to exploit the duality between the segmentation predictions and the boundary predictions, integrating shape information. Other works [11, 3, 5], instead, learn pairwise pixel-level affinity and monitor information flow across boundaries to preserve feature disparity for semantic boundaries and feature similarity for interior pixels.

While most of these methods [5, 20, 11] depend on the segmentation model and require re-training, extensive studies [26, 14] have proposed post-processing techniques to improve boundary details of segmentation results. DenseCRF [14] considers fully connected CRF models defined at the pixel level to improve segmentation accuracy around boundaries. SegFix [26], instead, proposes a model-agnostic method to refine segmentation maps, by training a separate network to transfer the label of interior pixels to boundary pixels. PointRend [13] presents a rendering approach to refine boundary information by performing point-based predictions at selected locations based on an iterative subdivision algorithm.

Boundary loss (BL) [12] and Active Boundary loss (ABL) [21], finally, propose a model-agnostic end-to-end trainable approach to tackle the problem of semantic segmentation at boundaries. BL promotes the refinement of the semantic boundaries by optimizing the sum of the linear combinations of the regional probability predictions and their distance transforms. ABL monitors the changes in the boundaries of the segmentation predictions and encourages the alignment between predicted boundaries and ground-truth boundaries, leveraging the distance transform of the prediction maps to regularize the network behavior.

Despite the empirical success of boundary-aware approaches in improving segmentation precision, there are still substantial segmentation errors at object boundaries. In this work, we investigate the reciprocal dependency between semantic segmentation and boundary-level objectives to increase the accuracy of semantic segmentation performance.

3 Method

Most of the existing semantic segmentation models can fail to provide correct predictions along semantic boundaries between two different classes, as widely used loss functions (like Cross-Entropy or Lovász-Softmax [2]) do not explicitly deal with the prediction of semantic region boundaries. With the aim of improving the prediction along boundaries in the case of indoor scene segmentation, we investigate the design of loss functions that explicitly model the prediction of semantic boundaries. In particular, we take inspiration from the Boundary loss [12] and the Active Boundary loss [21], two loss functions that already encode the presence of boundary regions in their formulation. Noticeably, all the functions we consider are model-agnostic and can be used during end-to-end training to improve boundary prediction.

Hereafter, we consider a segmentation setting characterized by C classes and input image resolution $H \times W$. $\boldsymbol{P} \in \mathbb{R}^{C \times H \times W}$, instead, will be used to indicate the class probability map predicted by the network. Thorough the rest of the section, given a tensor with spatial support $\boldsymbol{Z}$, the notation $\boldsymbol{Z}_i$ will be employed to denote the value(s) stored at the i-th spatial location of $\boldsymbol{Z}$, thus employing a "flattened" indexing of the two spatial dimensions.

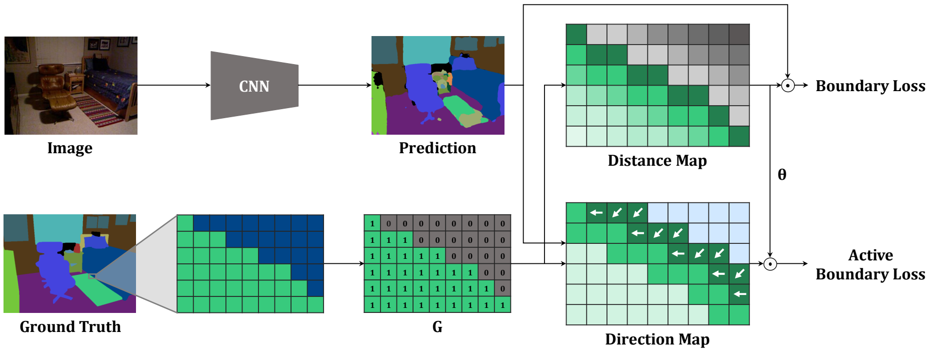
3.1 Boundary loss

The Boundary loss was originally proposed by Kervadec *et al.* [12]. It conceptually calculates an integral over the points between regions which capture the proximity of two shapes, and it is inspired by a discrete graph-based optimization technique for computing gradient flows, which introduces a non-symmetric ℓ_2 loss to regularize boundary deviation of the predicted segmentation mask relative to the ground truth. As such, it allows the incorporation of a weighting term between the estimated and expected pixels along a semantic boundary.

The loss can be seen as a weighted average of predicted probabilities over the entire image, as follows:

$$\mathbf{BL} = \frac{1}{N} \sum_{i}^{N} \boldsymbol{P}_i \boldsymbol{D}_i^{\intercal}, \tag{1}$$

where $\mathbf{BL}$ indicates the Boundary loss, N is the number of pixels of the input image, and $\boldsymbol{D} \in \mathbb{R}^{C \times H \times W}$ is a distance map that applies a probability weighting.



1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1

Fig. 1. We consider two loss functions for improving boundary-level predictions in semantic segmentation: (a) a *Boundary loss* which weights pixels predictions according to their distance to semantic boundaries; (b) an *Active Boundary* loss which promotes the alignment between predicted and ground truth boundaries. Best seen in color.

Negative values in $\boldsymbol{D}_i \in \mathbb{R}^C$ will increase the probability of predicting a given class in a pixel, while positive values will discourage the network from predicting a given class in a spatial location.

Given a one-hot ground-truth tensor $\boldsymbol{G} \in \{0,1\}^{C\times H\times W}$, the distance map is usually calculated by means of the distance transform operator, which computes for each positive pixel its distance to the closest zero-valued pixel on the same channel, *i.e.* the closest pixel which does not belong to a given class. In the original formulation of the Boundary loss [12], the distance map was defined as follows:

$$\boldsymbol{D}_i = -\mathrm{Dist}(\boldsymbol{G}_i) \odot \boldsymbol{G}_i + \mathrm{Dist}(1-\boldsymbol{G}_i) \odot (1-\boldsymbol{G}_i) \tag{2}$$

where $\odot$ indicates the element-wise multiplication and $\mathrm{Dist}(\cdot)$ is the distance transform. As it can be observed from the above formula, pixels that belong to a class are given a negative weight, thus promoting the prediction of high probability values for that class – while pixels that do not belong to a class are given a positive weight, thus discouraging the network from predicting the same class. When considering the magnitude of the weights, instead, it can be seen that pixels far from the boundaries, for which the $\mathrm{Dist}(\cdot)$ function produces high values, play a larger role in determining the loss in this formulation – while pixels close to the boundary are given less importance. In other words, the network is encouraged to give correct predictions in regions that do not lie close to the boundaries between classes and is allowed to be less precise in boundary regions.

With the aim of increasing the quality of predictions at the boundary level, we propose and investigate variations of the Boundary loss according to two principles: (i) we consider the different role of positive and negative pixels, and devise different weighting strategies for the two classes of pixels, instead of treating them equally as the original loss does; (ii) we replace the distance function with a *proximity* function, so that pixels close to a boundary are given greater importance, and regions that do not lie close to a boundary are given less importance – thus inverting the original spirit of the Boundary loss.

Following the first principle (*i.e.* treating positive and negative pixels differently), we devise two variations of the Boundary loss which correspond to the following distance maps:

$$\boldsymbol{D}_i^{+} = -\boldsymbol{G}_i + \mathrm{Dist}(1-\boldsymbol{G}_i) \odot (1-\boldsymbol{G}_i), \tag{3}$$

$$\boldsymbol{D}_i^{-} = -\mathrm{Dist}(\boldsymbol{G}_i) \odot \boldsymbol{G}_i + (1-\boldsymbol{G}_i). \tag{4}$$

As it can be observed, in the two above variants the distance map values are replaced with constant values which are independent of the distance from the boundary. This is done in the case of pixels that do not belong to the target class (*i.e.*, negative pixels) for $\boldsymbol{D}_i^{-}$, and in the case of pixels that belong to the target class (*i.e.*, positive pixels) for $\boldsymbol{D}_i^{+}$, respectively. In this manner, greater importance is also given to boundary pixels, compared to the original formulation.

According to the second principle, instead, we replace the concept of distance with that of proximity to the boundaries. To this aim, we devise an inversion function that translates distances to proximities. Our inversion function is defined as $\Phi(x) = \max(x) - x + 1$: as it can be seen, when applied to a distance transform, $\Phi(\cdot)$ returns the maximum value of the original map for pixels connected to a class boundary (for which $x = 1$ holds), and decreases linearly until reaching a minimum value of 1. According to this proximity function, we devise the following two variants of the Boundary loss:

$$\widetilde{\boldsymbol{D}}_i = \mathrm{ReLU}(K - \mathrm{Dist}(1-\boldsymbol{G}_i)) \odot (1-\boldsymbol{G}_i) - \mathrm{ReLU}(K - \mathrm{Dist}(\boldsymbol{G}_i)) \odot \boldsymbol{G}_i, \tag{5}$$

$$\widehat{\boldsymbol{D}}_i = \mathrm{Dist}(1-G_i) \odot (1-G_i) - \Phi(\mathrm{Dist}(G_i)) \odot G_i. \tag{6}$$

As it can be seen by comparing the two above formulations with the original loss, in the first case the distance function $\mathrm{Dist}(\cdot)$ is replaced with $\mathrm{ReLU}(K - \mathrm{Dist}(\cdot))$, *i.e.* with a proximity function that starts from K and decreases linearly until reaching 0 – while in the second case the full proximity function $\Phi(\cdot)$ is employed. Noticeably, in the second case, the maximum proximity value depends on the size of the object (being a function of the maximum distance in the ground-truth map), while in the first case it is constant.

3.2 Active Boundary loss

We now turn to the evaluation of a second boundary-aware loss function, namely the Active Boundary loss. This is formulated as a differentiable direction vector prediction problem, which gradually promotes the alignment between predicted boundaries (which in the following will be named, for brevity, PBs) and ground truth boundaries (for brevity again, GTBs). The pipeline for computing the loss can be conceptually divided into two phases.

Phase 1 During this phase, we compute the PBs starting from the probability map predicted by the network and devise a target direction map $\boldsymbol{D}^g$ which will be employed to align PBs with GTBs.

Specifically, boundary pixels of the predicted boundary map are recovered through the computation of the Kullback–Leibler (KL) divergence between the probabilities predicted for adjacent pixels. The i-th pixel of the PB is defined as

$$\boldsymbol{PB}_i = \begin{cases} 1 \text{ if } \exists \mathbb{KL}(\boldsymbol{P}_i || \boldsymbol{P}_j) > \epsilon,\, j \in \mathcal{N}_2(i); \\ 0 \qquad \text{otherwise}, \end{cases} \tag{7}$$

where $\mathcal{N}_2(\cdot)$ indicates the 2-neighborhood of a pixel, corresponding to the offset $\{\{1,0\},\{0,1\}\}$ (*i.e.*, the pixels to the right and below the current pixel). The threshold value ϵ is calculated dynamically to ensure that the number of boundary pixels in $\boldsymbol{PB}$ is less than $1/100$ of the area of the input image.

The pixels of GTBs are, accordingly, determined by applying Eq. 7 to the one-hot ground-truth tensor and replacing the KL divergence with a simpler equality condition on the class labels between the pixels in $\mathcal{N}_2(\cdot)$.

As a second point, we compute a target direction map containing offset vectors which will encourage pixels on the PBs to move towards pixels of the GTBs. In the original version of the Active Boundary loss, the offset was encoded as a one-hot vector. In our version, we encode the coordinate of the offset vector as a progressive index indicating its position within the 8-neighborhood of a pixel, ranging from 0 (*i.e.* offset $\{-1,-1\}$ or top-left corner) to 8 (*i.e.* offset $\{1,1\}$ or bottom-right corner) following the row-major order, and excluding index 4 which is associated with the central pixel itself.

Formally, the target direction map $\boldsymbol{D}^g \in \mathbb{R}^{H \times W}$ is computed by considering the offset direction which would move a pixel closer to a GTB, *i.e.*:

$$\boldsymbol{D}^g_i = \arg\min_j \boldsymbol{M}_{i+\Delta_j},\ j \in \{0, 1, ..., 7\}, \tag{8}$$

where $\boldsymbol{M} = \text{Dist}(\text{GTBs})$ is the result of the distance transform applied to GTBs and Δ_j represents the j-th element in the set of directions $\Delta = \{\{-1,-1\}$, $\{0,-1\}$, $\{1,-1\}$, $\{-1,0\}$, $\{1,0\}$, $\{-1,1\}$, $\{0,1\}$, $\{1,1\}\}$.

Phase 2 By using the KL divergence between the predictions for a pixel i and those for one of its neighbor pixels j as logits in a cross-entropy loss, the predicted boundary at pixel i is pushed towards the pixel j in a probabilistic way. The purpose is to increase the KL divergence between the class probability distribution of i and j while reducing the KL divergence between i and its 8-neighborhood pixels. To this aim, a predicted direction map $\boldsymbol{D}^p \in \mathbb{R}^{8 \times H \times W}$ is computed as follows:

$$\boldsymbol{D}^p_i = \left\{ \frac{\mathrm{e}^{\mathbb{KL}(\boldsymbol{P}_i, \boldsymbol{P}_{i+\Delta_k})}}{\sum_{h=0}^{7} \mathrm{e}^{\mathbb{KL}(\boldsymbol{P}_i, \boldsymbol{P}_{i+\Delta_h})}}, k \in \{0, 1, ..., 7\} \right\}, \tag{9}$$

Employing the predicted and the target direction map, the Active Boundary loss can be defined as a weighted cross-entropy (**CE**) loss, as follows:

$$\mathbf{ABL} = \left(\sum_i \Lambda(\boldsymbol{M}_i) \odot \mathbf{CE}(\boldsymbol{D}_i^p, \boldsymbol{D}_i^g) \odot \boldsymbol{PB}_i \right) \cdot \frac{1}{\sum_i \boldsymbol{PB}_i} \tag{10}$$

Through the weight function $\Lambda(x) = \frac{\min(x,\theta)}{\theta}$, the distance of the pixel i from the nearest boundary of GTBs is used as weight to penalize its divergence from the GTBs.

Managing collisions Noticeably, collisions between offset vectors of neighboring pixels are possible, especially in the case of complex boundary shapes. To address this problem, the original formulation of the Active Boundary loss [21] suggests detaching the gradient flow for all non-boundary pixels. As a result, the gradient is calculated only for the pixels on the predicted boundaries, ignoring all the other pixels.

To overcome any conflicts, we adopted an equivalent strategy. In our implementation, we multiply the result of the weighted cross-entropy loss by the predicted boundary map $\boldsymbol{PB}$, so that the only pixels that contribute to the loss calculation are the boundary pixels. The final value is the average calculated by dividing the sum of the weighted and masked values of the cross-entropy by the number of predicted boundary pixels.

Finally, the Active Boundary loss is regularized through label smoothing [19], to prevent the network from taking over-confident decisions. During label smoothing, the highest probability of the one-hot target distribution is set at 0.8, while the rest of the distribution is set to 0.2/7. Both values have been empirically determined during our preliminary experiments.

Applying proximity function As in the case of the Boundary loss, we propose to employ a proximity function in place of the distance function when weighting predicted boundary pixels (cfr. Eq. 10). Employing the previously defined proximity function Φ, we propose to modify the Active Boundary loss function as follows:

$$\widehat{\mathbf{ABL}} = \left(\sum_i \Lambda(\widehat{\boldsymbol{M}}_i) \odot \mathbf{CE}(\boldsymbol{D}_i^p, \boldsymbol{D}_i^g) \odot \boldsymbol{PB}_i \right) \cdot \frac{1}{\sum_i \boldsymbol{PB}_i}, \tag{11}$$

where $\widehat{M}$ is obtained by applying the proximity function to the distance transform applied to GTBs, *i.e.* $\widehat{M} = \Phi(\text{Dist}(\text{GTB}))$. As it can be observed, also in this case we give more importance to pixels lying close to object boundaries, in order to increase the quality of the prediction at the boundary level. This is in contrast with the original spirit of the Active Boundary loss, which instead promoted pixels far from the boundaries. As the maximum proximity value depends on the size of the ground-truth object mask, the application of the proposed proximity function encourages the network to concentrate on the boundaries of objects with a significant area.

4 Experiments

4.1 Dataset

We conduct our analyses on the image segmentation dataset NYU-Depth V2 [17], which provides densely annotated images of indoor environments. Specifically, the NYU-Depth V2 dataset consists of 1449 RGB-D frames showing interior scenes, acquired through the Microsoft Kinect sensor and with a size of 640×480. Since the distortion of the images has been corrected, they showcase a thin white border which we remove by cropping the original images to a size of 608×448 pixels. We use the segmentation labels provided in [7], in which all labels were mapped to 40 classes. We employ the standard training/test split with 795 and 654 images, respectively, and train our models on RGB images only.

In NYU-Depth V2, ground-truth labels are given as semantic regions, rather than pixel-level segmentation. This occasionally results in thin strips of unlabeled pixels between two adjacent regions and creates an issue when evaluating segmentation results at boundary level. To remedy the issue, we pre-processed the ground truth to remove small unlabeled regions through the median filtering strategy proposed in [23]. Overall, the NYUDv2 is a challenging dataset due to difficult lighting conditions and cluttered scenes.

4.2 Implementation details and evaluation protocol

We train our semantic segmentation models using two loss functions $\mathcal{L}_{bl}$ and $\mathcal{L}_{abl}$, both consisting of the traditional cross-entropy and IoU losses, which are paired with the considered boundary-level losses:

$$\begin{aligned} \mathcal{L}_{bl} &= \mathbf{CE} + \mathbf{IoU} + w_a \, \mathbf{BL}, \\ \mathcal{L}_{abl} &= \mathbf{CE} + \mathbf{IoU} + w_b \, \mathbf{ABL}. \end{aligned} \tag{12}$$

Here, **CE** is the cross-entropy loss and **IoU** refers to the lovász-softmax loss [2], a surrogate IoU loss. While the **CE** loss focuses on per-pixel classification, the lovász-softmax loss prevents small objects from being ignored. The weights w_a and w_b regulate the contribution of **BL** and **ABL** to the final loss, respectively. In particular, our experimental results are obtained by setting w_a to 1 both for the original version of BL and its proposed variants, while w_b is set to 0.8. The loss hyper-parameters K and θ are respectively set to 300 and 50.

In all experiments, we employ a DeepLabV3 [4] with ResNet-50 [8] as our default backbone architecture. Following the training protocol of [25], we use random scaling, crop, left-right flipping, and brightness jittering during data augmentation. We use a plain SGD optimizer, with an initial learning rate of 0.005 and weight decay equal to 0.0005. Training is performed with a mini-batch size of 4 and conducted for 200 training epochs. The learning rate is divided by 10 after 60, 80, 100, and 150 epochs.

Table 1. Quantitative results on the NYUDv2 dataset, when training with the Boundary loss and the proposed variations.

Loss function	Pixel Accuracy	Mean Accuracy	Mean IoU
CE	64.85	53.37	38.95
CE + IoU	65.16	54.73	39.68
CE + IoU + BL	65.10	**55.05**	39.49
CE + IoU + BL^-	64.97	54.15	39.39
CE + IoU + BL^+	65.24	54.71	39.55
CE + IoU + $\widetilde{\text{BL}}$	65.30	54.98	39.45
CE + IoU + $\widehat{\text{BL}}$	**65.36**	54.60	**39.84**

4.3 Quantitative Evaluation

Table 1 reports the results obtained on the NYUDv2 dataset when training with the Boundary loss, and with the four proposed variations, in terms of mean intersection-over-union, pixel accuracy, and mean accuracy [16]. As it can be seen, the combination of cross-entropy loss and IoU loss leads to improved results in terms of all metrics, proving that this combination is useful in the domain of indoor segmentation.

When turning to the evaluation of the losses based on BL, we first notice that the combination of cross-entropy, IoU, and Boundary loss leads to an improvement in terms of mean accuracy and to a decrease in pixel accuracy and mean IoU, highlighting that the original loss struggles to improve the results. The usage of the proposed variations that treat positive and negative pixels differently ($\boldsymbol{D}^+$ and $\boldsymbol{D}^-$ – indicated in Table 1, respectively, as BL^+ and BL^-), helps to recover this quantitative loss, leading to improved results in terms of accuracy and mean IoU. This also highlights that giving a constant weight to pixels close to the boundary works better than using a distance function which gives more importance to pixels far from a boundary.

Using the proposed variations that employ a proximity function in place of the distance function ($\tilde{\boldsymbol{D}}$ and $\hat{\boldsymbol{D}}$ – indicated in Table 1, respectively, as $\tilde{\text{BL}}$ and $\widehat{\text{BL}}$) leads to a further improvement in terms of pixel accuracy and mean IoU, with the full proximity function providing the best result on all metrics except the mean accuracy. Figure 2 reports some qualitative samples, comparing the predictions obtained with CE+IoU and those with CE+IoU+BL and CE+IoU+$\widehat{\text{BL}}$.

In Table 2, instead, we turn to the evaluation of the Active Boundary loss, and the proposed variant based on the proximity function. Firstly, we notice that in this case the ABL, in its original formulation, does not show a loss in performance when compared with the CE + IoU baseline. Indeed, a CE+IoU+ABL setting leads to an improvement in terms of pixel accuracy, mean accuracy, and mean IoU. Further, applying the proximity function in place of the distance function significantly increases the performance in terms of pixel accuracy and mean IoU,

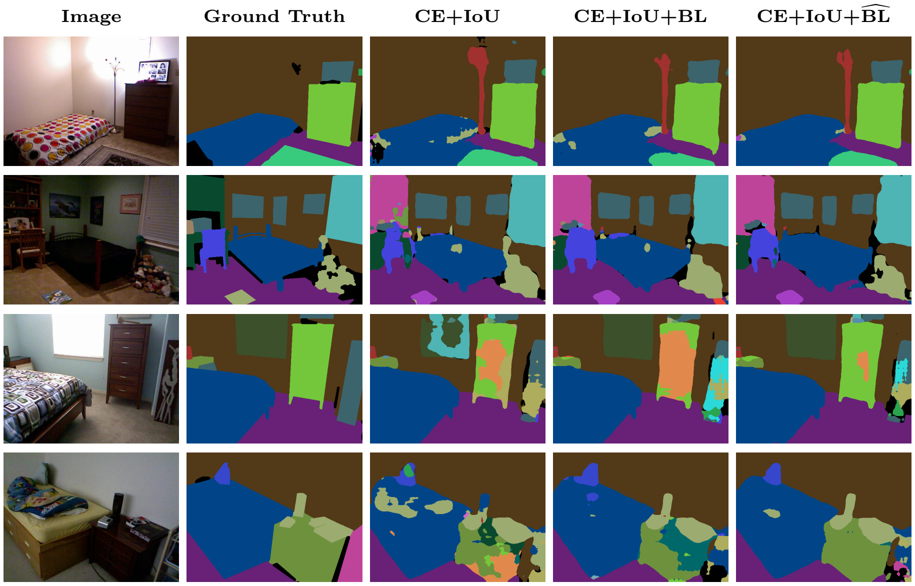


Fig. 2. Qualitative comparison between Boundary loss functions

Table 2. Quantitative results on the NYUDv2 dataset, when training with the Active Boundary loss and the proposed variation.

Loss function	Pixel Accuracy	Mean Accuracy	Mean IoU
CE	64.85	53.37	38.95
CE + IoU	65.16	54.73	39.68
CE + IoU + ABL	65.18	**54.76**	39.74
CE + IoU + $\widehat{\text{ABL}}$	**65.49**	54.60	**39.99**

thus confirming the appropriateness of using a proximity function that gives higher importance to boundary pixels. Finally, in Figure 3 we show qualitative samples comparing the results obtained when employing the CE+IoU baselines, in comparison with the ABL loss with distance and proximity functions.

5 Conclusion

We considered the usage of boundary loss functions when training segmentation models in indoor scenarios. To this end, we have considered two recently proposed boundary-level objectives, *i.e.* the Boundary loss, and Active Boundary loss, and proposed the application of a proximity function that gives higher importance to boundary pixels. Through quantitative and qualitative experiments on the NYUDv2 dataset, we have shown that the proposed variation can improve segmentation results at the boundary level.

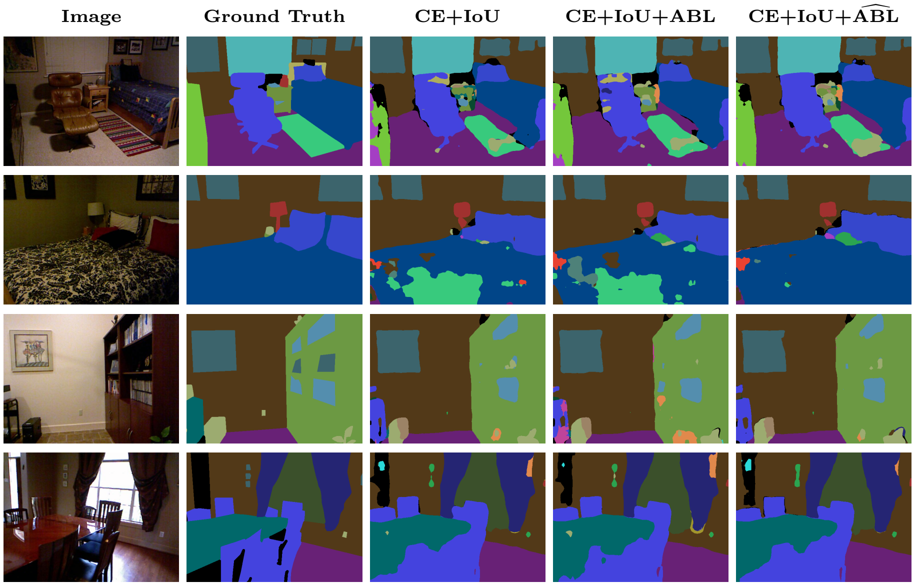


Fig. 3. Qualitative comparison between Active Boundary loss functions

References

1. Acuna, D., Kar, A., Fidler, S.: Devil is in the edges: Learning semantic boundaries from noisy annotations. In: CVPR (2019)
2. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: CVPR (2018)
3. Bertasius, G., Torresani, L., Yu, S.X., Shi, J.: Convolutional random walk networks for semantic image segmentation. In: CVPR (2017)
4. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. In: CVPR (2017)
5. Ding, H., Jiang, X., Liu, A.Q., Thalmann, N.M., Wang, G.: Boundary-aware feature propagation for scene segmentation. In: ICCV (2019)
6. Ding, H., Jiang, X., Shuai, B., Liu, A.Q., Wang, G.: Semantic correlation promoted shape-variant context for segmentation. In: CVPR (2019)
7. Gupta, S., Arbelaez, P., Malik, J.: Perceptual organization and recognition of indoor scenes from rgb-d images. In: CVPR (2013)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
9. Hu, W., Zhao, H., Jiang, L., Jia, J., Wong, T.T.: Bidirectional projection network for cross dimension scene understanding. In: CVPR (2021)
10. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: ICCV (2019)
11. Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: ECCV (2018)
12. Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ayed, I.B.: Boundary loss for highly unbalanced segmentation. In: MIDL (2019)

13. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: CVPR (2020)
14. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. NeurIPS (2011)
15. Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., Pantofaru, C.: Virtual multi-view fusion for 3d semantic segmentation. In: ECCV (2020)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
17. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
18. Pang, Y., Li, Y., Shen, J., Shao, L.: Towards bridging semantic gap to improve semantic segmentation. In: CVPR (2019)
19. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
20. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. In: ICCV (2019)
21. Wang, C., Zhang, Y., Cui, M., Liu, J., Ren, P., Yang, Y., Xie, X., Hua, X., Bao, H., Xu, W.: Active boundary loss for semantic segmentation. arXiv preprint arXiv:2102.02696 (2021)
22. Wang, L., Li, D., Zhu, Y., Tian, L., Shan, Y.: Dual super-resolution learning for semantic segmentation. In: CVPR (2020)
23. Xiaofeng, R., Bo, L.: Discriminatively trained sparse code gradients for contour detection. In: NeurIPS (2012)
24. Yu, Z., Feng, C., Liu, M.Y., Ramalingam, S.: Casenet: Deep category-aware semantic edge detection. In: CVPR (2017)
25. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: ECCV (2020)
26. Yuan, Y., Xie, J., Chen, X., Wang, J.: Segfix: Model-agnostic boundary refinement for segmentation. In: ECCV (2020)
27. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
28. Zhuo, W., Salzmann, M., He, X., Liu, M.: Indoor scene parsing with instance segmentation, semantic labeling and support relationship inference. In: CVPR (2017)