ES-CRF: Embedded Superpixel CRF for Semantic Segmentation

Jie Zhu
Peking University
Beijing, China
zhujie@stu.pku.edu.cn

Huabin Huang
Megvii Inc
Beijing, China
huanghuabin@megvii.com

Banghuai Li
Megvii Inc
Beijing, China
libanghuai@megvii.com

Leye Wang
Peking University
Beijing, China
leyewang@pku.edu.cn

Abstract

Modern semantic segmentation methods devote much attention to adjusting feature representations to improve the segmentation performance in various ways, such as metric learning, architecture design, etc. However, almost all those methods neglect the particularity of boundary pixels. These pixels are prone to obtain confusing features from both sides due to the continuous expansion of receptive fields in CNN networks. In this way, they will mislead the model optimization direction and make the class weights of such categories that tend to share many adjacent pixels lack discrimination, which will damage the overall performance. In this work, we dive deep into this problem and propose a novel method named Embedded Superpixel CRF (ES-CRF) to address it. ES-CRF involves two main aspects. On the one hand, ES-CRF innovatively fuses the CRF mechanism into the CNN network as an organic whole for more effective end-to-end optimization. It utilizes CRF to guide the message passing between pixels in high-level features to purify the feature representation of boundary pixels, with the help of inner pixels belong to the same object. On the other hand, superpixel is integrated into ES-CRF to exploit the local object prior for more reliable message passing. Finally, our proposed method yields new records on two challenging benchmarks, i.e., Cityscapes and ADE20K. Moreover, we make detailed theoretical analysis to verify the superiority of ES-CRF.

1 Introduction

Semantic segmentation, a pixel-wise classification task, plays an important role in practical applications such as autonomous driving, image editing, etc. Nowadays, CNN-based methods [2, 13, 58, 10, 31] tend to dominate this field, and they attempt to adjust the feature representation of the model itself to recognize each pixel correctly, but CNN networks have the natural defect for segmentation tasks. Generally speaking, most discriminative higher layers in the CNN network always have the larger receptive field, thus pixels around the boundary may obtain confusing features from both sides. As a result, these ambiguous boundary pixels will mislead the optimization direction of the model and make the class weights of such categories that tend to share adjacent pixels indistinguishable. For the convenience of illustration, we call this issue as Pernicious Favor on Boundary.
Figure 1: (a) Observations on ADE20K. We find a corresponding category that shares the most adjacent pixels for each class and calculate the similarity of their class weights. X-axis stands for the number of adjacent pixels for each class pair in descending order, and Y-axis represents the similarity of their class weights. Blue line denotes baseline model while orange line denotes ES-CRF. Apparently, two categories that share more adjacent pixels incline to have more similar class weights, while ES-CRF effectively decreases the similarity between adjacent categories and makes their class weights more discriminative. (b) Message passing procedure of ES-CRF. $F$ is the original feature maps of the CNN network. ES-CRF utilizes pairwise module $\psi^f_{p}$ and superpixel-based module $\psi^f_{s}$ on $F$ to obtain refined feature maps $F^p$ and $F^s$ respectively. Then $F^s$, $F^p$ and $F$ are fused as $F^*$ to further segment the image.

(PFB). We take DeeplabV3+ [5] as an example to train on ADE20K [50] dataset. Then, we count the number of adjacent pixels for each class pair and find a corresponding category that has the most adjacent pixels for each class. Fig.1(a) shows the similarity of the class weight between these pairs in descending order according to the number of adjacent pixels. It is clear that if two categories share more adjacent pixels, their class weights tend to be more similar, which actually indicates that PFB makes class representations lack discrimination and damages the overall segmentation performance. Many previous works aim to improve boundary pixel segmentation, but they seldom take PFB into consideration. Thus a question is naturally raised — Is there a properer approach that can refine boundary pixel segmentation and eliminate the PFB problem to boost the overall segmentation performance at the same time?

Considering the inherent drawback of CNN networks mentioned before, delving into the relationship between raw pixels becomes a potential alternative to eliminate the PFB problem, and Conditional Random Field (CRF) [2] stands out. It is generally known that pixels of the same object tend to share similar characteristics in the local area. Intuitively, CRF utilizes the local consistency between original image pixels to refine the boundary segmentation results with the help of inner pixels of the same object. CRF makes some boundary pixels that are misclassified by the CNN network quite easy to be recognized correctly. But almost all the CRF-based methods [2, 53] only adopt CRF as an offline post-processing module, we call it Vanilla-CRF, to refine the final segmentation results. They are incapable of relieving PFB problem as CRF and the CNN network are treated as two totally separate modules.

Based on [2, 3, 28, 32, 29], go a step further to unify the segmentation model and CRF in a single pipeline for end-to-end training and optimization. We call it Joint-CRF for simplicity. Joint-CRF can alleviate the PFB problem to some extent as the segmentation score refined by CRF directly involves in the backpropagation of the segmentation model. Joint-CRF inclines to rectify those misclassified boundary pixels via increasing the prediction score of the associated category. Afterwards, the disturbing gradients caused by those pixels will be relieved, which will promote the class representation learning. However, as shown in Fig.3 the effectiveness of Joint-CRF is restricted as it only optimizes the scale of the gradient and lacks the ability to purify class representations effectively due to the defective design. More theoretical analysis can be found in Sec. 3.3.

To overcome the aforementioned drawbacks, in this paper, we present a novel approach named Embedded Superpixel CRF(ES-CRF) to address the PFB problem more effectively. The superiority of ES-CRF lies in two main aspects. On the one hand, ES-CRF fuses CRF mechanism into the
Figure 2: Illustration of Joint-CRF and ES-CRF. The first row is the simplified structure of Joint-CRF, which unifies the CNN network and CRF in a single pipeline for end-to-end training. However, CRF only serves as a post-processing module. The second row is the overview of our ES-CRF, which fuses CRF into the CNN network as an organic whole to eliminate PFB problem.

Segmentation model itself. Different from previous CRF-based methods, ES-CRF utilizes the local consistency among original image pixels to guide the message passing on the high-level features. Each pixel pair that comes from the same object tends to obtain higher message passing weights. Therefore, the feature representation of the boundary pixels can be purified by the corresponding inner pixels from the same object. In turn, those pixels will further contribute to the discriminative class representation learning. In Sec. 3.3, we prove theoretically that ES-CRF outperforms other CRF-based methods on eliminating the PFB problem by optimizing both direction and scale of the disturbing gradient. On the other hand, ES-CRF adopts superpixel \[ \text{[47, 44, 36, 15, 35]} \] to further strengthen the reliability of the message passing to the boundary pixels. Superpixel is an unsupervised algorithm to group adjacent pixels that share similar characteristics to form a block. It is prone to achieve clear and smooth boundaries and increases the potential for higher segmentation performance. In ES-CRF, we average the feature representation of all inner pixels in the same superpixel block and then attach this local object prior to each pixel back to enhance the representation of boundary pixels.

To the best of our knowledge, we are the first to raise the PFB problem in semantic segmentation and propose an effective approach to solve it. We conduct extensive experiments on two challenging semantic segmentation benchmarks, i.e., Cityscapes \[ \text{[9]} \] and ADE20K \[ \text{[56]} \], and achieve new state-of-the-art (SOTA) results with \[ 46.83\% \text{mIoU on ADE20K and 82.74\% mIoU on Cityscapes, based on DeeplabV3+ [5] and ResNet-101 [18].} \]

In addition, we make an exhaustive theoretical analysis in Sec. 3.3 to prove the effectiveness of ES-CRF.

2 Related Work

Semantic Segmentation. Fully convolutional network (FCN) \[ \text{[33]} \] based methods have made great progress in semantic segmentation by leveraging the powerful convolutional features of classification networks \[ \text{[18, 22]} \] pre-trained on large-scale data \[ \text{[38]} \]. There are several model variants proposed to enhance contextual aggregation. For example, DeeplabV2 \[ \text{[3]} \] and DeeplabV3 \[ \text{[4]} \] take advantage of the astrous spatial pyramid pooling (ASPP) to embed contextual information, which consists of parallel dilated convolutions with different dilated rates to broaden the receptive field. Inspired by the encoder-decoder structures \[ \text{[37, 11]} \], DeeplabV3+ \[ \text{[5]} \] adds a decoder upon DeeplabV3 to refine the segmentation results especially along object boundaries. With the success of self-attention mechanism in natural language processing, Non-local \[ \text{[46]} \] first adopts self-attention mechanism as a module for computer vision tasks, such as video classification, object detection and instance segmentation. A²Net \[ \text{[2]} \] proposes the double attention block to distribute and gather informative global features from the entire spatio-temporal space of the images. DANet \[ \text{[12]} \] applies both spatial and channel attention to gather information around the feature maps, which costs even more computation and memory than the Non-local method.
**Conditional Random Fields.** Fully connected CRFs have been used for semantic image labeling in [34, 43], but inference complexity in fully connected models has restricted their application to sets of hundreds of image regions or fewer. To address this issue, densely connected pairwise potentials [24] facilitate interactions between all pairs of image pixels based on a mean field approximation to the CRF distribution. Chen et al. [2] show further improvements by post-processing the results of a CNN with a CRF. Subsequent works [28, 32, 29] have taken this idea further by incorporating a CRF as layers within a deep network and then learning parameters of both the CRF and CNN together via backpropagation. In terms of enhancements to conventional CRF models, Ladicky et al. [25] propose using an off-the-shelf object detector to provide additional cues for semantic segmentation.

**Superpixel.** Superpixel [36] is pixels with similar characteristics that are grouped together to form a large block. Since its introduction in 2003, there have been many mature algorithms [1, 47, 44] and mature evaluation metrics such as Boundary Recall and Undersegmentation Error. Now, publicly available superpixel algorithms have turned into standard tools in low-level vision. Owing to their representational and computational efficiency, superpixels are widely-used in computer vision algorithms such as target detection [41, 48], semantic segmentation [15, 39, 14], and saliency estimation [19, 35]. Yan et al. [48] convert object detection problem into superpixel labeling problem and conducts an energy function considering appearance, spatial context and numbers of labels. Gadde et al. [14] use superpixels to change how information is stored in the higher level of a CNN. In [19], superpixels are taken as input and contextual information is recovered among superpixels, which enables large context to be involved in the analysis efficiently.

Our approach is motivated by the successful combination of CRF and CNN in the scene segmentation task. But different from previous works, we no longer treat CRF as a post-processing module but apply it to the feature layer to guide the message passing, which aims to purify the feature representation of boundary pixels to relieve the PFB problem mentioned before. In addition, we introduce the superpixel algorithm to strengthen the reliability of the message passing to boundary pixels.

### 3 Method

In this section, we first revisit the basic mathematical principles of CRF briefly. Then, we make a detailed illustration of our design of ES-CRF, followed by the exhaustive theoretical analysis of the effectiveness of ES-CRF in relieving PFB.

#### 3.1 Revisiting Conditional Random Field (CRF)

CRF is a typical discriminative model suitable for prediction tasks where contextual information or the state of the neighbors affects the current prediction. Nowadays, it is widely adopted in the semantic segmentation field [2, 10]. CRF utilizes the correlation between original image pixels to refine the segmentation results by modeling this problem as the maximum a posteriori (MAP) inference in a conditional random field (CRF), defined over original image pixels. In practice, the most common way is to approximate CRF as a message passing procedure among pixels and it can be formulated as:

\[
Y_i^* = \frac{1}{Z_i} (\psi_u(i) + \sum_{j \neq i} \psi_p(i, j) Y_j) ,
\]

(1)

where \( Y_i \) and \( Y_i^* \) are defined as the classification scores of CNN model and CRF respectively for pixel \( i \), \( Z_i \) is the normalization factor known as the partition function, and \( \psi_u(i) \) is a unary function which often adopts \( Y_i \) as the default value. \( G \) is the associated pixel set with pixel \( i \). For example, DenseCRF [24] takes all other pixels except pixel \( i \) itself as the set \( G \). Moreover, the pairwise function \( \psi_p(i, j) \) is defined to measure the message passing weight from pixel \( j \) to pixel \( i \). It is formulated as : 

\[
\psi_p(i, j) = \mu(i, j) \sum_{m=1}^{M} \omega^{(m)} k^{(m)}(f_i, f_j) ,
\]

(2)

where \( \mu(i, j) \) is a label compatibility function that introduces the co-occurrence probability for a specific label pair assignment at pixel \( i \) and \( j \), while \( k(f_i, f_j) \) is a set of hand-designed Gaussian
Another important component in Eq.(4) is the appearance kernel, where
\[
k(f_i, f_j) = w^{(1)} \exp \left( -\frac{|p_i - p_j|^2}{2\theta_a^2} - \frac{|I_i - I_j|^2}{2\theta_b^2} \right) + w^{(2)} \exp \left( -\frac{|p_i - p_j|^2}{2\theta_c^2} \right).
\]

The appearance kernel is inspired by the observation that nearby pixels with similar colors are more likely to share the same class. \(\theta_a\) and \(\theta_b\) are scale factors to control the degree of these two elements, i.e., similarity and distance between two pixels. Apart from this, the smoothness kernel further removes the influence of some small isolated regions [24] and \(\theta_c\) is the associated scale factor. Notably, all these parameters are learnable during the model training.

Unfortunately, current CRF-based methods [2, 3, 28, 32, 29] for semantic segmentation always adopt CRF as a post-processing module. For example, Vanilla-CRF [2, 3] utilizes CRF to refine segmentation scores offline, which has no impacts on PFB since the CNN network and CRF are treated as two separate modules. Joint-CRF [28, 32, 29] works in a similar way although CRF involves in the backpropagation of CNN networks, which restricts its ability to relieve PFB.

### 3.2 Embedded Superpixel CRF

To solve the PFB problem in a more intrinsical way, we propose a novel method named Embedded Superpixel CRF (ES-CRF) to tackle the tough problem via fusing the CRF mechanism into the CNN network as an organic whole for more effective end-to-end training. An overview of ES-CRF can be found in Fig 2 and we formulate its core function based on Eq.(1) as:

\[
F_i^* = \frac{1}{Z_i} \left\{ \psi_u^f(i) + \sum_{j \neq i} \psi_p^f(i, j)F_j + F_i^S \right\}.
\]

Specifically, the first two terms are analogous to Eq.(1), but we perform CRF mechanism on the high-level features. \(F_i^*\) stands for the original output of feature extractors for pixel \(i\), \(\psi_u^f(i)\) and \(\psi_p^f(i, j)\) play the same role as they do in Eq.(1). \(\psi_u^f(i)\) takes \(F_i\) as the default value. In addition, we reformulate \(\psi_p^f(i, j)\) to perform message passing between pixel pairs in the high-level feature:

\[
\psi_p^f(i, j) = \mu^f(i, j)k(f_i, f_j).
\]

It is worth noting that \(k(f_i, f_j)\) is no longer hand-designed Gaussian kernels as it is in Eq.(1) but simple convolution operators instead to make the whole model more flexible for end-to-end training and optimization. Experiments in Sec4 prove this modified version is a properer choice:

\[
k(f_i, f_j) = f_i \cdot f_j = \text{conv}([I_i, p_i]) \cdot \text{conv}([I_j, p_j]),
\]

where \([x, y]\) denotes the concatenation operator. Different from Eq.(3), we normalize the input image \(I\) into the range [0, 1] to eliminate the scale variance between pixels and we replace original absolute position coordinates \(p\) with cosine position embeddings [45] to make it more compatible with CNN networks. ES-CRF encodes the appearance and position of pixels into more discriminative tokens via the flexible convolution operation, then the dot product is adopted to measure the similarity between pixel pairs. As indicated in Eq.(4), ES-CRF intends to make nearby pixel pairs that share same appearance to achieve higher \(k(f_i, f_j)\). Its intention is the same as Eq.(3). Correspondingly, we also adjust \(\mu^f(i, j)\) as the feature compatibility to measure the co-occur probability of \(F_i\) and \(F_j\):

\[
\mu^f(i, j) = \text{sigmoid}([\text{conv}(F_i, F_j)]).
\]

Another important component in Eq.(4) is \(F_i^S\), it relies on the superpixel algorithm [36, 47, 44, 14] to divide the whole image \(I\) into several non-overlapping blocks. Pixels in the same superpixel block tend to share the same characteristics. Thus, we adopt this local object prior to achieve the more effective message passing between pixels in the high-level layer. Concretely, we design \(F_i^S\) as:

\[
F_i^S = \sum_{l=1}^{Q} \psi_s^f(l)F_l = \sum_{l=1}^{Q} \frac{1}{n_l} F_l
\]
Through Eq.(9), (10) and (11), class weights in the next iteration will be updated. \(\theta_2\) is larger than \(\theta_1\). (c) ES-CRF aims to enhance the feature representation of \(F_k\) via the inner pixels like \(F_j\) from the same object. It adjusts both scale and direction of \(\nabla W\) to make \(\theta_3 > \theta_2 > \theta_1\).

\(Q\) is the associated superpixel block that contains pixel \(i\) and \(\psi^f_i(l)\) devotes the re-weighting factor for the features of pixel \(l\) in \(Q\). We adopt \(\psi^f_i(l) = \frac{1}{n}\) for simplicity and \(n\) is the total number of pixels in \(Q\). \(F^{iS}_{k}\) serves as a supplement in Eq.(4) to attach the local object prior to each pixel back, which increases the reliability of message passing in ES-CRF. What’s more, superpixel [13, 39, 14] always tends to generate clearer and smoother boundary segmentation results than traditional CNN networks or CRFs do, which also increases the potential for more accurate segmentation results. Detailed experiments can be found in Sec.4.

### 3.3 How ES-CRF Relieves PFB

In this section, without loss of generality, we take the simple two-class segmentation problem as an example to dive into the principle of ES-CRF from the perspective of gradient descent. Suppose \(F_k\) is the feature vector for a foreground boundary pixel \(k\) and its prediction probability is \(P_k\). Then, the typical cross-entropy loss \(L_k\) can be defined as:

\[
L_k = -\ln P_k = \ln \frac{1}{1 + e^{-Y_k}}, \quad Y_k = W^T \cdot F_k
\]  

where \(W\) is the class weight matrix and its gradient variation \(\nabla W\) can be formulated as:

\[
\nabla W = \frac{\partial L_k}{\partial W} = \frac{\partial L_k}{\partial P_k} \cdot \frac{\partial P_k}{\partial Y_k} \cdot \frac{\partial Y_k}{\partial W}
\]  

Through Eq.(9), (10) and (11), class weights in the next iteration will be updated:

\[
W^* = W - \nabla W = W + (1 - P_k) \cdot F_k
\]

As shown in Eq.(12), the direction of the gradient descent in pixel \(k\) keeps the same as \(F_k\) while the magnitude of the gradient is decided by \(P_k\). **What happens if we integrate the CRF into the segmentation pipeline?** As we have discussed in Sec.1, Vanilla-CRF has nothing to do with the optimization process of CNN networks, while if we adopt Joint-CRF, \(\nabla W\) can be reformulated as:

\[
-\nabla W = (1 - \hat{P}_k) \cdot F_k = (1 - \frac{1}{Z_k} \sum_{j \in G} w_j P_j + P_k) \cdot F_k
\]

where \(\hat{P}_k\) is the refined score by CRF, \(w_j\) is the message passing weight from pixel \(j\) to pixel \(k\) and \(P_j\) is the original score of pixel \(j\). In general, boundary pixel \(k\) is hard to classify correctly due to the confusing features from both sides. Thus the original probability \(P_k\) is always small. In contrast,

---

The detailed derivation process can be found in our Appendix.
other inner pixels of the same object are easy to recognize and tend to achieve a higher probability. Consequently, \( \hat{P}_k \) is usually larger than \( P_k \) and disturbing gradients caused by pixel \( k \) will be relieved to some extent, which makes inter-class distance further as shown in Fig.3(b). However, Eq.(13) only adjusts the scale of the gradient descent while the direction still keeps the same as \( F_k \), which weakens its effects for better representation learning. When it comes to our proposed ES-CRF, \( \nabla W \) can be further defined as:

\[
-\nabla W = (1 - P_k^*) \cdot F_k^* = (1 - P_k^*) \cdot \frac{1}{Z_k} \left( \sum_{j \in G} w_j F_j + F_k \right) \tag{14}
\]

\[
P_k^* = \text{sigmoid} \left( \sum_{j \in G} w_j Y_j + Y_k \right) \tag{15}
\]

where \( F_k^* \) is the refined feature representations by ES-CRF, and \( P_k^* \) is the refined score which is analogous to \( \hat{P}_k \) in Eq.(13). Comparing with Joint-CRF, it is clear that ES-CRF not only changes the scale of the gradient descent but also adjusts its optimization direction, which opens up room for more accurate segmentation performance. As depicted in Fig.3(c), assume \( W_1 \) is the class weight vector that pixel \( k \) belongs to, while \( W_2 \) is the other one which has a higher co-occurrent probability with \( W_1 \) in the same image. ES-CRF designs an effective message passing procedure to purify the feature representation of boundary pixels assisted by inner pixels from the same object (\( F_j \) in Fig.3(c)). In this way, it relieves the influence of disturbing gradients and makes the inter-class distance between \( W_1(W_1^*) \) and \( W_2 \) further, which means more discriminative feature representations.

4 Experiment

4.1 Experiment Setup

Datasets. We follow the previous works [2, 17, 58, 40, 5] and perform various experiments on two challenging semantic segmentation benchmarks, i.e., Cityscapes [9] and ADE20K [56]. Due to the space limit, a detailed description of these two datasets can be found in our Appendix.

Implementation Details. We adopt DeeplabV3+ [5] with ResNet [18] pretrained on ImageNet [38] as our baseline to implement ES-CRF. The detailed information follows standard settings in [2, 17, 58, 40, 5] and we add it into our Appendix. Specially, we employ SLIC [1], an excellent superpixel segmentation algorithm, to divide each image of ADE20K and Cityscapes into 200 and 600 blocks respectively. To verify the effectiveness of our approach for semantic segmentation, we adopt two common metrics in our experiments, i.e., class-wise mIoU to measure the overall segmentation performance and 1-pixel boundary F-score [42] to measure the boundary segmentation performance. All experiments are conducted on 8 2080Ti GPUs.

4.2 Ablation Study

4.2.1 Comparisons with Related Methods

As shown in Table 1, we compare our proposed ES-CRF with other traditional CRF-based methods, i.e., Vanilla-CRF and Joint-CRF. First of all, it is clear that all the CRF-based methods outperform the baseline model by a large margin, which well verifies the main claim in [2, 3, 28, 32, 29] that CRF is beneficial to boundary segmentation (F-score). What’s more, ES-CRF achieves the best result among all those methods, which surpasses the baseline model with up to 1.48% mIoU and 2.20% F-score improvements. ES-CRF fuses the CRF mechanism into the CNN network as an organic whole. It relieves the disturbing gradients caused by the PFB problem and adjusts the feature representations to boost the overall segmentation performance as well as the boundary segmentation. Fig.1(a) also proves that ES-CRF can decrease the inter-class similarity consistently which results in more discriminative feature representations. Similar experiments on Cityscapes dataset can be found in our Appendix.

4.2.2 Ablation Study on Message Passing Strategies

As we have discussed in Sec.3.2, two message passing components, i.e., pairwise module \( \psi^p \) and superpixel-based module \( \psi^s \), play vital roles in our proposed ES-CRF. Table 2 shows that \( \psi^p \) and \( \psi^s \)
Table 1: Comparisons with baseline, Vanilla-CRF and Joint-CRF on ADE20K val dataset. * results are reported in single scale testing.

| Method         | ResNet-50 | ResNet-101 |
|----------------|-----------|------------|
|                | F-score (%) | mIoU (%) | F-score (%) | mIoU (%) |
| DeeplabV3+     | 14.25      | 42.72     | 16.15      | 44.60    |
| Vanilla-CRF    | 16.26      | 43.18 (+0.46) | 17.89      | 45.14 (+0.54) |
| Joint-CRF      | 16.32      | 43.69 (+0.96) | 18.03      | 45.61 (+1.01) |
| ES-CRF (Ours)  | **16.45**  | **44.20 (+1.48)** | **18.32**  | **46.02 (+1.42)** |

Table 2: Comparisons between various message passing strategies, and ablation studies for different message passing components in ES-CRF, i.e., pairwise $\psi_f^p$ and superpixel-based $\psi_f^s$. * results are reported in single scale testing.

| Method         | $\psi_f^p$ | $\psi_f^s$ | mIoU(%) |
|----------------|------------|------------|---------|
| DeeplabV3+     | 42.72      |            | 44.60   |
| + Non-local    | 43.52 (+0.80) | 45.34 (+0.74) |         |
| ES-CRF         | ✓          | ✓          | 43.91 (+1.19) | 45.47 (+0.87) |
|                | ✓          | ✓          | 43.83 (+1.11) | 45.85 (+1.25) |
|                | ✓ ✓        | ✓ ✓        | **44.20 (+1.48)** | **46.02 (+1.42)** |

can boost the overall segmentation performance on ADE20K val dataset with up to 1.19% mIoU and 1.25% mIoU gains when integrated into the baseline model respectively. Moreover, if we fuse them as a whole into ES-CRF, they can further promote the segmentation performance by up to 1.48% mIoU improvements. We also attempt to introduce Non-local [46], another famous message passing method, into our experiments for comprehensive comparisons even though it actually has different design concepts from ours. Unfortunately, we find that although Non-local achieves improvements over the baseline, it is still inferior to our ES-CRF. Further intuitive visualization analysis for message passing components in ES-CRF is discussed in the next section.

4.2.3 Visualization Analysis for Pairwise Message Passing

In Sec.3.2, we mentioned that ES-CRF takes an equivalent transformation to replace hand-designed Gaussian kernels in Vanilla-CRF with simple convolution operators for more flexible end-to-end optimization. The convolution operation involves two aspects. One is the appearance similarity and the other one is the relative position between pixels. As depicted in Fig.4(a), we take a pixel $k$ in the stool for an example and show its relationship with other pixels in the image. Fig.4(b) and Fig.4(c) show its appearance similarity and cosine position embedding with other pixels respectively. It is clear that pixels share similar colors or close to the target pixel $k$ tend to be highlighted. Subsequently, pixels close to $k$ achieve higher values. (d) The visualization of $\psi_f^p$ in ES-CRF. $\psi_f^p$ focuses more on most relevant pixels compared with $k$.

![Image](image_url)

**Figure 4:** Visualization of pairwise message passing module $\psi_f^p$ in ES-CRF. (a) A target pixel $k$ of the stool in the image. (b) The appearance similarity between other pixels and pixel $k$. Pixels share the similar colors with $k$ tend to be highlighted. (c) The visualization of the relative position between $k$ and other pixels. Pixels close to $k$ achieve higher values. (d) The visualization of $\psi_f^p$ in ES-CRF. $\psi_f^p$ focuses more on most relevant pixels compared with $k$. 

8
### Table 3: Comparisons with other state-of-the-art methods on ADE20K val dataset.

| Method      | Reference | Backbone | mIoU (%) |
|-------------|-----------|----------|----------|
| CCNet [23]  | ICCV2019  | ResNet101| 45.22    |
| ANL [58]    | ICCV2019  | ResNet101| 45.24    |
| OCRNet [51] | ECCV2020  | ResNet101| 45.28    |
| EfficientFCN [31] | ECCV2020 | ResNet101| 45.28    |
| GFFNet [27] | AAAI2020  | ResNet101| 45.33    |
| APCNet [17] | ICCV2019  | ResNet101| 45.38    |
| OCNet [52]  | IJCV2021  | ResNet101| 45.45    |
| DMNet [16]  | ICCV2019  | ResNet101| 45.50    |
| RecoNet [6] | ECCV2020  | ResNet101| 45.54    |
| SPNet [21]  | CVPR2020  | ResNet101| 45.60    |
| DNL [49]    | ECCV2020  | HRNetV2  | 45.82    |
| ACNet [13]  | ICCV2019  | ResNet101| 45.90    |
| DNL [49]    | ECCV2020  | ResNet101| 45.97    |
| CaCNet [30] | ECCV2020  | ResNet101| 46.12    |
| CPNet [50]  | CVPR2020  | ResNet101| 46.27    |
| STLNet [57] | CVPR2021  | ResNet101| 46.48    |
| **ES-CRF (Ours)** | N/A | ResNet101 | **46.83** |

### Table 4: Comparisons with other state-of-the-art methods on Cityscapes val dataset.

| Method      | Reference | Backbone | mIoU (%) |
|-------------|-----------|----------|----------|
| SCF [20]    | ECCV2020  | ResNet101| 80.33    |
| CCNet [23]  | ICCV2019  | ResNet101| 81.3     |
| GFFNet [27] | AAAI2020  | ResNet101| 81.8     |
| RecoNet [6] | ECCV2020  | ResNet101| 81.6     |
| ACNet [13]  | ICCV2019  | ResNet101| 82.00    |
| RAANet [40] | NeurIPS2020| ResNet101| 81.9     |
| HANet [8]   | CVPR2020  | ResNet101| 82.05    |
| RPCNet [54] | CVPR2020  | ResNet101| 82.1     |
| SpyGR [26]  | CVPR2020  | ResNet101| 80.5     |
| SETR [55]   | CVPR2021  | T-Large  | 82.15    |
| STLNet [57] | CVPR2021  | ResNet101| 82.3     |
| **ES-CRF (Ours)** | N/A | ResNet101 | **82.74** |

in Fig.4(d), we directly visualize the results of our pairwise message passing module $\psi^j_p$ defined in Eq.5. We can find that $\psi^j_p$ becomes concentrated on the most relevant pixels compared with the pixel $k$, which verifies the reliability of our pairwise message passing design.

### 4.3 Comparisons with State-of-the-art Methods

**ADE20K.** We first compare our ES-CRF with existing methods on the widely used ADE20K val set. We follow standard settings in [23, 58, 51, 57, 6] to adopt multi-scale testing and left-right flipping strategies for a fair comparison. Results are shown in Table 3. Obviously, ES-CRF outperforms existing approaches with a dominant advantage.

**Cityscapes.** To verify the generalization of our proposed method, we further perform detailed comparisons with other SOTA methods on Cityscapes val set. Multi-scale testing and left-right flipping strategies are also adopted. The result is reported in Table 4. Remarkably, our algorithm achieves **82.74%** mIoU and outperforms previous methods by a large margin.
5 Conclusion

In this paper, we raise an important but ignored problem in the semantic segmentation, named as *Pernicious Favor on Boundary* (PFB) for simplicity. PFB is mainly caused by pixels around the object boundary due to the continuous expansion of receptive fields in the CNN network. We dive deep into it and propose a novel method, *ES-CRF*, via combining CNN network with CRF as an organic whole to refine the boundary segmentation and alleviate PFB at the same time. Eventually, our proposed method achieves new SOTA results on both two challenging benchmarks, i.e., Cityscapes and ADE20K. What’s more, *ES-CRF* focuses on boundary pixel segmentation, which may lack the ability to deal with other computer vision tasks, i.e., object detection.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Wanli Chen, Xinge Zhu, Ruqi Sun, Junjun He, Ruiyu Li, Xiaoyong Shen, and Bei Yu. Tensor low-rank reconstruction for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 52–69, Cham, 2020. Springer International Publishing.

[7] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A2-nets: Double attention networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[8] Sungta Choi, Joanne T. Kim, and Jaegul Choo. Cars can’t fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Henghui Ding, Xudong Jiang, Ai Qin Liu, Nadia Magenmat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6819–6829, 2019.

[11] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qin Liu, and Gang Wang. Context-foregrounded feature and gated multi-scale aggregation for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2393–2402, 2018.

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[13] Jun Fu, Jing Liu, Yuhang Wang, Yong Li, Yongjun Bao, Jinhui Tang, and Hanqing Lu. Adaptive context network for scene parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6748–6757, 2019.

[14] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *European conference on computer vision*, pages 597–613. Springer, 2016.

[15] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[16] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019.

[17] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7519–7528, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
[19] Shengfeng He, Rynson WH Lau, Wexi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *International journal of computer vision*, 115(3):330–344, 2015.

[20] Yang He, Bernt Schiele, and Mario Fritz. Synthetic convolutional features for improved semantic segmentation. In Adrien Bartoli and Andrea Fusillo, editors, *Computer Vision – ECCV 2020 Workshops*, pages 320–336, Cham, 2020. Springer International Publishing.

[21] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4003–4012, 2020.

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[23] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Cross-slice attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[24] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24:109–117, 2011.

[25] L'ubor Ladický, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip H. S. Torr. What, where and how many? combining object detectors and crfs. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 424–437, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[26] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[27] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11418–11425, 2020.

[28] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van den Hengel. Deeply learning the messages in message passing inference. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[29] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.

[30] Jianbo Liu, Junjun He, Yu Qiao, Jimmy S. Ren, and Hongsheng Li. Learning to predict context-adaptive convolution for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 769–786, Cham, 2020. Springer International Publishing.

[31] Jianbo Liu, Junjun He, Jiawei Zhang, Jimmy S. Ren, and Hongsheng Li. Efficientfcn: Holistically-guided decoding for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 1–17, Cham, 2020. Springer International Publishing.

[32] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pages 1377–1385, 2015.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[34] Nadia Payet and Sinisa Todorovic. (rf)² – random forest random field. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[35] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE conference on computer vision and pattern recognition*, pages 733–740. IEEE, 2012.

[36] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, IEEE International Conference on*, volume 2, pages 10–10. IEEE Computer Society, 2003.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[39] Abbasheke Sharma, Oncel Tuzel, and Ming-Yu Liu. Recursive context propagation network for semantic scene labeling. In *NIPS*, volume 1, page 1, 2014.

[40] Dingguo Shen, Yuanfeng Ji, Ping Li, Yi Wang, and Di Lin. Ranet: Region attention network for semantic segmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13927–13938. Curran Associates, Inc., 2020.

[41] Guang Shu, Afshin Dehghan, and Mubarak Shah. Improving an object detector and extracting regions using superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.
T. Takikawa, D. Acuna, V. Jampani, and S. Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2020.

Takahiro Toyoda and Osamu Hasegawa. Random field model for integration of local information and global information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(8):1483–1489, 2008.

Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In European conference on computer vision, pages 13–26. Springer, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, Computer Vision – ECCV 2020, pages 191–207, Cham, 2020. Springer International Publishing.

Changqian Yu, Jiachen Lu, Hengshuang Zhao, Xiaoou Tang, Xiatian Zhu, and Jianfeng Feng. Scene parsing through ade20k dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 593–602, 2019.

Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 593–602, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes] See Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work doesn’t have any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Data and instructions are included in section 4. Code will be released soon.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Results are reported in our experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 1.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [No] We use open source data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] It doesn’t contain personally identifiable information or offensive content.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]