Super-BPD: Super Boundary-to-Pixel Direction for Fast Image Segmentation

Anonymous CVPR submission

Paper ID 5969

Abstract

Image segmentation is a fundamental vision task and still remains a crucial step for many applications. In this paper, we propose a fast image segmentation method based on a novel super boundary-to-pixel direction (super-BPD) and a customized segmentation algorithm with super-BPD. Precisely, we define BPD on each pixel as a two-dimensional unit vector pointing from its nearest boundary to the pixel. In the BPD, nearby pixels from different regions have opposite directions departing from each other, and nearby pixels in the same region have directions pointing to the other or each other (i.e., around medial points). We make use of such property to partition image into super-BPDs, which are novel informative superpixels with robust direction similarity for fast grouping into segmentation regions. Extensive experimental results on BSDS500 and Pascal Context demonstrate the accuracy and efficiency of the proposed super-BPD in segmenting images. Specifically, we achieve comparable or superior performance with MCG while running at ~25fps vs 0.07fps. Super-BPD also exhibits a noteworthy transferability to unseen scenes.

1. Introduction

Image segmentation aims to decompose an image into non-overlapping regions, where pixels within each region share similar perceptual appearance, e.g., color, intensity, and texture. Image segmentation is a crucial step for many vision tasks such as object proposal generation [31, 43], object detection [20], semantic segmentation [17]. Though it has been explored for many years, an efficient and accurate segmentation still remains challenging.

There are many unsupervised image segmentation methods that can be roughly categorized into early merging and clustering methods [45, 13], active contours [21, 6, 7], variational approaches [28, 34], watersheds [39, 29], segmentation with graphical models [18, 35, 5]. Though these classical methods are mathematically rigorous and achieve desirable results in some specific applications, as depicted in Fig. 1, they usually do not perform well in segmenting natural images or are not very efficient. Superpixel segmentation [1, 38] is an efficient alternative that oversegments image into small and compact regions. A grouping process [24] is usually involved to produce final segmentation.

Thanks to convolutional neural networks (CNNs), semantic segmentation [25, 9, 44] that classifies each pixel into a predefined class category has witnessed a great progress in both accuracy and efficiency. Nevertheless, it does not generalize well to unseen object categories. Alternatively, for image segmentation, some methods [3, 31, 26, 16] resort to learn contours, followed by a transformation to bridge up the gap between contours and segmentation. As shown in Fig. 1, though these methods achieve impressive performances, the inevitable contour to segmentation transformation takes great effort to remedy leakage problem at weak boundaries, and is usually time-consuming.

Different from previous methods that directly learn contours and transform contours to segmentation, we propose a novel super boundary-to-pixel direction (super-BPD) and an efficient segmentation algorithm with super-BPD. Specifically, we introduce a boundary-to-pixel direction (BPD) on each pixel in terms of a two-dimensional unit vector pointing from its nearest boundary to the pixel.
vector, pointing from its nearest boundary to the underlying pixel. The BPD not only provides contour positions but also encodes the relative position of each pixel to the corresponding region boundary, and thus relationship of neighboring pixels. This allows us to efficiently partition an image into super-BPDs such that each pixel and the pixel it points to and having similar direction are in the same super-BPD. In fact, the super-BPD can be regarded as a novel alternative of classical superpixel, which provides robust direction for further grouping into segmentation regions.

The set of super-BPDs form a region adjacency graph (RAG), where the edges are weighted by the direction similarity along the boundaries of adjacent super-BPDs. Nearby pixels within different regions have approximately opposite BPD, and hence small direction similarity. Such property also holds even at weak boundaries, where the learned BPD smoothly diverges to approximately opposite directions along the direction (see Fig. 2 for an example). This equips the super-BPDs with robust direction similarity that helps to group similar super-BPDs within the same perceptual region and separate super-BPDs of different regions. We leverage such direction similarity between adjacent super-BPDs to partition the RAG into different clusters, resulting in a segmentation. As shown in Fig. 1, the proposed super-BPD achieves a good trade-off between accuracy and efficiency on PASCAL Context dataset [27].

The main contribution of this paper are two-fold: 1) We present a novel super boundary-to-pixel direction (super-BPD), which is a powerful alternative of classical superpixel. super-BPD provides robust direction similarity between adjacent super-BPDs, which allows for an efficient image segmentation. 2) We propose an efficient segmentation algorithm with super-BPDs in a coarse-to-fine way based on the direction similarity, leading to a good trade-off between segmentation accuracy and efficiency.

2. Related Work

We shortly review some works on image segmentation and other vision tasks leveraging direction information.

2.1. Image Segmentation

Unsupervised Methods. Many image segmentation methods have been proposed in the past two decades, and can be roughly classified into several categories. Early segmentation methods are driven by region merging and clustering methods. Typical examples are region competition [45] and mean shift [13]. Active contours [21, 6, 7] are another type of popular segmentation methods that evolve region contours by minimizing some energy functions. Variational approaches [28, 34] also attempt to minimize some energy functions based on some appropriate hypothesis about the underlying image (e.g., piece-wise constant in [28]). A set of watersheds [39, 29] have been proposed from the community of mathematical morphology. They segment image domain into catchment basins (i.e., regions) and watershed lines (i.e., contours). Another popular family of segmentation methods are based on graphical models [35, 5, 18], which model image domain as a graph and attempt to cut graphs based on some energy minimization. Besides these segmentation methods, superpixel methods [1, 33] aim to over-segment an image into small and compact regions.

Supervised Methods. A number of learning-based image segmentation methods have been proposed. Different from semantic segmentation that can be regarded as a pixel-wise category classification problem, the mainstream learning-based segmentation methods [3, 31, 26, 16] start with learning contours. They then resort to oriented watershed transformations and globalization via spectral clustering to alleviate the leakage problem at weak boundaries. However, such contour to segmentation process is usually time-consuming. In [41], the authors propose mutex watershed (MWS) by learning local attractive/repulsive affinities, followed by a modified maximum spanning tree to segment images. Another direction is to learn a feature embedding [24] for SLIC superpixels [1] such that superpixels within the same region (resp., different regions) have similar (resp., very different) embedded features. A simple merging algorithm based on the embedded features is then adopted to group superpixels into perceptual regions.

The proposed super-BPD falls into supervised methods. Different from the existing learning-based methods, super-BPD does not rely on contours and is free of the time-
Figure 3. Illustration of the proposed BPD and super-BPD. For each pixel \( p \), we find its nearest boundary pixel \( B_p \) within different regions as \( p \). The BPD \( D(p) \) is defined as the two-dimensional unit vector pointing from \( B_p \) to \( p \). We adopt a CNN to learn the BPD, based on which we partition the image into super-BPDs represented by the root pixels. We then apply a dilation to merge nearby root pixels.

### 2.2. Direction Cues for Vision Applications

The direction information has been recently explored in different vision tasks. Some methods rely on similar direction field defined on regions of interest. For instance, deep watershed transform [4] propose to learn the direction field on semantic segmentation, and then regress the distance to boundaries based on the direction information, followed by a classical watershed to produce instance segmentation. Textfield [42] and DeepFlux [40] defines similar direction field on text areas and skeleton context for scene text detection and skeleton extraction, respectively. The direction cue is also explored in MaskLab [8] and IRnet [2] for improving instance segmentation and weakly instance segmentation, respectively. PifPaf [23] and PVNet [30] leverage direction cue for 2D human pose estimation and 6DoF pose estimation, respectively.

The proposed super-BPD builds upon boundary-to-pixel direction (BPD), which is in spirit similar to [4, 42, 40] but defined on the whole image domain instead of regions of interest. The BPD learning confirms that it is possible to learn the direction field encoding the relative position of each pixel with respect to region contours in natural images. In this sense, the BPD can be seen as an extension of the flux notion introduced in [36] for binary object skeletonization, to natural images for image segmentation. super-BPD differs a lot with [4, 42, 40] in how to use BPD defined on the whole image. The major contribution is the extension of BPD to super-BPDs that enhances the robustness of direction information of BPD. In fact, the super-BPD is a powerful alternative of classical superpixel, and provides robust direction similarity between neighboring super-BPDs. Based on this, we propose an efficient coarse-to-fine RAG partition algorithm, leading to an accurate and efficient generic image segmentation.

### 3. Super Boundary-to-Pixel Direction

Current accurate segmentation methods achieve great performances with time-consuming post-processing, which limits their usages in practice. Whereas, efficient segmentation methods provide degenerated results. We propose to remedy this issue by introducing a novel super boundary-to-pixel direction (super-BPD). The boundary-to-pixel direction (BPD) is defined on each pixel \( p \) as the two-dimensional unit vector pointing from its nearest boundary pixel \( B_p \) to \( p \). Such BPD encodes the relative position between each pixel \( p \) and the region (containing \( p \)) boundary. We adopt a CNN to learn such BPD, which is then used to partition the image into super-BPDs, a powerful alternative of classical superpixels. In fact, super-BPDs provides robust direction similarity between adjacent super-BPDs, thus allowing fast image segmentation by partitioning the region (i.e., super-BPD) adjacency graph (RAG).

#### 3.1. Boundary-to-Pixel Direction (BPD)

**Definition.** As shown in Fig. 3, for each pixel in the image domain \( p \in \Omega \), we find its nearest boundary pixel \( B_p \), which lies outside the region containing \( p \). A two-dimensional unit vector pointing from \( B_p \) to \( p \) is then defined as the BPD \( D \) on pixel \( p \) given by

\[
D(p) = \frac{B_p - p}{|B_p - p|},
\]
where $|B_p|_p$ is the distance between $B_p$ and $p$. The BPD provides cues about contour positions and relative position of each pixel $p$ to its region boundary. It is worth to note that generating BPD from ground-truth annotation could be efficiently achieved by distance transform algorithm.

**Architecture and Learning.** We adopt a CNN to predict BPD in terms of a map of two-dimensional vectors having the same size as input image. For a fair comparison with other methods, we adopt VGG16 [37] as the backbone network, where the last max pooling layer and all following layers are discarded. We also leverage ASPP layer [9] to enlarge the receptive field, better coping with large regions. We extract features from different stages of VGG16 to aggregate multi-scale information. Specifically, we apply $1 \times 1$ convolutions to $conv3, conv4, conv5$, and ASPP layers, followed by a concatenation of these side output features after resizing them to the size of $conv3$. Finally, we apply three consecutive $1 \times 1$ convolutions on the fused feature maps, followed by an upsampling with bilinear interpolation to predict the BPD.

We define the loss function in terms of both $L_2$-norm distance and angle distance:

$$L = \sum_{p \in \Omega} w(p)(\|D(p) - \hat{D}(p)\|_2 + \alpha \|\cos^{-1}(D(p), \hat{D}(p))\|^2),$$

where the adaptive weight at pixel $p$ $w(p) = 1/\sqrt{|R_p|}$ is proportional to the inverse square root of the size of region $R_p$ containing $p$ and $\alpha$ the hyper-parameter (set to 1).

### 3.2. BPD Grouping into Super-BPDs

We first partition the image into super-BPDs (i.e., superpixels with direction) and then merge nearby root pixels.

**From BPD to super-BPD.** We partition the image into super-BPDs in terms of a forest encoded by a parent image $\mathcal{P}$ (see Fig. 3 and Line 5 to 9 in Algorithm 1). Precisely, for each pixel $p$, the direction $\angle \hat{D}(p)$ is binned into one of the eight directions, pointing to its neighbor $n_p$. If the angle between the directions of $\hat{D}(p)$ and $\hat{D}(n_p)$ is smaller than a given threshold $\theta_a$, we group them together by setting the parent of $p$ to $n_p$. Otherwise, $p$ is a root node representing a super-BPD. We insert $p$ into the set of roots $\mathcal{R}$.

**Merge nearby root pixels.** Since a pair of neighboring pixels on different sides of the region symmetry axis have very different directions pointing to each other, there are many nearby roots representing adjacent super-BPDs within the same region (See Fig. 3 for an example). We apply a simple dilation with $3 \times 3$ structuring element to group nearby roots (i.e., super-BPDs). This is efficiently achieved by updating the parenthood $\mathcal{P}$ of each root pixel $r$ to all root pixels within the bottom half of $3 \times 3$ window $N_3$ centered at $r$ (see Line 11 to 13 in Algorithm 1).

### 4. Image Segmentation with Super-BPD

Similar to superpixel-based methods, we take a graph-based approach for image segmentation with super-BPD. We first construct an adjacency graph of super-BPD, then define graph edge weights and apply the graph partitioning method to group super-BPDs into segments (see Fig. 4).

**Region Adjacency Graph.** Based on previous super-BPD partition encoded by the parent image $\mathcal{P}$, we construct a region adjacency graph $(V_R, E)$ (see Line 5 in Algorithm 2), where $V_R$ denotes the set of roots representing super-BPDs and $E$ stands for the set of edges linking the roots of adjacent super-BPDs.

**Graph Edge Weight.** We compute the area $A$ on each root $r \in V_R$ as the underlying super-BPD size, and the direction similarity $S$ on each edge. For each edge $e = (r_1, r_2) \in E$ linking two regions $R_1$ and $R_2$, let $B(e) = \{p_1^1, p_1^2\}$ be the set of neighboring pixels along the boundaries such that $p_1^1 \in R_1$ and $p_1^2 \in R_2$, we define the direction similarity on
Algorithm 2: Algorithm for segmentation with super-BPDs encoded in $P$. See Sec. 4 for details.

\textbf{Input:} $P$, threshold $\theta_1, \theta_s, a_t, a_s$

\textbf{Output:} Set of linking edges $E_l$

\begin{algorithmic}[1]
\Function{Super_BPD2SEG}{$P, \theta_1, \theta_s, a_t, a_s$}
\State // initialization
\State $E_l \leftarrow \emptyset$, $S \leftarrow 0$, $A \leftarrow 0$
\State // Construct region adjacency graph
\State $(V_R, E^s_A, E^t_A, S, A) \leftarrow \text{Get_RAG}(P)$
\State // merge similar large and small super-BPDs
\For{$e = (r_1, r_2) \in E^s_A$}
\If{$\mathcal{A}(r_1) > a_t$ and $\mathcal{A}(r_2) > a_t$ and $S(e) \theta(\mathcal{A}(r_1), \mathcal{A}(r_2))$ and not $\mathcal{Rep}(r_1, r_2)$ then}
\State Merge($r_1, r_2$), $E_l$.push($e$) // updating
\EndIf
\EndFor
\State // merge tiny super-BPDs
\For{$e = (r_1, r_2) \in E^t_A$}
\If{$\mathcal{A}(r_1) < a_t$ or $\mathcal{A}(r_2) < a_t$ and not $\mathcal{Rep}(r_1, r_2)$ then}
\State Merge($r_1, r_2$), $E_l$.push($e$) // updating
\EndIf
\EndFor
\State return $E_l$
\EndFunction
\end{algorithmic}

$e$ as following:

$$S(e) = 180 - \frac{\sum_{i=1}^{\mid B(e) \mid} \cos^{-1}\left(\hat{D}(P_s(p_1^i)), \hat{D}(P_s(p_2^i))\right)}{|B(e)|},$$
(3)

where $|B(e)|$ denotes the number of boundary points between $R_1$ and $R_2$, and $P_s(p)$ stands for the $s$-th ancestor starting from the pixel $p$ ($s = 0$ means the pixel itself).

We further divide the set of edges into attractive edges $E^s_A$ sorted in decreasing order of direction similarity and repulsive edges $E^t_A$ having small direction similarity ($S < 10$).

\textbf{Graph Partitioning.} Based on the RAG ($V_R, E^s_A \cup E^t_A$) and associated information, we progressively merge adjacent super-BPDs (from large to tiny) having large $S$ (see Line 7 to 13 in Algorithm 2). Inspired by [18] such that merging of large regions requires large similarity, we set different direction similarity threshold for merging adjacent super-BPDs of different sizes. Specifically, for each edge $e = (r_1, r_2)$ that links two adjacent super-BPDs $R_1$ and $R_2$, we adopt a piece-wise constant threshold function $\theta(\mathcal{A}(r_1), \mathcal{A}(r_2))$ with values set to $\theta_t$ if both $\mathcal{A}(r_1)$ and $\mathcal{A}(r_2)$ are larger than $a_s$, and set to $\theta_s < \theta_t$ if both of them are larger than $a_t$ and one of them is smaller than $a_s$.

We first merge similar large and small super-BPDs. For that, we consider each edge $e = (r_1, r_2) \in E^s_A$ in decreasing order of the direction similarity. If the direction similarity $S(e)$ is larger than $\theta(S(r_1), S(r_2))$ and the merge of $R_1$ and $R_2$ does not violate the repulsive rule, we merge $R_1$ and $R_2$, and insert $e$ to $E_l$. We then merge the tiny regions (smaller than $a_t$ pixels) to the neighboring super-BPD having highest direction similarity. Note that each merge should not violate the repulsive rule and trigger the update of repulsive information $\text{Rep}$ between super-BPDs and super-BPD size $A$.

The whole segmentation from BPD is composed of three stages: 1) Super-BPD partition (Line 5 to 9 in Algorithm 1). The complexity is $O(N)$, where $N$ denotes the number of pixels in image. 2) Nearby root pixels grouping (Line 11 to 13 in Algorithm 1), which has a linear complexity with the number of root pixels. 3) Super-BPD merging (Algorithm 2), which has a quasi-linear time complexity with respect to the number of edges (i.e., in hundreds order) in RAG. Therefore, the whole post-processing has a near linear complexity, and is thus efficient.

5. Experiments

We conduct generic image segmentation experiments on PASCAL Context [27] and BSDS500 [3] dataset.

5.1. Implementation Details

For training on BSDS500 dataset, we adopt the same data augmentation strategy used in [24]. Specifically, the training images are rotated to 16 angles and flipped at each angle, then we crop the largest rectangle from the transformed images, yielding 9600 training images. Since PASCAL Context dataset has enough images, We only randomly flip images during training. The proposed network is initialized with the VGG16 model pretrained on ImageNet [15] and optimized using ADAM [22]. Both models are trained for first 80k iterations with initial learning rate $10^{-5}$ for backbone layers and $10^{-4}$ for extra layers. We then decay the learning rates to $10^{-6}$ and $10^{-5}$, respectively, and continue to train the model for another 320k iterations on both BSDS500 PASCAL Context.

For the hyper-parameter settings, unless explicitly stated otherwise, we set $\theta_s$ to 45 for super-BPD partition. The threshold $a_t$ for small and $a_s$ for tiny regions are fixed to 1500 and 200. The step $s$ involved in computing the direction similarity in Eq. (3) is set to 3. The other two hyperparameters $\theta_t$ and $\theta_s$ for merging large and small regions are tuned for the optimal dataset setting (ODS) on each dataset.

The proposed super-BPD is implemented with PyTorch platform. All experiments are carried out on a workstation with an Intel Xeon 16-core CPU (3.5GHz), 64GB RAM, and a single Titan Xp GPU. Training on PASCAL Context using a batch size of 1 takes about 6 hours.

5.2. Comparing with State-of-the-art Methods

\textbf{Datasets.} PASCAL Context [27] contains precisely localized pixel-wise semantic annotations for the whole image. We ignore the semantics of each region for benchmarking
the generic image segmentation methods. This dataset is composed of 7605 trainval images and 2498 test images.

BSDS500 [3] is a benchmark dataset for image segmentation and boundary detection. It is divided into 200 training images, 100 val images, and 200 test images. Each image has 5-10 different segmentation ground-truth annotations.

**Metrics.** We use four standard benchmark measures adopted in [3]: $F_b$-measure for boundaries, $F_{op}$-measure for objects and parts $F_{op}$ introduced in [32].

**In-Dataset Evaluation.** We compare the proposed super-BPD with other state-of-the-art image segmentation methods on both PASCAL Context and BSDS500. Some qualitative comparisons are shown in Fig. 5. Super-BPD correctly segments images into perceptual regions on both datasets.

The quantitative comparison on PASCAL Context is depicted in Tab. 1. Super-BPD achieves a pleasant trade-off between accuracy and efficiency on segmenting images in PASCAL Context. Specifically, super-BPD performs competitively with COB [26] while being much faster based on...
Table 3. Quantitative evaluation in terms of ODS metrics for super-BPD and DEL [24] under cross-dataset validation.

<table>
<thead>
<tr>
<th>Methods</th>
<th>DEL-C</th>
<th>Super-BPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS F&lt;sub&gt;op&lt;/sub&gt;</td>
<td>0.328</td>
<td>0.347</td>
</tr>
<tr>
<td>Covering PRJ VI</td>
<td>0.58</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>1.73</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td>0.319</td>
<td>0.356</td>
</tr>
<tr>
<td></td>
<td>0.57</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>0.76</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4. Influence of ASPP module on performance in ODS F<sub>op</sub>.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ASPP</th>
<th>ODS F&lt;sub&gt;op&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL Context</td>
<td></td>
<td>0.454</td>
</tr>
<tr>
<td>BSDS500</td>
<td>✓</td>
<td>0.472</td>
</tr>
<tr>
<td>BSDS500</td>
<td>✓</td>
<td>0.348</td>
</tr>
</tbody>
</table>

The quantitative comparison on BSDS500 dataset is shown in Tab. 2. Super-BPD approaches the “upper bound” of ground-truth segmentation. It is noteworthy to mention that the lower accuracy on BSDS500 than on PASCAL Context is due to inconsistent annotations between different subjects.

Cross-Dataset Evaluation. To demonstrate the generalization ability of the proposed super-BPD, we evaluate the model trained on one dataset and test the trained model on another dataset. We mainly compare super-BPD with DEL-C [24], which is also dedicated for a good trade-off between accuracy and efficiency. As depicted in Tab. 3, super-BPD is robust in generalizing to unseen datasets. Specifically, super-BPD outperforms DEL-C [24] for both PASCAL Context to BSDS500 and BSDS500 to PASCAL Context setting. In fact, super-BPD even outperforms DEL-C [24] properly trained on the corresponding training set.

Runtime Analysis. Super-BPD has three stages: BPD inference, super-BPD partition, and segmentation with super-BPD. BPD inference on the GPU using VGG16 takes on average 22 ms for a PASCAL Context image, and super-BPD partition and segmentation with super-BPD require on average 22 ms for a PASCAL Context image on the CPU. As depicted in Tab. 1 and Tab. 2, super-BPD is much more efficient than the other competing methods while achieving comparable or superior performance.

5.3. Ablation Studies

ASPP Module. We first study the effect of ASPP module that increases receptive field for coping with large regions. As shown in Tab. 4, when the ASPP module is not used, the performance slightly decreases on both PASCAL Context and BSDS500 dataset.

Loss Functions. We study the effect of different loss functions to train the network on PASCAL Context dataset. As depicted in Tab. 5, both L<sub>2</sub> loss function and the angular...
6. Application: Object Proposal Generation

To evaluate the effectiveness of the proposed super-BPD in practical applications, we apply super-BPD to object proposal generation, which is a prerequisite step for a number of mid-level and high-level vision tasks such as object detection [19]. There are many types object proposal generation methods, among which an important category builds upon image segmentation. Following DEL [24], we replace EGB [18] in MTSE [10] with the proposed super-BPD to refine the bounding boxes produced by BING [12]. The detection recall with 0.8 IoU overlap versus the number of proposals is depicted in Fig. 6. Using super-BPD in MTSE significantly improves MTSE with DEL and EGB at high IoU overlap threshold. More experiments of applications using image segmentation is out of the scope of this paper and will be explored in our future work.

7. Conclusion and Limitations

We propose a fast image segmentation method based on a novel super boundary-to-pixel direction (super-BPD) and a customized segmentation with super-BPD. Specifically, the BPD allows a fast image partition into super-BPDs, a powerful alternative of classical superpixels with robust direction similarity. We then construct a region adjacency graph and merge super-BPDs based on the direction similarity along their boundaries, resulting in a fast image segmentation. The proposed super-BPD achieves a good compromise between accuracy and efficiency, and can separate nearby regions with weak boundaries. In particular, super-BPD achieves comparable or superior performance with MCG but being near real-time. Besides, super-BPD also has an appealing transferability to unseen scenes. This allows potential use of super-BPD in many vision tasks. For example, we have verified the usefulness of super-BPD in object proposal generation. In the future, we would like to explore super-BPD in other applications.

Though the proposed super-BPD achieves a pleasant trade-off between generic image segmentation accuracy and efficiency, it is still difficult for super-BPD to accurately segment small regions. This is because that the prediction of BPD around small regions is not very accurate due to dramatic changes of direction. Some failure cases are illustrated in Fig. 7, where small regions are not accurately segmented. The segmentation of very small regions also remains a problem for other image segmentation methods.

References


