Perceive, Excavate and Purify: A Novel Object Mining Framework for Instance Segmentation

Jinming Su, Ruihong Yin, Xingyue Chen and Junfeng Luo
Meituan
{sujinming, yinruihong, chenxingyue02, luojunfeng}@meituan.com

Abstract

Recently, instance segmentation has made great progress with the rapid development of deep neural networks. However, there still exist two main challenges including discovering indistinguishable objects and modeling the relationship between instances. To deal with these difficulties, we propose a novel object mining framework for instance segmentation. In this framework, we first introduce the semantics perceiving subnetwork to capture pixels that may belong to an obvious instance from the bottom up. Then, we propose an object excavating mechanism to discover indistinguishable objects. In the mechanism, preliminary perceived semantics are regarded as original instances with classifications and locations, and then indistinguishable objects around these original instances are mined, which ensures that hard objects are fully excavated. Next, an instance purifying strategy is put forward to model the relationship between instances, which pulls the similar instances close and pushes away different instances to keep intra-instance similarity and inter-instance discrimination. In this manner, the same objects are combined as the one instance and different objects are distinguished as independent instances. Extensive experiments on the COCO dataset show that the proposed approach outperforms state-of-the-art methods, which validates the effectiveness of the proposed object mining framework.

1. Introduction

Instance segmentation is a fundamental perception task, which aims to detect and segment pre-defined objects at the instance level. Over the past years, the task of instance segmentation has made significant progress with the development of deep neural networks [18] and has a wide range of applications in real-world scenarios such as autonomous driving [28] and video surveillance [40].

To address the task of instance segmentation, lots of learning-based methods have been proposed in recent years, achieving impressive performance. For the basic task, the existing methods can be summarized into three categories. (1) Top-down methods [2,12,15,35,39] address the problem based on object detection, i.e., detecting and segmenting the object in the box. (2) Bottom-up methods [9,11,29] deal with the problem in a labeling and clustering manner, i.e.,
learning the per-pixel embeddings first and then clustering them into groups. (3) The latest direct method [36, 37] aims at dealing with instance segmentation directly, without dependence on object detection or embedding-and-clustering. However, there still exist two challenges that hinder the development of instance segmentation. First, there usually exists indistinguishable objects in real-world scenarios, especially adjacent and overlapping objects as shown in Fig. 1 (a). Second, The relationships (such as the distance in feature space) between different instances are rarely studied while they are important for instance distinguishment as shown in Fig. 1 (b). Due to these two difficulties, instance segmentation remains a challenging vision task.

To address these two difficulties, many methods have made some efforts. To deal with indistinguishable objects, online hard example mining (OHEM) [32, 33] automatically select hard examples to make training more effective and efficient, which boosts the detection performance and is easy to transfer to instance segmentation. In addition, focal loss [25] focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. These methods mainly enhance the learning ability of difficult examples at the level of training strategies, which are indirect schemes without awareness at the object level. To address the second difficulty about modeling instance relationship, relation network [14] proposes an object relation module to process a set of objects simultaneously through interaction between their appearance feature and geometry, which allows modeling of their relations. Besides, IRNet [1] defines two kinds of relations between a pair of pixels: the displacement between their coordinates and their class equivalence, which are used to train the network effectively. These methods utilize neural networks to implicitly model the instance relationship, but it is still difficult to explore explicit representation for the relationship between instances.

Inspired by these observations and analysis, we propose a novel object mining framework for instance segmentation, as shown in Fig. 2. In this framework, we first introduce semantics perceiving to capture all pixels potentially belonging to an instance, and we select pixels with high category confidence as the original instance descriptors with semantics-level classification and location. In this manner, independent objects are easy to be perceived, and indistinguishable objects mainly exist in the case caused by object gathering. In order to discover indistinguishable objects, we propose an object excavating mechanism. In this mechanism, original descriptors are fed into an instance learning subnetwork to determine whether each pixel is the center of the instance around original descriptors. In this way, indistinguishable objects around origin instances are detected with classification and locations, which are named mined descriptors. After that, we put all the instance descriptors (including both original instance descriptors and mined descriptors) into the instance purifying graph to model the relationship between instances. In this graph, we constrain instance descriptors of the same objects to have the highest similarity but instance descriptors of different objects have
little similarity, which can pull the similar instances close and push away different instances. Therefore, the instance purifying graph can keep intra-instance similarity and inter-instance discrimination. Finally, all independent instances after purifying are utilized to generate final masks by mask learning. Experimental results on the COCO [26] dataset achieve state-of-the-art performance, which verifies the effectiveness of the proposed method.

The main contributions of this paper are as follows: 1) we propose an object mining framework for instance segmentation, which achieves state-of-the-art performance and validates the effectiveness of this perceiving, excavating, and purifying pipeline. 2) we put forward the object excavating mechanism, in which indistinguishable objects around original instances are discovered so that hard instances in an image are fully mined. 3) we introduce the instance purifying mechanism to model the relationship between instances, which maintains the intra-instance similarity and inter-instance discrimination.

2. Related Work

In this section, we review related works that aim to resolve the challenges of instance segmentation in three aspects.

2.1. Instance Segmentation

Instance segmentation is a basic vision task, which requires instance-level and pixel-level predictions. With the development of deep neural networks, many methods [9, 11, 29, 36, 37] have made great progress in the benchmark dataset [26]. For the challenging task, the existing methods can be classified into three folds. First, top-down methods are usually based on object detection to segment objects in a bounding box. For example, Mask R-CNN [12] extends the object detection network Faster R-CNN [30] by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. In particular, recent anchor-free methods (e.g., FCOS [35] and PolarMask [39]) also achieve good results with fast inference. The second kind of method is bottom-up methods, which are usually learning the feature of each pixel and clustering pixels into groups in feature space. For example, SSAP [11] proposes a pixel-pair affinity pyramid, which computes the probability that two pixels belong to the same instance in a hierarchical manner. The last kind of method is proposed in recent years, which aims at addressing the task of instance segmentation in direct ways. For example, a series of methods, like SOLO [36, 37], assign categories to each pixel within an instance according to the instance’s location and size, thus nicely converting instance segmentation into a single-shot classification-solvable problem. In the above methods, various strategies have improved the performance of instance segmentation. However, there still exist two main challenges including discovering indistinguishable objects and modeling the relationship between instances.

2.2. Hard Example Mining

Hard example mining is used to speed up the convergence of neural networks and enhance the discriminative power of the learned features. To reduce the computational overhead of identifying hard examples, existing works have been explored in two directions including an exact search within each mini-batch and an approximate search from the whole dataset [16, 25, 32, 33]. For example, OHEM [32] automatically selects hard examples in a mini-batch to make training more effective and more efficient, which is an intuitive algorithm that eliminates several heuristics and hyper-parameters in common use. In addition, UHEM [16] presents how large numbers of hard negatives can be obtained automatically by analyzing the output of a trained detector on video sequences, and describes simple procedures for mining large numbers of such hard negatives (and also hard positives) from unlabeled video data. Moreover, SCHEM [33] makes use of class signatures that keep track of feature embedding online with minor additional cost during training and identifies hard negative example candidates using the signatures. As for the above methods, they mainly enhance the learning ability with hard examples by means of training strategies, which are indirect schemes without awareness at the object level.

2.3. Object Relation

It is well believed that modeling relations between objects would make contributions to object recognition. Early works make use of object relations as a post-processing step [10], in which the detected objects are re-scored by considering object relationships. After stepping into the era of deep learning, learning-based object relation is widely explored. For example, relation network [14] introduces an object relation module to process a set of objects simultaneously through interaction between their appearance feature and geometry, thus allowing modeling of their relations. Additionally, IRNet [1] estimates rough areas of individual instances and detects boundaries between different object classes, which ensures to assign instance labels to the seeds and to propagate them within the boundaries. In this way, the entire areas of instances can be estimated accurately and IRNet is trained with inter-pixel relations on the attention maps, thus no extra supervision is required. In these methods, object relation is studied and proved to be effective for object detection or instance segmentation. Nevertheless, it is not clear which information has been learned in these relation modules.
we first introduce semantics perceiving subnetwork to cap-
distinguishable objects mainly exist in the case of object
kernels to learn the feature transferring.
subnetworks and mask learning subnetworks. For each sub-
technique extractor. After the feature extractor, features in each
are copied and concatenated with the coordinate to learn new instance descriptors. Note that \(\oplus\) means concatenation operation, and Coord means CoordConv.

3. The Proposed Approach

To address the aforementioned two difficulties (i.e., per-
ing indistinguishable objects and modeling the rela-
tionship between instances), we propose a novel object mining
framework for instance segmentation as depicted in Fig. 2,
denoted as PEP (Perceiving, Excavating, and Purifying). In
this framework, we first extract common features for fur-
ther learning. Then, the semantics perceiving subnetwork
is introduced to obtain original instance descriptors. The
object excavating subnetwork is also proposed to mine in-
distinguishable objects as mined instance descriptors. Next,
the instance purifying mechanism is used to model the re-
lationship between all instances. Finally, all the obtained
instance descriptors are fed into mask learning. Details of
the proposed approach are described as follows.

3.1. Feature Extractor

Inspired by recent methods for panoptic segmentation
and instance segmentation [23, 34, 36], we adopt the pyra-
mid convolutional network to extract the common features.
As shown in Fig. 2, the proposed method PEP takes feature
pyramid network [13, 24] as the feature extractor, which is
modified by removing the last two layers (i.e., classification
and global average pooling layers) for the pixel-level pre-
diction task. Feature extractor is composed of five stages
for feature encoding, named as \(C_s(\pi_s)\) with parameters \(\pi_s\),
where \(s(s = 1, 2, \ldots, 5)\) represent the \(s\)th stage in the fea-
ture extractor. After the feature extractor, features in each
stage are fed into two subnetworks: semantics perceiving
subnetworks and mask learning subnetworks. For each sub-

3.2. Semantics Perceiving

In fact, it is easy to perceive independent objects. In-
distinguishable objects mainly exist in the case of object
gathering. Therefore, in order to recognize all instances,
we first introduce semantics perceiving subnetwork to cap-
ture all pixels that may belong to an instance. Specifically,
we carry out semantic segmentation to classify each pixel
and then choose pixels with high category confidence as the
original instance descriptors with semantics-level classifi-
cation and locations.

For semantics perceiving subnetwork, we first conduct
the common semantic segmentation as semantics perceiving
branch \(P_s(\pi_s)\) with several convolution layers to ob-
tain the feature map with size \(C_{P_s} \times H_s \times W_s\), where
\(C_{P_s}, H_s, W_s\) means number of channel, height and width
of the feature. In this feature, each pixel \((c, h, w) \in \mathbb{R}^{C_{P_s} \times H_s \times W_s}\) may represent one object, where \(h\) and \(w\) are locations of each object in the feature, and \(c\) repre-
sents the confidence that the object belongs to each class,
and the maximum confidence determines the classification
of each object. To learn the locations and classification of
objects, we expect the output of the semantics perceiving
branch to approximate the ground-truth mask (represented
as \(G_{P_s}\), which is a one-hot encoding map in the channel
dimension and is generated from the ground-truth mask of
instance semantics) of classes and locations for each object
by minimizing the loss:

\[
\mathcal{L}_P = \sum_{s=1}^{5} CE(P_s, G_{P_s}), \tag{1}
\]

where \(CE(\cdot, \cdot)\) means the cross-entropy loss function with
the following formulation:

\[
CE(P, G) = -\sum_{i} \sum_{j} G_{i,j}\log P_{i,j}, \tag{2}
\]

where \(P_{i,j}\) represents the predicted probabilities that the \(i\)th
\((i \in \mathbb{R}^{H \times W})\) pixel belongs to \(j\)th \((0 \leq j \leq C, 0\) represents
background) class, \(G_{i,j}\) is the ground truth.

To represent instances better, we adopt descriptors to char-
acterize instances. Toward this end, we introduce in-
stance descriptor extracting branch \(D_s(\pi_s)\) with several
convolution layers. It can learn the instance descriptors rep-
resented by the output feature map with size \(C_{D_s} \times H_s \times W_s\),
in which these symbols have similar meanings as ones in
the object semantics perceiving branch. In the descriptor
branch, each object is represented as an instance descrip-
tor \(I\) with size \(C_{D_s} \times 1 \times 1\). In this manner, each object
is represented by a \(C_{D_s}\)-dimensional vector, which simulta-
neously ensures the consistency of features and reduces the
difficulty of instance representation during learning. By the
way, this branch does not directly learn an objective func-
tion but is indirectly supervised.

For the proposed semantics and descriptors, object repre-
sentation is redundant because each non-background pixel
can be represented as one object, which will bring extra
computation in the next procedures. To reduce the redun-
original instance descriptors, where we get a set of instance descriptors. After that, only instance descriptors with high category confidence are retained while others are discarded. Then, close instances are merged and different instances are distinguished to get the final instance descriptors.

3.3. Object Excavating

After semantics perceiving, independent objects can be perceived, and original instance descriptors with classifications and locations are obtained. In fact, original instances can be regarded as easily identifiable objects. Due to object gathering, the rest of the indistinguishable objects mainly exist around the easily identifiable objects. In order to perceive indistinguishable objects, we propose the object excavating subnetwork to mine indistinguishable objects around the original instances.

For mining indistinguishable objects, we introduce the object excavating subnetwork. In the subnetwork, each instance descriptor in \( \mathcal{D} = \{ \mathcal{I}_{\text{ind}} \}_{\text{ind}=1}^{N_{\text{ori}}} \) called original instance descriptors, where \( \mathcal{I}_{\text{ind}} \) means the instance descriptor with index \( \text{ind} \), and \( N_{\text{ori}} \) means the number of selected instance descriptors.

3.4. Instance Purifying

For the sake of improving the mutual promotion ability between different instances, we introduce the instance purifying mechanism to directly model the relationship between instances as displayed in Fig. 4. In this mechanism, we construct the relationship between each instance and other instances, in which each instance is represented by the instance descriptor \( \mathcal{I}_{\text{ind}} \) in \( \mathcal{D} \).

To model the relationship, we propose an instance purifying subnetwork to maintain intra-instance similarity and inter-instance discrimination. As shown in Fig. 4, we put all the instance descriptors from the semantic perceiving subnetwork and object excavating subnetwork into the instance purifying graph, in which each node is regarded as an instance descriptor, and the weight of the connection edge between nodes represents the feature distance between two instance descriptors. For the purifying graph, its adjacency matrix \( M \) can represent the relationship between instance descriptors. This matrix has the size of \( N_{\text{all}} \times N_{\text{all}} \). In the matrix, the rows and columns represent the indexes of instances, and each value in the matrix represents the similarities between instances of the corresponding row and column. We drive the matrix \( M \) to approach its ground truth \( G_M \) by minimizing the loss function

\[
\mathcal{L}_M = CE(M, G_M).
\]

As a result, through the object excavating subnetwork, we mine objects around each original instance to obtain imperceptible objects with the classification and descriptors. Supposing the number of newly mined objects is \( N_{\text{mined}} \), we add these new instance descriptors to the original descriptor set and obtain \( \mathcal{D} = \{ \mathcal{I}_{\text{ind}} \}_{\text{ind}=1}^{N_{\text{all}}} \), where \( N_{\text{all}} = N_{\text{ori}} + N_{\text{mined}} \). By the way, we denote the whole object excavating subnetwork as \( \mathcal{M}_s(\pi, M_s) \).

Figure 4. Structure of the instance purifying mechanism. Each instance descriptor learns the similarity with other instance descriptors in feature space, ensuring that the similar instances are close and different instances are far away. Then, close instances are merged and different instances are distinguished to get the final instance descriptors.
similarity, but different objects have no similarity. In other words, the graph learning pulls the similar instances close and pushes away different instances, which can keep intra-instance similarity and inter-instance discrimination.

During inference, we predict the similarity matrix in feature space. Then, we merge the close instances and distinguish different instances to obtain the final instance descriptors. For the convenience of description, we denote this instance purifying subnetwork as $IL$.

### 3.5. Mask Learning

After obtaining the instance descriptors $I_{ind}$, we use a general feature learning branch $B_s(\pi_{B_s})$ to produce a general feature with size $C_{B_s} \times H_s \times W_s$ to represent the feature space of object mask learning, which provides shared pixel-level feature bases for all instance descriptors. In this way, all descriptors are convolved with the general features $B_s\pi_{B_s}$ to obtain the instance segmentation result as follows.

$$M_{ind} = I_{ind} \otimes B,$$

where $\otimes$ means the convolutional operation. Instance segmentation result $M_{ind}$ is expected to approximate the ground-truth masks $G_{M,ind}$ by minimizing the loss

$$\mathcal{L}_M = \sum_{ind=1}^{N_{all}} CE(M_{ind}, G_{M,ind}).$$

By taking the losses of Eqs.(1),(3), (4), (5), and (7), the overall learning objective can be formulated as follows:

$$\min_{P} \mathcal{L}_P + \alpha \mathcal{L}_E + \beta \mathcal{L}_\pi_{DE} + \gamma \mathcal{L}_M + \delta \mathcal{L}_L,$$

where $P$ is the set of $\{\pi_s, \pi_{B_s}, \pi_{DE}, \pi_{M_s}, \pi_{IL}\}_{s=1}^{3}$ and $\pi_{IL}$ for convenience of presentation.

### 4. Experiments and Results

#### 4.1. Experimental Setup

**Dataset.** To evaluate the performance of the proposed method, we carry out all experiments on COCO dataset [26]. As set in previous work [12, 24], we train our network using the union of 118k train images $train2017$, conduct ablation studies on the 5k validation images $val2017$, and report final results on $test-dev2017$ for comparison with state-of-the-art methods.

**Metrics.** We report the results with the standard COCO metrics including AP (averaged over IoU thresholds), $AP_{50}$, $AP_{75}$, and $AP_S$, $AP_M$ and $AP_L$ (AP at different scales).

**Training and Inference.** Our proposed method can be trained end-to-end. We apply the stochastic gradient descent algorithm with a weight decay of 0.0001 and momentum of 0.9 to optimize the loss in Eq. 8. In the optimization process, parameters in the feature extractor are initialized by the pre-trained ResNet-101 [13] model, whose output dimensions for the second to fifth stages are 256, 512, 1024, and 2048 respectively. COCO $train2017$ dataset is used as the training set. The shorter side of the input image is resized to 800 pixels, and the longer side is less than or equal to 1333 pixels. We train our network for 36 epochs with a mini-batch of 16 images (8 GPUs $\times$ 2 mini-batch). “Step” learning rate policy is employed for all experiments. The initial learning rate is 0.01, which is decreased by a factor of 10 at 27 and 33 epoch respectively. In addition, we empirically set the weight $\alpha = \beta = \gamma = \delta = 1$ in Eq. 8. The dimension $C_{D_s}$ of instance descriptors is 256. To match the dimension in instance descriptor, the dimension $C_{B_s}$ of the general feature $B$ is 256, too. During inference, all supervisions are removed, and the final result is obtained from the output of mask learning.

#### 4.2. Comparisons with State-of-the-art Methods

We compare our approach with about 26 state-of-the-art algorithms in instance segmentation, including two-stage methods (MNC [8], FCIS [22], Mask R-CNN [12], MaskLab+ [5], MS R-CNN [15], HTC [4], DCT-Mask R-CNN [31], BMask R-CNN [7], and BCNet+Faster R-CNN [17]), and one-stage methods (YOLACT [2], TensorMask [6], ShapeMask [19], PolarMask [39], CenterMask-W [38], CenterMask-L [21], SOLO [36], SOLOv2 [37], CondInst [34] and BCNet+FCOS [17]) on COCO $test-dev2017$ dataset.

To prove the effectiveness of our method, we report our results based on ResNet-101 with FPN, as listed in Tab. 1. Comparing the proposed method with state-of-the-art methods, we can see that our method consistently outperforms both one-stage and two-stage methods across almost all metrics. For AP, compared with the second-best methods DCT-Mask R-CNN in two-stage methods and BCNet+FCOS in one-stage methods, our method is noticeably improved by 0.8% (from 40.1% to 40.9%) and 1.3% (from 39.6% to 40.9%) respectively. Additionally, it is worth noting that compared with DCT-Mask R-CNN, our method is significantly better on $AP_S$ (+1.7%), $AP_M$ (+7.9%) respectively. As for $AP_L$ in one-stage methods, we are below the methods (BCNet+Faster R-CNN and DCT-Mask R-CNN), which conduct more computations for fine feature learning and decoding. However, our method has fewer computational overheads. Compared with BCNet+FCOS, our method gains an 8.7% higher result in $AP_L$. Overall, these observations present the effectiveness and robustness of our proposed method and validate that the object mining frame-
Two-stage methods

<table>
<thead>
<tr>
<th></th>
<th>Backbone</th>
<th>AP</th>
<th>AP(_{50})</th>
<th>AP(_{75})</th>
<th>AP(_{S})</th>
<th>AP(_{M})</th>
<th>AP(_{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNC [8]</td>
<td>ResNet-101</td>
<td>24.6</td>
<td>44.3</td>
<td>24.8</td>
<td>4.7</td>
<td>25.9</td>
<td>43.6</td>
</tr>
<tr>
<td>FCIS [22]</td>
<td>ResNet-101</td>
<td>29.2</td>
<td>49.5</td>
<td>-</td>
<td>7.1</td>
<td>31.3</td>
<td>50.0</td>
</tr>
<tr>
<td>Mask R-CNN [12]</td>
<td>ResNet-101 FPN</td>
<td>35.7</td>
<td>58.0</td>
<td>37.8</td>
<td>15.5</td>
<td>38.1</td>
<td>52.4</td>
</tr>
<tr>
<td>Mask R-CNN [12]</td>
<td>ResNeXt-101 FPN</td>
<td>37.1</td>
<td>60.0</td>
<td>39.4</td>
<td>16.9</td>
<td>39.9</td>
<td>53.5</td>
</tr>
<tr>
<td>MaskLab+ [5]</td>
<td>ResNet-101 FPN</td>
<td>35.4</td>
<td>57.4</td>
<td>37.4</td>
<td>16.9</td>
<td>38.3</td>
<td>49.2</td>
</tr>
<tr>
<td>MS R-CNN [15]</td>
<td>ResNet-101 FPN</td>
<td>38.3</td>
<td>58.8</td>
<td>41.5</td>
<td>17.8</td>
<td>40.4</td>
<td>54.4</td>
</tr>
<tr>
<td>HTC [4]</td>
<td>ResNet-101 FPN</td>
<td>39.7</td>
<td>61.8</td>
<td>43.1</td>
<td>21.0</td>
<td>42.2</td>
<td>53.5</td>
</tr>
<tr>
<td>MS R-CNN†</td>
<td>ResNet-101 FPN</td>
<td>39.6</td>
<td>60.7</td>
<td>43.1</td>
<td>18.8</td>
<td>41.5</td>
<td>56.2</td>
</tr>
<tr>
<td>BMask R-CNN [7]</td>
<td>ResNet-101 FPN</td>
<td>37.7</td>
<td>59.3</td>
<td>40.6</td>
<td>16.8</td>
<td>39.9</td>
<td>54.6</td>
</tr>
<tr>
<td>BMask R-CNN with MS</td>
<td>ResNet-101 FPN</td>
<td>38.7</td>
<td>59.1</td>
<td>41.9</td>
<td>17.4</td>
<td>40.7</td>
<td>55.5</td>
</tr>
<tr>
<td>BCNet+R-CNN [17]</td>
<td>ResNet-101 FPN</td>
<td>39.6</td>
<td>61.5</td>
<td>43.1</td>
<td>22.7</td>
<td>42.4</td>
<td>51.1</td>
</tr>
<tr>
<td>BCNet+Faster R-CNN [17]</td>
<td>ResNet-101 FPN</td>
<td>39.6</td>
<td>61.5</td>
<td>43.1</td>
<td>22.7</td>
<td>42.4</td>
<td>51.1</td>
</tr>
</tbody>
</table>

One-stage methods

<table>
<thead>
<tr>
<th></th>
<th>Backbone</th>
<th>AP</th>
<th>AP(_{50})</th>
<th>AP(_{75})</th>
<th>AP(_{S})</th>
<th>AP(_{M})</th>
<th>AP(_{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLACT [2]</td>
<td>ResNet-101 FPN</td>
<td>31.2</td>
<td>50.6</td>
<td>32.8</td>
<td>12.1</td>
<td>33.3</td>
<td>47.1</td>
</tr>
<tr>
<td>ShapeMask [19]</td>
<td>ResNet-101 FPN</td>
<td>37.4</td>
<td>58.1</td>
<td>40.0</td>
<td>16.1</td>
<td>40.1</td>
<td>53.8</td>
</tr>
<tr>
<td>PolarMask [39]</td>
<td>ResNet-101 FPN</td>
<td>30.4</td>
<td>51.9</td>
<td>31.0</td>
<td>13.4</td>
<td>32.4</td>
<td>42.8</td>
</tr>
<tr>
<td>CenterMask-W [38]</td>
<td>Hourglass-104</td>
<td>34.5</td>
<td>56.1</td>
<td>36.3</td>
<td>16.3</td>
<td>37.4</td>
<td>48.4</td>
</tr>
<tr>
<td>CenterMask-L [21]</td>
<td>ResNet-101 FPN</td>
<td>38.3</td>
<td>-</td>
<td>-</td>
<td>17.7</td>
<td>40.8</td>
<td>54.5</td>
</tr>
<tr>
<td>SOLO [36]</td>
<td>ResNet-101 FPN</td>
<td>37.8</td>
<td>59.5</td>
<td>40.4</td>
<td>16.4</td>
<td>40.6</td>
<td>54.2</td>
</tr>
<tr>
<td>BlendMask [3]</td>
<td>ResNet-101 FPN</td>
<td>38.4</td>
<td>60.7</td>
<td>41.3</td>
<td>18.2</td>
<td>41.5</td>
<td>53.3</td>
</tr>
<tr>
<td>MEIns [41]</td>
<td>ResNet-101 FPN</td>
<td>33.9</td>
<td>56.2</td>
<td>35.4</td>
<td>19.8</td>
<td>36.1</td>
<td>42.3</td>
</tr>
<tr>
<td>D-SOLO [36]</td>
<td>ResNet-101 FPN</td>
<td>38.4</td>
<td>59.6</td>
<td>41.1</td>
<td>16.8</td>
<td>41.5</td>
<td>54.6</td>
</tr>
<tr>
<td>SOLOv2 [37]</td>
<td>ResNet-101 FPN</td>
<td>39.7</td>
<td>60.7</td>
<td>42.9</td>
<td>17.3</td>
<td>42.9</td>
<td>57.4</td>
</tr>
<tr>
<td>CondInst [34]</td>
<td>ResNet-101 FPN</td>
<td>39.1</td>
<td>60.9</td>
<td>42.0</td>
<td>21.5</td>
<td>41.7</td>
<td>50.9</td>
</tr>
<tr>
<td>BCNet+FCOS [17]</td>
<td>ResNet-101 FPN</td>
<td>39.6</td>
<td>61.2</td>
<td>42.7</td>
<td>22.3</td>
<td>42.3</td>
<td>51.0</td>
</tr>
<tr>
<td>BoundaryFormer [20]</td>
<td>ResNet-101 FPN</td>
<td>39.4</td>
<td>60.9</td>
<td>42.6</td>
<td>22.1</td>
<td>42.0</td>
<td>51.2</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-101 FPN</td>
<td>40.9</td>
<td>61.5</td>
<td>43.3</td>
<td>18.2</td>
<td>44.4</td>
<td>59.7</td>
</tr>
</tbody>
</table>

Table 1. Comparisons with state-of-the-art methods on COCO test-dev2017 dataset. The best result is in **bold** fonts.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>AP</th>
<th>AP(_{50})</th>
<th>AP(_{75})</th>
<th>AP(_{S})</th>
<th>AP(_{M})</th>
<th>AP(_{L})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td>38.5</td>
<td>59.1</td>
<td>41.4</td>
<td>17.2</td>
<td>42.8</td>
<td>57.4</td>
</tr>
<tr>
<td>Baseline + excavating</td>
<td></td>
<td>39.8</td>
<td>60.9</td>
<td>42.1</td>
<td>17.5</td>
<td>43.3</td>
<td>58.7</td>
</tr>
<tr>
<td>Baseline + purifying</td>
<td></td>
<td>40.1</td>
<td>60.8</td>
<td>42.3</td>
<td>17.6</td>
<td>43.9</td>
<td>58.4</td>
</tr>
</tbody>
</table>

PEP | 40.6 | 61.0 | 43.2 | 17.7 | 44.3 | 59.8 |

Table 2. Performance of different settings of the proposed method on COCO val2017 dataset.

work is useful for the task of instance segmentation.

Some examples generated by our approach are shown in Fig. 5. We can see that instances can be distinguished and segmented with accurate location and precise shape by the proposed method in both easy (like large objects) and complex situations (small objects, occluded and densely distributed scenes). These visualizations indicate that the proposed method has a good ability of object excavating and instance purifying. Besides, it also shows the effectiveness and superiority of the proposed object mining method.

### 4.3. Ablation Analysis

To validate the effectiveness of different components, we carry out several experiments on COCO val2017 dataset and compare the performance variations of our framework PEP.

Firstly, to investigate the effectiveness of the proposed object excavating mechanism, we conduct ablation experiments and introduce two different models for comparisons. The first setting only consists of the feature extractor, semantics perceiving, and the masking learning, which is regarded as the “Baseline” model. In addition, we carry out another model (“Baseline+excavating”) by adding object excavating subnetwork to Baseline. The comparisons of above two models are listed in Tab. 2. As for the metric AP, the object excavating mechanism can improve the performance of Baseline from 38.5% to 39.8% (+1.3%).
Figure 5. Visualization examples of the proposed approach.

sistently improve the performance of Baseline model on all metrics by 0.3%-1.8%, which verifies the effectiveness of the proposed object excavating mechanism.

Secondly, to explore the effectiveness of the instance purifying mechanism, we conduct another experiment by combining the instance purifying subnetwork with Baseline as “Baseline+purifying”. From the third row of Tab. 2, we find that the performance of instance segmentation can be consistently improved (from 38.5% to 40.1% on AP), which validates that the instance purifying mechanism is effective.

Thirdly, comparing the proposed framework PEP with other models in Tab. 2, we can also find that if combining both object excavating and instance purifying mechanisms in PEP, our method can gain further performance. This shows that our proposed object excavating and instance purifying mechanisms are compatible with each other.

5. Conclusion

In this paper, we rethink the difficulties that hinder the development of instance segmentation and propose an object mining framework. In this framework, we introduce a novel object excavating mechanism to mine the objects around each instance, which ensures to fully excavate hard samples. In addition, the instance purifying mechanism is introduced to model the relationship between instances, which pulls similar instances close and pushes away different instances. Extensive experiments on benchmark datasets validate the effectiveness of the proposed approach and show that the perspective of object mining is useful for the task.
References


ceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017. 2, 3
[34] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In European conference on computer vision, 2020. 4, 6, 7
[40] Bo Zhang and Jian Zhang. A traffic surveillance system for obtaining comprehensive information of the passing vehicles based on instance segmentation. IEEE Transactions on Intelligent Transportation Systems, 2020. 1