

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,

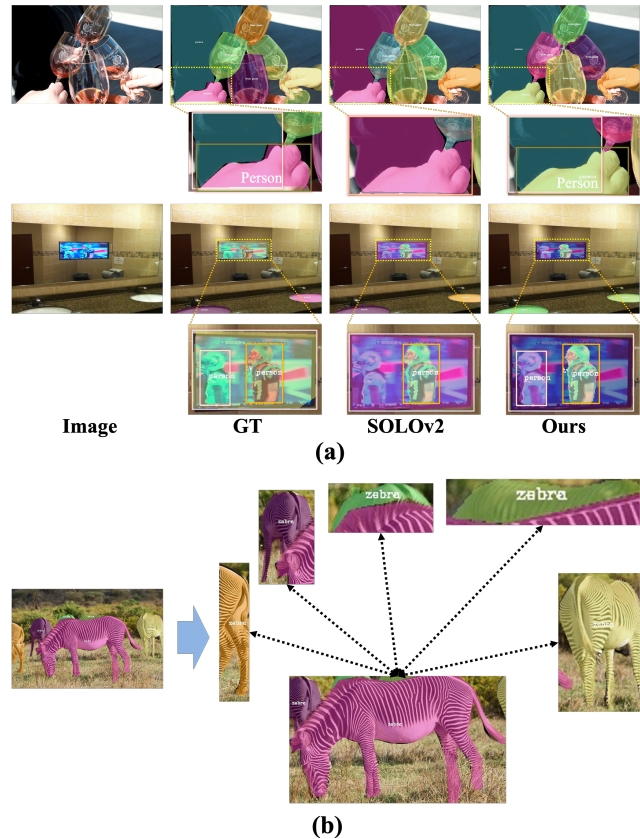


Figure 1. Challenges of instance segmentation. (a) Indistinguishable objects. It is difficult for adjacent and overlapping objects (e.g., missed persons from SOLOv2 [37]) to locate and segment. (b) Underresearched relationship between instances. The relationships (e.g., feature distance) between different instances are rarely studied while they are important for instance distinguishing.

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.*,

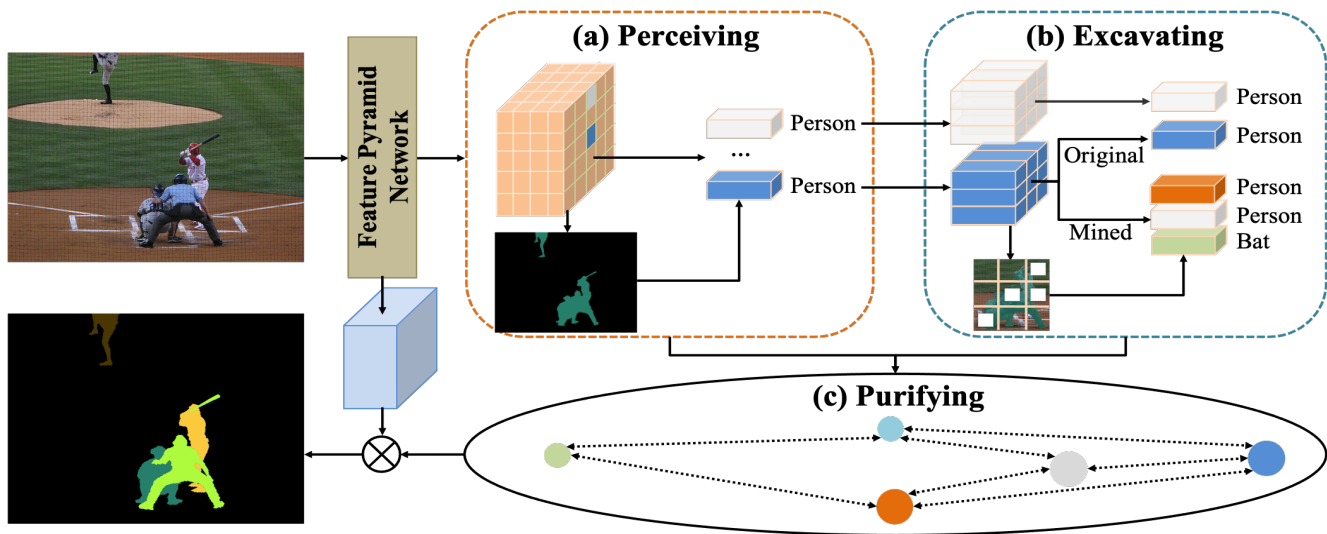


Figure 2. Framework of our approach. We first extract pyramid features by the feature pyramid network, which provides features for semantics perceiving and mask learning. Next, a semantics perceiving subnetwork based on semantic segmentation is introduced to capture pixels that may belong to an obvious instance called original instance descriptors with classifications and locations. Then, by learning instances from the region around original descriptors, the object excavating subnetwork is used to discover indistinguishable objects called mined descriptors. After excavating, all descriptors are modeled in a relationship graph according to feature distances, which can link all instances with each other and obtain final independent instances. Note that \otimes means convolutional operation.

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 mod-

eling 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.

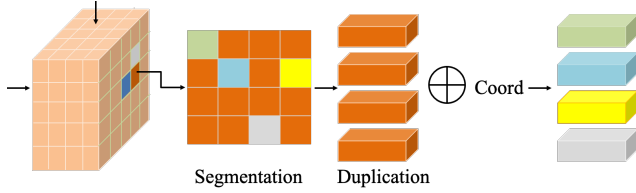


Figure 3. Structure of the object excavating subnetwork. The original instance descriptor is fed into the instance-level semantic segmentation to detect objects around original instances. In order to learn instance descriptors of the newly detected instances, instance descriptors of original objects 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.*, perceiving indistinguishable objects and modeling the relationship 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 further learning. Then, the semantics perceiving subnetwork is introduced to obtain original instance descriptors. The object excavating subnetwork is also proposed to mine indistinguishable objects as mined instance descriptors. Next, the instance purifying mechanism is used to model the relationship 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 pyramid 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 prediction task. Feature extractor is composed of five stages for feature encoding, named as $\mathcal{C}_s(\pi_s)$ with parameters π_s , where $s(s = 1, 2, \dots, 5)$ represent the s th stage in the feature extractor. After the feature extractor, features in each stage are fed into two subnetworks: semantics perceiving subnetworks and mask learning subnetworks. For each subnetwork, there are four 3×3 convolution layers with 256 kernels to learn the feature transferring.

3.2. Semantics Perceiving

In fact, it is easy to perceive independent objects. Indistinguishable 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 classification and locations.

For semantics perceiving subnetwork, we first conduct the common semantic segmentation as semantics perceiving branch $\mathcal{P}_s(\pi_{\mathcal{P}_s})$ with several convolution layers to obtain the feature map with size $C_{\mathcal{P}_s} \times H_s \times W_s$, where $C_{\mathcal{P}_s}, H_s, W_s$ means number of channel, height and width of the feature. In this feature, each pixel $(c, h, w) \in \mathbb{R}^{C_{\mathcal{P}_s} \times H_s \times W_s}$ may represent one object, where h and w are locations of each object in the feature, and c represents 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 $G_{\mathcal{P}_s}$, 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:

$$\mathcal{L}_{\mathcal{P}} = \sum_{s=1}^5 CE(\mathcal{P}_s, G_{\mathcal{P}_s}), \quad (1)$$

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

$$CE(P, G) = - \sum_i^{H \times W} \sum_j^C G_{i,j} \log P_{i,j}, \quad (2)$$

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 characterize instances. Toward this end, we introduce instance descriptor extracting branch $\mathcal{D}_s(\pi_{\mathcal{D}_s})$ with several convolution layers. It can learn the instance descriptors represented by the output feature map with size $C_{\mathcal{D}_s} \times H_s \times W_s$, 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 descriptor \mathcal{I} with size $C_{\mathcal{D}_s} \times 1 \times 1$. In this manner, each object is represented by a $C_{\mathcal{D}_s}$ -dimensional vector, which simultaneously 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 function but is indirectly supervised.

For the proposed semantics and descriptors, object representation 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-

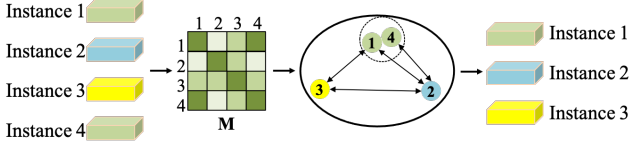


Figure 4. Structure of the instance purifying mechanism. Each instance descriptor learns the similarity with other instance descriptors in feature space, ensuring that the similar instances are close and different instances are far away. Then, close instances are merged and different instances are distinguished to get the final instance descriptors.

dancy, only instance descriptors with high category confidence are retained while others are discarded. After that, we get a set of instance descriptors $\mathbb{D} = \{\mathcal{I}_{ind}\}_{ind=1}^{N_{ori}}$ called original instance descriptors, where \mathcal{I}_{ind} means the instance descriptor with index ind , and N_{ori} means the number of selected 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 \mathbb{D} is fed into an instance-level semantic segmentation subnetwork to detect whether each pixel is the center of a new instance around origin instances as shown in Fig. 3. The output \mathcal{E}_{ind} of instance-level semantic segmentation around the original instance \mathcal{I}_{ind} is easy to learn with several convolutional layers and then supervised by the ground truth of instance-level semantic segmentation $G_{\mathcal{E}_{ind}}$ (one binary map, in which the locations are set to 1 if the pixel is the center of the instance while others representing background are set to 0). To learn the feature map \mathcal{E}_{ind} , the minimizing optimization objective is as follows:

$$\mathcal{L}_{\mathcal{E}} = \sum_{ind=1}^{N_{ori}} CE(\mathcal{E}_{ind}, G_{\mathcal{E}_{ind}}). \quad (3)$$

In this manner, indistinguishable objects around each original instance are detected as the key pixels.

Then, we learn the corresponding instance descriptor from each key pixel by following operations. Firstly, descriptors of the original instance are copied $N_{\mathcal{E}_{ind}}$ times, which is the same as the number of key pixels. Then, the

corresponding coordinate information on each copied feature descriptor is concatenated by CoordConv [27]. Then, the new instance descriptors are learned by convolutional layers. At the same time, similar classification learning is carried out for each key pixel $\mathcal{E}_{ind}^k (e = 1, \dots, N_{\mathcal{E}_{ind}})$ to gain semantic information. To achieve this, we expect the output $\mathcal{P}_{\mathcal{E}_{ind}^e}$ of classification module for the key pixel to approximate its ground truth (represented as $\mathcal{G}_{\mathcal{E}_{ind}^e}$, which is a similar map as $G_{\mathcal{P}_s}$) by minimizing the loss:

$$\mathcal{L}_{\mathcal{P}_{\mathcal{E}}} = \sum_{ind=1}^{N_{ori}} \sum_{e=1}^{N_{\mathcal{E}_{ind}}} CE(\mathcal{P}_{\mathcal{E}_{ind}^e}, \mathcal{G}_{\mathcal{E}_{ind}^e}). \quad (4)$$

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_{mined} , we add these new instance descriptors to the original descriptor set and obtain $\mathbb{D} = \{\mathcal{I}_{ind}\}_{ind=1}^{N_{all}}$ where $N_{all} = N_{ori} + N_{mined}$. By the way, we denote the whole object excavating subnetwork as $\mathcal{M}_s(\pi_{\mathcal{M}_s})$.

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 $\mathcal{I}_{ind} \in \mathbb{D}$.

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 $N_{all} \times N_{all}$. 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

$$\mathcal{L}_M = CE(M, G_M). \quad (5)$$

It is worth noting that modeling pixel-level relationships are usually difficult due to a large amount of computation. However, the number of instance descriptors is usually limited ($N_{all} \leq 30$), so the computation is efficient. By graph learning, we constrain the same object to have the highest

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 $\mathcal{I}\mathcal{L}(\pi_{\mathcal{I}\mathcal{L}})$.

3.5. Mask Learning

After obtaining the instance descriptors \mathcal{I}_{ind} , we use a general feature learning branch $\mathcal{B}_s(\pi_{\mathcal{B}_s})$ to produce a general feature with size $C_{\mathcal{B}_s} \times H_s \times W_s$ 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 \mathcal{B} to obtain the instance segmentation result as follows.

$$M_{ind} = \mathcal{I}_{ind} \otimes \mathcal{B}, \quad (6)$$

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

$$\mathcal{L}_M = \sum_{ind=1}^{N_{all}} CE(M_{ind}, G_{M_{ind}}). \quad (7)$$

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

$$\min_{\mathbb{P}} \mathcal{L}_{\mathcal{P}} + \alpha \mathcal{L}_{\mathcal{E}} + \beta \mathcal{L}_{\mathcal{P}_{\mathcal{E}}} + \gamma \mathcal{L}_M + \delta \mathcal{L}_M, \quad (8)$$

where \mathbb{P} is the set of $\{\pi_s, \pi_{\mathcal{B}_s}, \pi_{\mathcal{P}_s}, \pi_{\mathcal{D}_s}, \pi_{\mathcal{M}_s}\}_{s=1}^5$ and $\pi_{\mathcal{I}\mathcal{L}}$ 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 $C_{\mathcal{D}_s}$ of instance descriptors is 256. To match the dimension in instance descriptor, the dimension $C_{\mathcal{B}_s}$ of the general feature \mathcal{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_M (+1.7%), AP_L (+7.9%) respectively. As for AP_S 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-

	Backbone	AP	AP₅₀	AP₇₅	AP_S	AP_M	AP_L	
Two-stage methods								
	MNC [8]	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
	FCIS [22]	ResNet-101	29.2	49.5	-	7.1	31.3	50.0
	Mask R-CNN [12]	ResNet-101 FPN	35.7	58.0	37.8	15.5	38.1	52.4
	Mask R-CNN [12]	ResNeXt-101 FPN	37.1	60.0	39.4	16.9	39.9	53.5
	MaskLab+ [5]	ResNet-101 FPN	35.4	57.4	37.4	16.9	38.3	49.2
	MS R-CNN [15]	ResNet-101 FPN	38.3	58.8	41.5	17.8	40.4	54.4
	HTC [4]	ResNet-101 FPN	39.7	61.8	43.1	21.0	42.2	53.5
	MS R-CNN [†]	ResNet-101 FPN	39.6	60.7	43.1	18.8	41.5	56.2
	BMask R-CNN [7]	ResNet-101 FPN	37.7	59.3	40.6	16.8	39.9	54.6
	BMask R-CNN with MS	ResNet-101 FPN	38.7	59.1	41.9	17.4	40.7	55.5
	BCNet+Faster R-CNN [17]	ResNet-101 FPN	39.8	61.5	43.1	22.7	42.4	51.1
	DCT-Mask R-CNN [31]	ResNet-101 FPN	40.1	61.2	43.6	22.7	42.7	51.8
One-stage methods								
	YOLACT [2]	ResNet-101 FPN	31.2	50.6	32.8	12.1	33.3	47.1
	TensorMask [6]	ResNet-101 FPN	37.1	59.3	39.4	17.4	39.1	51.6
	ShapeMask [19]	ResNet-101 FPN	37.4	58.1	40.0	16.1	40.1	53.8
	PolarMask [39]	ResNet-101 FPN	30.4	51.9	31.0	13.4	32.4	42.8
	CenterMask-W [38]	Hourglass-104	34.5	56.1	36.3	16.3	37.4	48.4
	CenterMask-L [21]	ResNet-101 FPN	38.3	-	-	17.7	40.8	54.5
	SOLO [36]	ResNet-101 FPN	37.8	59.5	40.4	16.4	40.6	54.2
	BlendMask [3]	ResNet-101 FPN	38.4	60.7	41.3	18.2	41.5	53.3
	MEIns [41]	ResNet-101 FPN	33.9	56.2	35.4	19.8	36.1	42.3
	D-SOLO [36]	ResNet-101 FPN	38.4	59.6	41.1	16.8	41.5	54.6
	SOLOv2 [37]	ResNet-101 FPN	39.7	60.7	42.9	17.3	42.9	57.4
	CondInst [34]	ResNet-101 FPN	39.1	60.9	42.0	21.5	41.7	50.9
	BCNet+FCOS [17]	ResNet-101 FPN	39.6	61.2	42.7	22.3	42.3	51.0
	BoundaryFormer [20]	ResNet-101 FPN	39.4	60.9	42.6	22.1	42.0	51.2
	Ours	ResNet-101 FPN	40.9	61.5	44.3	18.2	44.4	59.7

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

	AP	AP₅₀	AP₇₅	AP_S	AP_M	AP_L
Baseline	38.5	59.1	41.4	17.2	42.8	57.4
Baseline + excavating	39.8	60.9	42.1	17.5	43.3	58.7
Baseline + purifying	40.1	60.8	42.3	17.6	43.9	58.4
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%). We also observe that the object excavating mechanism can con-



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 pu-

rifying 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

- [1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2019. 2, 3
- [2] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019. 1, 6, 7
- [3] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8573–8581, 2020. 7
- [4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiao-xiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 6, 7
- [5] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018. 6, 7
- [6] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2061–2069, 2019. 6, 7
- [7] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *European conference on computer vision*, pages 660–676. Springer, 2020. 6, 7
- [8] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3150–3158, 2016. 6, 7
- [9] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 1, 3
- [10] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *2009 IEEE Conference on computer vision and Pattern Recognition*, pages 1271–1278. IEEE, 2009. 3
- [11] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 642–651, 2019. 1, 3
- [12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 3, 6, 7
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 6
- [14] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 2, 3
- [15] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019. 1, 6, 7
- [16] SouYoung Jin, Aruni RoyChowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 307–324, 2018. 3
- [17] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4019–4028, 2021. 6, 7
- [18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1
- [19] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9207–9216, 2019. 6, 7
- [20] Justin Lazarow, Weijian Xu, and Zhuowen Tu. Instance segmentation with mask-supervised polygonal boundary transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4382–4391, 2022. 7
- [21] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 6, 7
- [22] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017. 6, 7
- [23] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 4
- [24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4, 6
- [25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Pro-*

- ceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 3
- [26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 6
- [27] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, 2018. 5
- [28] Pauline Luc, Camille Couprie, Yann Lecun, and Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599, 2018. 1
- [29] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in neural information processing system*, 2017. 1, 3
- [30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 3
- [31] Xing Shen, Jirui Yang, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Dct-mask: Discrete cosine transform mask representation for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8720–8729, 2021. 6, 7
- [32] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 2, 3
- [33] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7251–7259, 2019. 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
- [35] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019. 1, 3
- [36] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, pages 649–665. Springer, 2020. 2, 3, 4, 6, 7
- [37] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 6, 7
- [38] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. Centermask: single shot instance segmentation with point representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9313–9321, 2020. 6, 7
- [39] Enze Xie, Peize Sun, Xiaohe Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12193–12202, 2020. 1, 3, 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
- [41] Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. Mask encoding for single shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10226–10235, 2020. 7