

Zero-Shot Instance Segmentation

Ye Zheng^{1,2} Jiahong Wu³ Yongqiang Qin³ Faen Zhang³ Li Cui¹
 Institute of Computing Technology, Chinese Academy of Sciences¹
 University of Chinese Academy of Sciences² AInnovation Technology Co., Ltd³
 {zhengye@ict.ac.cn, wujiahong@ainnovation.com, qinyongqiang@ainnovation.com
 zhangfaen@ainnovation.com, lcui@ict.ac.cn}

Abstract

Deep learning has significantly improved the precision of instance segmentation with abundant labeled data. However, in many areas like medical and manufacturing, collecting sufficient data is extremely hard and labeling this data requires high professional skills. We follow this motivation and propose a new task set named zero-shot instance segmentation (ZSI). In the training phase of ZSI, the model is trained with seen data, while in the testing phase, it is used to segment all seen and unseen instances. We first formulate the ZSI task and propose a method to tackle the challenge, which consists of Zero-shot Detector, Semantic Mask Head, Background Aware RPN and Synchronized Background Strategy. We present a new benchmark for zero-shot instance segmentation based on the MS-COCO dataset. The extensive empirical results in this benchmark show that our method not only surpasses the state-of-the-art results in zero-shot object detection task but also achieves promising performance on ZSI. Our approach will serve as a solid baseline and facilitate future research in zero-shot instance segmentation. Code available at [ZSI](#).

1. Introduction

In recent years, deep learning based instance segmentation methods [1, 2, 3, 4, 5, 6] have made great progress. These supervised learning paradigm methods strongly rely on large-scale labeled data. However, for many real-world applications, e.g., the medical and manufacturing, collecting and labeling data are very time consuming and need professional annotators, which results in that we always do not have labeled data for unseen classes in these tasks. Besides, for open-set [7, 8] instance segmentation tasks, we can not label all categories, so there are many unlabeled unseen classes that need to be segmented. Without abundant labeled data, modern instance segmentation approaches are not competent to train a deep neural network to segment unseen instances. In recent years, many zero-shot learning

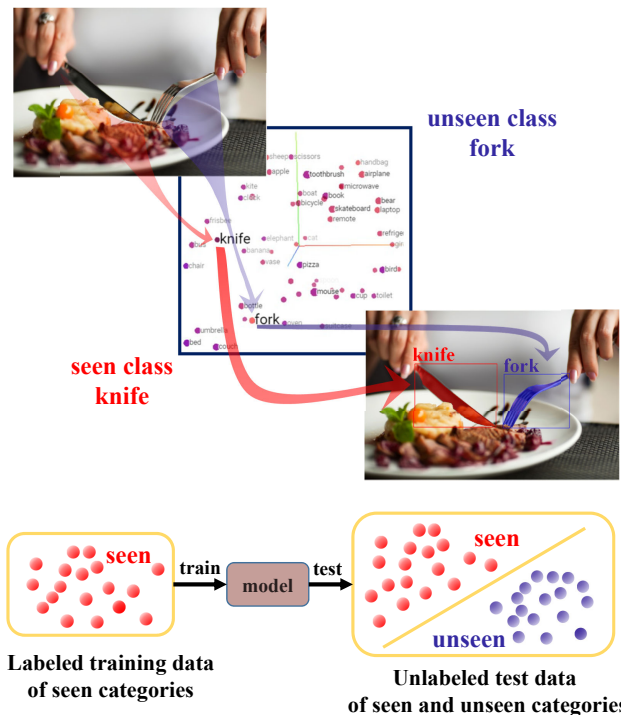


Figure 1. In zero-shot instance segmentation, we can only use the labeled data of seen categories for training but predict the instance segmentation results for both seen and unseen categories. In our method, we use the seen classes data, e.g., “knife” to establish the mapping relationship between visual and semantic concepts during training and then transfer it to segment unseen instances, e.g., “fork” in inference.

methods have been proposed. The latest achievements of zero-shot learning focus on the problem of zero-shot classification [7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Limited by the existing benchmarks [20, 21, 22], the zero-shot classification approaches focus on reasoning a single dominant unseen object in input image, which makes it unsuitable for real scenes. In the real world, several unseen objects belonging to different classes may appear at the same time. Therefore, zero-shot object detection (ZSD) [23, 24] and zero-shot semantic segmentation (ZSS) [25] have been pro-

posed. ZSD is aiming to simultaneously localize and recognize unseen objects and ZSS is used to segment unseen classes, these tasks are more practical for the real-world scenarios. However, the bounding box results obtained from ZSD and the segmentation results of the entire image from ZSS are still not fine enough when we need pixel-level segmentation results of each instance. In order to meet this demand for finer results, we introduce a new problem setting, called zero-shot instance segmentation (ZSI). As illustrated in Fig 1, the goal for ZSI is not only to detect all unseen objects, but also to further precisely segment each unseen instance.

There are two main challenges for ZSI task. (i) How to do instance segmentation for unseen classes. Without unseen classes data, we can not train a deep neural network to segment unseen instances. We introduce extra semantic knowledge contained in pre-trained word-vectors to correlate the seen and unseen classes. We use the semantic word-vector and image data of seen classes to establish the visual-semantic mapping relationship in a detection-segmentation manner and transfer it from seen to unseen classes. We propose the zero-shot detector and Semantic Mask Head (SMH) to detect and segment each unseen instance. We discuss the details in Section 3.2.1 and Section 3.2.2. (ii) How to reduce the confusion between background and unseen classes. Unlike the zero-shot classification task that only has one domain object in each image and does not need to consider the background class, ZSI needs to distinguish foreground and background. Since the unseen data are not observed during training, the model is likely to identify the unseen objects as the background, which has a great impact on the performance. We believe that the representation of background class is the key to solve this problem. We find that the current representation of the background class has two major drawbacks: (i) The existing semantic representation of background is unreasonable. Previous works [23, 24] used the word-vector for the “background” word to represent background class. However, in computer vision task, this simple word-vector learned from unlabeled text data is not enough to represent the complex background; (ii) The existing semantic representation of the background class is fixed, which makes it difficult to adapt to the changing background in different images. To obtain a reasonable and dynamic adaptive word-vector for background class, We propose Background Aware RPN (BA-RPN) and Synchronized Background Strategy (Sync-bg). We introduce the details in Section 3.2.3.

To facilitate the research of zero-shot instance segmentation, we propose the experimental protocol and benchmark for ZSI based on the challenging MS-COCO dataset. Because instance segmentation has always been regarded as a downstream task of object detection, we set up the dataset for ZSI based on the existing ZSD benchmark in [23, 26].

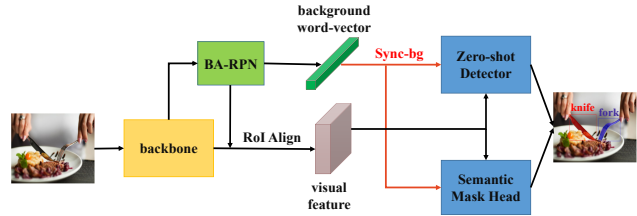


Figure 2. The whole architecture for our zero-shot instance segmentation framework. For an input image, we obtain the visual feature and background word-vector for each proposal from backbone and BA-RPN through RoI Align. Then we use Sync-bg to synchronize the word-vector for background class in Zero-shot Detector and Semantic Mask Head. We can get the instance segmentation results from these structures.

To summarize, the main contributions of this paper are as follows: (i) we introduce the problem of zero-shot instance segmentation in real-world settings; (ii) we propose a novel method which tackles ZSI problem in a background aware detection-segmentation manner, including Zero-shot Detector, Semantic Mask Network, Background Aware RPN and Synchronized Background Strategy. (iii) we put forward a new experimental benchmark for ZSI to evaluate the performance of the model; (iv) we provide extensive experimental and ablation studies to highlight the benefits of the proposed methods, and the results show that our method surpasses the state-of-the-art ZSD works and achieve promising performance on ZSI.

2. Related Works

2.1. General Instance Segmentation

Instance segmentation is a classical problem in the field of computer vision. Deep learning based instance segmentation methods have made great progress in the past several years, *e.g.*, Mask R-CNN [1], FCIS [2], YOLCAT [3], HTC [4], PolarMask [5] and SOLO [6]. Some of them solve this problem in a detection-segmentation manner, *e.g.*, as a typical work, Mask R-CNN adds a mask branch to Faster R-CNN which detects all objects first and then segments all pixels in each object. The above methods work in an intensive supervision manner and they are difficult to extend to novel categories without training samples.

2.2. Zero-shot Learning

Zero-shot learning is a classical problem in the research community of machine learning, which aims at using seen classes data to train the network and reason about unseen classes. For the past few years, several achievements have been proposed for zero-shot learning [7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Most of them have employed the transfer learning method that transfers the knowledge learned from the seen classes to the unseen classes through various intermediate representations, *e.g.*,

the semantic word-vectors learned with unsupervised fashion from unannotated text data and the attributes of manual design. However, these methods focus on zero-shot classification task which only recognizes one domain object of the input image, and it is far from the real world.

In recent years, some work of ZSD have been reported. Rahman *et al.* [24] introduce the max-margin loss into Faster R-CNN to distinguish different classes by using semantic information. Bansal *et al.* [23] use the linear projection to map the proposals from R-CNN to a word-vector and then classify them. They also develop an EM-like method to learn the word-vector for background class in an iteration manner. Zhu *et al.* [27] develop the ZS-Yolo based on the one-step Yolo detector. Rahman *et al.* [26] propose polarity loss to increase the distance between classes and adopt RetinaNet [28] as base detector for ZSD. Li *et al.* [29] use textual descriptions as the extra information and develop an attention mechanism to address ZSD problem. Zhao *et al.* [30] use Generative Adversarial Networks to synthesize semantic representations for unseen objects to help detect unseen objects. Zhu *et al.* [31] propose DELO that synthesizes visual features for unseen objects from semantic information and incorporate the detection for seen and unseen objects. Zheng *et al.* [32] boost the ZSD performance with a cascade structure to progressively refine the visual-semantic mapping relationship and propose BLRPN to learn the word-vector for background class. In terms of the background representation, SB [23] and PL [26] use the fixed word-vector for “background” word, DSES [23] and BLC [32] try to learn a more reasonable representation. However, the training methods for DSES [23] and BLC [32] are multi-step processes. The background word-vector learned from these non-end-to-end training processes is a local optimum result and it is used as a fixed representation. In our method, we learn the background word-vector in a full end-to-end jointly training manner, which can get an optimal result. In addition, benefiting from our Synchronized Background Strategy, the background word-vector learned from our method is dynamically adaptive compared to the previous fixed form in the other methods, which can greatly improve the recall rate for unseen instances. We note that Bucher *et al.* [25] propose the ZSS task that performs semantic segmentation of the entire image, which is different from our method aiming to do instance segmentation for each individual unseen instance.

3. Methodology

3.1. Problem Definition

The setting of zero-shot instance segmentation problem in this paper is described below. Suppose we are given images and word-vectors from two non-overlapping sets of classes: seen classes C_s and unseen classes C_u . The

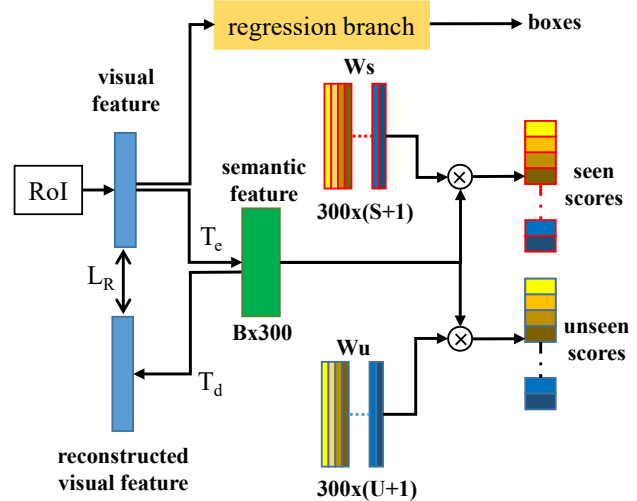


Figure 3. The details for zero-shot detector. It is trained in an encoder-decoder manner and we only use the encoder T_e in testing process. W_s is the word-vectors of all seen classes and background class. W_u is the word-vectors of all unseen classes and background class. S is the number of seen classes and U is the number of unseen classes. Each class has a 300-dimensional word-vector. B is batch size.

training set D_{train} is constructed from C_s which consists of images x_s and seen classes word-vectors w_s . The testing set D_{test} is build from C_s and C_u , which consists of images x and word-vectors w , where the seen and unseen instances may appear at the same image. In training, we use $\theta = \arg \max_{\theta} \sum_{i=1}^{D_{train}} \log p(y_{si} \in C_s | x_{si}, w_s, \theta)$ to train the network, where we use the D_{train} which only contains the instance for seen classes to optimize the parameters θ of the network. For inference, the target is changed to $\arg \max_{\theta} \sum_{i=1}^{D_{test}} \log p(y_{si} \in C_s, y_{ui} \in C_u | x_i, w, \theta)$, which means using the trained network θ to get the precisely instance segmentation results for both the seen and unseen classes. In summary, we learn the θ from seen instance and use it to inference unseen instances.

3.2. Zero-Shot Instance Segmentation

We propose an end-to-end network that adopts the semantic word-vector to detect and segment unseen instances. The idea of visual-semantic mapping relationship is embodied in the entire network, including the BA-RPN, Zero-shot Detector and SMH. Fig 2 shows the whole architecture of our network. We build our zero-shot detector based on Faster R-CNN with the visual-semantic alignment. Then, we introduce our SMH into zero-shot detector to enable the instance segmentation for unseen classes by learning visual-semantic relationship with an encoder-decoder structure. In addition, we develop BA-RPN and Sync-bg to learn a dynamically adaptive word-vector for background class.

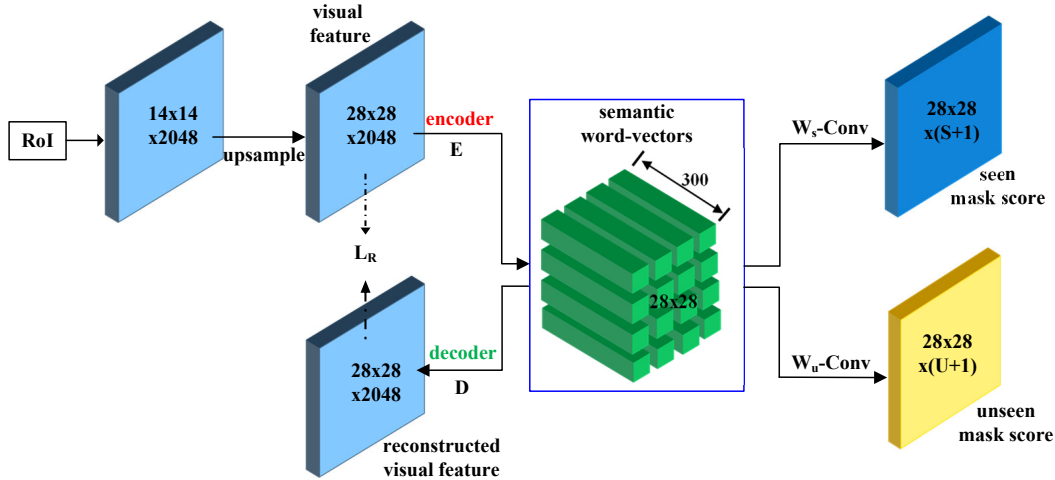


Figure 4. Our Semantic Mask Head is an encoder-decoder structure. In training, we use the encoder E to encode the visual feature into the semantic word-vectors. Then we adopt the decoder D to decode the semantic word-vectors back to reconstructed visual feature and use the loss function \mathcal{L}_R to minimize the difference between the two visual features. D is removed in inference. W_s -Conv and W_u -Conv are both fixed convolutional layers and we use them to perform pixel-by-pixel convolution on the semantic word-vectors to get the seen and unseen classes instance segmentation results.

3.2.1 Zero-Shot Detector

The main idea for our Zero-Shot Detector is learning the relationship between visual and semantic concepts from seen classes data and transferring it to detect unseen objects. To this end, we replace the classification branch in Faster R-CNN with a new semantic-classification branch. Fig 3 indicates its details. The semantic-classification branch is an encoder-decoder structure. In that, we use T_e to encode the visual feature for the input RoI into the semantic feature and use T_d to decode the semantic feature back into visual feature during training. \mathcal{L}_R is the reconstruction loss function to reduce the reconstruction error for visual feature and reconstructed visual feature, which is formulated in Eq 1. The decoder module and reconstruction loss can push the network to learn a more discriminative visual-semantic alignment. In inference, T_d is removed and we can get the scores for seen and unseen classes by performing the matrix multiplication of the matrices semantic feature and W_s , semantic feature and W_u , respectively.

3.2.2 Semantic Mask Head

In order to do instance segmentation for unseen instance, we focus on how to segment them using the visual-semantic mapping relationship. We propose Semantic Mask Head to learn this relationship and transform it from seen classes to segmenting the unseen instances. The details for our Semantic Mask Head are illustrated in Fig 4. The overall architecture is an encoder-decoder structure. The encoder module E is a single 1×1 convolutional layer structure, which encodes the visual features into semantic space, and then we can get the segmentation result from these seman-

tic word-vectors. Considering that this single forward encoder structure is not enough to learn a tight visual-semantic alignment, we develop a decoder structure D with a reconstruction loss function \mathcal{L}_R to further improve the quality of the mapping relationship between visual features and semantic word-vectors. Our decoder module decodes the semantic word-vectors back to visual features and we use \mathcal{L}_R to minimize the difference between the reconstructed visual feature and the original visual feature. For this purpose, the decoder module has a symmetrical structure relative to the encoder module, which also includes a single 1×1 convolutional layer, in which the number of input and output channels are opposite to E . We use a 300-dimensional word-vector as the semantic representation for each class, so E is responsible for converting the input visual feature into the semantic feature with a channel dimension of 300. In this $300 \times 28 \times 28$ semantic feature tensor, each channel represents a dimension of the word-vector and each 300×1 element is a word-vector. To get the classification score for each element, we need to calculate the similarity between the word-vector of each element and the word-vectors of all seen and unseen classes, so as to find the closest category. To this propose, we add a classification module after the encoder. This module includes two branches, one for seen classes and the other one for unseen classes. In it, W_s -Conv indicates a fixed 1×1 convolutional layer and we adopt the word-vectors of all seen classes and background class W_s as the weight of it. W_u -Conv is also a fixed 1×1 convolutional layer with the weight of W_u , which indicates the word-vectors of all unseen classes and background class. The W_s and W_u are the same in our zero-shot detector. We merge the results of seen and unseen classes through scores.

We use mse loss for the reconstruction loss \mathcal{L}_R in zero-shot detector and SMH. As shown in Equation 1, we measure the mean square error (squared ℓ_2 norm) between each element in the original visual feature O and the reconstructed visual feature R .

$$L_R = \sum_{i=1}^F (O_i - R_i)^2 \quad (1)$$

3.2.3 BA-RPN and Synchronized Background

In our zero-shot detector, word-vectors W_s and W_u for seen and unseen classes are used as the weights value for the fixed FC layers to classify the objects into $n + 1$ categories. Where n is the number of seen or unseen classes and the 1 represents the background class. In our Semantic Mask Head, W_s and W_u are used again as the weights value of W_s -Conv and W_u -Conv to classify the pixels into $n + 1$ categories, which means that our Semantic Mask Head also classifies each pixel to background class or other target classes. From the above discussions, we can learn that the word-vector of background class in W_s and W_u directly affect the classification result of the background class. However, the existing word-vector for background class is learned from large-scale text data without using the visual information, and can not effectively represents complex background class. To this end, we propose the Background Aware Region Proposal Network (BA-RPN), which introduces the visual-semantic learning process into the original RPN to learn a more reasonable word-vector for background class from images. The architecture of BA-RPN is indicated in Fig 5, which uses an FC layer T to transform the input visual features into semantic features. We use a 300×2 dimensional FC layer W_{bf} to get the background-foreground binary classification score from the input visual feature. The weight for W_{bf} is a 300×2 vectors, which represents the two word-vectors for foreground and background to learn. W_{bf} will be optimized during training so that we can learn a new word-vector v_b for background class.

Now we have a new word-vector for background class which can be used to replace the original one in our zero-shot detector and SMH. However, we believe this is still not the best way to solve the problem of background representation. The background class has different forms in different images, while the background word-vector learned from BA-RPN is still a fixed one. The training process for BA-RPN and the whole ZSI framework are divided, which leads to the background learning limited to BA-RPN, and the benefits of zero-shot detector and SMH are not well utilized. To fix these issues, we propose the Synchronized Background Strategy and use this synchronous operation in training and inference. Algorithm 1 indicates the details for our Sync-bg in training process. In each training step, we first forward

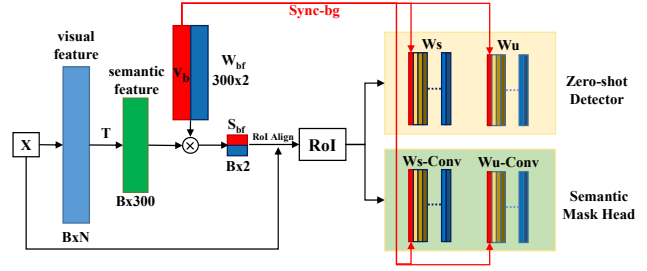


Figure 5. This is the details for BA-RPN and Sync-bg. In BA-RPN, we use T to transform the $B \times N$ dimensional visual feature to the $B \times 300$ dimensional semantic feature, where B is batch size and N is the dimension of the visual feature. Then we use W_{bf} which contain the background class word-vector v_b to get the foreground-background classification score S_{bf} . We use the Sync-bg to synchronize the v_b in the W_s and W_u for Zero-shot Detector, the W_s -Conv and W_u -Conv for Semantic Mask Head.

Algorithm 1 Synchronized Background in Training.

Input: The visual features from backbone network x ;

- 1: **for all** training iterative steps **do**
 - 2: Forward x into BA-RPN to get background word-vector v_b and features x_p for all proposals;
 - 3: Update the W_s in zero-shot detector with v_b ;
 - 4: Update the W_s -Conv in SMH with v_b ;
 - 5: Forward x_p through zero-shot detector and SMH;
 - 6: Calculate loss and back propagate gradient to update v_b in BA-RPN.;
 - 7: **end for**
-

the visual features from backbone into BA-RPN and get the word-vector v_b for background class and the visual features for each proposal. Then we update the W_s in zero-shot detector and the W_s -Conv in SMH by replacing the word-vector for background with v_b . After we forward the visual features into zero-shot detector and SMH, we calculate the loss and propagate the gradient back to update all trainable parameters, including the v_b in BA-RPN. These end-to-end jointly training process can learn a more reasonable v_b . In inference, for an input image, BA-RPN can adaptively output the corresponding background word-vector to it and the Synchronized Background Strategy synchronizes this adaptive background word-vector to update the W_s , W_u in zero-shot detector and W_s -Conv, W_u -Conv in SMH. Using this dynamic adaptive background word-vector in our zero-shot detector and SMH significantly improves the performance.

4. Loss Function.

As shown in Eq 2, the whole loss function \mathcal{L}_{ZSI} for our network has three components: the loss of BA-RPN \mathcal{L}_{BA} , the loss of zero-shot detector \mathcal{L}_{ZSD} and the loss for SMH \mathcal{L}_{SMH} . \mathcal{L}_{BA} includes foreground-background classification cross-entropy loss and the smooth ℓ_1 regression

Table 1. Effects of each component in our work. Results are reported on 48/17 split and 65/15 split of MS-COCO, respectively.

	ZSD	SMH		Det Decoder	BA-RPN & Sync-bg	ZSI			ZSD	
		Encoder	Decoder			Recall@100			mAP	Recall@100
						0.4	0.5	0.6	0.5	0.5
48/17	✓					-	-	-	-	47.2
	✓	✓				43.8	38.5	32.7	7.5	48.1
	✓	✓	✓			46.1	41.2	35.5	8.4	48.6
	✓	✓	✓	✓		46.8	41.8	35.9	8.6	49.3
	✓	✓	✓	✓	✓	50.3	44.9	38.7	9.0	53.9
65/15	✓					-	-	-	-	52.9
	✓	✓				48.9	42.6	35.5	9.1	53.4
	✓	✓	✓			51.7	45.8	38.7	10.1	54.1
	✓	✓	✓	✓		52.4	47.0	40.5	10.3	55.0
	✓	✓	✓	✓	✓	55.8	50.0	42.9	10.5	58.9

loss. \mathcal{L}_{ZSD} is consist of the cross-entropy classification loss, the smooth ℓ_1 regression loss and the reconstruction loss. \mathcal{L}_{SMH} includes a per-pixel binary classification loss and the reconstruction loss. CE is the cross-entropy loss and BCE is binary cross-entropy loss. λ_{ZSD} and λ_{SMH} are hyperparameters.

$$\begin{aligned}
 \mathcal{L}_{ZSI} &= \mathcal{L}_{BA} + \mathcal{L}_{ZSD} + \mathcal{L}_{SMH} \\
 \mathcal{L}_{BA} &= \ell_1(\mathbf{r}, \hat{\mathbf{r}}) + CE(c, \hat{c}) \\
 \mathcal{L}_{ZSD} &= \ell_1(\mathbf{r}, \hat{\mathbf{r}}) + CE(c, \hat{c}) + \lambda_{ZSD} \mathcal{L}_R(O, R) \\
 \mathcal{L}_{SMH} &= BCE(c, \hat{c}) + \lambda_{SMH} \mathcal{L}_R(O, R)
 \end{aligned} \tag{2}$$

5. Experiments

5.1. Datasets

Considering that instance segmentation task is always regarded as a downstream task for object detection, we develop the datasets for ZSI by following the previous ZSD works [23, 26]. We use MS-COCO as the basic dataset because it is widely used as the common benchmark for object detection and instance segmentation. We choose the 2014 version for MS-COCO, because this version has more data in validation set than the 2017 version, so we can have more data to evaluate our method. In order to construct the benchmark, we give two division methods of seen and unseen classes: 48/17 split and 65/15 split, which means we split the MS-COCO into 48 seen classes with 17 unseen classes and 65 seen classes with 15 unseen classes. For the training set, we first select all images which contains seen classes objects from the training set of MS-COCO. Then we remove the image in this selected set if it contains any unseen object to guarantee that the unseen objects will not be observed by the network during training. For test set, we extract all images which contains unseen objects from the validation set for MS-COCO. The images in our test set may contain seen and unseen objects together. A detailed description of the datasets can found in the appendix.

5.2. Evaluation Protocol

We evaluate the performance for ZSI with two settings: the ZSI setting and the generalize zero-shot instance segmentation (GZSI) setting. When using ZSI setting, the network only needs to predict the results for unseen instances. For GZSI setting, the results for seen and unseen classes need to be predicted together. GZSI is closer to the situation in real world because the seen and unseen instances may appear at the same time. Referring to the previous works of ZSD [23, 26, 32, 30], we use Recall@100 across different IoU thresholds (0.4, 0.5, 0.6) as the main metric and the 100 means we select the top 100 score results for evaluation. In addition, we also give the results of mean average precision (IoU threshold is 0.5) as a reference.

5.3. Implement Details

We adopt word2vec [33] as our semantic word-vector and normalize it with a ℓ_2 normalization. In zero-shot detector, T_e is an FC layer with output dimension 300 and T_d is a 2048 output dimensional FC layer. W_s and W_u are two fixed FC layer which are not updated during training. In SMH, E and D are two 1×1 convolutional layers. W_s -Conv and W_u -Conv are both fixed 1×1 convolutional layers and the weight for them are the word-vectors for seen and unseen classes. The weight for reconstruction loss function in zero-shot detector and SMH are both set to 0.5.

5.4. Component-wise Analysis

We investigate the contributions of the main components for our method. “ZSD” means our Zero-shot Detector, “SMH” denotes the Semantic Mask Head, “Encoder” and “Decoder” under the “SMH” mean the encoder and decoder module in Semantic Mask Head, respectively. “Det Decoder” represents adding the decoder module into zero-shot detector and “BA-RPN&Sync-bg” indicates the BA-RPN and Synchronized Background Strategy. The results for 48/17 and 65/15 splits are shown in Table 1, respectively. Compared with the baseline, our method brings 6.4%

Table 2. Comparison of our method with the previous state-of-the-art ZSD works on two splits of COCO. Seen/Unseen refers to the split of datasets. Our method significantly surpasses all other works.

Method	Seen/Unseen	Recall@100			mAP
		0.4	0.5	0.6	0.5
		SB [23]	48/17	34.46	22.14
DSES [23]	48/17	40.23	27.19	13.63	0.54
TD [29]	48/17	45.50	34.30	18.10	-
PL [26]	48/17	-	43.59	-	10.1
Gtnet [30]	48/17	47.30	44.60	35.50	-
DELO [31]	48/17	-	33.50	-	7.6
BLC [32]	48/17	49.63	46.39	41.86	9.9
ZSI	48/17	57.4	53.9	48.3	11.4
PL [26]	65/15	-	37.72	-	12.4
BLC [32]	65/15	54.18	51.65	47.86	13.1
ZSI	65/15	61.9	58.9	54.4	13.6

Table 3. This table shows the performances in Recall@100 and mAP (IoU threshold=0.5) for our method and other state of the art over GZSD task. HM denotes the harmonic average for seen and unseen classes.

Method	Seen/Unseen	seen		unseen		HM	
		mAP	Recall	mAP	Recall	mAP	Recall
DSES [23]	48/17	-	15.02	-	15.32	-	15.17
PL [26]	48/17	35.92	38.24	4.12	26.32	7.39	31.18
BLC [32]	48/17	42.10	57.56	4.50	46.39	8.20	51.37
ZSI	48/17	46.51	70.76	4.83	53.85	8.75	61.16
PL [26]	65/15	34.07	36.38	12.40	37.16	18.18	36.76
BLC [32]	65/15	36.00	56.39	13.10	51.65	19.20	53.92
ZSI	65/15	38.68	67.11	13.60	58.93	20.13	62.76

and 7.4% improvement for ZSI, 6.7% and 7.2% for ZSD in terms of Recall@100 for 48/17 and 65/15 splits, respectively.

5.5. Comparison with Other ZSD Methods

We compare our method with the state-of-the-art zero-shot detection approaches on two split benchmarks in Table 2 over ZSD setting. We can observe that: (i) for 48/17 split, we compare our approaches with SB [23], DSES [23], TD [29], PL [26], Gtnet [30], DELO [31] and BLC [32]. Our method surpasses all of them, brings up to 36.99% and 11.08% gain in terms of Recall@100 and mAP; (ii) for 65/15 split, compared with PL [26] and BLC [32], our method still gets the best performance and brings 21.18% gain for Recall@100 and 1.2% improvement for mAP. In addition, we also compare the performance under GZSD setting [23] in Table 3, which require our network to predict seen and unseen at the same time. We can learn that our method obtains up to 46% and 26% improvement for seen and unseen classes in Recall@100 compared with DSES [23], PL [26] and BLC [32].

5.6. Generalized Zero-Shot Instance Segmentation

For the generalized zero-shot instance segmentation setting, we need to segment the instances for seen and unseen

Table 4. This table shows the performances in Recall@100 and mAP (IoU threshold=0.5) for our method over GZSI task. HM denotes the harmonic average for seen and unseen classes.

Method	Seen/Unseen	seen		unseen		HM	
		mAP	Recall	mAP	Recall	mAP	Recall
baseline	48/17	41.11	62.30	3.01	38.14	5.61	47.31
ZSI	48/17	43.04	64.48	3.65	44.90	6.73	52.94
baseline	65/15	36.43	63.59	9.06	42.63	14.52	51.04
ZSI	65/15	38.68	67.11	13.60	58.93	20.13	62.76

Table 5. Effectiveness for BA-RPN and Synchronized Background Strategy. The Recall@100 results for ZSI and ZSD are reported on 48/17 split and 65/15 split of MS-COCO, respectively.

	Method	ZSI			ZSD
		0.4	0.5	0.6	0.5
48/17	RPN	43.8	38.5	32.7	48.1
	BA-RPN	43.1	37.9	31.4	48.9
	BA-RPN&Sync-bg in mask	44.1	38.8	32.7	48.6
	BA-RPN&Sync-bg in det	46.7	40.9	35.0	51.9
	BA-RPN&Sync-bg	47.9	42.1	35.4	54.2
65/15	RPN	48.9	42.6	35.5	53.4
	BA-RPN	49.0	42.4	35.4	55.3
	BA-RPN&Sync-bg in mask	49.0	42.5	35.4	55.0
	BA-RPN&Sync-bg in det	51.9	45.3	37.9	57.6
	BA-RPN&Sync-bg	53.7	47.4	39.9	58.5

classes as the same time, which means GZSI task is more challenging than ZSI. We build a baseline that has no decoder in detector and SMH, no BARPN, and no Sync-bg. We report the GZSI results in Table 4. We can learn that our method brings up to 5.61% for mAP and 11.72% for Recall@100 improvements on the harmonic average for seen and unseen classes over GZSI setting compared with the baseline.

5.7. Ablation Studies for BA-RPN and Sync-bg

We report the experimental results about the BA-RPN and Sync-bg to discuss the effectiveness for them. We build a baseline network by only combining the zero-shot detector and the encoder module in SMH. We conducted a series of ablation experiments on this basis, see Table 5. ‘‘RPN’’ means above baseline, ‘‘BA-RPN’’ is replacing RPN with our Background Aware RPN, ‘‘BA-RPN&Sync-bg in det’’ denotes we only use the Sync-bg strategy in BA-RPN and zero-shot detector, ‘‘BA-RPN&Sync-bg in mask’’ indicates that we use the Sync-bg strategy in BA-RPN and SMH, ‘‘BA-RPN&Sync-bg’’ represents that we use Sync-bg strategy through BA-RPN, zero-shot detector and SMH. From these results, we can learn that: (i) using BA-RPN alone is even worse than the original RPN without Sync-bg strategy, it degrades ZSI performance by 0.4% to 0.6%, which verifies that using BA-RPN alone does not bring benefits; (ii) we need synchronize the background word-vector learned

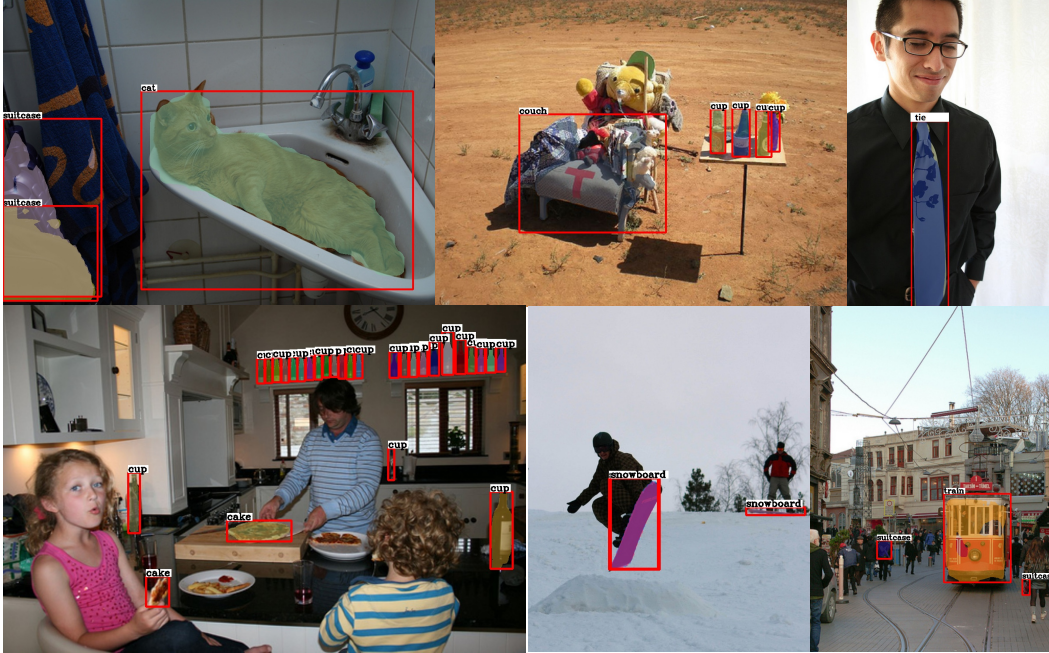


Figure 6. Examples for the results of zero-shot instance segmentation from our method. All of these instances belong to unseen classes.

Table 6. Importance of the semantic information. The semantic information needs to contain prior knowledge.

word-vectors	has knowledge	Recall@100	mAP
random baseline	No	0.2	0
one-hot	No	0.3	0
word-vec (in our method) trained from unannotated text data	Yes	58.9	13.6

from BA-RPN to the other components. If we only use Sync-bg in SMH, it does not bring much improvement for ZSI and reduces the ZSD performance. But when we use Sync-bg in detector, we can get a significantly improved for both ZSD and ZSI. We believe that the reason for this phenomenon is that in the prediction process, we first use the zero-shot detector to obtain the bounding boxes, and then perform instance segmentation on these boxes. If we only synchronize the background word-vector in semantic mask head without zero-shot detector, the inconsistency of the background word-vector in zero-shot detector and semantic mask head will decrease ZSI performance. When we maintain the consistency of the background word-vector that synchronize the background word-vector in whole framework, we can get the best performance.

5.8. The effect of the semantic information

We add a experiment to explore the effect of the semantic information. The results is shown in Table 6. We can learn that if one-hot vectors were used for the word vectors, the performance on unseen classes should be equivalent to a random baseline because the knowledge in semantic information is necessary.

5.9. Qualitative Results

To intuitively evaluating the qualitative results, we give some instance segmentation results of unseen instances in Fig 6 for our method. We find that our method can precisely detect and segment unseen classes in different situations. For example, our method detects and segments instances under densely packed scenes, e.g., “cup”. It is noteworthy that multiple instances are also segmented from a messy background like “couch”, “cup” and “snowboard”.

6. Conclusion

We introduce a novel zero-shot instance segmentation task and provide the evaluation protocol and benchmarks for it. To tackle this new task, we propose a zero-shot instance segmentation network with Zero-shot Detector, Semantic Mask Head, BA-RPN and Synchronized Background Strategy. By making use of the semantic information and learning more reasonable adaptive background representation, our method outperforms other state-of-the-art zero-shot detection approaches and provide a solid baseline for zero-shot instance segmentation task. Worthwhile and related future work can be spawned from our new task with the above technical contributions.

Acknowledgement

The paper is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61672498.

References

- [1] 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, 2
- [2] 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. 1, 2
- [3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 9157–9166, 2019. 1, 2
- [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 conference on computer vision and pattern recognition*, pages 4974–4983, 2019. 1, 2
- [5] Enze Xie, Peize Sun, Xiaoge 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, 2
- [6] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488*, 2019. 1, 2
- [7] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 1, 2
- [8] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4016–4025, 2019. 1
- [9] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5327–5336, 2016. 1, 2
- [10] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3174–3183, 2017. 1, 2
- [11] Shafin Rahman, Salman Khan, and Fatih Porikli. A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing*, 27(11):5652–5667, 2018. 1, 2
- [12] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning—the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017. 1, 2
- [13] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 4166–4174, 2015. 1, 2
- [14] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6034–6042, 2016. 1, 2
- [15] Eloi Zablocki, Patrick Bordes, Laure Soulier, Benjamin Piwowarski, and Patrick Gallinari. Context-aware zero-shot learning for object recognition. In *International Conference on Machine Learning*, pages 7292–7303. PMLR, 2019. 1, 2
- [16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 1, 2
- [17] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2188–2196, 2018. 1, 2
- [18] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289, 2018. 1, 2
- [19] Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. Meta-learning for generalized zero-shot learning. In *AAAI*, pages 6062–6069, 2020. 1, 2
- [20] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 1
- [21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1
- [22] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1
- [23] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018. 1, 2, 3, 6, 7
- [24] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *Asian Conference on Computer Vision*, pages 547–563. Springer, 2018. 1, 2, 3
- [25] Maxime Bucher, VU Tuan-Hung, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 468–479, 2019. 1, 3

- [26] Shafin Rahman, Salman Khan, and Nick Barnes. Improved visual-semantic alignment for zero-shot object detection. *34th AAAI Conference on Artificial Intelligence*, 2020. [2](#), [3](#), [6](#), [7](#)
- [27] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. [3](#)
- [28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. [3](#)
- [29] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8690–8697, 2019. [3](#), [7](#)
- [30] Shizhen Zhao, Changxin Gao, Yuanjie Shao, Lerenhan Li, Changqian Yu, Zhong Ji, and Nong Sang. Gtnet: Generative transfer network for zero-shot object detection. *arXiv preprint arXiv:2001.06812*, 2020. [3](#), [6](#), [7](#)
- [31] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don’t even look once: Synthesizing features for zero-shot detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11693–11702, 2020. [3](#), [7](#)
- [32] Ye Zheng, Ruoran Huang, Chuanqi Han, Xi Huang, and Li Cui. Background learnable cascade for zero-shot object detection. *arXiv preprint arXiv:2010.04502*, 2020. [3](#), [6](#), [7](#)
- [33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. [6](#)