Background Learnable Cascade for Zero-Shot Object Detection

Ye Zheng[1,2][0000−0003−1618−6834], Ruoran Huang[1,2][0000−0001−9014−761X], Chuanqi Han[1,2][0000−0002−3482−0475], Xi Huang[1][0000−0003−1953−5809], and Li Cui[1][0000−0002−4125−2138]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100190, China

Abstract. Zero-shot detection (ZSD) is crucial to large-scale object detection with the aim of simultaneously localizing and recognizing unseen objects. There remain several challenges for ZSD, including reducing the ambiguity between background and unseen objects as well as improving the alignment between visual and semantic concept. In this work, we propose a novel framework named Background Learnable Cascade (BLC) to improve ZSD performance. The major contributions for BLC are as follows: (i) we propose a multi-stage cascade structure named Cascade Semantic R-CNN to progressively refine the alignment between visual and semantic of ZSD; (ii) we develop the semantic information flow structure and directly add it between each stage in Cascade Semantic R-CNN to further improve the semantic feature learning; (iii) we propose the background learnable region proposal network (BLRPN) to learn an appropriate word vector for background class and use this learned vector in Cascade Semantic R-CNN, this design makes "Background Learnable" and reduces the confusion between background and unseen classes. Our extensive experiments show BLC obtains significantly performance improvements for MS-COCO over state-of-the-art methods. [3]

Keywords: Zero-shot object detection, Multi-stage structure, Background learnable, Semantic information flow

1 Introduction

Zero-shot learning (ZSL) is widely used to reason about objects belonging to unseen classes that have never been observed during training. Traditional ZSL researches focus on the classification problem of unseen objects and achieve high classification accuracy [1]. However, there still exists a large gap between ZSL settings and real-world scenarios. ZSL just focuses on recognizing unseen objects, not detecting them. For example, most of datasets used as ZSL benchmark only have one dominant object in each sample [2,3,4], while in real-world, various objects may appear in a single image without being precisely localized.

[3] Code has been made available at https://github.com/zhengye1995/BLC

To simultaneously localize and recognize unseen objects, some preliminary attempts [5,6,7,8] for zero-shot object detection (ZSD) have been reported. ZSD introduces a more practical setting to detect novel objects that are not observed during training. On this foundation, Rahman et al. [9], Li et al. [10], Zhao et al. [11] and Zhu et al. [12] make improvements to boost ZSD performance. These achievements combine the visual-semantic mapping relationship in ZSL with the deep learning based detection model in traditional object detection methods to detect unseen objects. However, these works still have their limitations: (i) can not gradually optimize the visual-semantic alignment to properly map visual features to semantic information; (ii) lack of a handy pipeline to learn a discriminative background class semantic embedding representation, while this representation is important for reducing the confusion between background and unseen classes; (iii) rely on pre-trained weights that were learned from seen or unseen datasets.

We therefore propose a novel framework named Background Learnable Cascade (BLC) for ZSD, including three components: Cascade semantic R-CNN, semantic information flow and BLRPN. BLC is motivated on the cognitive science about how humans reason objects through semantic information. Humans can use semantic information such as words to describe the characteristics of objects, and conversely, humans can also reason the categories for objects from the semantic description. Based on the past life experience, humans have established an abstract visual-semantic mapping relationship for seen objects and transfer it to recognize unseen objects. For example, humans can recognize the zebra with the language description "a horse with black and white stripes" and the visual memory of horse even if they had never seen a zebra before. Inspirited by this, BLC develops a visual-semantic alignment substructure named semantic branch to learn the visual-semantic relationship between seen objects' images and word vectors. Then transfers this alignment from seen classes to unseen classes to detect unseen objects. In order to progressively refine the visual-semantic alignment, BLC develops Cascade Semantic R-CNN by integrating the semantic branch in a multi-stage architecture based on Cascade R-CNN [13]. This combination can take advantage of the cascade structure and multi-stage refinement policy. In Cascade Semantic R-CNN, the semantic branches in later stages only benefit from better localized bounding boxes without direct semantic information connections. To remedy this problem, BLC further designs semantic information flow structure to improve the semantic information flow by directly connecting semantic branches in each cascade stage. The semantic feature in the current stage will be modulated through fully connected layers and fed to the next stage. This design promotes the circulation of semantic information between each stage and is beneficial to learn a proper visual-semantic relationship. Due to the coarse word vector for background class used in semantic branch is inability to exactly represent the complex background, BLC develops a novel framework denoted as background learnable region proposal network (BLRPN) to learn an appropriate word vector for background class. Our study shows that replacing

the coarse background word vector in semantic branch with the new one learned from BLRPN can effectively increases the recall rate for unseen classes.

Our main contributions of Background Learnable Cascade (BLC) are: (i) we develop Cascade Semantic R-CNN, which effectively integrates multi-stage structure and cascade strategy into zero-shot object detection by first integrating cascade with the semantic branch; (ii) we develop semantic information flow structure among each cascade stage to improve the semantic feature learning; (iii) we develop a background learnable region proposal network (BLRPN) to learn a more appropriate background class semantic word vector reducing the confusion between background and unseen classes; (iv) extensive experiments on two different MS-COCO splits show significant performance improvement in terms of mAP and recall.

2 Related Work

Zero-shot Recognition. In the past few years, several works have been proposed [14,15,16,17,18,19,20,21,2,22,23,24,25,1,26] for zero-shot image recognition. Most approaches of ZSL [27,28,21,29,30,31,32,33,34,35,36] have employed the relationship between seen and unseen classes to optimize recognition of unseen objects. The most classic way is to learn the alignment between the visual and semantic information by using extra source data. This alignment can classify unseen image categories by using labeled image data and semantic representations trained with unsupervised fashion from unannotated text data. In our work, we follow this methodology to detect objects for unseen classes.

Object detection. Deep learning based object detection methods have made great progress in the past several years, e.g., YOLO [37], SSD [38], RetinaNet [39], Faster R-CNN [40], R-FCN [41], MASK R-CNN [42], DCN [43], CornerNet [44], CenterNet [45] and FCOS [46]. The recent multi-stage structures have further boosted performance for object detection, e.g., Cascade R-CNN [13] and Cascade RPN [47]. The multi-stage cascade strategy progressively refine the results and we also adopt this strategy to refine visual-semantic alignment in our BLC.

Recent achievements for ZSD. In recent years, some ZSD approaches have been proposed. Rahman et al. [7] combine ConSE [22] and Faster R-CNN [40] with a max-margin loss and a meta-class clustering loss to tackle the problem of ZSD. Bansal et al. [5] employ a background-aware model to solve the confusion for background class in ZSD, and they use additional data to densely sample training classes. They also propose a generalization version of ZSD called generalized zero-shot object detection (GZSD) which aims to detect seen and unseen objects together. Demirel et al. [6] adopt the hybrid region embedding to improve performance. Zhu et al. [8] introduce ZS-YOLO, which is built on a one-step YOLOv2 [48] detector. Rahman et al. [9] propose polarity loss to cluster semantic and develop an end-to-end network based RetinaNet [39]. Li et

al. [10] address ZSD with textual descriptions by jointly learning visual units, visual-unit attention and word-level attention.

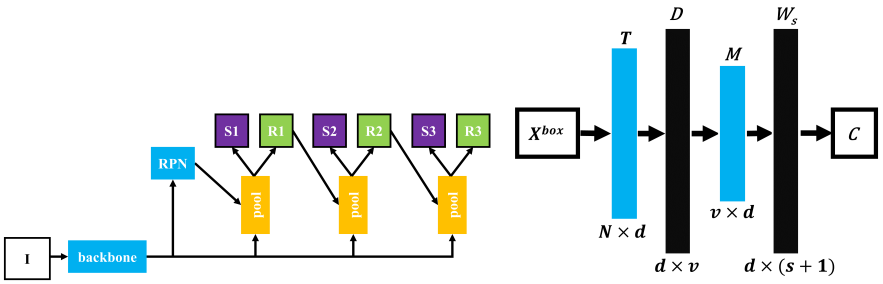
There are some key differences between our work and previous works: (i) to the choice of evaluation datasets, Rahman et al. [7] and Zhu et al. [8] use the ILSVRC-2017 detection dataset [3]. This dataset is restrictive for evaluate ZSD, in comparison with our choice — MS-COCO [49]. Because each image in ILSVRC-2017 detection dataset only has one dominant object, which exists a big gap with the real scene. We follow the choices and splits for dataset introduced by Bansal et al. [5] and Rahman et al. [9] in MS-COCO). These dataset splits are more challenging and closer to the real scene settings. (ii) for the representation of background class, most of them just use a trivial representation for background class, e.g., the semantic vectors for 'background' word [5] and the mean vectors for all seen classes [7]. These representations are not the optimal solution to address the confusion between background and unseen classes. Bansal et al. [5] propose a background-aware approach based on an iterative EM-like training procedure, but it is complex and inefficient for datasets with a small number of categories like MS-COCO. In contrast, our BLRPN, as an end-to-end framework, can learn a reasonable representation for background class through only one training process without iterations while not be affected by the sparsity of category; (iii) in the aspect of the optimization strategy, all of these previous works just refine the visual-semantic alignment once, which may not enough to optimize this alignment. In BLC, we adopt multi-stage architecture to progressively refine this alignment to improve the performance of ZSD. (iv) for the training process, most of them need fine tune their model based on additional pre-trained weights, which are learned from seen or unseen-class data, while our work, as stated above, just needs a simple and straightforward training process without any additional pre-trained weights on seen or unseen data.

3 Background Learnable Cascade

In this section, we elaborate Background Learnable Cascade (BLC). We first introduce our semantic branch about learning the alignment between the visual and semantic information. Then we introduce Cascade Semantic R-CNN which integrates our semantic branch with a multi-stage cascade structure. Since Cascade Semantic R-CNN does not use the semantic information between each stage, we develop semantic information flow structure via incorporating a direct path to reinforce the information flow among semantic branches. Moreover, in consideration of further reducing the confusion between background and unseen classes, we develop BLRPN to learn a discriminative word-embedding representation for background objects. Finally, we describe the details of training process, loss function and inference settings.

3.1 Model Architecture

Semantic Branch. We propose semantic branch to learn the alignment between the visual and semantic information. The details about our semantic



(a) Cascade Semantic R-CNN (b) Semantic branch

Fig. 1. The architecture for Cascade Semantic R-CNN. (a) is the overview architecture and (b) indicates the details for semantic branch. In figure (b), T, M are trainable FC layers and D, W_s are fixed FC layers. For an input image I, a backbone network (ResNet) is used to obtain the features. Then these features will be forwarded to the Region Proposal Network (RPN) to generate a set of object proposals. After we use a RoI pooling layer to map the proposals' features to a set of fix size objective features, we forward them through the semantic branches (purple *S1*,*S2* and *S3*) and the regression branch (green *R1*,*R2* and *R3*) in 3 cascade stages to get category scores and bounding boxes for objects.

branch denoted as S are illustrated in Fig. 1(b). The basic idea is derived from [9] which uses the relationship between the visual features and the semantic embedding as the bridge to detect unseen objects. There are four main components in semantic branch. $W_s \in \mathbb{R}^{d\times(s+1)}$ is a fixed FC layer, whose parameters are the stacked semantic word vectors of background and seen classes. More specifically, d is the dimension of word vector for each class, s denotes the number of seen classes and 1 denotes the background class. As shown in Fig. 4, each class has a corresponding word vector v_c ($1 \times d$ dimension) in W_s. For background class, we use the mean word vector $v_b = \frac{1}{s}\sum_{c=1}^{s} v_c$ in our baseline and this v_b will be improved in our BLRPN. Since the word vector quantity for W_s is limited and causes the serious sparsity of semantic representation, we add an external vocabulary $D \in \mathbb{R}^{d\times v}$ to enhance the richness of semantic information, where v is the number of words in this external vocabulary. D is also implemented by a fixed FC layer like W_s. To overcome the limitation of fixed semantic representation of W_s and D, we make an updatable representation by introducing an adjustable FC layer M to semantic branch which can be regarded as an attention mechanism in visual-semantic alignment. With this adaptive M whose dimension is $v \times d$, semantic branch can update the semantic word embedding space to learn a more flexible and reliable alignment. $T \in \mathbb{R}^{N\times d}$ is an FC layer which is used to adjust the dimension of input objective feature $\mathbf{x}^{box}$ to fit the subsequent model. In detail, it transforms $\mathbf{x}^{box}$ from N dimension to d dimension. With these above components, our semantic branch projects the input visual feature tensors to the semantic space and then gets the category score $\mathbf{c}$. The calculation process is

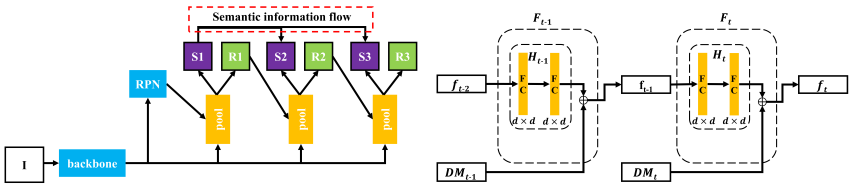


Fig. 2. The architecture for semantic information flow. (a) indicates adding semantic information flow into Cascade Semantic R-CNN and (b) shows the details of semantic information flow.

summarized as follows:

$$\begin{aligned} S &= \delta(W_s MDT), \\ \mathbf{c} &= \sigma(S(\mathbf{x}^{box})), \\ &= \sigma(\delta(W_s MDT)\mathbf{x}^{box}). \end{aligned} \tag{1}$$

Where, $\delta(\cdot)$ denotes a tanh activation function, $\sigma(\cdot)$ is the softmax activation function and $\mathbf{c}$ represents the category score.

Cascade Semantic R-CNN. In order to gradually refine the visual-semantic alignment, we integrate above semantic branch into Cascade R-CNN to develop Cascade Semantic R-CNN. We replace the classification branch of each stage for Cascade R-CNN with our semantic branch, as shown in Fig. 1. In particular, the semantic branches for each stage do not share parameter weights. This framework progressively refines predictions through the semantic branches and bounding box regression branches. The whole pipeline is summarized as follows:

$$\begin{aligned} \mathbf{x}_t^{box} = \mathcal{P}(\mathbf{x}, \mathbf{r}_{t-1}), \qquad \mathbf{r}_t = \mathcal{R}_t(\mathbf{x}_t^{box}), \\ \mathbf{c}_t = \sigma(\mathbf{S}_t(\mathbf{x}_t^{box})) = \sigma(\delta(W_s M_t DT_t)\mathbf{x}_t^{box}). \end{aligned} \tag{2}$$

Here, $\mathbf{x}$ represents the visual feature from backbone network which is based on ResNet-50 [50] and the Feature Pyramid Networks (FPN) [51]. $\mathbf{r}_{t-1}$ is the RoIs for $(t-1)$-th stage and $\mathbf{x}_t^{box}$ represents the objective feature derived from $\mathbf{x}$ and the input RoIs $\mathbf{r}_{t-1}$. $\mathcal{P}(\cdot)$ is a pooling operator and we use RoI Align [42] here. $\mathcal{R}_t$ and $\mathcal{S}_t$ indicate the bounding box regression branch and the semantic branch at the t-th stage, respectively. $\mathbf{c}_t$ represents category score predictions for t-th stage. This process will be iterated in each stage.

Semantic Information Flow. In Cascade Semantic R-CNN, the visual-semantic alignment in semantic branches of each stage is purely based on the visual objective features $\mathbf{x}_t^{box}$. This design does not have direct information flow between semantic branches for each stage, failing to make full use of the relevance of

semantic information in different stages and progressively refine semantic representing. With the aim of making up this issue, we develop a semantic information flow structure between semantic branches among each cascade stage by forwarding the modulated semantic information from previous stages to current stage, as illustrated in Fig. 2. We show the calculation process for semantic information flow as follows:

$$\begin{aligned} \mathbf{f}_1 &= DM_1 \\ \mathbf{f}_2 &= \mathcal{F}_2(\mathbf{f}_1, DM_2) \\ &\vdots \\ \mathbf{f}_t &= \mathcal{F}_t(\mathbf{f}_{t-1}, DM_t)). \end{aligned} \tag{3}$$

Where, $\mathbf{f}_t$ represents the semantic information for t-th stage derived from $\mathcal{F}_t$ which combines the semantic information of current stage and the preceding one. DM_t indicates the local semantic information for t-th stage. $\mathcal{F}$ is a function which fuses the semantic information for last stage and current stage with two steps. First, modulating the input semantic information for preceding stage $\mathbf{f}_{t-1}$ with two FC layers $\mathcal{H}_t$. Then, adding this modulated feature with the semantic information of current stage DM_t in an element-wise manner. The calculation details for $\mathcal{F}$ in t-th stage are:

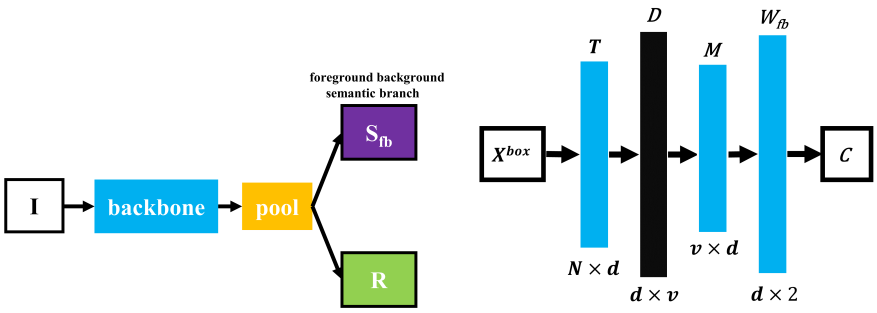
$$\mathcal{F}_t(\mathbf{f}_{t-1}, DM_t)) = \mathcal{H}_t(\mathbf{f}_{t-1}) + DM_t \tag{4}$$

After adding the semantic information flow into Cascade Semantic R-CNN, the calculation process for $\mathbf{c}_t$ in Equation 2 will be changed with replacing original DM_t with new $\mathbf{f}_t$:

$$\mathbf{c}_t = \sigma(\mathbf{S}_t(\mathbf{x}_t^{box})) = \sigma(\delta(W_s \mathbf{f}_t T_t)\mathbf{x}_t^{box}). \tag{5}$$

The semantic features will benefit from this approach and can help to learn a robust visual-semantic alignment and improve zero shot detection performance.

Background Learnable RPN (BLRPN). In Cascade Semantic R-CNN, the W_s in semantic branch adopts a coarse mean word vector v_b for background class, which may not reasonably represent the background class and further reduce the confusion between background and unseen classes. We need a new background semantic vector to replace the old one because this "replace" strategy can avoid modifying Cascade Semantic R-CNN structure and introducing extra computation. Since the background visual concept is very complex, the better idea is to learn background semantic vector from various background visual data. In order to ensure that the learned background class word vector can directly replace the original one, the learning process needs to be consistent with the process it Cascade Semantic R-CNN. Based on above analysis, we develop Background Learnable RPN to learn this new background semantic vector and use it to



(a) BLRPN (b) foreground-background semantic branch

Fig. 3. The architecture for BLRPN. (a) is the overview architecture and (b) indicates the details about foreground-background semantic branch S_{fb}. In S_{fb}, D is fixed while T, M and W_{fb} are trainable FC layers. c is the foreground background binary classification score.

replace the coarse one in W_s. In Fig. 3, we develop a foreground-background semantic branch S_{fb} and integrate it into the original RPN. S_{fb} is modified from our semantic branch for consistency, and the details are illustrated in Fig. 3(b). The only difference between S_{fb} and semantic branch S is that the W_s in S is replaced by the W_{fb} in S_{fb}. We implement W_{fb} with an FC layer without bias and make it trainable. The parameters of $W_{fb} \in \mathbb{R}^{d\times 2}$ contain two word vectors, one is v_b for background class and the other is v_f for foreground class, so v_b as the new background word vector will be updated during training. v_f is initialized with a uniform random distribution and the v_b is initialized with the mean word vectors for all seen classes, which is the same as W_s. During training, we feed the visual features derived from the backbone network to the foreground-background branch and get the foreground-background classification score. The details are:

$$\begin{aligned} S_{fb} &= \delta(W_{fb}MDT), \\ \mathbf{c} &= \sigma(S_{fb}(\mathbf{x}^{box})), \\ &= \sigma(\delta(W_{fb}MDT)\mathbf{x}^{box}). \end{aligned} \tag{6}$$

After calculating the loss, we back propagate all gradients to update trainable parameters includes W_{fb}. W_{fb} will be updated means that we can learn the target background class semantic vector v_b in the course of training BLRPN. As shown in Fig. 4, we use this new v_b to replace the old one for background class in W_s. Finally, we retrain our Cascade Semantic R-CNN model with this new W_s and effectively improve the performance for unseen objects. Overall, BLRPN learns the new v_b by establishing the alignment between visual concepts and semantic representation of background classes.

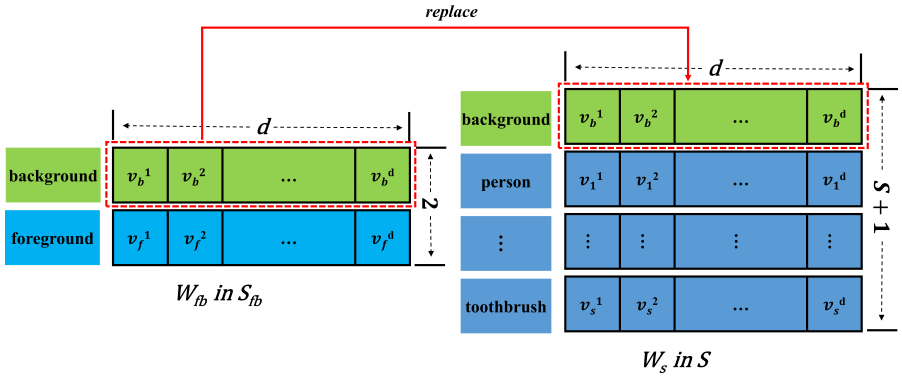


Fig. 4. W_s is the word vectors for background and other seen classes, it includes 1 background class and s seen classes, each class has a $1 \times d$ dimensional word vector. W_{fb} is the word vectors for background and foreground classes, it includes 1 background class and 1 foreground class, each class has a $1 \times d$ word vector. Here, we replace the v_b in W_s with that in W_{fb} learned from BLRPN.

3.2 Learning

Training Process. Compared with previous achievements [5,6,7,8] needing multi-step training and pre-trained weights on seen or unseen data, the training process of our model is very simple and convenient with a two step manner. First we train BLRPN to get v_b and use it to obtain a new W_s. Then we train our Cascade Semantic R-CNN equipped with semantic information flow with this new W_s. It needs to be emphasized that we only adopt the ImageNet pre-trained weights in the above training processes without any pre-trained weights of seen-class data.

Loss Function. First, we introduce the loss function of Cascade Semantic R-CNN. In each stage t for Cascade Semantic R-CNN, the box regression branch predicts the RoIs $\mathbf{r}_t$ and the semantic branch predicts category score c_t. The loss function L_{cs} is:

$$\begin{aligned} L_{cs} &= \sum_{t=1}^{3} \alpha_t (L_t^{reg} + L_t^{sem}), \\ L_t^{reg}(\mathbf{r}_t, \widehat{\mathbf{r}}_t) &= \ell_1(\mathbf{r}_t, \widehat{\mathbf{r}}_t), \\ L_t^{sem}(c_t, \widehat{c}_t) &= CE(c_t, \widehat{c}_t). \end{aligned} \tag{7}$$

Here, L_t^{sem} represents classification loss for semantic branch which adopts cross-entropy (CE) loss function. L_t^{reg} is the loss of the boxes predictions at stage t, which uses smooth $\ell 1$ loss. The coefficient α_t is the loss weight for each stage, we follow the settings in Cascade R-CNN [13] and set α_t to [1,0.5,0.25] for 3 stages.

The loss function of BLRPN denoted as L_{blrpn} is consists of the classification loss L^{fbsem} in foreground-background semantic branch and the box regression loss L^{reg} in regression branch:

$$\begin{aligned} L_{blrpn} &= L^{reg} + L^{fbsem}, \\ L^{reg}(\mathbf{r}, \widehat{\mathbf{r}}) &= \ell_1(\mathbf{r}, \widehat{\mathbf{r}}), \\ L^{fbsem}(c, \widehat{c}) &= CE(c, \widehat{c}). \end{aligned} \tag{8}$$

Inference. We forward the input images through Cascade Semantic R-CNN to get the boxes and categories for all objects, then we apply Non-Maximum Suppression (NMS) to get the final results. In addition to the original inference process for the seen class in Equation 5, we add an extra calculation process to inference unseen objects. The extra process is as follows:

$$\mathbf{c}_{unseen} = W_u W_s^{\mathsf{T}} \sigma(\delta(W_s f T)\mathbf{x}^{box}). \tag{9}$$

Where, $W_u \in \mathbb{R}^{d\times(u+1)}$ denotes the stacking word vectors for background and unseen classes, u indicates the number of unseen classes. The other components are same as Equation 5. For an input object feature $\mathbf{x}^{box}$, we first map this visual feature to the category probability of seen classes. Then we use the transpose of W_s to transform this probability back to semantic space, finally we get unseen category score from the semantic space through W_u. For GZSD task, we simultaneously execute the above two reasoning process, so as to achieve the simultaneous reasoning of seen and unseen objects.

4 Experiments

4.1 Datasets

We perform experiments on MS-COCO dataset [49]. MS-COCO (2014) includes 82783 training images and 40504 validation images with 80 classes. We follow the datasets settings in [5] and [9] for MS-COCO. We divide the dataset with two different splits: (i) 48 seen classes an 17 unseen classes; (ii) 65 seen classes and 15 unseen classes. The seen classes are training set and unseen classes are test set. Both splits remove all images from the training set which contain any object from seen classes. Specially, the images for unseen classes in test set still have objects for seen classes in order to maintain the number of samples in the test set. Following [9], we use extra vocabulary from NUS-WIDE [52] and remove MS-COCO classes names and all tags with no word-vectors. We use a 300 dimensional word2vec [53] with a ℓ_2 normalization for MS-COCO classes and extra vocabulary.

4.2 Evaluation Protocol

We report the evaluation results on ZSD and GZSD task like previous work [5,9] over two splits for MS-COCO. We use recall and mAP as metrics, these metrics

Table 1. Comparison of the proposed BLC with the previous state-of-the-art zsd work on two splits of COCO. Seen/Unseen refers to the split of datasets. The proposed BLC can achieve 10.6 mAP and 48.87 Recall@100 for 48/17 split, 14.7 mAP and 54.68 Recall@100 for 65/15 split, significantly surpasses all other work. "ms" indicates multi-scale training and test.

Method	Seen/Unseen	Recall@100			mAP
		0.4	0.5	0.6	0.5
SB [5]	48/17	34.46	22.14	11.31	0.32
DSES [5]	48/17	40.23	27.19	13.63	0.54
TD [10]	48/17	45.50	34.30	18.10	-
PL [9]	48/17	-	43.59	-	10.10
BLC	48/17	**49.63**	**46.39**	**41.86**	9.90
BLC (ms)	48/17	**51.33**	**48.87**	**45.03**	**10.60**
PL [9]	65/15	-	37.72	-	12.40
BLC	65/15	**54.18**	**51.65**	**47.86**	**13.10**
BLC (ms)	65/15	**57.23**	**54.68**	**51.22**	**14.70**

for boxes are all evaluated across IoU thresholds in 0.4, 0.5 and 0.6. In particular, the evaluation for recall is based on Recall@K [5], which means the recall when only the top K detections are selected from an image, we set K to 100 by following the settings in [5].

4.3 Implementation Details

In all experiments, we adopt ResNet-50 [50] as the backbone network with FPN [51]. We train all models with 4 GPUs (two images per GPU) for 12 epochs with a SGD optimizer which momentum is 0.9 and weight-decay is 0.0001. The initial learning rate for the optimizer is set to 0.01, and decreased by 0.1 after 8 and 11 epochs. The long edge and short edge of images are resized to 1333 and 800 without changing the aspect ratio. We use horizontal flip during training and the multi-scale for training is set to [400, 1400]. We implement our model in PyTorch [54] and the pre-trained model is from PyTorch official model zoo.

4.4 Quantitative Results

Results in Benchmarks. We compare Background Learnable Cascade with the state-of-the-art zero-shot detection approaches on two splits of MS-COCO in Table 1. We can observe that: (i) for 48/17 split, we compare our approaches with SB [5], DSES [5], TD [10] and PL [9]. Our BLC surpasses all of them in Recall@100 and mAP, brings up to 33.72% (4×) and 10.28% (33×) gain in terms of Recall@100 and mAP; (ii) for 65/15 split, compared with PL [9], our BLC brings 16.96% gain for Recall@100 and 2.3% improvement for mAP. Moreover, in other previous works, Recall@100 drops severely as IoU threshold increasing

Table 2. Effects of each component in our work. Results are reported on 48/17 split and 65/15 split of MS-COCO, respectively.

	Cascade Semantic	Semantic Info	BLRPN	Recall@100			mAP
				0.4	0.5	0.6	0.5
48/17	✓			40.96	38.75	35.25	9.3
	✓	✓		43.84	41.73	38.11	9.5
	✓		✓	48.52	45.41	41.04	9.6
	✓	✓	✓	**49.63**	**46.39**	**41.86**	**9.9**
65/15	✓			49.75	47.28	43.87	12.4
	✓	✓		51.49	49.05	45.07	12.7
	✓		✓	53.38	51.03	47.39	12.9
	✓	✓	✓	**54.18**	**51.65**	**47.86**	**13.1**

while our BLC can still maintain a high Recall@100 indicating our approach is more robust for stringent IoU threshold.

Component-wise Analysis. We investigate the contributions of the main components for BLC. "Cascade Semantic" means the baseline Cascade Semantic R-CNN, "Semantic Flow" denotes the semantic information flow, "BLRPN" represents the new background class word vector learned from our background learnable region proposal network. The results for 48/17 and 65/15 splits are shown in Table 2, respectively.

Class-wise Performance. We report the Recall@100 on two splits of MS-COCO for each unseen classes in Table 3. Our BLC makes significant improvement on both splits: (i) for the split of 48/17, BLC substantially boosts baseline in the most of classes. For the classes which are hard to detect, BLC achieves 2.1×, 1.6×, 3.7×, 1.4×, 2.1×, 1.5× and 2.5× improvement on Recall@100 for "skateboard", "cup", "knife", "cake", "keyboard", "sink" and "scissors" classes, respectively; (ii) for the split of 65/15, BLC also obtains further improvement compared with baseline. We also note that BLC is unable to detect any true positive for the class "umbrella" and "tie", the Recall@100 rate for the class "hair drier" is also unsatisfying. The main reason is that there are fewer classes are semantically similar with these poor classes in training dataset, which makes them difficult to detect.

Generalized Zero-Shot Detection (GZSD) Results. The generalized zero-shot detection task is more realistic that both seen and unseen classes are presented during evaluation. We report the performance for GZSD in Table 4 under on both splits over MS-COCO. The score threshold is 0.2 for seen classes and 0.05 for unseen classes, respectively. The IoU threshold for mAP is 0.5. Our BLC exceeds other stat-of-the-art methods in terms of mAP and recall@100.

Table 3. Class-wise Recall@100 for 48/17 and 65/15 splits of MS-COCO with the IoU threshold is 0.5. Our BLC achieves significant improvement in most of unseen classes compared with Cascade Semantic R-CNN baseline.

48/17 split of MS-COCO

Method	Overall	bus	dog	cow	elephant	umbrella	tie	skateboard	cup	knife	cake	couch	keyboard	sink	scissors	airplane	cat	snowboard
baseline	38.73	72.9	**94.6**	67.3	68.1	0.0	0.0	19.9	24.0	12.4	24.0	63.7	11.6	9.2	8.3	48.3	70.7	**63.4**
BLC	**46.39**	**77.4**	88.4	**71.9**	**77.2**	0.0	0.0	**41.7**	**38.0**	**45.6**	**34.3**	**65.2**	**23.8**	**14.1**	**20.8**	48.3	**79.9**	61.8

65/15 split of MS-COCO

Method	Overall	airplane	train	parking meter	cat	bear	suitcase	frisbee	snowboard	fork	sandwich	hot dog	toilet	mouse	toaster	hair drier
baseline	47.28	53.9	70.6	5.9	90.2	85.1	40.7	25.9	59.9	33.7	76.9	64.4	33.2	3.3	**64.1**	1.4
BLC	**51.28**	**58.7**	**72.0**	**10.2**	**96.1**	**91.6**	**46.9**	**44.1**	**65.4**	**37.9**	**82.5**	**73.6**	**43.8**	**7.9**	35.9	**2.7**

Table 4. This table shows Recall@100 and mAP (IoU threshold=0.5) for our BLC and other stat 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 [5]	48/17	-	15.02	-	15.32	-	15.17
PL [9]	48/17	35.92	38.24	4.12	26.32	7.39	31.18
BLC	48/17	**42.10**	**57.56**	**4.50**	**46.39**	**8.20**	**51.37**
PL [9]	65/15	34.07	36.38	12.40	37.16	18.18	36.76
BLC	65/15	**36.00**	**56.39**	**13.10**	**51.65**	**19.20**	**53.92**

4.5 Qualitative Results

For intuitively evaluating the qualitative results, we give some detection results in Fig. 5 for BLC on two splits of MS-COCO. We find that BLC can precisely detect unseen classes under different situations. For example, BLC detects objects under densely packed scenes, e.g., “airplanes”, “elephants” and “hot dogs”, as well as successfully captures small objects like the tiny “airplane”. It is noteworthy that multiple objects are also detected by BLC from messy background like “cat” and “couch”. The main issue in BLC is the misclassification for unseen objects which belong to the same meta class due to lacking of enough information to distinguish them, and we can see it as cases of ”elephant” and ”cat”.

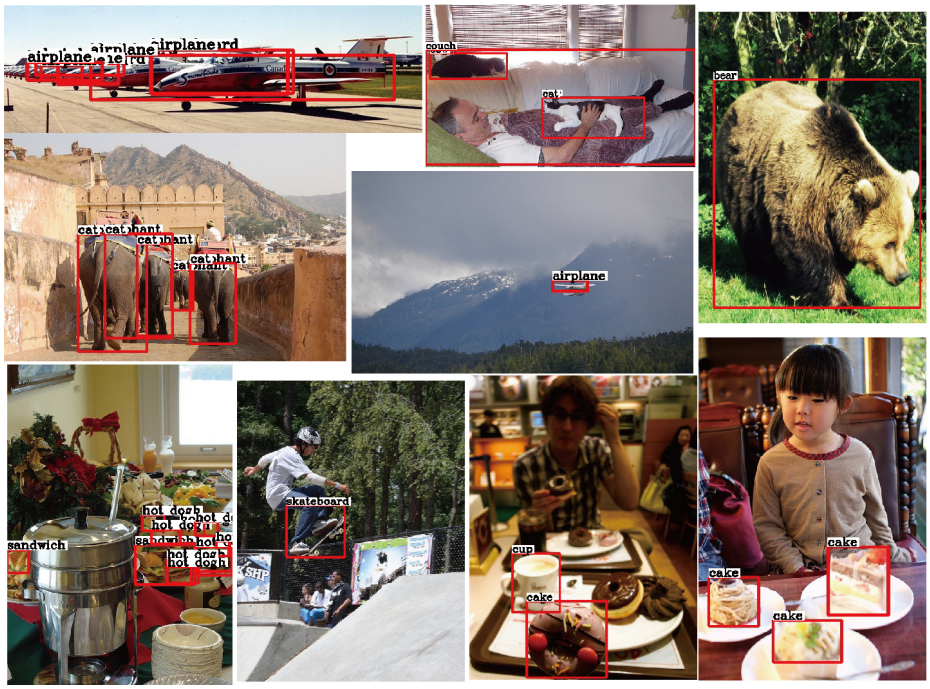


Fig. 5. Examples for detection results of BLC on 48/17 and 65/15 splits of MS-COCO. All these objects are belong unseen classes.

5 Conclusions

In this paper, we propose a novel framework for ZSD named Background Learnable Cascade (BLC), which includes Cascade Semantic R-CNN, semantic information flow and BLRPN. Cascade Semantic R-CNN progressively refines the visual-semantic alignment, semantic information flow improves the semantic feature learning and BLRPN learns a appropriate word vector for background class to reduce the confusion between background and unseen classes. Experiments in two splits of MS-COCO show that BLC outperforms several state of the art under both ZSD and GZSD tasks.

Acknowledgement

The paper is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61672498 and the National Key Research and Development Program of China under Grant No. 2016YFC0302300.

References

1. Zhang, Z., Saligrama, V.: Zero-shot learning via joint latent similarity embedding. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 6034–6042
2. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE (2008) 722–729
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115** (2015) 211–252
4. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200. (2010)
5. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 384–400
6. Demirel, B., Cinbis, R.G., Ikizler-Cinbis, N.: Zero-shot object detection by hybrid region embedding. arXiv preprint arXiv:1805.06157 (2018)
7. Rahman, S., Khan, S., Porikli, F.: Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In: Asian Conference on Computer Vision, Springer (2018) 547–563
8. Zhu, P., Wang, H., Saligrama, V.: Zero shot detection. IEEE Transactions on Circuits and Systems for Video Technology (2019)
9. Rahman, S., Khan, S., Barnes, N.: Improved visual-semantic alignment for zero-shot object detection. 34th AAAI Conference on Artificial Intelligence (2020)
10. Li, Z., Yao, L., Zhang, X., Wang, X., Kanhere, S., Zhang, H.: Zero-shot object detection with textual descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8690–8697
11. Zhao, S., Gao, C., Shao, Y., Li, L., Yu, C., Ji, Z., Sang, N.: Gtnet: Generative transfer network for zero-shot object detection. arXiv preprint arXiv:2001.06812 (2020)
12. Zhu, P., Wang, H., Saligrama, V.: Don't even look once: Synthesizing features for zero-shot detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 11693–11702
13. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 6154–6162
14. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 1563–1572
15. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 5327–5336
16. Elhoseiny, M., Saleh, B., Elgammal, A.: Write a classifier: Zero-shot learning using purely textual descriptions. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 2584–2591
17. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems. (2013) 2121–2129
18. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: European Conference on Computer Vision, Springer (2014) 393–409

19. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 3174–3183
20. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2009) 951–958
21. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE transactions on pattern analysis and machine intelligence **36** (2013) 453–465
22. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650 (2013)
23. Rahman, S., Khan, S., Porikli, F.: A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. IEEE Transactions on Image Processing **27** (2018) 5652–5667
24. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning-the good, the bad and the ugly. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4582–4591
25. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: Proceedings of the IEEE international conference on computer vision. (2015) 4166–4174
26. Zhang, Z., Saligrama, V.: Zero-shot recognition via structured prediction. In: European conference on computer vision, Springer (2016) 533–548
27. Al-Halah, Z., Stiefelhagen, R.: Automatic discovery, association estimation and learning of semantic attributes for a thousand categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 614–623
28. Al-Halah, Z., Tapaswi, M., Stiefelhagen, R.: Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5975–5984
29. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence **41** (2018) 2251–2265
30. Zablocki, E., Bordes, P., Soulier, L., Piwowarski, B., Gallinari, P.: Context-aware zero-shot learning for object recognition. In: International Conference on Machine Learning, PMLR (2019) 7292–7303
31. Luo, R., Zhang, N., Han, B., Yang, L.: Context-aware zero-shot recognition. arXiv preprint arXiv:1904.09320 (2019)
32. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123** (2017) 32–73
33. Mishra, A., Krishna Reddy, S., Mittal, A., Murthy, H.A.: A generative model for zero shot learning using conditional variational autoencoders. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2018) 2188–2196
34. Kumar Verma, V., Arora, G., Mishra, A., Rai, P.: Generalized zero-shot learning via synthesized examples. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4281–4289

35. Verma, V.K., Rai, P.: A simple exponential family framework for zero-shot learning. In: Joint European conference on machine learning and knowledge discovery in databases, Springer (2017) 792–808
36. Verma, V.K., Brahma, D., Rai, P.: Meta-learning for generalized zero-shot learning. In: AAAI. (2020) 6062–6069
37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 779–788
38. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision, Springer (2016) 21–37
39. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. (2017) 2980–2988
40. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
41. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. (2016) 379–387
42. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2017) 2961–2969
43. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 764–773
44. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 734–750
45. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 6569–6578
46. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9627–9636
47. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 6154–6162
48. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 7263–7271
49. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
50. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
51. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2117–2125
52. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: Proceedings of the ACM international conference on image and video retrieval. (2009) 1–9

53. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119
54. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)